# Exploring and pre-processing a dataset using Pandas

Estimated time needed: **30** minutes

## Objectives

After completing this lab you will be able to:

- Explore the dataset
- Pre-process dataset as required (may be for visualization)

## Introduction

The aim of this lab is to provide you a refresher on the **Pandas** library, so that you can pre-process and anlyse the datasets before applying data visualization techniques on it. This lab will work as acrash course on *pandas*. if you are interested in learning more about the *pandas* library, detailed description and explanation of how to use it and how to clean, munge, and process data stored in a *pandas* dataframe are provided in our course **Data Analysis with Python** and **Python for Applied Data Science**

---

## Table of Contents

# Exploring Datasets with *pandas*

*pandas* is an essential data analysis toolkit for Python. From their website:

> *pandas* is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, **real world** data analysis in Python.

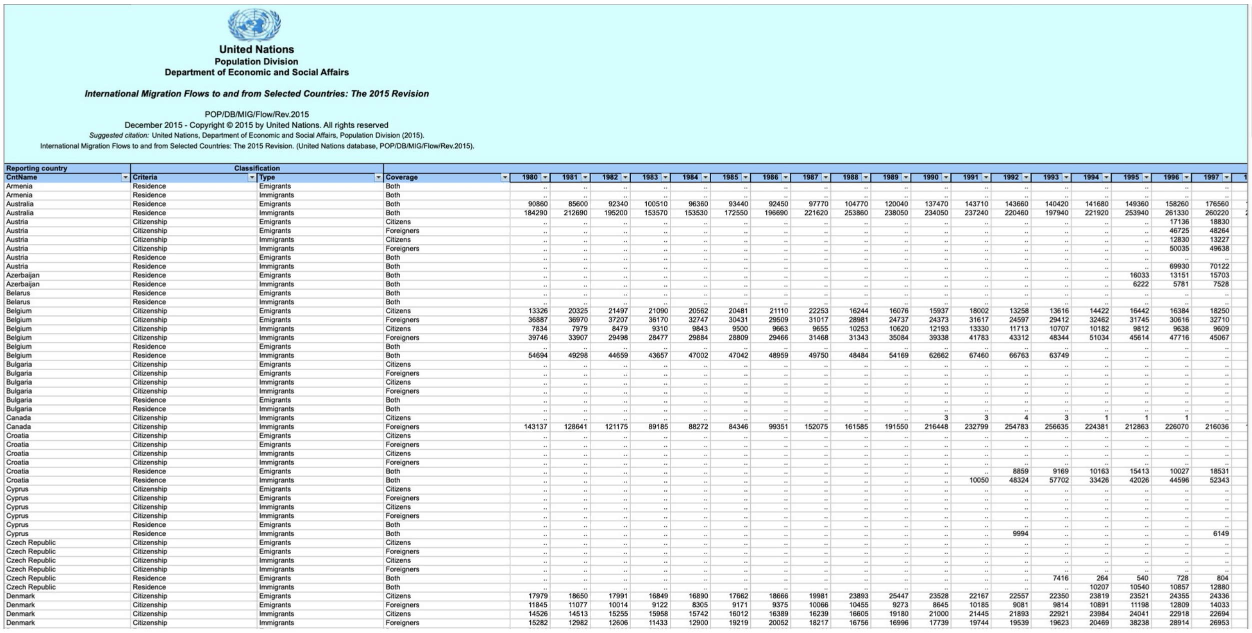
The course heavily relies on *pandas* for data wrangling, analysis, and visualization. We encourage you to spend some time and familiarize yourself with the *pandas* API Reference: http://pandas.pydata.org/pandas-docs/stable/api.html.

## The Dataset: Immigration to Canada from 1980 to 2013

Dataset Source: International migration flows to and from selected countries - The 2015 revision.

The dataset contains annual data on the flows of international immigrants as recorded by the countries of destination. The data presents both inflows and outflows according to the place of birth, citizenship or place of previous / next residence both for foreigners and nationals. The current version presents data pertaining to 45 countries.

In this lab, we will focus on the Canadian immigration data.



United Nations
Population Division
Department of Economic and Social Affairs

*International Migration Flows to and from Selected Countries: The 2015 Revision*

POP/DB/MIG/Flow/Rev.2015
December 2015 - Copyright © 2015 by United Nations. All rights reserved
*Suggested citation:* United Nations, Department of Economic and Social Affairs, Population Division (2015).
International Migration Flows to and from Selected Countries: The 2015 Revision. (United Nations database, POP/DB/MIG/Flow/Rev.2015).

| Reporting country | Classification | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CntName | Criteria | Type | Coverage | 1980 | 1981 | 1982 | 1983 | 1984 | 1985 | 1986 | 1987 | 1988 | 1989 | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 |
| Armenia | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Armenia | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Australia | Residence | Emigrants | Both | 90860 | 85600 | 92340 | 100510 | 96360 | 93440 | 92450 | 97770 | 104770 | 120040 | 137470 | 143710 | 143660 | 140420 | 141680 | 149360 | 158260 | 176560 |
| Australia | Residence | Immigrants | Both | 184290 | 212690 | 195200 | 153570 | 153530 | 172550 | 196690 | 221620 | 253860 | 238050 | 234050 | 237240 | 220460 | 197940 | 221920 | 253940 | 261330 | 260220 |
| Austria | Citizenship | Emigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 17136 | 18830 |
| Austria | Citizenship | Emigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 46725 | 48264 |
| Austria | Citizenship | Immigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 12830 | 13227 |
| Austria | Citizenship | Immigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 50035 | 49638 |
| Austria | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Austria | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 69930 | 70122 |
| Azerbaijan | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 16033 | 13151 | 15703 |
| Azerbaijan | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 6222 | 5781 | 7528 |
| Belarus | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Belarus | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Belgium | Citizenship | Emigrants | Citizens | 13326 | 20325 | 21497 | 21090 | 20562 | 20481 | 21110 | 22253 | 16244 | 16076 | 15937 | 18002 | 13258 | 13616 | 14422 | 16442 | 16384 | 18250 |
| Belgium | Citizenship | Emigrants | Foreigners | 36887 | 36970 | 37207 | 36170 | 32747 | 30431 | 29509 | 31017 | 28981 | 24737 | 24373 | 31617 | 24597 | 29412 | 32462 | 31745 | 30616 | 32710 |
| Belgium | Citizenship | Immigrants | Citizens | 7834 | 7979 | 8479 | 9310 | 9843 | 9500 | 9663 | 9655 | 10253 | 10620 | 12193 | 13330 | 11713 | 10707 | 10182 | 9812 | 9638 | 9609 |
| Belgium | Citizenship | Immigrants | Foreigners | 39746 | 33907 | 29498 | 28477 | 29884 | 28809 | 29466 | 31468 | 31343 | 35084 | 39338 | 41783 | 43312 | 48344 | 51034 | 45614 | 47716 | 45067 |
| Belgium | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Belgium | Residence | Immigrants | Both | 54694 | 49298 | 44659 | 43657 | 47002 | 47042 | 48959 | 49750 | 48484 | 54169 | 62662 | 67460 | 66763 | 63749 | .. | .. | .. | .. |
| Bulgaria | Citizenship | Emigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Bulgaria | Citizenship | Emigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Bulgaria | Citizenship | Immigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Bulgaria | Citizenship | Immigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Bulgaria | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Bulgaria | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Canada | Citizenship | Immigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 3 | 3 | 4 | 3 | 1 | 1 | 1 | .. |
| Canada | Citizenship | Immigrants | Foreigners | 143137 | 128641 | 121175 | 89185 | 88272 | 84346 | 99351 | 152075 | 161585 | 191550 | 216448 | 232799 | 254783 | 256635 | 224381 | 212863 | 226070 | 216036 |
| Croatia | Citizenship | Emigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Croatia | Citizenship | Emigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Croatia | Citizenship | Immigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Croatia | Citizenship | Immigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Croatia | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 8859 | 9169 | 10163 | 15413 | 10027 | 18531 |
| Croatia | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 10050 | 48324 | 57702 | 33426 | 42026 | 44596 | 52343 |
| Cyprus | Citizenship | Emigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Cyprus | Citizenship | Emigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Cyprus | Citizenship | Immigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Cyprus | Citizenship | Immigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Cyprus | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Cyprus | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 9994 | .. | .. | .. | .. | 6149 |
| Czech Republic | Citizenship | Emigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Czech Republic | Citizenship | Emigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Czech Republic | Citizenship | Immigrants | Citizens | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Czech Republic | Citizenship | Immigrants | Foreigners | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Czech Republic | Residence | Emigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 7416 | 264 | 540 | 728 | 804 |
| Czech Republic | Residence | Immigrants | Both | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 10207 | 10540 | 10857 | 12880 |
| Denmark | Citizenship | Emigrants | Citizens | 17979 | 18650 | 17991 | 16849 | 16890 | 17662 | 18666 | 19981 | 23893 | 25447 | 23528 | 22167 | 22557 | 22350 | 23819 | 23521 | 24355 | 24336 |
| Denmark | Citizenship | Emigrants | Foreigners | 11845 | 11077 | 10014 | 9122 | 8305 | 9171 | 9375 | 10066 | 10455 | 9273 | 8645 | 10185 | 9081 | 9814 | 10891 | 11198 | 12809 | 14033 |
| Denmark | Citizenship | Immigrants | Citizens | 14526 | 14513 | 15255 | 15958 | 15742 | 16012 | 16389 | 16239 | 16605 | 19180 | 21000 | 21445 | 21893 | 22921 | 23984 | 24041 | 22918 | 22694 |
| Denmark | Citizenship | Immigrants | Foreigners | 15282 | 12982 | 12606 | 11433 | 12900 | 19219 | 20052 | 18217 | 16756 | 16996 | 17739 | 19744 | 19539 | 19623 | 20469 | 38238 | 28914 | 26953 |

