Hands-on Activity 12.1, CPE311, Anton Paala

May 24, 2025

0.1 Hands-on Activity 12.1 CPE311-CPE22S3, Anton Paala

[13]:
```python
import pandas as pd

df = pd.
 ↪read_csv('Indicator_11_1_Physical_Risks_Climate_related_disasters_frequency_721256391239001
 ↪csv')
df.head()

# Loading the dataset into a dataframe
```

[13]:
```
   ObjectId                      Country ISO2 ISO3  \
0         1  Afghanistan, Islamic Rep. of   AF  AFG
1         2  Afghanistan, Islamic Rep. of   AF  AFG
2         3  Afghanistan, Islamic Rep. of   AF  AFG
3         4  Afghanistan, Islamic Rep. of   AF  AFG
4         5  Afghanistan, Islamic Rep. of   AF  AFG

                                           Indicator       Unit  \
0  Climate related disasters frequency, Number of…  Number of
1  Climate related disasters frequency, Number of…  Number of
2  Climate related disasters frequency, Number of…  Number of
3  Climate related disasters frequency, Number of…  Number of
4  Climate related disasters frequency, Number of…  Number of

                                             Source CTS Code  \
0  The Emergency Events Database (EM-DAT) , Centr…     ECCD
1  The Emergency Events Database (EM-DAT) , Centr…     ECCD
2  The Emergency Events Database (EM-DAT) , Centr…     ECCD
3  The Emergency Events Database (EM-DAT) , Centr…     ECCD
4  The Emergency Events Database (EM-DAT) , Centr…     ECCD

                              CTS Name  \
0  Climate Related Disasters Frequency
1  Climate Related Disasters Frequency
2  Climate Related Disasters Frequency
3  Climate Related Disasters Frequency
4  Climate Related Disasters Frequency
```

```
                                   CTS Full Descriptor  …  2015  2016  2017  \
0  Environment, Climate Change, Adaptation, Clima…  …   NaN   NaN   NaN
1  Environment, Climate Change, Adaptation, Clima…  …   NaN   NaN   NaN
2  Environment, Climate Change, Adaptation, Clima…  …   1.0   4.0   1.0
3  Environment, Climate Change, Adaptation, Clima…  …   4.0   NaN   2.0
4  Environment, Climate Change, Adaptation, Clima…  …   NaN   NaN   2.0

   2018  2019  2020  2021  2022  2023  2024
0   1.0   NaN   NaN   1.0   NaN   NaN   NaN
1   NaN   NaN   NaN   NaN   NaN   1.0   1.0
2   3.0   6.0   5.0   2.0   5.0   2.0   5.0
3   1.0   1.0   1.0   1.0   1.0   NaN   2.0
4   NaN   NaN   1.0   NaN   NaN   NaN   NaN

[5 rows x 55 columns]
```

[14]:
```
df.info() # Getting information about the dataset; the datatypes, null values,␣
 ↪and more.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1972 entries, 0 to 1971
Data columns (total 55 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ObjectId              1972 non-null   int64
 1   Country               1972 non-null   object
 2   ISO2                  1920 non-null   object
 3   ISO3                  1972 non-null   object
 4   Indicator             1972 non-null   object
 5   Unit                  1972 non-null   object
 6   Source                1972 non-null   object
 7   CTS Code              986 non-null    object
 8   CTS Name              986 non-null    object
 9   CTS Full Descriptor   986 non-null    object
 10  1980                  238 non-null    float64
 11  1981                  262 non-null    float64
 12  1982                  278 non-null    float64
 13  1983                  378 non-null    float64
 14  1984                  272 non-null    float64
 15  1985                  286 non-null    float64
 16  1986                  262 non-null    float64
 17  1987                  364 non-null    float64
 18  1988                  364 non-null    float64
 19  1989                  310 non-null    float64
 20  1990                  370 non-null    float64
 21  1991                  362 non-null    float64
 22  1992                  318 non-null    float64
 23   1993                  404 non-null    float64
 24   1994                  406 non-null    float64
 25   1995                  418 non-null    float64
 26   1996                  414 non-null    float64
 27   1997                  502 non-null    float64
 28   1998                  510 non-null    float64
 29   1999                  598 non-null    float64
 30   2000                  604 non-null    float64
 31   2001                  594 non-null    float64
 32   2002                  660 non-null    float64
 33   2003                  578 non-null    float64
 34   2004                  544 non-null    float64
 35   2005                  678 non-null    float64
 36   2006                  526 non-null    float64
 37   2007                  642 non-null    float64
 38   2008                  572 non-null    float64
 39   2009                  580 non-null    float64
 40   2010                  682 non-null    float64
 41   2011                  498 non-null    float64
 42   2012                  596 non-null    float64
 43   2013                  514 non-null    float64
 44   2014                  472 non-null    float64
 45   2015                  614 non-null    float64
 46   2016                  512 non-null    float64
 47   2017                  598 non-null    float64
 48   2018                  574 non-null    float64
 49   2019                  608 non-null    float64
 50   2020                  610 non-null    float64
 51   2021                  640 non-null    float64
 52   2022                  688 non-null    float64
 53   2023                  718 non-null    float64
 54   2024                  618 non-null    float64
dtypes: float64(45), int64(1), object(9)
memory usage: 847.5+ KB
```

