```
% Richmond
```

```
%Distance
```

```
% Set up the Import Options and import the data
opts3 = spreadsheetImportOptions("NumVariables", 14);

% Specify sheet and range
opts3.Sheet = "dist_cepii";
opts3.DataRange = "A2:N50177";

% Specify column names and types
opts3.VariableNames = ["iso_o", "iso_d", "contig", "comlang_off",
"comlang_ethno", "colony", "comcol", "curcol", "col45", "smctry", "dist",
"distcap", "distw", "distwces"];
opts3.VariableTypes = ["categorical", "string", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double"];

% Specify variable properties
opts3 = setvaropts(opts3, "iso_d", "WhitespaceRule", "preserve");
opts3 = setvaropts(opts3, ["iso_o", "iso_d"], "EmptyFieldRule", "auto");

% Import the data
dist_cepii3 = readtable("/Users/richmondessieku/Desktop/FALL2023/Data/
dist_cepii.xls", opts3, "UseExcel", false);
dist_cepii2 = dist_cepii3
```

```
dist_cepii2 = 50176×14 table
```

...

	iso_o	iso_d	contig	comlang_off	comlang_ethno	colony	comcol
1	ABW	"ABW"	0	0	0	0	0
2	ABW	"AFG"	0	0	0	0	0
3	ABW	"AGO"	0	0	0	0	0
4	ABW	"AIA"	0	0	1	0	0
5	ABW	"ALB"	0	0	0	0	0
6	ABW	"AND"	0	1	0	0	0
7	ABW	"ANT"	0	1	1	0	1
8	ABW	"ARE"	0	0	0	0	0
9	ABW	"ARG"	0	1	0	0	0
10	ABW	"ARM"	0	0	0	0	0
11	ABW	"ATG"	0	0	1	0	0
12	ABW	"AUS"	0	0	1	0	0

	iso_o	iso_d	contig	comlang_off	comlang_ethno	colony	comcol
13	ABW	"AUT"	0	0	0	0	0
14	ABW	"AZE"	0	0	0	0	0
15	ABW	"BDI"	0	0	0	0	0
16	ABW	"BEL"	0	1	1	0	0
17	ABW	"BEN"	0	0	0	0	0
18	ABW	"BFA"	0	0	0	0	0
19	ABW	"BGD"	0	0	0	0	0
20	ABW	"BGR"	0	0	0	0	0
21	ABW	"BHR"	0	0	0	0	0
22	ABW	"BHS"	0	0	1	0	0
23	ABW	"BIH"	0	0	0	0	0
24	ABW	"BLR"	0	0	0	0	0
25	ABW	"BLZ"	0	1	1	0	0
26	ABW	"BMU"	0	0	1	0	0
27	ABW	"BOL"	0	1	0	0	0
28	ABW	"BRA"	0	0	0	0	0
29	ABW	"BRB"	0	0	1	0	0
30	ABW	"BRN"	0	0	0	0	0
31	ABW	"BTN"	0	0	0	0	0
32	ABW	"BWA"	0	0	1	0	0
33	ABW	"CAF"	0	0	0	0	0
34	ABW	"CAN"	0	0	1	0	0
35	ABW	"CCK"	0	0	1	0	0
36	ABW	"CHE"	0	0	0	0	0
37	ABW	"CHL"	0	1	0	0	0
38	ABW	"CHN"	0	0	0	0	0
39	ABW	"CIV"	0	0	0	0	0
40	ABW	"CMR"	0	0	1	0	0
41	ABW	"COG"	0	0	0	0	0
42	ABW	"COK"	0	0	1	0	0
43	ABW	"COL"	0	1	0	0	0
44	ABW	"COM"	0	0	0	0	0
45	ABW	"CPV"	0	0	0	0	0

	iso_o	iso_d	contig	comlang_off	comlang_ethno	colony	comcol
46	ABW	"CRI"	0	1	0	0	0
47	ABW	"CUB"	0	1	0	0	0
48	ABW	"CXR"	0	0	1	0	0
49	ABW	"CYM"	0	0	1	0	0
50	ABW	"CYP"	0	0	0	0	0
51	ABW	"CZE"	0	0	0	0	0
52	ABW	"DEU"	0	0	0	0	0
53	ABW	"DJI"	0	0	0	0	0
54	ABW	"DMA"	0	0	1	0	0
55	ABW	"DNK"	0	0	0	0	0
56	ABW	"DOM"	0	1	0	0	0
57	ABW	"DZA"	0	0	0	0	0
58	ABW	"ECU"	0	1	0	0	0
59	ABW	"EGY"	0	0	1	0	0
60	ABW	"ERI"	0	0	1	0	0
61	ABW	"ESH"	0	0	0	0	0
62	ABW	"ESP"	0	1	0	0	0
63	ABW	"EST"	0	0	0	0	0
64	ABW	"ETH"	0	0	0	0	0
65	ABW	"FIN"	0	0	0	0	0
66	ABW	"FJI"	0	0	1	0	0
67	ABW	"FLK"	0	0	1	0	0
68	ABW	"FRA"	0	0	0	0	0
69	ABW	"FRO"	0	0	0	0	0
70	ABW	"FSM"	0	0	1	0	0
71	ABW	"GAB"	0	0	0	0	0
72	ABW	"GBR"	0	0	1	0	0
73	ABW	"GEO"	0	0	0	0	0
74	ABW	"GHA"	0	0	1	0	0
75	ABW	"GIB"	0	1	1	0	0
76	ABW	"GIN"	0	0	0	0	0
77	ABW	"GLP"	0	0	0	0	0
78	ABW	"GMB"	0	0	1	0	0

