

ENSF 592 Spring 2023 Project Report

June 13th, 2023

Tejpreet Bal

Soumini Mohandas

This program aims to determine the countries most affected by climate change and the rate at which they were affected historically. Datasets of three key indicators of climate change were used to build the data frame of this program, which include:

- Annual Surface Temperature Change [1]
- Land Cover and Land Cover Altering Indicator (Carbon Sequestration) [2]
- Climate-related Disaster Frequency [3]

To ensure data consistency, only the common years for which the data was present were merged into the data frame. The index used to identify countries is the ISO 3 code which is what the user is required to enter. The full names of the countries can be long, and tedious to type, which is why ISO 3 is used. For instance, the country Madagascar, Rep. of can be entered as MDG.

The program starts by asking for two inputs:

1. The ISO 3 code of the first country.
2. The ISO 3 code of the second country to compare with.

If an invalid input is entered, the user is prompted to re-enter a valid input. The user may also enter Exit to terminate the program, which is handled using `exit()` function in the `Sys` module. After the selection, the means for each parameter are printed for the 29-year period for both countries. Please refer to Figure 1 in Appendix A to view the screenshot when Canada and USA were selected as an example.

Next, the user is asked to select a parameter from the menu for which a pivot table is printed, and a plot generated. Temperature change is selected for USA and Canada as a continuance of the previous example, the results can be seen in Appendix B.

The final two prompts allow the user to enter Y for the aggregate statistics for the entire climate dataset, and how many countries experienced more than 20 climate-related disasters in a year. If the user pleases to skip any of these prompts, any key can be entered. The results of entering Y are shown in Appendix C. After these two prompts, an Excel file named `ClimateDataExport.xlsx` is exported.

To view how the specifications are met in this program, please refer to Appendix D.

References

- [1] Annual Surface Temperature Change, FAO Temperature Change, February 27, 2021. [Online]. Available: <https://climatedata.imf.org/datasets/4063314923d74187be9596f10d034914/explore>

- [2] Land Cover Accounts, FAO Land Cover; IMF staff calculations, October 26, 2021. [Online]. Available: <https://climatedata.imf.org/datasets/b1e6c0ea281f47b285addae0cbb28f4b/explore>

- [3] Climate-related Disasters Frequency, EM-DAT, CRED / UCLouvain, Brussels, Belgium, February 27, 2021. [Online]. Available: <https://climatedata.imf.org/datasets/b13b69ee0dde43a99c811f592af4e821/explore>

Appendix A – Initial Prompts

```
ENSF592 Climate Data Statistics
Influence of climate change on various indicators for countries across several years

Please enter the ISO3 code of the country (example: CAN, USA) or enter 'exit' to exit the program: CAN
Now pick a second country to compare with the first country.
Please enter the ISO3 code of the country (example: CAN, USA) or enter 'exit' to exit the program: usa

You chose the countries Canada and United States

The mean aggregate statistics for the 2 countries over a period of 29 years is shown below:
For Canada:
Temperature Change          1.158759
Land Cover Index            99.571262
Extreme Temperature         NaN
Wildfires                   NaN
Storms                      1.400000
Landslides                  NaN
Droughts                    NaN
Floods                      1.500000
Total Disasters              2.827586
Fractional Total Disaster    0.979499
Fractional Land Cover Index  0.513028

For United States:
Temperature Change          0.863345
Land Cover Index            99.168032
Extreme Temperature         NaN
Wildfires                   NaN
Storms                      14.827586
Landslides                  1.000000
Droughts                    1.181818
Floods                      4.571429
Total Disasters              23.172414
Fractional Total Disaster    8.066846
Fractional Land Cover Index  0.510900
```