The Canada Immigration dataset can be fetched from here.

---

# *pandas* Basics

The first thing we'll do is install **openpyxl** (formerly **xlrd**), a module that *pandas* requires to read Excel files.

```
!mamba install openpyxl==3.0.9 -y
```

```
                  __    __    __    __
                 /  \  /  \  /  \  /  \
                /    \/    \/    \/    \
███████████████/  /██/  /██/  /██/  /████████████████████████
              /  / \   / \   / \   / \  \____
             /  /   \_/   \_/   \_/   \    o \__,
            / _/                       \_____/  `
            |/
        ███╗   ███╗ █████╗ ███╗   ███╗██████╗  █████╗
        ████╗ ████║██╔══██╗████╗ ████║██╔══██╗██╔══██╗
        ██╔████╔██║███████║██╔████╔██║██████╔╝███████║
        ██║╚██╔╝██║██╔══██║██║╚██╔╝██║██╔══██╗██╔══██║
        ██║ ╚═╝ ██║██║  ██║██║ ╚═╝ ██║██████╔╝██║  ██║
        ╚═╝     ╚═╝╚═╝  ╚═╝╚═╝     ╚═╝╚═════╝ ╚═╝  ╚═╝
        ██║ ╚═╝ ██║██║  ██║██║ ╚═╝ ██║██████╔╝██║  ██║
        ╚═╝     ╚═╝╚═╝  ╚═╝╚═╝     ╚═╝╚═════╝ ╚═╝  ╚═╝

        mamba (1.4.2) supported by @QuantStack

        GitHub:  https://github.com/mamba-org/mamba
        Twitter: https://twitter.com/QuantStack

█████████████████████████████████████████████████████████████


Looking for: ['openpyxl==3.0.9']

ain/linux-64 ━━━━━━━━━━━━━╸━━━━━━━━━━━━   0.0 B /  ??.?MB @  ??.?MB/s
0.1s
pkgs/main/noarch   ━━━━━━━━━━━━━╸━━━━━━━━━━━   0.0 B /  ??.?MB @  ??.?
MB/s  0.1s
pkgs/r/linux-64    ━━━━━━━━━━━━━━━╸━━━━━━━━━   0.0 B /  ??.?MB @  ??.?
MB/s  0.1s
pkgs/r/noarch      ━━━━━━━╸━━━━━━━━━━━━━━━╸━━   0.0 B /  ??.?MB @  ??.?
MB/s  0.1sain/linux-64 ━━━━━━━━━━━━━━━━╸━━━━━━━━━  16.4kB /  ??.?MB @
106.7kB/s  0.2s
pkgs/main/noarch   ━━━━━━━━━━━━━━━━╸━━━━━━━━  41.0kB /  ??.?MB @
266.6kB/s  0.2s
pkgs/r/linux-64    ━━━━━━━━━╸━━━━━━━━━━━━━━━━  45.1kB /  ??.?MB @
292.7kB/s  0.2s
pkgs/r/noarch      ━━━━━━━━━╸━━━━━━━━━━━━━━━━  12.3kB /  ??.?MB @
79.9kB/s  0.2sain/linux-64 ━━━━━━━━━━╸━━━━━━━━━━━━━━━ 323.6kB /  ??.?MB
@   1.3MB/s  0.3s
pkgs/main/noarch   ━━━━━━━━━━╸━━━━━━━━━━━━━━━ 524.3kB /  ??.?MB @
2.1MB/s  0.3s
pkgs/r/linux-64    ━━━━━━━━━━━╸━━━━━━━━━━━━━━ 507.9kB /  ??.?MB @
2.0MB/s  0.3s
pkgs/r/noarch      ━━━━━━━━━━━━╸━━━━━━━━━━━━━ 475.1kB /  ??.?MB @
1.9MB/s  0.3sain/noarch                                   873.1kB @
2.7MB/s  0.3s
[+] 0.4s
pkgs/main/linux-64 ━━━━━━━━━━━━╸━━━━━━━━━━━━━ 872.5kB /  ??.?MB @
2.4MB/s  0.4s
pkgs/r/linux-64    ━━━━━━━━━━━━━━╸━━━━━━━━━━━ 987.1kB /  ??.?MB @
2.8MB/s  0.4s
pkgs/r/noarch      ━━━━━━━━━━━━━━╸━━━━━━━━━━━ 921.6kB /  ??.?MB @
2.6MB/s  0.4sain/linux-64 ━━━━━━━━━━━━━━━╸━━━━━━━━━━   1.3MB /  ??.?MB
@   2.9MB/s  0.5s
pkgs/r/linux-64    ━━━━━━━━━━━━━━━━╸━━━━━━━━━   1.5MB /  ??.?MB @
3.2MB/s  0.5s
pkgs/r/noarch      ━━━━━━━━╸━━━━━━━━━━━━━━━━━   1.4MB /  ??.?MB @
3.0MB/s  0.5sain/linux-64 ╸━━━━━━━━━━━━━━━━╸━━━━━━━━   1.8MB /  ??.?MB
@   3.2MB/s  0.6s
pkgs/r/noarch      ━━━━━━━━━━━╸━━━━━━━━━━━━━━   1.9MB /  ??.?MB @
```

```
3.3MB/s  0.6sain/linux-64 ━━╸━━━━━━━━━━━━━━━╸━━━━━   2.3MB @   3.4MB/s
0.7s
pkgs/r/noarch       ━━━━━━━━━━━━━━━━━━━━━━━━   2.3MB @   3.5MB/s
Finalizing  0.7sain/linux-64 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━   2.8MB /  ??.?
MB @   3.6MB/s  0.8sain/linux-64 ━━━━━━╸━━━━━━━━━━━━━━━╸━
3.4MB /  ??.?MB @   3.9MB/s  0.9sain/linux-64
━━━━━━━━━━━━━━━━━━━━━━━━━   3.8MB /  ??.?MB @   3.9MB/s
1.0sain/linux-64 ━━━━━━━━━━━━╸━━━━━━━━━━━━   4.3MB /  ??.?MB @
3.9MB/s  1.1sain/linux-64 ━━━━━━━━━━━━━━━╸━━━━━━━━━   4.8MB /  ??.?MB
@   4.1MB/s  1.2sain/linux-64 ━━━━━━━━━━━━━━━━━━━━━━━━━
5.3MB /  ??.?MB @   4.1MB/s  1.3sain/linux-64
━━━━━━━━━━━━━━━━━━━━━━━━━━   5.8MB /  ??.?MB @   4.2MB/s
1.4sain/linux-64 ━━━━━━━━━━━━━━╸━━━━━━━━━━   6.3MB /  ??.?MB @
4.2MB/s  1.5sain/linux-64 ╸━━━━━━━━━━━━━━━╸━━━━━━━━   6.8MB /  ??.?MB
@   4.3MB/s  1.6sain/linux-64 ━━━━━━━━━━━━━━━━━━━━━━━━   7.0MB @
4.3MB/s Finalizing  1.7sain/linux-64                                 @
4.3MB/s  1.7s
e/jupyterlab/conda/envs/python

  Updating specs:

   - openpyxl==3.0.9
   - ca-certificates
   - certifi
   - openssl


  Package              Version  Build           Channel
Size
──────────────────────────────────────────────────────────────────────
────────
  Install:
──────────────────────────────────────────────────────────────────────
────────

  + et_xmlfile           1.1.0  py37h06a4308_0  pkgs/main/linux-64
10kB
  + openpyxl             3.0.9  pyhd3eb1b0_0    pkgs/main/noarch
168kB

  Upgrade:
──────────────────────────────────────────────────────────────────────
────────

  - ca-certificates   2023.5.7  hbcca054_0      conda-forge

  + ca-certificates  2024.3.11  h06a4308_0      pkgs/main/linux-64
130kB
  - openssl             1.1.1t  h0b41bf4_0      conda-forge
  + openssl             1.1.1w  h7f8727e_0      pkgs/main/linux-64
4MB

  Summary:

  Install: 2 packages
  Upgrade: 2 packages

  Total download: 4MB

──────────────────────────────────────────────────────────────────────
────────


━━━━━━━━━━━━━━━━━━━━━━━   0.0 B ca-certificates            0.0s
Extracting       ━━━━━━━━━━━━━━━━━━━━━━━       0
0.0s━━━━━━━━━━━━━━━━━━━━━━━   0.0 B ca-certificates            0.1s
Extracting       ━━━━━━━━━━━━━━━━━━━━━━━       0
0.0slfile                                         10.2kB @  61.9kB/s
0.2s
[+] 0.2s
Downloading      ━━━━━━━━━━━━━━━━━━━━━━━   4.2MB
0.2s
Extracting   (4) ╸━━━━━━━━━━━━━━━╸━━━━━━       0 ca-certificates
0.0s━╸━━━━━━━━━━━━━━━╸━━━━━       0 ca-certificates
0.1s━━╸━━━━━━━━━━━━━━━╸━━━━       0 ca-certificates
0.2s━━━━╸━━━━━━━━━━━━━━━╸━━       0 ca-certificates
0.3s━━━━━╸━━━━━━━━━━━━━━━╸━       0 et_xmlfile
0.4s━━━━━━╸━━━━━━━━━━━━━━━━       0 et_xmlfile
0.5s━━━━━━━╸━━━━━━━━━━━━━━━       0 et_xmlfile
0.6s━━━━━━━━╸━━━━━━━━━━━━━━       0 et_xmlfile
0.7s━━━━━━━━━╸━━━━━━━━━━━━━       0 openpyxl
0.8s━━━━━━━━━━╸━━━━━━━━━━━━       0 openpyxl
0.9s━━━━━━━━━━━╸━━━━━━━━━━━       0 openpyxl
1.0s━━━━━━━━━━━━╸━━━━━━━━━━       0 openpyxl
1.1s━━━━━━━╸━━━━━━━━━━━━━━━       0 openssl
1.2s━━━━━━━━━╸━━━━━━━━━━━━━       0 openssl
1.3s━━━━━━━━━━╸━━━━━━━━━━━━       0 openssl
1.4s━━━━━━━━━━━━━━━━━━       1 openssl
1.5s━━━━━━━━━━━━━━━━━━       1 ca-certificates            1.6s━━━━━━
3 openssl                    1.7s━━━━━━       3 openssl
1.8s
```