	iso_o	iso_d	contig	comlang_off	comlang_ethno	colony	comcol
79	ABW	"GNB"	0	0	0	0	0
80	ABW	"GNQ"	0	1	0	0	0
81	ABW	"GRC"	0	0	0	0	0
82	ABW	"GRD"	0	0	1	0	0
83	ABW	"GRL"	0	0	0	0	0
84	ABW	"GTM"	0	1	0	0	0
85	ABW	"GUF"	0	0	0	0	0
86	ABW	"GUY"	0	0	1	0	0
87	ABW	"HKG"	0	0	1	0	0
88	ABW	"HND"	0	1	0	0	0
89	ABW	"HRV"	0	0	0	0	0
90	ABW	"HTI"	0	0	0	0	0
91	ABW	"HUN"	0	0	0	0	0
92	ABW	"IDN"	0	0	0	0	1
93	ABW	"IND"	0	0	1	0	0
94	ABW	"IRL"	0	0	1	0	0
95	ABW	"IRN"	0	0	0	0	0
96	ABW	"IRQ"	0	0	0	0	0
97	ABW	"ISL"	0	0	0	0	0
98	ABW	"ISR"	0	0	1	0	0
99	ABW	"ITA"	0	0	0	0	0
100	ABW	"JAM"	0	0	1	0	0

⋮

```
% Convert to output type
iso_o2 = dist_cepii3.iso_o;
iso_d2 = dist_cepii3.iso_d;
contig3 = dist_cepii3.contig;
comlang_off3 = dist_cepii3.comlang_off;
comlang_ethno3 = dist_cepii3.comlang_ethno;
colony3 = dist_cepii3.colony;
comcol3 = dist_cepii3.comcol;
curcol3 = dist_cepii3.curcol;
col453 = dist_cepii3.col45;
smctry3 = dist_cepii3.smctry;
dist2 = dist_cepii3.dist;
distcap3 = dist_cepii3.distcap;
distw3 = dist_cepii3.distw;
```

```
distwces3 = dist_cepii3.distwces;

% Select Countries

countries =
{'USA','NLD','JPN','DEU','GBR','IND','NOR','ITA','CAN','BEL','HUN','BRA','AU
S','ESP','POL'};
[~,idxO] = ismember(iso_o2, countries);
[~,idxD] = ismember(iso_d2, countries);

iso_sele = sort(iso_o2(idxO & idxD),1)
```

```
iso_sele = 225×1 categorical
AUS
AUS
AUS
AUS
AUS
AUS
AUS
AUS
AUS
AUS
     ⋮
```

```
iso_d_sel = sort(iso_d2(idxO & idxD),1)
```

```
iso_d_sel = 225×1 string
"AUS"
"AUS"
"AUS"
"AUS"
"AUS"
"AUS"
"AUS"
"AUS"
"AUS"
"AUS"
     ⋮
```

```
dist_sel = dist2(idxO & idxD)
```

```
dist_sel = 225×1
10^4 ×
    0.1043
    1.6760
    1.3372
    1.5587
    1.6563
    1.7699
    1.7011
    1.5796
    1.0435
    1.6333
     ⋮
```

```
matrix_dis = reshape(dist_sel,[],15)
```

```
matrix_dis = 15×15
10^4 ×
    0.1043    1.6760    1.3372    1.5587    1.6563    1.7699    1.7011    1.5796 ···
    1.6760    0.0068    0.9666    0.6032    0.0197    0.1317    0.0324    0.1130
    1.3372    0.9666    0.1097    0.8191    0.9848    0.8390    0.9502    1.0267
    1.5587    0.6032    0.8191    0.1188    0.6161    0.6040    0.5716    0.7131
    1.6563    0.0197    0.9848    0.6161    0.0225    0.1479    0.0495    0.0976
    1.7699    0.1317    0.8390    0.6040    0.1479    0.0268    0.1263    0.1978
    1.7011    0.0324    0.9502    0.5716    0.0495    0.1263    0.0186    0.1454
    1.5796    0.1130    1.0267    0.7131    0.0976    0.1978    0.1454    0.0115
    1.0435    0.6420    1.4441    1.1644    0.6230    0.7282    0.6721    0.5370
    1.6333    0.1175    0.9483    0.7089    0.1146    0.1367    0.1438    0.0811
      ⋮
```

```
% Export
```

```
% Set up the Import Options and import the data
opts = delimitedTextImportOptions("NumVariables", 15);

% Specify range and delimiter
opts.DataLines = [1, Inf];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["x0", "x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8",
"x9", "x10", "x11", "x12", "x13", "x14"];
opts.VariableTypes = ["double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, ["x0", "x1", "x2", "x3", "x4", "x5", "x6", "x7",
"x8", "x9", "x10", "x11", "x12", "x13", "x14"], "TrimNonNumeric", true);
opts = setvaropts(opts, ["x0", "x1", "x2", "x3", "x4", "x5", "x6", "x7",
"x8", "x9", "x10", "x11", "x12", "x13", "x14"], "ThousandsSeparator", ",");

% Import the data
Export_popo1 = readtable("/Users/richmondessieku/Desktop/FALL2023/Data/
Export_popo1.csv", opts);
Trade_data_r =Export_popo1
```

```
Trade_data_r = 15×15 table
                                                                    ...
```