[15]:
```
dfcopy = df.copy() # Always save a backup for your dataframe to prevent
                   # changing the original dataset (and loading it again)
```

[16]:
```
df_filled = df.fillna(0) # Fill NaN values with zeroes so that they still get␣
 ↪values despite none
```

[17]:
```
df_filled.info() # Rechecking if the fill was successful
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1972 entries, 0 to 1971
Data columns (total 55 columns):
 #    Column                Non-Null Count  Dtype
---   ------                --------------  -----
```

```
0   ObjectId              1972 non-null   int64
1   Country               1972 non-null   object
2   ISO2                  1972 non-null   object
3   ISO3                  1972 non-null   object
4   Indicator             1972 non-null   object
5   Unit                  1972 non-null   object
6   Source                1972 non-null   object
7   CTS Code              1972 non-null   object
8   CTS Name              1972 non-null   object
9   CTS Full Descriptor   1972 non-null   object
10  1980                  1972 non-null   float64
11  1981                  1972 non-null   float64
12  1982                  1972 non-null   float64
13  1983                  1972 non-null   float64
14  1984                  1972 non-null   float64
15  1985                  1972 non-null   float64
16  1986                  1972 non-null   float64
17  1987                  1972 non-null   float64
18  1988                  1972 non-null   float64
19  1989                  1972 non-null   float64
20  1990                  1972 non-null   float64
21  1991                  1972 non-null   float64
22  1992                  1972 non-null   float64
23  1993                  1972 non-null   float64
24  1994                  1972 non-null   float64
25  1995                  1972 non-null   float64
26  1996                  1972 non-null   float64
27  1997                  1972 non-null   float64
28  1998                  1972 non-null   float64
29  1999                  1972 non-null   float64
30  2000                  1972 non-null   float64
31  2001                  1972 non-null   float64
32  2002                  1972 non-null   float64
33  2003                  1972 non-null   float64
34  2004                  1972 non-null   float64
35  2005                  1972 non-null   float64
36  2006                  1972 non-null   float64
37  2007                  1972 non-null   float64
38  2008                  1972 non-null   float64
39  2009                  1972 non-null   float64
40  2010                  1972 non-null   float64
41  2011                  1972 non-null   float64
42  2012                  1972 non-null   float64
43  2013                  1972 non-null   float64
44  2014                  1972 non-null   float64
45  2015                  1972 non-null   float64
46  2016                  1972 non-null   float64
47  2017                  1972 non-null   float64
```

```
 48   2018                  1972 non-null   float64
 49   2019                  1972 non-null   float64
 50   2020                  1972 non-null   float64
 51   2021                  1972 non-null   float64
 52   2022                  1972 non-null   float64
 53   2023                  1972 non-null   float64
 54   2024                  1972 non-null   float64
dtypes: float64(45), int64(1), object(9)
memory usage: 847.5+ KB
```

[19]:
```
df_filled.ISO2.unique() # Checking the unique values of each categorical column
```