Figure 1: Selecting Canada and USA using ISO 3

Appendix B: Pivot Table and Figure

```
Please select the parameter or climate change indicator you want to compare from the list.
Please enter the code for the parameter you want to choose from the list below or enter 'exit' to exit the program
0: Temperature Change
1: Land Cover Index
2: Total Disasters
0
You picked the Temperature Change parameter to compare the two countries.

A pivot table showing the Temperature Change statistics for the 2 countries for the period from 1992 to 2020 is displayed below and a corresponding bar graph is generated as well showing the trend:
Country  Canada  United States
Year
F1992    0.100    0.395
F1993    0.342    0.004
F1994    0.467    0.466
F1995    0.938    0.630
F1996   -0.125   -0.011
F1997    0.426    0.384
F1998    2.470    1.272
F1999    1.694    0.758
F2000    1.291    1.000
F2001    1.421    0.786
F2002    0.540    0.951
F2003    1.240    0.990
F2004    0.485    0.863
F2005    1.282    1.153
F2006    2.343    1.026
F2007    1.333    1.143
F2008    0.851    0.212
F2009    0.539    0.398
F2010    2.915    0.695
F2011    1.438    0.537
F2012    2.144    1.448
F2013    1.182    0.614
F2014    0.288    0.501
F2015    1.231    1.531
F2016    2.373    2.224
F2017    1.480    1.433
F2018    0.477    1.276
F2019    1.311    1.034
F2020    1.128    1.324
Plotting the data.
```

Figure 2: Pivot table showing yearly temperature change for Canada and USA

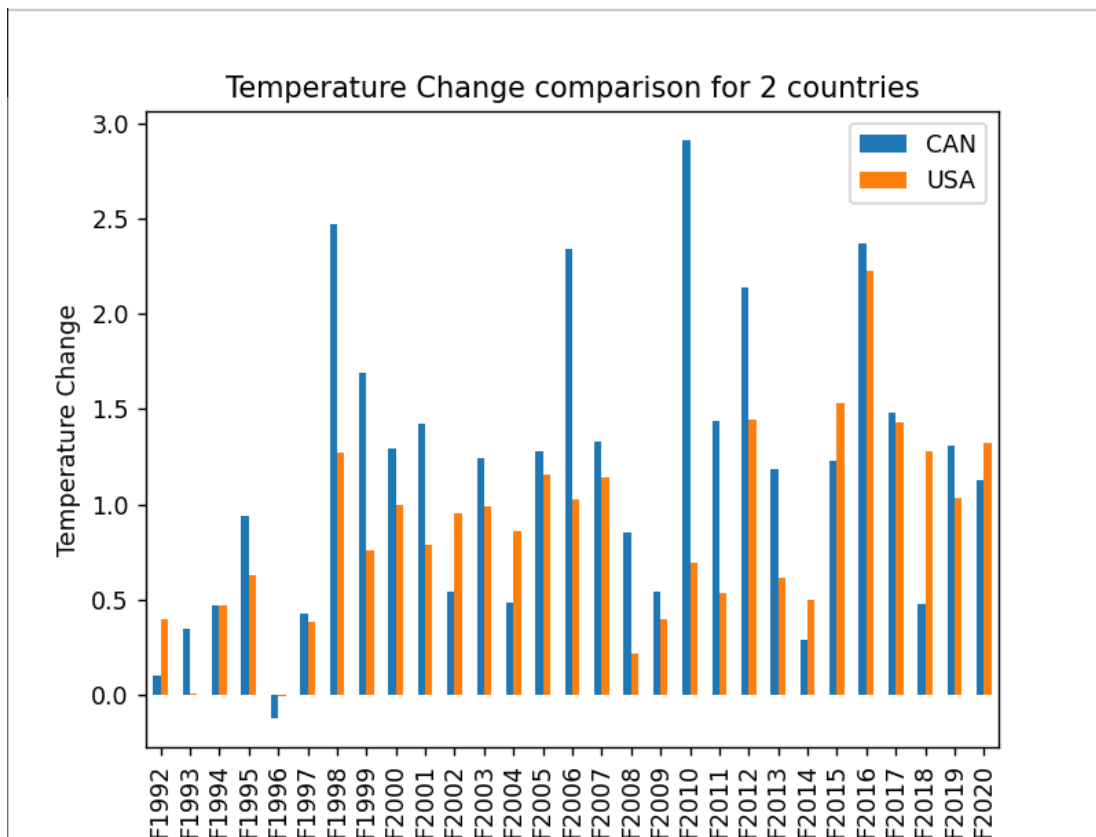


Figure 3: Plot generated using Matplotlib

Appendix C

```
Would you be interested in knowing the aggregate statistics for the entire climate data dataset?
Enter 'Y' for yes or enter any other key to skip this step and continue: y

The aggregate stats for the entire Climate Data dataset is:
  Temperature Change  Land Cover Index  Extreme Temperature  ...  Total Disasters  Fractional Total Disaster  Fractional Land Cover Index
count      5771.000000      5771.000000      5771.000000  ...  5771.000000      5771.000000      5771.000000
mean         0.855102        97.576986         0.000000  ...  1.511003         0.502513         0.502513
std         0.583697        30.235265         0.000000  ...  3.092509         1.050495         0.158677
min        -1.310000         0.000000         0.000000  ...  0.000000         0.000000         0.000000
25%         0.471000        95.276030         0.000000  ...  0.000000         0.000000         0.492700
50%         0.822000        99.693218         0.000000  ...  1.000000         0.272480         0.504119
75%         1.208500       100.570033         0.000000  ...  2.000000         0.606061         0.517393
max         3.691000       668.209843         0.000000  ...  34.000000        17.575758         3.669702

[8 rows x 11 columns]
```