	x0	x1	x2	x3	x4	x5	x6	x7
1	0	0	0	0	0	0	0	0
2	4.5201e+11	600818980	516074789	1.4466e+09	2.0240e+09	318249463	1.0298e+10	38434561
3	1.5309e+09	3.5388e+11	1.8422e+09	2.8742e+09	4.9020e+10	8.3493e+09	1.7806e+10	1.9245e+09
4	466879251	2.6404e+09	2.2025e+12	4.2299e+09	4.1237e+09	4.0569e+09	2.5442e+09	53535443
5	1.5897e+09	2.0435e+09	1.5944e+09	1.3175e+12	4.4926e+09	1.0607e+09	1.4954e+10	47349813
6	9.7881e+09	4.9419e+10	9.7568e+09	1.0731e+10	3.3867e+12	4.2847e+10	7.6269e+10	2.8144e+10
7	1.6940e+09	8.2890e+09	2.5881e+09	2.1556e+09	3.3785e+10	3.9615e+11	1.8754e+10	1.8832e+09
8	5.1796e+09	1.3637e+10	2.2267e+09	7.5076e+09	4.1598e+10	1.1115e+10	1.1840e+12	1.5535e+09
9	334964680	3.0014e+09	183319539	247869365	3.3478e+10	3.3136e+09	3.7146e+09	2.3203e+11
10	3.4631e+09	4.5674e+09	3.6759e+09	2.8114e+09	7.6572e+09	3.0850e+09	7.7906e+09	455983503
11	4.2345e+09	1.6977e+10	4.1711e+09	4.8900e+09	6.3620e+10	2.3373e+10	2.5686e+10	5.0342e+09
12	1.2143e+10	6.5535e+09	2.9546e+09	7.2427e+09	1.7575e+10	2.1808e+09	1.0718e+10	1.5862e+09
13	3.5134e+09	5.6697e+10	2.4901e+09	3.7312e+09	1.2148e+11	1.6463e+10	4.0192e+10	4.2401e+09
14	223722273	3.1286e+09	463463270	679275367	9.6435e+09	1.5163e+09	1.4512e+10	77029119
15	881528067	6.0967e+09	425326274	1.0561e+09	7.3513e+10	6.4370e+09	1.4568e+10	6.3820e+09

```
myMatrix_exp = table2array(Trade_data_r)
```

```
myMatrix_exp = 15×15
10^12 ×
         0         0         0         0         0         0         0         0···
    0.4520    0.0006    0.0005    0.0014    0.0020    0.0003    0.0103    0.0000
    0.0015    0.3539    0.0018    0.0029    0.0490    0.0083    0.0178    0.0019
    0.0005    0.0026    2.2025    0.0042    0.0041    0.0041    0.0025    0.0001
    0.0016    0.0020    0.0016    1.3175    0.0045    0.0011    0.0150    0.0000
    0.0098    0.0494    0.0098    0.0107    3.3867    0.0428    0.0763    0.0281
    0.0017    0.0083    0.0026    0.0022    0.0338    0.3962    0.0188    0.0019
    0.0052    0.0136    0.0022    0.0075    0.0416    0.0111    1.1840    0.0016
    0.0003    0.0030    0.0002    0.0002    0.0335    0.0033    0.0037    0.2320
    0.0035    0.0046    0.0037    0.0028    0.0077    0.0031    0.0078    0.0005
      ⋮
```

```
%
% Example dataMatrix (replace with your actual data)
dataMatrix = myMatrix_exp
```

```
dataMatrix = 15×15
10^12 ×
         0         0         0         0         0         0         0         0···
    0.4520    0.0006    0.0005    0.0014    0.0020    0.0003    0.0103    0.0000
    0.0015    0.3539    0.0018    0.0029    0.0490    0.0083    0.0178    0.0019
    0.0005    0.0026    2.2025    0.0042    0.0041    0.0041    0.0025    0.0001
    0.0016    0.0020    0.0016    1.3175    0.0045    0.0011    0.0150    0.0000
    0.0098    0.0494    0.0098    0.0107    3.3867    0.0428    0.0763    0.0281
    0.0017    0.0083    0.0026    0.0022    0.0338    0.3962    0.0188    0.0019
    0.0052    0.0136    0.0022    0.0075    0.0416    0.0111    1.1840    0.0016
    0.0003    0.0030    0.0002    0.0002    0.0335    0.0033    0.0037    0.2320
    0.0035    0.0046    0.0037    0.0028    0.0077    0.0031    0.0078    0.0005
      ⋮
```

```
% Replace with your actual data

% Get unique countries
countries = unique(dataMatrix(:,1));

% Extract columns for country1, country2, value
c1 = dataMatrix(:,1);
c2 = dataMatrix(:,2);
val =dataMatrix(:,3);

% Create index vectors for reshaping
idx1 = ismember(c1, countries);
idx2 = ismember(c2, countries);
```

```
%GDP
```

```
% Set up the Import Options and import the data
opts4 = delimitedTextImportOptions("NumVariables", 17, "Encoding", "UTF-8");

% Specify range and delimiter
opts4.DataLines = [2, Inf];
opts4.Delimiter = ",";

% Specify column names and types
opts4.VariableNames = ["LOCATION", "Country", "TRANSACT", "Transaction",
"MEASURE", "Measure", "TIME", "Year", "UnitCode", "Unit",
"PowerCodeCode", "PowerCode", "ReferencePeriodCode", "ReferencePeriod",
"Value", "FlagCodes", "Flags"];
opts4.VariableTypes = ["categorical", "categorical", "double",
"categorical", "categorical", "categorical", "double", "double",
"categorical", "categorical", "double", "categorical", "double", "double",
"double", "categorical", "categorical"];

% Specify file level properties
opts4.ExtraColumnsRule = "ignore";
opts4.EmptyLineRule = "read";

% Specify variable properties
opts4 = setvaropts(opts4, ["LOCATION", "Country", "Transaction", "MEASURE",
"Measure", "UnitCode", "Unit", "PowerCode", "FlagCodes", "Flags"],
"EmptyFieldRule", "auto");
```

```
opts4 = setvaropts(opts4, "TRANSACT", "TrimNonNumeric", true);
opts4 = setvaropts(opts4, "TRANSACT", "ThousandsSeparator", ",");

% Import the data
GDP4 = readtable("/Users/richmondessieku/Desktop/FALL2023/Data/GDP.csv",
opts4);

% Display results
% Clean GDP data
GDP_data = GDP4
```

```
GDP_data = 415×17 table
```