[19]:
```
array(['AF', 'AL', 'DZ', 'AS', 'AO', 'AI', 'AG', 'AR', 'AM', 'AU', 'AT',
       'AZ', nan, 'BS', 'BD', 'BB', 'BY', 'BE', 'BZ', 'BJ', 'BM', 'BT',
       'BO', 'BA', 'BW', 'BR', 'VG', 'BN', 'BG', 'BF', 'BI', 'CV', 'KH',
       'CM', 'CA', 'KY', 'CF', 'TD', 'CL', 'HK', 'MO', 'CN', 'CO', 'KM',
       'CD', 'CG', 'CK', 'CR', 'CI', 'HR', 'CU', 'CY', 'CZ', 'DK', 'DJ',
       'DM', 'DO', 'EC', 'EG', 'SV', 'ER', 'EE', 'SZ', 'ET', 'FJ', 'FI',
       'FR', 'PF', 'GA', 'GM', 'GE', 'DE', 'GH', 'GR', 'GD', 'GU', 'GT',
       'GN', 'GW', 'GY', 'HT', 'HN', 'HU', 'IS', 'IN', 'ID', 'IR', 'IQ',
       'IE', 'IM', 'IL', 'IT', 'JM', 'JP', 'JO', 'KZ', 'KE', 'KI', 'KP',
       'KR', 'KW', 'KG', 'LA', 'LV', 'LB', 'LS', 'LR', 'LY', 'LT', 'LU',
       'MG', 'MW', 'MY', 'MV', 'ML', 'MT', 'MH', 'MR', 'MU', 'MX', 'FM',
       'MD', 'MN', 'ME', 'MS', 'MA', 'MZ', 'MM', 'NP', 'AN', 'NL', 'NC',
       'NZ', 'NI', 'NE', 'NG', 'MK', 'MP', 'NO', 'OM', 'PK', 'PW', 'PA',
       'PG', 'PY', 'PE', 'PH', 'PL', 'PT', 'QA', 'RO', 'RU', 'RW', 'SH',
       'WS', 'ST', 'SA', 'SN', 'CS', 'RS', 'SC', 'SL', 'SX', 'SK', 'SI',
       'SB', 'SO', 'ZA', 'SS', 'ES', 'LK', 'KN', 'LC', 'VC', 'SD', 'SR',
       'SE', 'CH', 'SY', 'TW', 'TJ', 'TZ', 'TH', 'TL', 'TG', 'TK', 'TO',
       'TT', 'TN', 'TR', 'TM', 'TC', 'TV', 'UG', 'UA', 'AE', 'GB', 'US',
       'VI', 'UY', 'UZ', 'VU', 'VE', 'VN', 'WF', 'PS', 'YE', 'ZM', 'ZW'],
      dtype=object)
```

[20]:
```
df_filled.ISO3.unique()
```

[20]:
```
array(['AFG', 'ALB', 'DZA', 'ASM', 'AGO', 'AIA', 'ATG', 'ARG', 'ARM',
       'AUS', 'AUT', 'AZE', 'AZO', 'BHS', 'BGD', 'BRB', 'BLR', 'BEL',
       'BLZ', 'BEN', 'BMU', 'BTN', 'BOL', 'BIH', 'BWA', 'BRA', 'VGB',
       'BRN', 'BGR', 'BFA', 'BDI', 'CPV', 'KHM', 'CMR', 'CAN', 'SPI',
       'CYM', 'CAF', 'TCD', 'CHL', 'HKG', 'MAC', 'CHN', 'COL', 'COM',
       'COD', 'COG', 'COK', 'CRI', 'CIV', 'HRV', 'CUB', 'CYP', 'CZE',
       'DNK', 'DJI', 'DMA', 'DOM', 'ECU', 'EGY', 'SLV', 'ERI', 'EST',
       'SWZ', 'ETH', 'FJI', 'FIN', 'FRA', 'PYF', 'GAB', 'GMB', 'GEO',
       'DEU', 'DDR', 'DFR', 'GHA', 'GRC', 'GRD', 'GUM', 'GTM', 'GIN',
       'GNB', 'GUY', 'HTI', 'HND', 'HUN', 'ISL', 'IND', 'IDN', 'IRN',
       'IRQ', 'IRL', 'IMN', 'ISR', 'ITA', 'JAM', 'JPN', 'JOR', 'KAZ',
       'KEN', 'KIR', 'PRK', 'KOR', 'KWT', 'KGZ', 'LAO', 'LVA', 'LBN',
       'LSO', 'LBR', 'LBY', 'LTU', 'LUX', 'MDG', 'MWI', 'MYS', 'MDV',
       'MLI', 'MLT', 'MHL', 'MRT', 'MUS', 'MEX', 'FSM', 'MDA', 'MNG',
       'MNE', 'MSR', 'MAR', 'MOZ', 'MMR', 'NAM', 'NPL', 'ANT', 'NLD',
       'NCL', 'NZL', 'NIC', 'NER', 'NGA', 'MKD', 'MNP', 'NOR', 'OMN',
       'PAK', 'PLW', 'PAN', 'PNG', 'PRY', 'PER', 'PHL', 'POL', 'PRT',
       'QAT', 'ROU', 'RUS', 'RWA', 'BLM', 'SHN', 'MAF', 'WSM', 'STP',
       'SAU', 'SEN', 'SCG', 'SRB', 'SYC', 'SLE', 'SXM', 'SVK', 'SVN',
       'SLB', 'SOM', 'ZAF', 'SSD', 'SUN', 'ESP', 'LKA', 'KNA', 'LCA',
       'VCT', 'SDN', 'SUR', 'SWE', 'CHE', 'SYR', 'TWN', 'TJK', 'TZA',
       'THA', 'TLS', 'TGO', 'TKL', 'TON', 'TTO', 'TUN', 'TUR', 'TKM',
       'TCA', 'TUV', 'UGA', 'UKR', 'ARE', 'GBR', 'USA', 'VIR', 'URY',
       'UZB', 'VUT', 'VEN', 'VNM', 'WLF', 'PSE', 'YEM', 'ZMB', 'ZWE'],
      dtype=object)
```