Next, we'll do is import two key data analysis modules: *pandas* and *numpy*.

```python
import numpy as np  # useful for many scientific computing in Python
import pandas as pd # primary data structure library
```

Let's download and import our primary Canadian Immigration dataset using *pandas*'s `read_excel()` method.

```
df_can = pd.read_excel(
    'https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DV0101EN-
SkillsNetwork/Data%20Files/Canada.xlsx',
    sheet_name='Canada by Citizenship',
    skiprows=range(20),
    skipfooter=2)

print('Data read into a pandas dataframe!')
```

```
Data read into a pandas dataframe!
```

Let's view the top 5 rows of the dataset using the `head()` function.

```
df_can.head()
# tip: You can specify the number of rows you'd like to see as
follows: df_can.head(10)
```

```
         Type    Coverage          OdName  AREA AreaName   REG  \
0  Immigrants  Foreigners     Afghanistan   935     Asia  5501
1  Immigrants  Foreigners         Albania   908   Europe   925
2  Immigrants  Foreigners         Algeria   903   Africa   912
3  Immigrants  Foreigners  American Samoa   909  Oceania   957
4  Immigrants  Foreigners         Andorra   908   Europe   925

           RegName  DEV             DevName  1980  ...  2004  2005
2006  \
0    Southern Asia  902  Developing regions    16  ...  2978  3436
3009
1  Southern Europe  901   Developed regions     1  ...  1450  1223
856
2  Northern Africa  902  Developing regions    80  ...  3616  3626
4807
3        Polynesia  902  Developing regions     0  ...     0     0
1
4  Southern Europe  901   Developed regions     0  ...     0     0
1

   2007  2008  2009  2010  2011  2012  2013
0  2652  2111  1746  1758  2203  2635  2004
1   702   560   716   561   539   620   603
2  3623  4005  5393  4752  4325  3774  4331
3     0     0     0     0     0     0     0
4     1     0     0     0     0     1     1

[5 rows x 43 columns]
```

We can also view the bottom 5 rows of the dataset using the `tail()` function.

```
df_can.tail()

           Type    Coverage          OdName  AREA AreaName  REG  \
190  Immigrants  Foreigners        Viet Nam   935     Asia  920
191  Immigrants  Foreigners  Western Sahara   903   Africa  912
192  Immigrants  Foreigners           Yemen   935     Asia  922
193  Immigrants  Foreigners          Zambia   903   Africa  910
194  Immigrants  Foreigners        Zimbabwe   903   Africa  910

                RegName  DEV             DevName  1980  ...  2004
2005  2006  \
190  South-Eastern Asia  902  Developing regions  1191  ...  1816
1852  3153
191     Northern Africa  902  Developing regions     0  ...     0
0     1
192        Western Asia  902  Developing regions     1  ...   124
161   140
193      Eastern Africa  902  Developing regions    11  ...    56
91    77
194      Eastern Africa  902  Developing regions    72  ...  1450
615   454

     2007  2008  2009  2010  2011  2012  2013
190  2574  1784  2171  1942  1723  1731  2112
191     0     0     0     0     0     0     0
192   122   133   128   211   160   174   217
193    71    64    60   102    69    46    59
194   663   611   508   494   434   437   407

[5 rows x 43 columns]
```

When analyzing a dataset, it's always a good idea to start by getting basic information about your dataframe. We can do this by using the `info()` method.