...

	LOCATION	Country	TRANSACT	Transaction
1	AUS	Australia	1	Gross domestic product (expenditure approach)
2	AUS	Australia	1	Gross domestic product (expenditure approach)
3	AUS	Australia	1	Gross domestic product (expenditure approach)
4	AUS	Australia	1	Gross domestic product (expenditure approach)
5	AUS	Australia	1	Gross domestic product (expenditure approach)
6	AUS	Australia	1	Gross domestic product (expenditure approach)
7	AUS	Australia	1	Gross domestic product (expenditure approach)
8	AUT	Austria	1	Gross domestic product (expenditure approach)
9	AUT	Austria	1	Gross domestic product (expenditure approach)
10	AUT	Austria	1	Gross domestic product (expenditure approach)
11	AUT	Austria	1	Gross domestic product (expenditure approach)
12	AUT	Austria	1	Gross domestic product (expenditure approach)
13	AUT	Austria	1	Gross domestic product (expenditure approach)
14	AUT	Austria	1	Gross domestic product (expenditure approach)
15	BEL	Belgium	1	Gross domestic product (expenditure approach)
16	BEL	Belgium	1	Gross domestic product (expenditure approach)
17	BEL	Belgium	1	Gross domestic product (expenditure approach)
18	BEL	Belgium	1	Gross domestic product (expenditure approach)
19	BEL	Belgium	1	Gross domestic product (expenditure approach)
20	BEL	Belgium	1	Gross domestic product (expenditure approach)
21	BEL	Belgium	1	Gross domestic product (expenditure approach)
22	CAN	Canada	1	Gross domestic product (expenditure approach)
23	CAN	Canada	1	Gross domestic product (expenditure approach)
24	CAN	Canada	1	Gross domestic product (expenditure approach)
25	CAN	Canada	1	Gross domestic product (expenditure approach)