Figure 4: Statistics of the entire climate data

```
Would you be interested in knowing how many countries experienced more than 20 climate-related disasters in a year?
Enter 'Y' for yes or enter any other key to skip this step and continue: Y

The countries (along with their ISO3 codes and year) that experienced more than 20 climate-related disasters a year are:
('CHN', 'China, P.R.: Mainland', 'F2000')
('CHN', 'China, P.R.: Mainland', 'F2001')
('CHN', 'China, P.R.: Mainland', 'F2002')
('CHN', 'China, P.R.: Mainland', 'F2005')
('CHN', 'China, P.R.: Mainland', 'F2006')
('CHN', 'China, P.R.: Mainland', 'F2008')
('CHN', 'China, P.R.: Mainland', 'F2009')
('CHN', 'China, P.R.: Mainland', 'F2010')
('CHN', 'China, P.R.: Mainland', 'F2012')
('CHN', 'China, P.R.: Mainland', 'F2013')
('CHN', 'China, P.R.: Mainland', 'F2014')
('CHN', 'China, P.R.: Mainland', 'F2015')
('CHN', 'China, P.R.: Mainland', 'F2016')
('CHN', 'China, P.R.: Mainland', 'F2017')
('IDN', 'Indonesia', 'F2020')
('IND', 'India', 'F2005')
('IND', 'India', 'F2018')
('PHL', 'Philippines', 'F2009')
('PHL', 'Philippines', 'F2011')
('USA', 'United States', 'F1992')
('USA', 'United States', 'F1993')
('USA', 'United States', 'F1997')
('USA', 'United States', 'F1998')
('USA', 'United States', 'F1999')
('USA', 'United States', 'F2000')
('USA', 'United States', 'F2001')
('USA', 'United States', 'F2002')
('USA', 'United States', 'F2003')
('USA', 'United States', 'F2006')
('USA', 'United States', 'F2007')
('USA', 'United States', 'F2008')
('USA', 'United States', 'F2011')
('USA', 'United States', 'F2012')
('USA', 'United States', 'F2013')
('USA', 'United States', 'F2015')
('USA', 'United States', 'F2016')
('USA', 'United States', 'F2017')
('USA', 'United States', 'F2020')
Please wait. The program is exporting the dataframe to an excel file.
Exporting completed. Thank you for your patience.
```

Figure 5: More than 20 climate-related disasters in a year

Appendix D – Specification Justification

Table 1: Table summarizing the compliance of the program with the given specifications

<p>Stage 1: Dataset Selection</p> <ul style="list-style-type: none"> ○ You must use at least three separate Excel sheets or files that can be related in some way. ○ Your final combined dataset (see next stage) must have at least ten columns and 200 rows. ○ You may edit the given datasets before you begin coding, but your program should not modify the Excel files directly. ○ You may not hard-code/copy-paste any information into your program except for the Excel column names. 	<ul style="list-style-type: none"> • 3 Excel file names: Annual_Surface_Temperature_Change.csv, Climate-related_Disasters_Frequency.csv, Land_Cover_Accounts.csv • The data frame has: [5771 rows x 12 columns] • The Excel files were not modified and nothing was hard coded.
<p>Stage 2: DataFrame Creation</p> <ul style="list-style-type: none"> ○ Import your chosen data into a Pandas DataFrame. ○ You must use at least two merge/join operations and you must delete any duplicated columns/rows that result from the merge. ○ You must create a hierarchical index of at least two levels (row or column). 	<ul style="list-style-type: none"> • After importing and formatting the data using Pandas, outer merge operations were done at line 73 in the climate_data.py file: <pre> 71 # All the above sub datasets of disaster are joined to create a dataset of disasters. 72 dis_1 = pd.merge(d_temp, d_wildfire, on = ['ISO3', 'Country', 'Year'], how = 'outer') 73 dis_2 = pd.merge(dis_1, d_landslide, on = ['ISO3', 'Country', 'Year'], how = 'outer') 74 dis_3 = pd.merge(dis_2, d_drought, on = ['ISO3', 'Country', 'Year'], how = 'outer') 75 dis_4 = pd.merge(dis_3, d_flood, on = ['ISO3', 'Country', 'Year'], how = 'outer') 76 dis_5 = pd.merge(dis_4, d_total, on = ['ISO3', 'Country', 'Year'], how = 'outer') 77 disaster_data = pd.merge(dis_5, d_total, on = ['ISO3', 'Country', 'Year'], how = 'outer') </pre> • Hierarchical index was created at line 144: • All data was sorted according to the country name and the years ranging from 1992 to 2020. <pre> 143 # Indexed and sorted the combined dataset 144 # A hierarchical indexing is created 145 climate_data = climate_data.set_index(['ISO3', 'Country', 'Year']) 146 climate_data = climate_data.sort_index() </pre> • Data was imported in the main function by calling the create_dataframe() function. • All null values were replaced with zeros.