This method can be used to get a short summary of the dataframe.

```
df_can.info(verbose=False)

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Columns: 43 entries, Type to 2013
dtypes: int64(37), object(6)
memory usage: 65.6+ KB
```

To get the list of column headers we can call upon the data frame's `columns` instance variable.

```
df_can.columns

Index([    'Type', 'Coverage',   'OdName',     'AREA', 'AreaName',
'REG',
        'RegName',      'DEV',  'DevName',       1980,       1981,
1982,
             1983,       1984,       1985,       1986,       1987,
1988,
             1989,       1990,       1991,       1992,       1993,
1994,
             1995,       1996,       1997,       1998,       1999,
2000,
             2001,       2002,       2003,       2004,       2005,
2006,
             2007,       2008,       2009,       2010,       2011,
2012,
             2013],
      dtype='object')
```

Similarly, to get the list of indices we use the `.index` instance variables.

```
df_can.index

RangeIndex(start=0, stop=195, step=1)
```

Note: The default type of intance variables `index` and `columns` are **NOT** `list`.

```
print(type(df_can.columns))
print(type(df_can.index))

<class 'pandas.core.indexes.base.Index'>
<class 'pandas.core.indexes.range.RangeIndex'>
```

To get the index and columns as lists, we can use the `tolist()` method.

```
df_can.columns.tolist()

['Type',
 'Coverage',
 'OdName',
 'AREA',
 'AreaName',
 'REG',
 'RegName',
 'DEV',
 'DevName',
 1980,
 1981,
 1982,
 1983,
 1984,
 1985,
 1986,
 1987,
 1988,
 1989,
 1990,
 1991,
 1992,
 1993,
 1994,
 1995,
 1996,
 1997,
 1998,
 1999,
 2000,
 2001,
 2002,
 2003,
 2004,
 2005,
 2006,
 2007,
 2008,
 2009,
 2010,
 2011,
 2012,
 2013]
```

```
df_can.index.tolist()
```

```
[0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 29,
 30,
 31,
 32,
 33,
 34,
 35,
 36,
 37,
 38,
 39,
 40,
 41,
 42,
 43,
 44,
 45,
 46,
 47,
 48,
 49,
 50,
 51,
 52,
 53,
 54,
 55,
 56,
 57,
 58,
 59,
 60,
 61,
 62,
 63,
 64,
 65,
 66,
 67,
 68,
 69,
 70,
 71,
 72,
 73,
 74,
 75,
 76,
 77,
 78,
 79,
 80,
 81,
 82,
 83,
 84,
 85,
 86,
 87,
 88,
 89,
 90,
 91,
 92,
 93,
 94,
 95,
 96,
 97,
 98,
 99,
 100,
 101,
 102,
 103,
 104,
 105,
 106,
 107,
 108,
 109,
 110,
 111,
 112,
 113,
 114,
 115,
 116,
 117,
118,
119,
120,
121,
122,
123,
124,
125,
126,
127,
128,
129,
130,
131,
132,
133,
134,
135,
136,
137,
138,
139,
140,
141,
142,
143,
144,
145,
146,
147,
148,
149,
150,
151,
152,
153,
154,
155,
156,
157,
158,
159,
160,
161,
162,
163,
164,
165,
166,
 167,
 168,
 169,
 170,
 171,
 172,
 173,
 174,
 175,
 176,
 177,
 178,
 179,
 180,
 181,
 182,
 183,
 184,
 185,
 186,
 187,
 188,
 189,
 190,
 191,
 192,
 193,
 194]
```

```
print(type(df_can.columns.tolist()))
print(type(df_can.index.tolist()))
```

```
<class 'list'>
<class 'list'>
```

To view the dimensions of the dataframe, we use the `shape` instance variable of it.

```
# size of dataframe (rows, columns)
df_can.shape
```

```
(195, 43)
```

**Note**: The main types stored in *pandas* objects are `float`, `int`, `bool`, `datetime64[ns]`, `datetime64[ns, tz]`, `timedelta[ns]`, `category`, and `object` (string). In addition, these dtypes have item sizes, e.g. `int64` and `int32`.

Let's clean the data set to remove a few unnecessary columns. We can use *pandas* `drop()` method as follows:

```
# in pandas axis=0 represents rows (default) and axis=1 represents
columns.
df_can.drop(['AREA','REG','DEV','Type','Coverage'], axis=1,
inplace=True)
df_can.head(2)

        OdName AreaName            RegName             DevName  1980
1981  \
0  Afghanistan     Asia     Southern Asia  Developing regions     16
39
1      Albania   Europe   Southern Europe   Developed regions      1
0

   1982  1983  1984  1985  ...  2004  2005  2006  2007  2008  2009
2010  \
0    39    47    71   340  ...  2978  3436  3009  2652  2111  1746
1758
1     0     0     0     0  ...  1450  1223   856   702   560   716
561

   2011  2012  2013
0  2203  2635  2004
1   539   620   603

[2 rows x 38 columns]
```

Let's rename the columns so that they make sense. We can use `rename()` method by passing in a dictionary of old and new names as follows:

```
df_can.rename(columns={'OdName':'Country', 'AreaName':'Continent',
'RegName':'Region'}, inplace=True)
df_can.columns

Index([  'Country', 'Continent',    'Region',   'DevName',
1980,
              1981,        1982,        1983,        1984,
1985,
              1986,        1987,        1988,        1989,
1990,
              1991,        1992,        1993,        1994,
1995,
              1996,        1997,        1998,        1999,
2000,
              2001,        2002,        2003,        2004,
2005,
              2006,        2007,        2008,        2009,
2010,
              2011,        2012,        2013],
      dtype='object')
```

We will also add a 'Total' column that sums up the total immigrants by country over the entire period 1980 - 2013, as follows:

```
df_can['Total'] = df_can.sum(axis=1)
df_can['Total']
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/
ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns
in DataFrame reductions (with 'numeric_only=None') is deprecated; in a
future version this will raise TypeError.  Select only valid columns
before calling the reduction.
  """Entry point for launching an IPython kernel.
```

```
0      58639
1      15699
2      69439
3          6
4         15
       ...
190    97146
191        2
192     2985
193     1677
194     8598
Name: Total, Length: 195, dtype: int64
```

We can check to see how many null objects we have in the dataset as follows:

```
df_can.isnull().sum()
```

```
Country      0
Continent    0
Region       0
DevName      0
1980         0
1981         0
1982         0
1983         0
1984         0
1985         0
1986         0
1987         0
1988         0
1989         0
1990         0
1991         0
1992         0
1993         0
1994         0
1995         0
```

```
1996         0
1997         0
1998         0
1999         0
2000         0
2001         0
2002         0
2003         0
2004         0
2005         0
2006         0
2007         0
2008         0
2009         0
2010         0
2011         0
2012         0
2013         0
Total        0
dtype: int64
```

Finally, let's view a quick summary of each column in our dataframe using the `describe()` method.

```
df_can.describe()
```

```
               1980          1981          1982          1983
1984  \
count    195.000000    195.000000    195.000000    195.000000
195.000000
mean     508.394872    566.989744    534.723077    387.435897
376.497436
std     1949.588546   2152.643752   1866.997511   1204.333597
1198.246371
min        0.000000      0.000000      0.000000      0.000000
0.000000
25%        0.000000      0.000000      0.000000      0.000000
0.000000
50%       13.000000     10.000000     11.000000     12.000000
13.000000
75%      251.500000    295.500000    275.000000    173.000000
181.000000
max    22045.000000  24796.000000  20620.000000  10015.000000
10170.000000

              1985         1986          1987          1988
1989  \
count   195.000000   195.000000    195.000000    195.000000
195.000000
```

```
mean    358.861538   441.271795    691.133333    714.389744
843.241026
std    1079.309600  1225.576630   2109.205607   2443.606788
2555.048874
min       0.000000     0.000000      0.000000      0.000000
0.000000
25%       0.000000     0.500000      0.500000      1.000000
1.000000
50%      17.000000    18.000000     26.000000     34.000000
44.000000
75%     197.000000   254.000000    434.000000    409.000000
508.500000
max    9564.000000  9470.000000  21337.000000  27359.000000
23795.000000

       ...          2005          2006          2007          2008  \
count  ...    195.000000    195.000000    195.000000    195.000000
mean   ...   1320.292308   1266.958974   1191.820513   1246.394872
std    ...   4425.957828   3926.717747   3443.542409   3694.573544
min    ...      0.000000      0.000000      0.000000      0.000000
25%    ...     28.500000     25.000000     31.000000     31.000000
50%    ...    210.000000    218.000000    198.000000    205.000000
75%    ...    832.000000    842.000000    899.000000    934.500000
max    ...  42584.000000  33848.000000  28742.000000  30037.000000

               2009          2010          2011          2012
2013  \
count    195.000000    195.000000    195.000000    195.000000
195.000000
mean    1275.733333   1420.287179   1262.533333   1313.958974
1320.702564
std     3829.630424   4462.946328   4030.084313   4247.555161
4237.951988
min        0.000000      0.000000      0.000000      0.000000
0.000000
25%       36.000000     40.500000     37.500000     42.500000
45.000000
50%      214.000000    211.000000    179.000000    233.000000
213.000000
75%      888.000000    932.000000    772.000000    783.000000
796.000000
max    29622.000000  38617.000000  36765.000000  34315.000000
34129.000000

               Total
count     195.000000
mean    32867.451282
std     91785.498686
min         1.000000
25%       952.000000
```

```
50%       5018.000000
75%      22239.500000
max     691904.000000

[8 rows x 35 columns]
```