	LOCATION	Country	TRANSACT	Transaction
26	CAN	Canada	1	Gross domestic product (expenditure approach)
27	CAN	Canada	1	Gross domestic product (expenditure approach)
28	CAN	Canada	1	Gross domestic product (expenditure approach)
29	CZE	Czech Republic	1	Gross domestic product (expenditure approach)
30	CZE	Czech Republic	1	Gross domestic product (expenditure approach)
31	CZE	Czech Republic	1	Gross domestic product (expenditure approach)
32	CZE	Czech Republic	1	Gross domestic product (expenditure approach)
33	CZE	Czech Republic	1	Gross domestic product (expenditure approach)
34	CZE	Czech Republic	1	Gross domestic product (expenditure approach)
35	CZE	Czech Republic	1	Gross domestic product (expenditure approach)
36	DNK	Denmark	1	Gross domestic product (expenditure approach)
37	DNK	Denmark	1	Gross domestic product (expenditure approach)
38	DNK	Denmark	1	Gross domestic product (expenditure approach)
39	DNK	Denmark	1	Gross domestic product (expenditure approach)
40	DNK	Denmark	1	Gross domestic product (expenditure approach)
41	DNK	Denmark	1	Gross domestic product (expenditure approach)
42	DNK	Denmark	1	Gross domestic product (expenditure approach)
43	FIN	Finland	1	Gross domestic product (expenditure approach)
44	FIN	Finland	1	Gross domestic product (expenditure approach)
45	FIN	Finland	1	Gross domestic product (expenditure approach)
46	FIN	Finland	1	Gross domestic product (expenditure approach)
47	FIN	Finland	1	Gross domestic product (expenditure approach)
48	FIN	Finland	1	Gross domestic product (expenditure approach)
49	FIN	Finland	1	Gross domestic product (expenditure approach)
50	FRA	France	1	Gross domestic product (expenditure approach)
51	FRA	France	1	Gross domestic product (expenditure approach)
52	FRA	France	1	Gross domestic product (expenditure approach)
53	FRA	France	1	Gross domestic product (expenditure approach)
54	FRA	France	1	Gross domestic product (expenditure approach)
55	FRA	France	1	Gross domestic product (expenditure approach)
56	FRA	France	1	Gross domestic product (expenditure approach)
57	DEU	Germany	1	Gross domestic product (expenditure approach)
58	DEU	Germany	1	Gross domestic product (expenditure approach)

	LOCATION	Country	TRANSACT	Transaction
59	DEU	Germany	1	Gross domestic product (expenditure approach)
60	DEU	Germany	1	Gross domestic product (expenditure approach)
61	DEU	Germany	1	Gross domestic product (expenditure approach)
62	DEU	Germany	1	Gross domestic product (expenditure approach)
63	DEU	Germany	1	Gross domestic product (expenditure approach)
64	GRC	Greece	1	Gross domestic product (expenditure approach)
65	GRC	Greece	1	Gross domestic product (expenditure approach)
66	GRC	Greece	1	Gross domestic product (expenditure approach)
67	GRC	Greece	1	Gross domestic product (expenditure approach)
68	GRC	Greece	1	Gross domestic product (expenditure approach)
69	GRC	Greece	1	Gross domestic product (expenditure approach)
70	GRC	Greece	1	Gross domestic product (expenditure approach)
71	ISL	Iceland	1	Gross domestic product (expenditure approach)
72	ISL	Iceland	1	Gross domestic product (expenditure approach)
73	ISL	Iceland	1	Gross domestic product (expenditure approach)
74	ISL	Iceland	1	Gross domestic product (expenditure approach)
75	ISL	Iceland	1	Gross domestic product (expenditure approach)
76	ISL	Iceland	1	Gross domestic product (expenditure approach)
77	ISL	Iceland	1	Gross domestic product (expenditure approach)
78	NOR	Norway	1	Gross domestic product (expenditure approach)
79	NOR	Norway	1	Gross domestic product (expenditure approach)
80	NOR	Norway	1	Gross domestic product (expenditure approach)
81	NOR	Norway	1	Gross domestic product (expenditure approach)
82	NOR	Norway	1	Gross domestic product (expenditure approach)
83	NOR	Norway	1	Gross domestic product (expenditure approach)
84	NOR	Norway	1	Gross domestic product (expenditure approach)
85	ESP	Spain	1	Gross domestic product (expenditure approach)
86	ESP	Spain	1	Gross domestic product (expenditure approach)
87	ESP	Spain	1	Gross domestic product (expenditure approach)
88	ESP	Spain	1	Gross domestic product (expenditure approach)
89	ESP	Spain	1	Gross domestic product (expenditure approach)
90	ESP	Spain	1	Gross domestic product (expenditure approach)
91	ESP	Spain	1	Gross domestic product (expenditure approach)

	LOCATION	Country	TRANSACT	Transaction
92	CHE	Switzerland	1	Gross domestic product (expenditure approach)
93	CHE	Switzerland	1	Gross domestic product (expenditure approach)
94	CHE	Switzerland	1	Gross domestic product (expenditure approach)
95	CHE	Switzerland	1	Gross domestic product (expenditure approach)
96	CHE	Switzerland	1	Gross domestic product (expenditure approach)
97	CHE	Switzerland	1	Gross domestic product (expenditure approach)
98	CHE	Switzerland	1	Gross domestic product (expenditure approach)
99	GBR	United Kingdom	1	Gross domestic product (expenditure approach)
100	GBR	United Kingdom	1	Gross domestic product (expenditure approach)