[23]:
```
df_filled.Country.unique() # There are a lot of countries inside this dataset.
                           # We should only fetch ASEAN countries.
```

[23]:
```
array(['Afghanistan, Islamic Rep. of', 'Albania', 'Algeria',
       'American Samoa', 'Angola', 'Anguilla', 'Antigua and Barbuda',
       'Argentina', 'Armenia, Rep. of', 'Australia', 'Austria',
       'Azerbaijan, Rep. of', 'Azores Island', 'Bahamas, The',
       'Bangladesh', 'Barbados', 'Belarus, Rep. of', 'Belgium', 'Belize',
       'Benin', 'Bermuda', 'Bhutan', 'Bolivia', 'Bosnia and Herzegovina',
       'Botswana', 'Brazil', 'British Virgin Islands',
       'Brunei Darussalam', 'Bulgaria', 'Burkina Faso', 'Burundi',
       'Cabo Verde', 'Cambodia', 'Cameroon', 'Canada', 'Canary Island',
       'Cayman Islands', 'Central African Rep.', 'Chad', 'Chile',
       'China, P.R.: Hong Kong', 'China, P.R.: Macao',
       'China, P.R.: Mainland', 'Colombia', 'Comoros, Union of the',
       'Congo, Dem. Rep. of the', 'Congo, Rep. of', 'Cook Islands',
       'Costa Rica', "Côte d'Ivoire", 'Croatia, Rep. of', 'Cuba',
       'Cyprus', 'Czech Rep.', 'Denmark', 'Djibouti', 'Dominica',
       'Dominican Rep.', 'Ecuador', 'Egypt, Arab Rep. of', 'El Salvador',
       'Eritrea, The State of', 'Estonia, Rep. of',
       'Eswatini, Kingdom of', 'Ethiopia, The Federal Dem. Rep. of',
       'Fiji, Rep. of', 'Finland', 'France', 'French Polynesia', 'Gabon',
       'Gambia, The', 'Georgia', 'Germany', 'Germany Dem Rep (former)',
       'Germany Fed Rep (former)', 'Ghana', 'Greece', 'Grenada', 'Guam',
       'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti',
       'Honduras', 'Hungary', 'Iceland', 'India', 'Indonesia',
       'Iran, Islamic Rep. of', 'Iraq', 'Ireland', 'Isle of Man',
       'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan',
       'Kazakhstan, Rep. of', 'Kenya', 'Kiribati',
       "Korea, Dem. People's Rep. of", 'Korea, Rep. of', 'Kuwait',
       'Kyrgyz Rep.', "Lao People's Dem. Rep.", 'Latvia', 'Lebanon',
       'Lesotho, Kingdom of', 'Liberia', 'Libya', 'Lithuania',
       'Luxembourg', 'Madagascar, Rep. of', 'Malawi', 'Malaysia',
       'Maldives', 'Mali', 'Malta', 'Marshall Islands, Rep. of the',
       'Mauritania, Islamic Rep. of', 'Mauritius', 'Mexico',
       'Micronesia, Federated States of', 'Moldova, Rep. of', 'Mongolia',
       'Montenegro', 'Montserrat', 'Morocco', 'Mozambique, Rep. of',
       'Myanmar', 'Namibia', 'Nepal', 'Netherlands Antilles',
       'Netherlands, The', 'New Caledonia', 'New Zealand', 'Nicaragua',
       'Niger', 'Nigeria', 'North Macedonia, Republic of ',
       'Northern Mariana Islands', 'Norway', 'Oman', 'Pakistan',
       'Palau, Rep. of', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru',
       'Philippines', 'Poland, Rep. of', 'Portugal', 'Qatar', 'Romania',
       'Russian Federation', 'Rwanda', 'Saint Barthélemy', 'Saint Helena',
       'Saint Martin (French Part)', 'Samoa',
       'São Tomé and Príncipe, Dem. Rep. of', 'Saudi Arabia', 'Senegal',
       'Serbia and Montenegro', 'Serbia, Rep. of', 'Seychelles',
       'Sierra Leone', 'Sint Maarten, Kingdom of the Netherlands',
       'Slovak Rep.', 'Slovenia, Rep. of', 'Solomon Islands', 'Somalia',
       'South Africa', 'South Sudan, Rep. of', 'Soviet Union (former)',
       'Spain', 'Sri Lanka', 'St. Kitts and Nevis', 'St. Lucia',
       'St. Vincent and the Grenadines', 'Sudan', 'Suriname', 'Sweden',
       'Switzerland', 'Syrian Arab Rep.', 'Taiwan Province of China',
       'Tajikistan, Rep. of', 'Tanzania, United Rep. of', 'Thailand',
       'Timor-Leste, Dem. Rep. of', 'Togo', 'Tokelau', 'Tonga',
       'Trinidad and Tobago', 'Tunisia', 'Türkiye, Rep. of',
       'Turkmenistan', 'Turks and Caicos Islands', 'Tuvalu', 'Uganda',
       'Ukraine', 'United Arab Emirates', 'United Kingdom',
       'United States', 'United States Virgin Islands', 'Uruguay',
       'Uzbekistan, Rep. of', 'Vanuatu', 'Venezuela, Rep. Bolivariana de',
       'Vietnam', 'Wallis and Futuna Islands', 'West Bank and Gaza',
       'Yemen, Rep. of', 'Zambia', 'Zimbabwe'], dtype=object)
```