---

# *pandas* Intermediate: Indexing and Selection (slicing)

## Select Column

**There are two ways to filter on a column name:**

Method 1: Quick and easy, but only works if the column name does NOT have spaces or special characters.

```
    df.column_name               # returns series
```

Method 2: More robust, and can filter on multiple columns.

```
    df['column']                  # returns series

    df[['column 1', 'column 2']]  # returns dataframe
```

---

Example: Let's try filtering on the list of countries ('Country').

```
df_can.Country  # returns a series

0         Afghanistan
1             Albania
2             Algeria
3      American Samoa
4             Andorra
            ...
190          Viet Nam
191    Western Sahara
192             Yemen
193            Zambia
194          Zimbabwe
Name: Country, Length: 195, dtype: object
```

Let's try filtering on the list of countries ('Country') and the data for years: 1980 - 1985.

```
df_can[['Country', 1980, 1981, 1982, 1983, 1984, 1985]] # returns a
dataframe
```

```
# notice that 'Country' is string, and the years are integers.
# for the sake of consistency, we will convert all column names to
string later on.

            Country  1980  1981  1982  1983  1984  1985
0       Afghanistan    16    39    39    47    71   340
1           Albania     1     0     0     0     0     0
2           Algeria    80    67    71    69    63    44
3    American Samoa     0     1     0     0     0     0
4           Andorra     0     0     0     0     0     0
..              ...   ...   ...   ...   ...   ...   ...
190        Viet Nam  1191  1829  2162  3404  7583  5907
191  Western Sahara     0     0     0     0     0     0
192           Yemen     1     2     1     6     0    18
193          Zambia    11    17    11     7    16     9
194        Zimbabwe    72   114   102    44    32    29

[195 rows x 7 columns]
```

## Select Row

There are main 2 ways to select rows:

```
    df.loc[label]    # filters by the labels of the index/column
    df.iloc[index]   # filters by the positions of the index/column
```

Before we proceed, notice that the default index of the dataset is a numeric range from 0 to 194. This makes it very difficult to do a query by a specific country. For example to search for data on Japan, we need to know the corresponding index value.

This can be fixed very easily by setting the 'Country' column as the index using `set_index()` method.

```
df_can.set_index('Country', inplace=True)
# tip: The opposite of set is reset. So to reset the index, we can use
df_can.reset_index()

df_can.head(3)

            Continent           Region            DevName  1980  1981
1982  \
Country

Afghanistan      Asia    Southern Asia  Developing regions    16    39
39
Albania        Europe  Southern Europe   Developed regions     1     0
0
Algeria        Africa  Northern Africa  Developing regions    80    67
71
```

```
             1983  1984  1985  1986  ...  2005  2006  2007  2008  2009
2010  \
Country                                ...

Afghanistan    47    71   340   496  ...  3436  3009  2652  2111  1746
1758
Albania         0     0     0     1  ...  1223   856   702   560   716
561
Algeria        69    63    44    69  ...  3626  4807  3623  4005  5393
4752

             2011  2012  2013  Total
Country
Afghanistan  2203  2635  2004  58639
Albania       539   620   603  15699
Algeria      4325  3774  4331  69439

[3 rows x 38 columns]

# optional: to remove the name of the index
df_can.index.name = None
```

Example: Let's view the number of immigrants from Japan (row 87) for the following scenarios: 1. The full row data (all columns) 2. For year 2013 3. For years 1980 to 1985

```
# 1. the full row data (all columns)
df_can.loc['Japan']

Continent                 Asia
Region            Eastern Asia
DevName      Developed regions
1980                       701
1981                       756
1982                       598
1983                       309
1984                       246
1985                       198
1986                       248
1987                       422
1988                       324
1989                       494
1990                       379
1991                       506
1992                       605
1993                       907
1994                       956
1995                       826
1996                       994
1997                       924
1998                       897
```

```
1999                          1083
2000                          1010
2001                          1092
2002                           806
2003                           817
2004                           973
2005                          1067
2006                          1212
2007                          1250
2008                          1284
2009                          1194
2010                          1168
2011                          1265
2012                          1214
2013                           982
Total                        27707
Name: Japan, dtype: object
```

```
# alternate methods
df_can.iloc[87]
```

```
Continent                     Asia
Region                Eastern Asia
DevName          Developed regions
1980                           701
1981                           756
1982                           598
1983                           309
1984                           246
1985                           198
1986                           248
1987                           422
1988                           324
1989                           494
1990                           379
1991                           506
1992                           605
1993                           907
1994                           956
1995                           826
1996                           994
1997                           924
1998                           897
1999                          1083
2000                          1010
2001                          1092
2002                           806
2003                           817
2004                           973
2005                          1067
```

```
2006                         1212
2007                         1250
2008                         1284
2009                         1194
2010                         1168
2011                         1265
2012                         1214
2013                          982
Total                       27707
Name: Japan, dtype: object
```

```
df_can[df_can.index == 'Japan']
```

```
      Continent         Region             DevName  1980  1981  1982
1983  \
Japan      Asia   Eastern Asia  Developed regions   701   756   598
309

       1984  1985  1986  ...   2005  2006  2007  2008  2009  2010  2011
2012  \
Japan   246   198   248  ...   1067  1212  1250  1284  1194  1168  1265
1214

       2013  Total
Japan   982  27707

[1 rows x 38 columns]
```

```
# 2. for year 2013
df_can.loc['Japan', 2013]
```

```
982
```

```
# alternate method
# year 2013 is the last column, with a positional index of 36
df_can.iloc[87, 36]
```

```
982
```

```
# 3. for years 1980 to 1985
df_can.loc['Japan', [1980, 1981, 1982, 1983, 1984, 1984]]
```

```
1980    701
1981    756
1982    598
1983    309
1984    246
1984    246
Name: Japan, dtype: object
```

```
# Alternative Method
df_can.iloc[87, [3, 4, 5, 6, 7, 8]]
```

```
1980    701
1981    756
1982    598
1983    309
1984    246
1985    198
Name: Japan, dtype: object
```

**Exercise:** Let's view the number of immigrants from **Haiti** for the following scenarios: 1. The full row data (all columns) 2. For year 2000 3. For years 1990 to 1995

```
df_can.loc['Haiti']
df_can.loc['Haiti', 2000]
df_can.loc['Haiti', [1990, 1991, 1992, 1993, 1994, 1995]]
```

```
1990    2379
1991    2829
1992    2399
1993    3655
1994    2100
1995    2014
Name: Haiti, dtype: object
```

Column names that are integers (such as the years) might introduce some confusion. For example, when we are referencing the year 2013, one might confuse that when the 2013th positional index.