⋮

```
GDP_data = removevars(GDP_data,{'TRANSACT','MEASURE','TIME',...

'UnitCode','PowerCodeCode','ReferencePeriodCode','FlagCodes','Flags','Measure','ReferencePeriod','Transaction','PowerCode','Unit'});


GDP_data =GDP_data(GDP_data.Year == 2020,:);
countries = {'USA','NLD','JPN','DEU','GBR','IND','NOR','ITA','CAN','BEL','HUN','BRA','AUS','ESP','POL'};

% Use the logical index to select the rows or specific countries  from the table
logical_index_gdp = ismember(GDP_data.LOCATION,countries);
GDP_data = GDP_data(logical_index_gdp,:);

GDP_data = removevars(GDP_data,"Year")
```

```
GDP_data = 15×3 table
```

	LOCATION	Country	Value
1	AUS	Australia	1.2357e+06
2	BEL	Belgium	5.2805e+05
3	CAN	Canada	1.6499e+06
4	DEU	Germany	4.0073e+06
5	NOR	Norway	3.2882e+05
6	ESP	Spain	1.5918e+06
7	GBR	United Kingdom	2.6691e+06
8	JPN	Japan	5.1111e+06
9	POL	Poland	1.1958e+06

	LOCATION	Country	Value
10	NLD	Netherlands	8.9865e+05
11	IND	India	8.5383e+06
12	HUN	Hungary	2.9650e+05
13	USA	United States	1.9377e+07
14	ITA	Italy	2.1302e+06
15	BRA	Brazil	2.8555e+06

```
GDP_data = removevars(GDP_data,"Country")
```

```
GDP_data = 15×2 table
```

	LOCATION	Value
1	AUS	1.2357e+06
2	BEL	5.2805e+05
3	CAN	1.6499e+06
4	DEU	4.0073e+06
5	NOR	3.2882e+05
6	ESP	1.5918e+06
7	GBR	2.6691e+06
8	JPN	5.1111e+06
9	POL	1.1958e+06
10	NLD	8.9865e+05
11	IND	8.5383e+06
12	HUN	2.9650e+05
13	USA	1.9377e+07
14	ITA	2.1302e+06
15	BRA	2.8555e+06

```
% Converting the Table into Martrix
Matrix_GDP = GDP_data.Value
```

```
Matrix_GDP = 15×1
10^7 ×
    0.1236
    0.0528
    0.1650
    0.4007
    0.0329
    0.1592
    0.2669
    0.5111
    0.1196
    0.0899
      ⋮
```

```
%myMatrix_GDP = table2array(Matrix_GDP)
% Check if it's a matrix
vectorColumn2gdp = Matrix_GDP
```

```
vectorColumn2gdp = 15×1
10^7 ×
    0.1236
    0.0528
    0.1650
    0.4007
    0.0329
    0.1592
    0.2669
    0.5111
    0.1196
    0.0899
      ⋮
```

```
gdpValues = vectorColumn2gdp; % GDP values in the same order as countryNames
Final_GDP = reshape(vectorColumn2gdp,[],15);


% Specify column names and types
opts5.VariableNames = ["Var1", "Var2", "Var3", "Var4", "Var5", "Var6",
"Var7", "Var8", "Var9", "Var10", "Var11", "Var12", "Var13", "Var14",
"Var15", "Var16", "Var17", "Var18", "Var19", "Var20", "Var21", "Var22",
"Var23", "Var24", "Var25", "Var26", "Var27", "Var28", "Var29", "Var30",
"Var31"];
opts5.VariableTypes = ["double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double", "double", "double"];

% Clear temporary variables
clear opts5;
% Set up the Import Options and import the data
opts2 = spreadsheetImportOptions("NumVariables", 32);

% Specify sheet and range
opts2.Sheet = "Sheet1";
opts2.DataRange = "A1:AF15";

% Specify column names and types
opts2.VariableNames = ["Var1", "Var2", "Var3", "Var4", "Var5", "Var6",
"Var7", "Var8", "Var9", "Var10", "Var11", "Var12", "Var13", "Var14",
"Var15", "Var16", "Var17", "Var18", "Var19", "Var20", "Var21", "Var22",
"Var23", "Var24", "Var25", "Var26", "Var27", "Var28", "Var29", "Var30",
"Var31", "Var32"];
opts2.VariableTypes = ["double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double", "double",
"double", "double", "double", "double", "double", "double"];

% Import the data
Concatenated_Data1 = readtable("/Users/richmondessieku/Desktop/FALL2023/
Data/Concatenated_Data1.xlsx", opts2, "UseExcel", false);

% Clear temporary variables
clear opts2

% Display results
Concatenated_Data1
```

```
Concatenated_Data1 = 15×32 table
```

...