[24]:
```
asean_countries = ['Brunei Darussalam',
                   'Cambodia',
                   'Indonesia',
                   "Lao People's Dem. Rep.",
                   'Malaysia', 'Myanmar', 'Philippines',
                   'Thailand', 'Timor-Leste, Dem. Rep. of', 'Vietnam']
# Putting the ASEAN countries in a list so that we could just call it in the␣
 ↪query
dfasean = df_filled.query('Country == @asean_countries') # Call all ASEAN␣
 ↪Countries in the dataset
```

[27]:
```
dfasean.Country.unique() # Checking if the query was successful
```

[27]:
```
array(['Brunei Darussalam', 'Cambodia', 'Indonesia',
       "Lao People's Dem. Rep.", 'Malaysia', 'Myanmar', 'Philippines',
       'Thailand', 'Timor-Leste, Dem. Rep. of', 'Vietnam'], dtype=object)
```

[37]:
```
dfasean.head() # Checking the dataset by looking at the first five observations.
```

```
[37]:      ObjectId            Country ISO2 ISO3  \
     258        259  Brunei Darussalam   BN  BRN
     259        260  Brunei Darussalam   BN  BRN
     260        261  Brunei Darussalam   BN  BRN
     261        262  Brunei Darussalam   BN  BRN
     298        299           Cambodia   KH  KHM

                                                Indicator       Unit  \
     258  Climate related disasters frequency, Number of…  Number of
     259  Climate related disasters frequency, Number of…  Number of
     260  Climate related disasters frequency, Number of…  Number of
     261  Climate related disasters frequency, Number of…  Number of
     298  Climate related disasters frequency, Number of…  Number of

                                                   Source CTS Code  \
     258  The Emergency Events Database (EM-DAT) , Centr…     ECCD
     259  The Emergency Events Database (EM-DAT) , Centr…     ECCD
     260  The Emergency Events Database (EM-DAT) , Centr…        0
     261  The Emergency Events Database (EM-DAT) , Centr…        0
     298  The Emergency Events Database (EM-DAT) , Centr…     ECCD

                                     CTS Name  \
     258  Climate Related Disasters Frequency
     259  Climate Related Disasters Frequency
     260                                    0
     261                                    0
     298  Climate Related Disasters Frequency

                                      CTS Full Descriptor  …  2015  2016  2017  \
     258  Environment, Climate Change, Adaptation, Clima…  …   0.0   0.0   0.0
     259  Environment, Climate Change, Adaptation, Clima…  …   0.0   0.0   0.0
     260                                                0  …   0.0   0.0   0.0
     261                                                0  …   0.0   0.0   0.0
     298  Environment, Climate Change, Adaptation, Clima…  …   0.0   1.0   0.0

          2018  2019  2020  2021  2022  2023  2024
     258   0.0   0.0   0.0   0.0   0.0   0.0   0.0
     259   0.0   0.0   0.0   0.0   0.0   0.0   0.0
     260   0.0   0.0   0.0   0.0   0.0   0.0   0.0
     261   0.0   0.0   0.0   0.0   0.0   0.0   0.0
     298   0.0   0.0   0.0   0.0   0.0   0.0   0.0

     [5 rows x 55 columns]
```

```
[36]: dfasean.Indicator.unique() # In this column, we can see a lot of indicators for␣
        ↪the dataset.
      # We can choose many of these for our analysis.
# For my analysis, I will take a look at the total number of Disasters␣
↪throughout 2015-2024
```

```
[36]: array(['Climate related disasters frequency, Number of Disasters: TOTAL',
       'Climate related disasters frequency, Number of Disasters: Wildfire',
       'Climate related disasters frequency, Number of People Affected: TOTAL',
       'Climate related disasters frequency, Number of People Affected:
Wildfire',
       'Climate related disasters frequency, Number of Disasters: Drought',
       'Climate related disasters frequency, Number of Disasters: Flood',
       'Climate related disasters frequency, Number of Disasters: Storm',
       'Climate related disasters frequency, Number of People Affected:
Drought',
       'Climate related disasters frequency, Number of People Affected: Flood',
       'Climate related disasters frequency, Number of People Affected: Storm',
       'Climate related disasters frequency, Number of Disasters: Landslide',
       'Climate related disasters frequency, Number of People Affected:
Landslide',
       'Climate related disasters frequency, Number of Disasters: Extreme
temperature',
       'Climate related disasters frequency, Number of People Affected: Extreme
temperature'],
      dtype=object)
```

```
[38]: df_total = dfasean.query("Indicator == 'Climate related disasters frequency,␣
↪Number of Disasters: TOTAL'")
# Filter out the indicator
```

```
[40]: df_total.Indicator.unique()
# Checking if the indicator got filtered out successfully.
```

```
[40]: array(['Climate related disasters frequency, Number of Disasters: TOTAL'],
      dtype=object)
```

```
[42]: df_total.info() # There are actually more unneccessary columns in the dataset.␣
↪For example, the ISO2 and ISO3,
                # which is just the abbrevations of the country.
                # The Unit column just has the value, "Number of" which doesn't␣
↪really produce insights.
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, 258 to 1928
Data columns (total 55 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ObjectId            10 non-null     int64
 1   Country             10 non-null     object
 2   ISO2                10 non-null     object
```

```
3   ISO3                 10 non-null     object
4   Indicator            10 non-null     object
5   Unit                 10 non-null     object
6   Source               10 non-null     object
7   CTS Code             10 non-null     object
8   CTS Name             10 non-null     object
9   CTS Full Descriptor  10 non-null     object
10  1980                 10 non-null     float64
11  1981                 10 non-null     float64
12  1982                 10 non-null     float64
13  1983                 10 non-null     float64
14  1984                 10 non-null     float64
15  1985                 10 non-null     float64
16  1986                 10 non-null     float64
17  1987                 10 non-null     float64
18  1988                 10 non-null     float64
19  1989                 10 non-null     float64
20  1990                 10 non-null     float64
21  1991                 10 non-null     float64
22  1992                 10 non-null     float64
23  1993                 10 non-null     float64
24  1994                 10 non-null     float64
25  1995                 10 non-null     float64
26  1996                 10 non-null     float64
27  1997                 10 non-null     float64
28  1998                 10 non-null     float64
29  1999                 10 non-null     float64
30  2000                 10 non-null     float64
31  2001                 10 non-null     float64
32  2002                 10 non-null     float64
33  2003                 10 non-null     float64
34  2004                 10 non-null     float64
35  2005                 10 non-null     float64
36  2006                 10 non-null     float64
37  2007                 10 non-null     float64
38  2008                 10 non-null     float64
39  2009                 10 non-null     float64
40  2010                 10 non-null     float64
41  2011                 10 non-null     float64
42  2012                 10 non-null     float64
43  2013                 10 non-null     float64
44  2014                 10 non-null     float64
45  2015                 10 non-null     float64
46  2016                 10 non-null     float64
47  2017                 10 non-null     float64
48  2018                 10 non-null     float64
49  2019                 10 non-null     float64
50  2020                 10 non-null     float64
```

```
 51   2021                    10 non-null     float64
 52   2022                    10 non-null     float64
 53   2023                    10 non-null     float64
 54   2024                    10 non-null     float64
dtypes: float64(45), int64(1), object(9)
memory usage: 4.4+ KB
```

[44]:
```
df_total['CTS Code'].unique() # Like this column, it only has ECCD.
```

[44]:
```
array(['ECCD'], dtype=object)
```

[46]:
```
df_total['CTS Name'].unique() # This one too
```

[46]:
```
array(['Climate Related Disasters Frequency'], dtype=object)
```

[53]:
```
df_total['CTS Full Descriptor'].unique() # This is just the description of the␣
  ↪CTS Code
```

[53]:
```
array(['Environment, Climate Change, Adaptation, Climate Related Disasters
Frequency'],
      dtype=object)
```