<ul style="list-style-type: none"> ○ All data should be presented in the correctly sorted order, depending on the index. ○ You may not use global variables. You must import the data within your main function. ○ Remember to check for null values or data mismatches. 	
<p>Stage 3: User Entry</p> <ul style="list-style-type: none"> ○ Your application must return useful information. Design an interface that allows users to search based on some sort of criteria or keywords. ○ The user must provide at least two pieces of information/selection (e.g. "school name" and "grade"). ○ Give the user clear input instructions. If an invalid entry is given, use try/except statements to handle the error and continue to prompt for user input. ○ You must not hard-code any data values (the data within your spreadsheets could be changed!). ○ Any output information must 	<ul style="list-style-type: none"> ● Please see Appendices A, B and C for the user entry stage. ● For incorrect inputs, try/except is used to print the appropriate message and allow for re-entry. The input is case-insensitive, and any space characters before or after the code are trimmed. ● Nothing was hard coded, and the output is clearly defined.

	<p>be clearly defined using printed headers (DataFrame tables) or sentences (scalar values).</p>
<p>Stage 4: Analysis and Calculations</p> <ul style="list-style-type: none"> You may choose what data trends to present from your data. However, you must meet the following specifications. Use the describe method to print aggregate stats for the entire dataset. Add at least two columns to the combined dataset. Use an aggregation computation for a subset of the data. Use a masking operation. Use the groupby operation at least once. Create and print a pivot table. Include at least two user-defined functions or a class that contains two methods. 	<ul style="list-style-type: none"> The describe method was used at line 188: <pre> 187 188 agg_stats = df.dropna().describe() 189 return agg_stats </pre> <ul style="list-style-type: none"> Two columns were added at line 128 and below: <pre> 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 </pre> <ul style="list-style-type: none"> At line 218, aggregation computation for a subset of the data was done using .mean(). Masking operation was done at line 390: <pre> 388 # Aggregate stats for the 2 countries selected 389 print(f"The mean aggregate statistics for the 2 countries over a period of 29 years is") 390 country_yearly_disasters = masking_operation_subset_2(climate_data) 391 total_disasters_yearly = masking_operation_subset_1(climate_data) 392 df = analysis_part_1(climate_data, country_yearly_disasters, total_disasters_yearly) 393 country_stats = choice_country_1(df, iso3_1) 394 country_1_stats = country_1_agg_statistics(country_stats) 395 print(f"for {country_1}:\n{country_1_stats.to_string()}") </pre> <ul style="list-style-type: none"> Pivot table at line 269: <pre> 268 # Creation and printing of a pivot table 269 comparison_stats = two_country_stats.pivot_table(pa 270 return comparison_stats </pre> <ul style="list-style-type: none"> The two user defined function are called when the user enters two ISO 3 codes for comparison
<ul style="list-style-type: none"> Stage 5: Export and Matplotlib <ul style="list-style-type: none"> At the conclusion of your program, export your entire merged, hierarchical dataset to an Excel file in the working directory. Be sure to 	<ul style="list-style-type: none"> The Excel file is exported at the conclusion of the program and a plot can also be generated as shown Appendix B.

<p>include the index and header values. The TAs will use this to verify the structure of your dataset and your added columns.</p> <ul style="list-style-type: none">○ Use your data to create at least one plot using Matplotlib. Save the plot as a .png file and upload to the repository.	
--	--