	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8
1	4.5201e+11	1.0428e+03	1.6760e+04	1.3372e+04	1.5587e+04	1.6563e+04	1.7699e+04	1.7011e+04
2	1.5309e+09	1.6760e+04	68.4447	9.6657e+03	6.0315e+03	196.8761	1.3166e+03	323.7796
3	466880000	1.3372e+04	9.6657e+03	1.0974e+03	8.1914e+03	9.8477e+03	8.3898e+03	9.5017e+03
4	1.5897e+09	1.5587e+04	6.0315e+03	8.1914e+03	1.1880e+03	6.1606e+03	6.0405e+03	5.7157e+03
5	9.7881e+09	1.6563e+04	196.8761	9.8477e+03	6.1606e+03	224.8358	1.4793e+03	495.3633
6	1.6940e+09	1.7699e+04	1.3166e+03	8.3898e+03	6.0405e+03	1.4793e+03	267.5404	1.2634e+03
7	5.1796e+09	1.7011e+04	323.7796	9.5017e+03	5.7157e+03	495.3633	1.2634e+03	185.8346
8	334960000	1.5796e+04	1.1300e+03	1.0267e+04	7.1312e+03	976.2567	1.9777e+03	1.4537e+03
9	3.4631e+09	1.0435e+04	6.4196e+03	1.4441e+04	1.1644e+04	6.2296e+03	7.2820e+03	6.7206e+03
10	4.2345e+09	1.6333e+04	1.1749e+03	9.4825e+03	7.0892e+03	1.1460e+03	1.3668e+03	1.4384e+03
11	1.2143e+10	7.8314e+03	9.4633e+03	1.8550e+04	1.0358e+04	9.2983e+03	1.0777e+04	9.5742e+03
12	3.5134e+09	1.6658e+04	173.0333	9.8108e+03	5.9882e+03	173.5239	1.4814e+03	360.3150
13	223720000	1.5964e+04	1.0878e+03	1.0641e+04	5.9416e+03	969.2816	2.3908e+03	1.1571e+03
14	881530000	1.5608e+04	1.1606e+03	1.0669e+04	6.9259e+03	965.9211	2.2930e+03	1.4516e+03
15	7.2345e+09	1.6009e+04	5.8917e+03	7.6943e+03	548.3946	6.0353e+03	5.7703e+03	5.5702e+03

```
% Display results
%Concatenated_Data
concatenated_matrix = Concatenated_Data
```

```
concatenated_matrix = 15×31 table
```

...