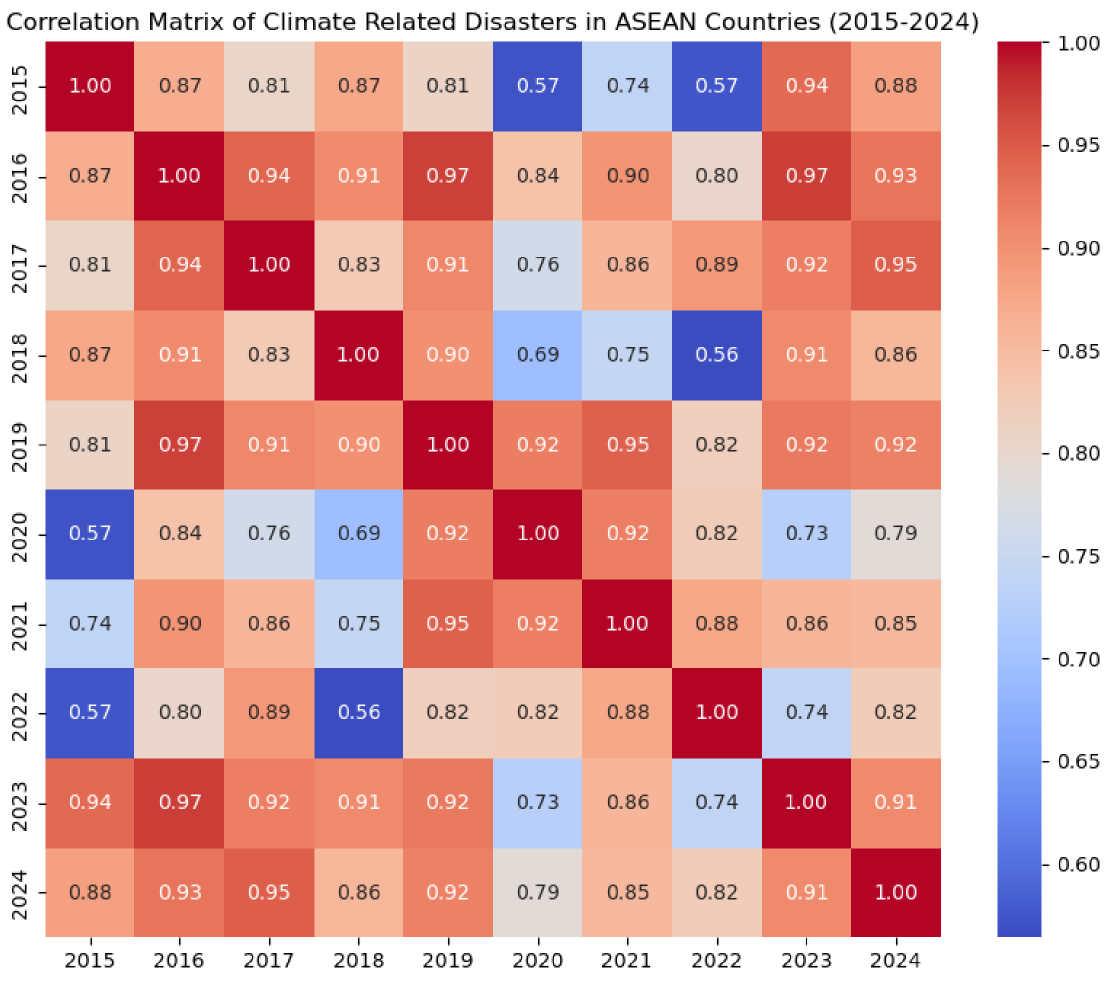
[54]:
```
df_total_copy = df_total.copy()
```

[77]:
```
df2 = df_total_copy.drop(columns=['ISO2', 'ISO3', 'Unit', 'Source', 'CTS Code',␣
  ↪'CTS Name', 'CTS Full Descriptor'])
```

[78]:
```
import matplotlib.pyplot as plt # Use the library to create vizualizations for␣
  ↪data analysis
import seaborn as sns
# Let's focus mainly on the total disasters from 2015 to 2024.
disasters = df2[['Country'] + [str(year) for year in range(2015, 2025)]] #␣
  ↪Select relevant columns for the years 2015 to 2024
correlation_matrix = disasters.corr(numeric_only=True)
# Set the size of the plot
plt.figure(figsize=(10, 8))

# Create a heatmap
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm',␣
  ↪square=True)

# Show the plot
plt.title('Correlation Matrix of Climate Related Disasters in ASEAN Countries␣
  ↪(2015-2024)')
plt.show()
```