To avoid this ambuigity, let's convert the column names into strings: '1980' to '2013'.

```
df_can.columns = list(map(str, df_can.columns))
# [print (type(x)) for x in df_can.columns.values] #<-- uncomment to
check type of column headers
```

Since we converted the years to string, let's declare a variable that will allow us to easily call upon the full range of years:

```
# useful for plotting later on
years = list(map(str, range(1980, 2014)))
years
```

```
['1980',
 '1981',
 '1982',
 '1983',
 '1984',
 '1985',
 '1986',
 '1987',
 '1988',
 '1989',
 '1990',
 '1991',
 '1992',
 '1993',
 '1994',
 '1995',
 '1996',
 '1997',
 '1998',
 '1999',
 '2000',
 '2001',
 '2002',
 '2003',
 '2004',
 '2005',
 '2006',
 '2007',
 '2008',
 '2009',
 '2010',
 '2011',
 '2012',
 '2013']
```

**Exercise:** Create a list named 'year' using map function for years ranging from 1990 to 2013. Then extract the data series from the dataframe df_can for Haiti using year list.

```python
year = list(map(str, range(1990, 2014)))
haiti = df_can.loc['Haiti', year]
```

## Filtering based on a criteria

To filter the dataframe based on a condition, we simply pass the condition as a boolean vector.