	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8
1	1.6563e+04	1.7699e+04	1.7011e+04	1.5796e+04	1.0428e+03	1.6760e+04	1.3372e+04	1.5587e+04
2	196.8761	1.3166e+03	323.7796	1.1300e+03	1.6760e+04	68.4447	9.6657e+03	6.0315e+03
3	9.8477e+03	8.3898e+03	9.5017e+03	1.0267e+04	1.3372e+04	9.6657e+03	1.0974e+03	8.1914e+03
4	6.1606e+03	6.0405e+03	5.7157e+03	7.1312e+03	1.5587e+04	6.0315e+03	8.1914e+03	1.1880e+03
5	224.8358	1.4793e+03	495.3633	976.2567	1.6563e+04	196.8761	9.8477e+03	6.1606e+03
6	1.4793e+03	267.5404	1.2634e+03	1.9777e+03	1.7699e+04	1.3166e+03	8.3898e+03	6.0405e+03
7	495.3633	1.2634e+03	185.8346	1.4537e+03	1.7011e+04	323.7796	9.5017e+03	5.7157e+03
8	1.5796e+04	1.1300e+03	1.0267e+04	7.1312e+03	976.2567	1.9777e+03	1.4537e+03	114.7229
9	1.0435e+04	6.4196e+03	1.4441e+04	1.1644e+04	6.2296e+03	7.2820e+03	6.7206e+03	5.3698e+03
10	1.6333e+04	1.1749e+03	9.4825e+03	7.0892e+03	1.1460e+03	1.3668e+03	1.4384e+03	811.4083
11	7.8314e+03	9.4633e+03	1.8550e+04	1.0358e+04	9.2983e+03	1.0777e+04	9.5742e+03	9.0589e+03
12	1.6658e+04	173.0333	9.8108e+03	5.9882e+03	173.5239	1.4814e+03	360.3150	1.1459e+03
13	1.5964e+04	1.0878e+03	1.0641e+04	5.9416e+03	969.2816	2.3908e+03	1.1571e+03	1.4835e+03
14	1.5608e+04	1.1606e+03	1.0669e+04	6.9259e+03	965.9211	2.2930e+03	1.4516e+03	546.8060
15	1.6009e+04	5.8917e+03	7.6943e+03	548.3946	6.0353e+03	5.7703e+03	5.5702e+03	7.0115e+03

```
%concatenated_matrix = Concatenated_Data1
concatenated_matrix = table2array(concatenated_matrix)
```

```
concatenated_matrix = 15×31
10^13 ×
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ···
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
      ⋮
```

```
% Regression
```

```

X_var = concatenated_matrix(:,16:end-1)
```

```
X_var = 15×15
10^13 ×
    0.0452    0.0002    0.0000    0.0010    0.0001    0.0001    0.0001    0.0000 ···
    0.0002    0.0049    0.0008    0.0018    0.0354    0.0002    0.0003    0.0002
    0.0000    0.0004    0.0004    0.0003    0.0003    0.2203    0.0004    0.0000
    0.0002    0.0004    0.0001    0.0015    0.0002    0.0002    0.1317    0.0000
    0.0010    0.3387    0.0043    0.0076    0.0049    0.0010    0.0011    0.0028
    0.0002    0.0034    0.0396    0.0019    0.0008    0.0003    0.0002    0.0002
    0.0005    0.0042    0.0011    0.1184    0.0014    0.0002    0.0008    0.0002
    0.0000    0.0003    0.0000    0.0000    0.0033    0.0003    0.0004    0.0232
    0.0003    0.0005    0.0004    0.0003    0.0008    0.0003    0.0008    0.0000
    0.0004    0.0017    0.0004    0.0005    0.0064    0.0023    0.0026    0.0005
     ⋮
```

```
dis = concatenated_matrix(:,1:15)
```

```
dis = 15×15
10^4 ×
    1.6563    1.7699    1.7011    1.5796    0.1043    1.6760    1.3372    1.5587⋯
    0.0197    0.1317    0.0324    0.1130    1.6760    0.0068    0.9666    0.6032
    0.9848    0.8390    0.9502    1.0267    1.3372    0.9666    0.1097    0.8191
    0.6161    0.6040    0.5716    0.7131    1.5587    0.6032    0.8191    0.1188
    0.0225    0.1479    0.0495    0.0976    1.6563    0.0197    0.9848    0.6161
    0.1479    0.0268    0.1263    0.1978    1.7699    0.1317    0.8390    0.6040
    0.0495    0.1263    0.0186    0.1454    1.7011    0.0324    0.9502    0.5716
    1.5796    0.1130    1.0267    0.7131    0.0976    0.1978    0.1454    0.0115
    1.0435    0.6420    1.4441    1.1644    0.6230    0.7282    0.6721    0.5370
    1.6333    0.1175    0.9483    0.7089    0.1146    0.1367    0.1438    0.0811
     ⋮
```

```
GDP_use = concatenated_matrix(:,end)
```

```
GDP_use = 15×1
10^13 ×
    0.1236
    0.0528
    0.1650
    0.4007
    0.0329
    0.1592
    0.2669
    0.5111
    0.1196
    0.0899
     ⋮
```

```
lnX_var = log(X_var);
lnX_var = lnX_var(:);
lnGDPi = log(GDP_use);
lnd = log(dis);

N = 15;
ones_col = ones(N^2, 1);
lnGDP_rep = repmat(lnGDPi, N, 1);
lnGDP_transpose = lnGDPi'
```

```
lnGDP_transpose = 1×15
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624⋯
```

```
lnGDP_rep_1 = repmat(lnGDP_transpose, 15, 15)
```

```
lnGDP_rep_1 = 15×225
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624 ···
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
   27.8427   26.9925   28.1317   29.0191   26.5188   28.0959   28.6127   29.2624
      ⋮
```

```
lnGDP_col = lnGDP_rep(:,1)
```

```
lnGDP_col = 225×1
   27.8427
   26.9925
   28.1317
   29.0191
   26.5188
   28.0959
   28.6127
   29.2624
   27.8099
   27.5242
      ⋮
```

```
lnd_vec = lnd(:);

% Regression
A_B = [ones_col lnGDP_rep lnGDP_col lnd_vec]
```

```
A_B = 225×4
     1.0000   27.8427   27.8427    9.7149
     1.0000   26.9925   26.9925    5.2826
     1.0000   28.1317   28.1317    9.1950
     1.0000   29.0191   29.0191    8.7259
     1.0000   26.5188   26.5188    5.4154
     1.0000   28.0959   28.0959    7.2993
     1.0000   28.6127   28.6127    6.2053
     1.0000   29.2624   29.2624    9.6675
     1.0000   27.8099   27.8099    9.2529
     1.0000   27.5242   27.5242    9.7010
      ⋮
```

```

[B, Bt, ~, ~, stats] = regress(lnX, A)
```

```
Warning: X is rank deficient to within machine precision.
B = 4×1
   35.9565
   -0.3126
         0
   -0.5479
Bt = 4×2
```

```
   29.5363   42.3768
   -0.5413   -0.0839
         0         0
   -0.7395   -0.3563
stats = 1×4
    0.1670   22.2583    0.0000    3.6835
```

```
% Calculate residuals
resid = lnX - A_B * B
```

```
resid = 225×1
    4.9065
   -3.4756
   -2.1634
   -0.9178
   -1.6956
   -1.9244
   -1.2447
   -1.8831
   -0.2284
    0.1288
      ⋮
```

```
% Calculate fitted values
X_var_hat = A_B*B
```

```
X_var_hat = 225×1
   21.9305
   24.6247
   22.1250
   22.1046
   24.7000
   23.1748
   23.6126
   21.5126
   22.1938
   22.0377
      ⋮
```

```
% Finding the MSE
MSE = sum(resid.^2) / (225 - 4)
```

```
MSE = 3.7002
```

```
% Finding SE
SE = sqrt(MSE * diag(inv(X_var' * X_var)))
```

```
SE = 15×1
10^-10 ×
    0.0426
    0.0065
    0.0489
    0.0163
    0.0562
    0.0087
    0.0146
    0.0834
    0.0025
    0.0868
     ⋮
```