	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
2015	1.00	0.87	0.81	0.87	0.81	0.57	0.74	0.57	0.94	0.88
2016	0.87	1.00	0.94	0.91	0.97	0.84	0.90	0.80	0.97	0.93
2017	0.81	0.94	1.00	0.83	0.91	0.76	0.86	0.89	0.92	0.95
2018	0.87	0.91	0.83	1.00	0.90	0.69	0.75	0.56	0.91	0.86
2019	0.81	0.97	0.91	0.90	1.00	0.92	0.95	0.82	0.92	0.92
2020	0.57	0.84	0.76	0.69	0.92	1.00	0.92	0.82	0.73	0.79
2021	0.74	0.90	0.86	0.75	0.95	0.92	1.00	0.88	0.86	0.85
2022	0.57	0.80	0.89	0.56	0.82	0.82	0.88	1.00	0.74	0.82
2023	0.94	0.97	0.92	0.91	0.92	0.73	0.86	0.74	1.00	0.91
2024	0.88	0.93	0.95	0.86	0.92	0.79	0.85	0.82	0.91	1.00

In this correlation matrix or heatmap, we can see the differences between the counts of total disasters in all ASEAN countries in different years. We can see that comparing 2015 to 2022 and 2020 shows that there were less climate disasters in the span of 5 years. The same goes for 2018 and 2022. Additionally, comparing 2016 to 2023 and 2019, shows that there were more consistent number of total disasters in those years. We will know more by looking at the line chart vizualization later.

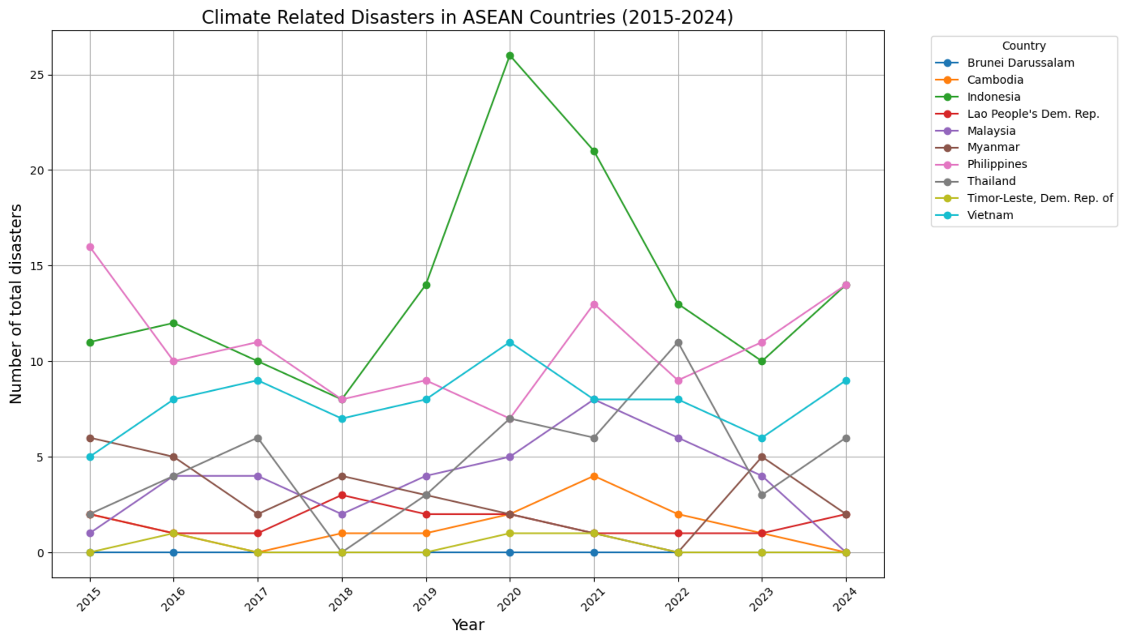
[79]:
```python
# Set 'Country' as the index
disasters.set_index('Country', inplace=True)
```

[83]:
```python
# Set the size of the plot
plt.figure(figsize=(14, 8))

# Plot each country's disaster data
for country in disasters.index:
    plt.plot(disasters.columns, disasters.loc[country], marker='o',␣
 ↪label=country)
```

```python
# Add titles and labels
plt.title('Climate Related Disasters in ASEAN Countries (2015-2024)',␣
 ↪fontsize=16)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Number of total disasters', fontsize=14)
plt.xticks(rotation=45)
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid()

# Show the plot
plt.tight_layout()
plt.show()
```



In this line chart, we can see that in the year 2020, Indonesia had the highest total of climate-related disasters compared to other ASEAN countries. We can also see that Philippines is the second highest total of climate-related disasters in 2015. Moreover, the total number of climate related disasters in countries such as Vietnam and Thailand show fluctations throughout the years.