For example, Let's filter the dataframe to show the data on Asian countries (AreaName = Asia).

```python
# 1. create the condition boolean series
condition = df_can['Continent'] == 'Asia'
print(condition)
```

```
Afghanistan        True
Albania           False
Algeria           False
American Samoa    False
Andorra           False
                  ...
Viet Nam           True
```

```
Western Sahara    False
Yemen              True
Zambia            False
Zimbabwe          False
Name: Continent, Length: 195, dtype: bool

# 2. pass this condition into the dataFrame
df_can[condition]

                                               Continent
Region  \
Afghanistan                                         Asia
Southern Asia
Armenia                                             Asia
Western Asia
Azerbaijan                                          Asia
Western Asia
Bahrain                                             Asia
Western Asia
Bangladesh                                          Asia
Southern Asia
Bhutan                                              Asia
Southern Asia
Brunei Darussalam                                   Asia  South-
Eastern Asia
Cambodia                                            Asia  South-
Eastern Asia
China                                               Asia
Eastern Asia
China, Hong Kong Special Administrative Region      Asia
Eastern Asia
China, Macao Special Administrative Region          Asia
Eastern Asia
Cyprus                                              Asia
Western Asia
Democratic People's Republic of Korea               Asia
Eastern Asia
Georgia                                             Asia
Western Asia
India                                               Asia
Southern Asia
Indonesia                                           Asia  South-
Eastern Asia
Iran (Islamic Republic of)                          Asia
Southern Asia
Iraq                                                Asia
Western Asia
Israel                                              Asia
Western Asia
Japan                                               Asia
```

```
Eastern Asia
Jordan                                              Asia
Western Asia
Kazakhstan                                          Asia
Central Asia
Kuwait                                              Asia
Western Asia
Kyrgyzstan                                          Asia
Central Asia
Lao People's Democratic Republic                    Asia  South-
Eastern Asia
Lebanon                                             Asia
Western Asia
Malaysia                                            Asia  South-
Eastern Asia
Maldives                                            Asia
Southern Asia
Mongolia                                            Asia
Eastern Asia
Myanmar                                             Asia  South-
Eastern Asia
Nepal                                               Asia
Southern Asia
Oman                                                Asia
Western Asia
Pakistan                                            Asia
Southern Asia
Philippines                                         Asia  South-
Eastern Asia
Qatar                                               Asia
Western Asia
Republic of Korea                                   Asia
Eastern Asia
Saudi Arabia                                        Asia
Western Asia
Singapore                                           Asia  South-
Eastern Asia
Sri Lanka                                           Asia
Southern Asia
State of Palestine                                  Asia
Western Asia
Syrian Arab Republic                                Asia
Western Asia
Tajikistan                                          Asia
Central Asia
Thailand                                            Asia  South-
Eastern Asia
Turkey                                              Asia
Western Asia
```

```
Turkmenistan                                        Asia
Central Asia
United Arab Emirates                                Asia
Western Asia
Uzbekistan                                          Asia
Central Asia
Viet Nam                                            Asia  South-
Eastern Asia
Yemen                                               Asia
Western Asia

                                                           DevName
1980  \
Afghanistan                                     Developing regions
16
Armenia                                         Developing regions
0
Azerbaijan                                      Developing regions
0
Bahrain                                         Developing regions
0
Bangladesh                                      Developing regions
83
Bhutan                                          Developing regions
0
Brunei Darussalam                               Developing regions
79
Cambodia                                        Developing regions
12
China                                           Developing regions
5123
China, Hong Kong Special Administrative Region  Developing regions
0
China, Macao Special Administrative Region      Developing regions
0
Cyprus                                          Developing regions
132
Democratic People's Republic of Korea           Developing regions
1
Georgia                                         Developing regions
0
India                                           Developing regions
8880
Indonesia                                       Developing regions
186
Iran (Islamic Republic of)                      Developing regions
1172
Iraq                                            Developing regions
262
```

```
Israel                                          Developing regions
1403
Japan                                            Developed regions
701
Jordan                                          Developing regions
177
Kazakhstan                                      Developing regions
0
Kuwait                                          Developing regions
1
Kyrgyzstan                                      Developing regions
0
Lao People's Democratic Republic                Developing regions
11
Lebanon                                         Developing regions
1409
Malaysia                                        Developing regions
786
Maldives                                        Developing regions
0
Mongolia                                        Developing regions
0
Myanmar                                         Developing regions
80
Nepal                                           Developing regions
1
Oman                                            Developing regions
0
Pakistan                                        Developing regions
978
Philippines                                     Developing regions
6051
Qatar                                           Developing regions
0
Republic of Korea                               Developing regions
1011
Saudi Arabia                                    Developing regions
0
Singapore                                       Developing regions
241
Sri Lanka                                       Developing regions
185
State of Palestine                              Developing regions
0
Syrian Arab Republic                            Developing regions
315
Tajikistan                                      Developing regions
0
Thailand                                        Developing regions
```

```
56
Turkey                                          Developing regions
481
Turkmenistan                                    Developing regions
0
United Arab Emirates                            Developing regions
0
Uzbekistan                                      Developing regions
0
Viet Nam                                        Developing regions
1191
Yemen                                           Developing regions
1

                                                1981  1982  1983  1984
1985  \
Afghanistan                                       39    39    47    71
340
Armenia                                            0     0     0     0
0
Azerbaijan                                         0     0     0     0
0
Bahrain                                            2     1     1     1
3
Bangladesh                                        84    86    81    98
92
Bhutan                                             0     0     0     1
0
Brunei Darussalam                                  6     8     2     2
4
Cambodia                                          19    26    33    10
7
China                                           6682  3308  1863  1527
1816
China, Hong Kong Special Administrative Region     0     0     0     0
0
China, Macao Special Administrative Region         0     0     0     0
0
Cyprus                                           128    84    46    46
43
Democratic People's Republic of Korea              1     3     1     4
3
Georgia                                            0     0     0     0
0
India                                           8670  8147  7338  5704
4211
Indonesia                                        178   252   115   123
100
Iran (Islamic Republic of)                      1429  1822  1592  1977
```

```
1648
Iraq                                             245   260   380   428
231
Israel                                          1711  1334   541   446
680
Japan                                            756   598   309   246
198
Jordan                                           160   155   113   102
179
Kazakhstan                                         0     0     0     0
0
Kuwait                                             0     8     2     1
4
Kyrgyzstan                                         0     0     0     0
0
Lao People's Democratic Republic                   6    16    16     7
17
Lebanon                                         1119  1159   789  1253
1683
Malaysia                                         816   813   448   384
374
Maldives                                           0     0     1     0
0
Mongolia                                           0     0     0     0
0
Myanmar                                           62    46    31    41
23
Nepal                                              1     6     1     2
4
Oman                                               0     0     8     0
0
Pakistan                                         972  1201   900   668
514
Philippines                                     5921  5249  4562  3801
3150
Qatar                                              0     0     0     0
0
Republic of Korea                               1456  1572  1081   847
962
Saudi Arabia                                       0     1     4     1
2
Singapore                                        301   337   169   128
139
Sri Lanka                                        371   290   197  1086
845
State of Palestine                                 0     0     0     0
0
Syrian Arab Republic                             419   409   269   264
385
```

```
Tajikistan                                         0     0     0     0
0
Thailand                                          53   113    65    82
66
Turkey                                           874   706   280   338
202
Turkmenistan                                       0     0     0     0
0
United Arab Emirates                               2     2     1     2
0
Uzbekistan                                         0     0     0     0
0
Viet Nam                                        1829  2162  3404  7583
5907
Yemen                                              2     1     6     0
18

                                                1986  ...   2005
2006  \
Afghanistan                                      496  ...   3436
3009
Armenia                                            0  ...    224
218
Azerbaijan                                         0  ...    359
236
Bahrain                                            0  ...     12
12
Bangladesh                                       486  ...   4171
4014
Bhutan                                             0  ...      5
10
Brunei Darussalam                                 12  ...      4
5
Cambodia                                           8  ...    370
529
China                                           1960  ...  42584
33518
China, Hong Kong Special Administrative Region     0  ...    729
712
China, Macao Special Administrative Region         0  ...     21
32
Cyprus                                            48  ...      7
9
Democratic People's Republic of Korea              0  ...     14
10
Georgia                                            0  ...    114
125
India                                           7150  ...  36210
33848
```

```
Indonesia                                        127  ...    632
613
Iran (Islamic Republic of)                      1794  ...   5837
7480
Iraq                                             265  ...   2226
1788
Israel                                          1212  ...   2446
2625
Japan                                            248  ...   1067
1212
Jordan                                           181  ...   1940
1827
Kazakhstan                                         0  ...    506
408
Kuwait                                             4  ...     66
35
Kyrgyzstan                                         0  ...    173
161
Lao People's Democratic Republic                  21  ...     42
74
Lebanon                                         2576  ...   3709
3802
Malaysia                                         425  ...    593
580
Maldives                                           0  ...      0
0
Mongolia                                           0  ...     59
64
Myanmar                                           18  ...    210
953
Nepal                                             13  ...    607
540
Oman                                               0  ...     14
18
Pakistan                                         691  ...  14314
13127
Philippines                                     4166  ...  18139
18400
Qatar                                              1  ...     11
2
Republic of Korea                               1208  ...   5832
6215
Saudi Arabia                                       5  ...    198
252
Singapore                                        205  ...    392
298
Sri Lanka                                       1838  ...   4930
4714
State of Palestine                                 0  ...    453
```

```
627
Syrian Arab Republic                             493  ...   1458
1145
Tajikistan                                         0  ...     85
46
Thailand                                          78  ...    575
500
Turkey                                           257  ...   2065
1638
Turkmenistan                                       0  ...     40
26
United Arab Emirates                               5  ...     31
42
Uzbekistan                                         0  ...    330
262
Viet Nam                                        2741  ...   1852
3153
Yemen                                              7  ...    161
140

                                                 2007   2008   2009
2010  \
Afghanistan                                      2652   2111   1746
1758
Armenia                                           198    205    267
252
Azerbaijan                                        203    125    165
209
Bahrain                                            22      9     35
28
Bangladesh                                       2897   2939   2104
4721
Bhutan                                              7     36    865
1464
Brunei Darussalam                                  11     10      5
12
Cambodia                                          460    354    203
200
China                                           27642  30037  29622
30391
China, Hong Kong Special Administrative Region    674    897    657
623
China, Macao Special Administrative Region         16     12     21
21
Cyprus                                              4      7      6
18
Democratic People's Republic of Korea               7     19     11
45
Georgia                                           132    112    128
```

```
126
India                                                 28742  28261  29456
34235
Indonesia                                               657    661    504
712
Iran (Islamic Republic of)                             6974   6475   6580
7477
Iraq                                                   2406   3543   5450
5941
Israel                                                 2401   2562   2316
2755
Japan                                                  1250   1284   1194
1168
Jordan                                                 1421   1581   1235
1831
Kazakhstan                                              436    394    431
377
Kuwait                                                   62     53     68
67
Kyrgyzstan                                              135    168    173
157
Lao People's Democratic Republic                         53     32     39
54
Lebanon                                                3467   3566   3077
3432
Malaysia                                                600    658    640
802
Maldives                                                  2      1      7
4
Mongolia                                                 82     59    118
169
Myanmar                                                1887    975   1153
556
Nepal                                                   511    581    561
1392
Oman                                                     16     10      7
14
Pakistan                                              10124   8994   7217
6811
Philippines                                           19837  24887  28573
38617
Qatar                                                     5      9      6
18
Republic of Korea                                      5920   7294   5874
5537
Saudi Arabia                                            188    249    246
330
Singapore                                               690    734    366
805
```

```
Sri Lanka                                        4123   4756   4547
4422
State of Palestine                                441    481    400
654
Syrian Arab Republic                             1056    919    917
1039
Tajikistan                                         44     15     50
52
Thailand                                          487    519    512
499
Turkey                                           1463   1122   1238
1492
Turkmenistan                                       37     13     20
30
United Arab Emirates                               37     33     37
86
Uzbekistan                                        284    215    288
289
Viet Nam                                         2574   1784   2171
1942
Yemen                                             122    133    128
211

                                                 2011   2012   2013
Total
Afghanistan                                      2203   2635   2004
58639
Armenia                                           236    258    207
3310
Azerbaijan                                        138    161     57
2649
Bahrain                                            21     39     32
475
Bangladesh                                       2694   2640   3789
65568
Bhutan                                           1879   1075    487
5876
Brunei Darussalam                                   6      3      6
600
Cambodia                                          196    233    288
6538
China                                           28502  33024  34129
659962
China, Hong Kong Special Administrative Region    591    728    774
9327
China, Macao Special Administrative Region         13     33     29
284
Cyprus                                              6     12     16
1126
Democratic People's Republic of Korea              97     66     17
```

```
388
Georgia                                            139    147    125
2068
India                                            27509  30933  33087
691904
Indonesia                                          390    395    387
13150
Iran (Islamic Republic of)                        7479   7534  11291
175923
Iraq                                              6196   4041   4918
69789
Israel                                            1970   2134   1945
66508
Japan                                             1265   1214    982
27707
Jordan                                            1635   1206   1255
35406
Kazakhstan                                         381    462    348
8490
Kuwait                                              58     73     48
2025
Kyrgyzstan                                         159    278    123
2353
Lao People's Democratic Republic                    22     25     15
1089
Lebanon                                           3072   1614   2172
115359
Malaysia                                           409    358    204
24417
Maldives                                             3      1      1
30
Mongolia                                           103     68     99
952
Myanmar                                            368    193    262
9245
Nepal                                             1129   1185   1308
10222
Oman                                                10     13     11
224
Pakistan                                          7468  11227  12603
241600
Philippines                                      36765  34315  29544
511391
Qatar                                                3     14      6
157
Republic of Korea                                 4588   5316   4509
142581
Saudi Arabia                                       278    286    267
3425
```

```
Singapore                                         219    146    141
14579
Sri Lanka                                        3309   3338   2394
148358
State of Palestine                                555    533    462
6512
Syrian Arab Republic                             1005    650   1009
31485
Tajikistan                                         47     34     39
503
Thailand                                          396    296    400
9174
Turkey                                           1257   1068    729
31781
Turkmenistan                                       20     20     14
310
United Arab Emirates                               60     54     46
836
Uzbekistan                                        162    235    167
3368
Viet Nam                                         1723   1731   2112
97146
Yemen                                             160    174    217
2985

[49 rows x 38 columns]

# we can pass multiple criteria in the same line.
# let's filter for AreaNAme = Asia and RegName = Southern Asia

df_can[(df_can['Continent']=='Asia') & (df_can['Region']=='Southern
Asia')]

# note: When using 'and' and 'or' operators, pandas requires we use
'&' and '|' instead of 'and' and 'or'
# don't forget to enclose the two conditions in parentheses

                           Continent         Region
DevName  1980  \
Afghanistan                     Asia  Southern Asia  Developing
regions    16
Bangladesh                      Asia  Southern Asia  Developing
regions    83
Bhutan                          Asia  Southern Asia  Developing
regions     0
India                           Asia  Southern Asia  Developing
regions  8880
Iran (Islamic Republic of)      Asia  Southern Asia  Developing
regions  1172
Maldives                        Asia  Southern Asia  Developing
```

```
regions     0
Nepal                           Asia  Southern Asia  Developing
regions     1
Pakistan                        Asia  Southern Asia  Developing
regions   978
Sri Lanka                       Asia  Southern Asia  Developing
regions   185

                            1981  1982  1983  1984  1985  1986  ...
2005  \
Afghanistan                   39    39    47    71   340   496  ...
3436
Bangladesh                    84    86    81    98    92   486  ...
4171
Bhutan                         0     0     0     1     0     0  ...
5
India                       8670  8147  7338  5704  4211  7150  ...
36210
Iran (Islamic Republic of)  1429  1822  1592  1977  1648  1794  ...
5837
Maldives                       0     0     1     0     0     0  ...
0
Nepal                          1     6     1     2     4    13  ...
607
Pakistan                     972  1201   900   668   514   691  ...
14314
Sri Lanka                    371   290   197  1086   845  1838  ...
4930

                             2006   2007   2008   2009   2010   2011
2012  \
Afghanistan                  3009   2652   2111   1746   1758   2203
2635
Bangladesh                   4014   2897   2939   2104   4721   2694
2640
Bhutan                         10      7     36    865   1464   1879
1075
India                       33848  28742  28261  29456  34235  27509
30933
Iran (Islamic Republic of)   7480   6974   6475   6580   7477   7479
7534
Maldives                        0      2      1      7      4      3
1
Nepal                         540    511    581    561   1392   1129
1185
Pakistan                    13127  10124   8994   7217   6811   7468
11227
Sri Lanka                    4714   4123   4756   4547   4422   3309
3338
```

```
                             2013   Total
Afghanistan                  2004   58639
Bangladesh                   3789   65568
Bhutan                        487    5876
India                       33087  691904
Iran (Islamic Republic of)  11291  175923
Maldives                        1      30
Nepal                        1308   10222
Pakistan                    12603  241600
Sri Lanka                    2394  148358

[9 rows x 38 columns]
```

**Exercise:** Fetch the data where AreaName is 'Africa' and RegName is 'Southern Africa'. Display the dataframe and find out how many instances are there?

```
df_can[(df_can['Continent']=='Africa') & (df_can['Region']=='Southern
Africa')]

             Continent           Region             DevName  1980
1981  1982  \
South Africa    Africa  Southern Africa  Developing regions  1026
1118   781
Botswana        Africa  Southern Africa  Developing regions    10
1     3
Namibia         Africa  Southern Africa  Developing regions     0
5     5
Lesotho         Africa  Southern Africa  Developing regions     1
1     1
Swaziland       Africa  Southern Africa  Developing regions     4
1     1

              1983  1984  1985  1986  ...  2005  2006  2007  2008
2009  2010  \
South Africa   379   271   310   718  ...   988  1111  1200  1123
1188  1238
Botswana         3     7     4     2  ...     7    11     8    28
15    42
Namibia          3     2     1     1  ...     6    19    13    26
14    16
Lesotho          2     7     5     3  ...     4     0     4     1
8     7
Swaziland        0    10     7     1  ...     7     7     5     6
10     3

              2011  2012  2013  Total
South Africa   959  1243  1240  40568
Botswana        53    64    76    396
Namibia         23    24    83    320
```

```
Lesotho          1     0     6    107
Swaziland       13    17    39    188

[5 rows x 38 columns]
```

## Sorting Values of a Dataframe or Series

You can use the `sort_values()` function is used to sort a DataFrame or a Series based on one or more columns. You to specify the column(s) by which you want to sort and the order (ascending or descending). Below is the syntax to use it:- `df.sort_values(col_name, axis=0, ascending=True, inplace=False, ignore_index=False)` col_nam - the column(s) to sort by.  axis - axis along which to sort. 0 for sorting by rows (default) and 1 for sorting by columns. ascending - to sort in ascending order (True, default) or descending order (False). inplace - to perform the sorting operation in-place (True) or return a sorted copy (False, default). ignore_index - to reset the index after sorting (True) or keep the original index values (False, default).

Let's sort out dataframe df_can on 'Total' column, in descending order to find out the top 5 countries that contributed the most to immigration to Canada.

```
df_can.sort_values(by='Total', ascending=False, axis=0, inplace=True)
top_5 = df_can.head(5)
top_5

                                                   Continent  \
India                                                   Asia   
China                                                   Asia   
United Kingdom of Great Britain and Northern Ir...    Europe   
Philippines                                             Asia   
Pakistan                                                Asia   

                                                                Region
\
India                                                    Southern Asia

China                                                     Eastern Asia

United Kingdom of Great Britain and Northern Ir...     Northern Europe

Philippines                                         South-Eastern Asia

Pakistan                                                 Southern Asia


                                                               DevName
1980  \
India                                               Developing regions
8880   
China                                               Developing regions
5123   
```

```
United Kingdom of Great Britain and Northern Ir...   Developed regions
22045
Philippines                                         Developing regions
6051
Pakistan                                            Developing regions
978

                                                     1981   1982
1983  \
India                                                8670   8147
7338
China                                                6682   3308
1863
United Kingdom of Great Britain and Northern Ir...  24796  20620
10015
Philippines                                          5921   5249
4562
Pakistan                                              972   1201
900

                                                     1984  1985  1986
...  \
India                                                5704  4211  7150
...
China                                                1527  1816  1960
...
United Kingdom of Great Britain and Northern Ir...  10170  9564  9470
...
Philippines                                          3801  3150  4166
...
Pakistan                                              668   514   691
...

                                                     2005   2006
2007  \
India                                               36210  33848
28742
China                                               42584  33518
27642
United Kingdom of Great Britain and Northern Ir...   7258   7140
8216
Philippines                                         18139  18400
19837
Pakistan                                            14314  13127
10124

                                                     2008   2009
2010  \
India                                               28261  29456
34235
```

```
China                                               30037  29622
30391
United Kingdom of Great Britain and Northern Ir...   8979   8876
8724
Philippines                                         24887  28573
38617
Pakistan                                             8994   7217
6811

                                                     2011   2012
2013  \
India                                               27509  30933
33087
China                                               28502  33024
34129
United Kingdom of Great Britain and Northern Ir...   6204   6195
5827
Philippines                                         36765  34315
29544
Pakistan                                             7468  11227
12603

                                                     Total
India                                               691904
China                                               659962
United Kingdom of Great Britain and Northern Ir...  551500
Philippines                                         511391
Pakistan                                            241600

[5 rows x 38 columns]
```

**Exercise:** Find out top 3 countries that contributes the most to immigration to Canda in the year 2010. Display the country names with the immigrant count in this year

```
df_can.sort_values(by='2010', ascending=False, axis=0, inplace=True)
top3_2010 = df_can['2010'].head(3)
top3_2010

Philippines    38617
India          34235
China          30391
Name: 2010, dtype: int64
```

Congratulations! you have learned how to wrangle data with Pandas. You will be using alot of these commands to preprocess the data before its can be used for data visualization.

Thank you for completing this lab!

## Author

Alex Aklson

### Other Contributors

Jay Rajasekharan, Ehsan M. Kermani, Slobodan Markovic, Weiqing Wang, Dr. Pooja

## Change Log

| Date (YYYY-MM-DD) | Version | Changed By | Change Description |
|---|---|---|---|
| 2023-06-08 | 2.5 | Dr. Pooja | Separated from original lab |
| 2021-05-29 | 2.4 | Weiqing Wang | Fixed typos and code smells. |
| 2021-01-20 | 2.3 | Lakshmi Holla | Changed TOC cell markdown |
| 2020-11-20 | 2.2 | Lakshmi Holla | Changed IBM box URL |
| 2020-11-03 | 2.1 | Lakshmi Holla | Changed URL and info method |
| 2020-08-27 | 2.0 | Lavanya | Moved Lab to course repo in GitLab |