Stundent Name: Vijay kumar Ramavath

Student ID:21031708

In [ ]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import wbgapi as wb
```

In [2]:

```python
wb.source.info()
```

Out[2]:

id	name	code	concepts	lastupdated
1	Doing Business	DBS	3	2021-08-18
2	World Development Indicators	WDI	3	2022-04-08
3	Worldwide Governance Indicators	WGI	3	2021-09-27
5	Subnational Malnutrition Database	SNM	3	2016-03-21
6	International Debt Statistics	IDS	4	2022-01-14
11	Africa Development Indicators	ADI	3	2013-02-22
12	Education Statistics	EDS	3	2020-12-20
13	Enterprise Surveys	ESY	3	2022-03-25
14	Gender Statistics	GDS	3	2022-03-01
15	Global Economic Monitor	GEM	3	2020-07-27
16	Health Nutrition and Population Statistics	HNP	3	2022-02-16
18	IDA Results Measurement System	IDA	3	2021-07-23
19	Millennium Development Goals	MDG	3	2018-09-19
20	Quarterly Public Sector Debt	PSD	3	2022-04-08
22	Quarterly External Debt Statistics SDDS	QDS	3	2022-04-15
23	Quarterly External Debt Statistics GDDS	QDG	3	2022-04-15
24	Poverty and Equity	POV	3	2021-10-05
25	Jobs	JOB	3	2022-04-08
27	Global Economic Prospects	GEP	3	2022-01-11
28	Global Financial Inclusion	FDX	3	2018-10-15
29	The Atlas of Social Protection: Indicators of Resilience and Equity	GSP	3	2021-12-20
30	Exporter Dynamics Database – Indicators at Country-Year Level	ED1	3	2016-03-31
31	Country Policy and Institutional Assessment	CPI	3	2021-09-16
32	Global Financial Development	GFD	3	2021-12-03
33	G20 Financial Inclusion Indicators	G2F	3	2019-02-27
34	Global Partnership for Education	GPE	3	2013-04-12
35	Sustainable Energy for All	SE4	3	2018-06-30
36	Statistical Capacity Indicators	BBS	3	2021-02-03
37	LAC Equity Lab	LEL	3	2016-04-06
38	Subnational Poverty	SNP	3	2015-04-27
39	Health Nutrition and Population Statistics by Wealth Quintile	HNQ	3	2021-12-13

id	name	code	concepts	lastupdated
40	Population estimates and projections	PQP	3	2022-02-17
41	Country Partnership Strategy for India (FY2013 - 17)	CPS	3	2015-05-22
43	Adjusted Net Savings	WAT	3	2021-04-27
45	Indonesia Database for Policy and Economic Research	IDD	3	2019-05-07
46	Sustainable Development Goals	SDG	3	2021-07-02
50	Subnational Population	SNT	3	2017-09-21
54	Joint External Debt Hub	JED	3	2022-02-17
57	WDI Database Archives	WDA	4	2022-03-02
58	Universal Health Coverage	UHC	3	2019-04-25
59	Wealth Accounts	WAC	3	2021-10-27
60	Economic Fitness	EFT	3	2018-04-19
61	PPPs Regulatory Quality	PRQ	3	2019-04-25
62	International Comparison Program (ICP) 2011	ICP	4	2020-08-24
63	Human Capital Index	HCI	3	2020-09-21
64	Worldwide Bureaucracy Indicators	WBI	3	2021-05-26
65	Health Equity and Financial Protection Indicators	HPI	3	2019-11-06
66	Logistics Performance Index	LPI	3	2019-10-29
67	PEFA 2011	PF1	4	2022-02-10
68	PEFA 2016	PF6	4	2022-03-08
69	Global Financial Inclusion and Consumer Protection Survey	RFA	3	2019-06-27
70	Economic Fitness 2	EF2	3	2019-04-26
71	International Comparison Program (ICP) 2005	IC5	4	2021-09-10
73	Global Financial Inclusion and Consumer Protection Survey (Internal)	RFI	3	2019-06-27
75	Environment, Social and Governance (ESG) Data	ESG	3	2022-02-22
76	Remittance Prices Worldwide (Sending Countries)	RWS	3	2020-12-07
77	Remittance Prices Worldwide (Receiving Countries)	RWR	3	2020-12-07
78	ICP 2017	IC7	4	2020-10-21
79	PEFA_GRPFM	GRP	4	2022-03-28
80	Gender Disaggregated Labor Database (GDLD)	GDL	4	2020-07-25
81	International Debt Statistics: DSSI	DSI	4	2021-12-16
82	Global Public Procurement	GPP	3	2021-03-24
83	Statistical Performance Indicators (SPI)	SPI	3	2021-04-01
84	Education Policy	EDP	3	2021-05-11
85	PEFA_2021_SNG	SNG	4	2022-03-28
86	Global Jobs Indicators Database (JOIN)	JON	3	2021-09-24
87	Country Climate and Development Report (CCDR)	CCD	3	2022-03-30
88	Food Prices for Nutrition	FPN	4	2022-01-31
	68 elements			

In [3]:

```
wb.series.info(q='arable')
```

Out[3]:

id	value
AG.LND.ARBL.HA	Arable land (hectares)
AG.LND.ARBL.HA.PC	Arable land (hectares per person)

id	value
AG.LND.ARBL.ZS	Arable land (% of land area)
AG.LND.TRAC.ZS	Agricultural machinery, tractors per 100 sq. km of arable land
	4 elements

In [4]:

```python
wb.series.info(q='forest')
```

Out[4]:

id	value
AG.LND.FRST.K2	Forest area (sq. km)
AG.LND.FRST.ZS	Forest area (% of land area)
NV.AGR.EMPL.KD	Agriculture, forestry, and fishing, value added per worker (constant 2015 US$)
NV.AGR.TOTL.CD	Agriculture, forestry, and fishing, value added (current US$)
NV.AGR.TOTL.CN	Agriculture, forestry, and fishing, value added (current LCU)
NV.AGR.TOTL.KD	Agriculture, forestry, and fishing, value added (constant 2015 US$)
NV.AGR.TOTL.KD.ZG	Agriculture, forestry, and fishing, value added (annual % growth)
NV.AGR.TOTL.KN	Agriculture, forestry, and fishing, value added (constant LCU)
NV.AGR.TOTL.ZS	Agriculture, forestry, and fishing, value added (% of GDP)
NY.ADJ.DFOR.CD	Adjusted savings: net forest depletion (current US$)
NY.ADJ.DFOR.GN.ZS	Adjusted savings: net forest depletion (% of GNI)
NY.GDP.FRST.RT.ZS	Forest rents (% of GDP)
	12 elements

In [6]:

```python
Arable= wb.data.DataFrame('AG.LND.ARBL.HA', wb.region.members('EMU'))
```

In [7]:

```python
Arable
```

Out[7]:

economy	YR1960	YR1961	YR1962	YR1963	YR1964	YR1965	YR1966	YR1967	YR1968	YR19…
AUT	NaN	1689000.0	1681000.0	1656000.0	1657000.0	1660000.0	1616000.0	1600000.0	1601000.0	158500(
BEL	NaN	1007000.0	1004000.0	1001000.0	998000.0	996000.0	994000.0	990000.0	989000.0	98800(
CYP	NaN	329000.0	327000.0	325000.0	323000.0	322000.0	320000.0	318000.0	316000.0	31400(
DEU	NaN	12220000.0	12185000.0	12165000.0	12161000.0	12153000.0	12155000.0	12049000.0	12032000.0	1200100(
ESP	NaN	16246000.0	16296000.0	16205000.0	15919000.0	15966000.0	15509000.0	15740000.0	15692000.0	1608200(
EST	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
FIN	NaN	2664700.0	2679100.0	2695500.0	2709600.0	2720500.0	2733400.0	2742400.0	2745100.0	261150(
FRA	NaN	19606000.0	19530000.0	19455000.0	19078000.0	18796000.0	18463000.0	18076000.0	17579000.0	1756500(
GRC	NaN	2794000.0	2863000.0	3057000.0	3001000.0	2991000.0	2995000.0	3020000.0	3053000.0	301700(
IRL	NaN	1590000.0	1567000.0	1561000.0	1526000.0	1492000.0	1429000.0	1424000.0	1394000.0	138100(
ITA	NaN	12862000.0	12795000.0	12674000.0	12598000.0	12524000.0	12444000.0	12389000.0	12346000.0	1222100(
LTU	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
LUX	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
LVA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
MLT	NaN	17000.0	16000.0	15000.0	14000.0	13000.0	13000.0	13000.0	13000.0	1300(

	YR1960	YR1961	YR1962	YR1963	YR1964	YR1965	YR1966	YR1967	YR1968	YR19...
MLT	NaN	17000.0	16000.0	15000.0	14000.0	13000.0	13000.0	13000.0	13000.0	13000...
NLD economy	NaN	992000.0	966000.0	949000.0	937000.0	928000.0	897000.0	873000.0	864000.0	845000...
PRT	NaN	2507000.0	2503000.0	2499000.0	2495000.0	2491000.0	2487000.0	2483000.0	2479000.0	2475000...
SVK	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na...
SVN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na...

19 rows × 61 columns

In [10]:

```
Forest= wb.data.DataFrame('AG.LND.FRST.K2', wb.region.members('EMU'))
```

In [11]:

```
Forest
```

Out[11]:

economy	YR1960	YR1961	YR1962	YR1963	YR1964	YR1965	YR1966	YR1967	YR1968	YR1969	...	YR2011	YR2012
AUT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	38667.98	38703.96
BEL	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	6897.56	6896.42
CYP	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1728.14	1727.88
DEU	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	114110.00	114130.00
ESP	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	185465.08	185476.76
EST	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	23530.18	23700.16
FIN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	222754.00	223088.00
FRA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	165024.00	165858.00
GRC	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	39018.03	39018.03
IRL	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	7272.38	7340.96
ITA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	90818.48	91356.56
LTU	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	21734.00	21768.00
LUX	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	887.00	887.00
LVA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	33759.84	33798.48
MLT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.50	3.50
NLD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3717.50	3700.20
PRT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	32640.00	32760.00
SVK	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	19186.78	19194.46
SVN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	12472.00	12474.00

19 rows × 61 columns
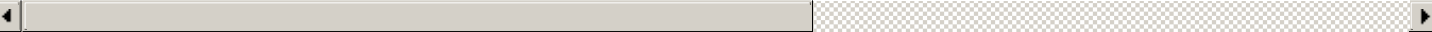
In [25]:

```
indicator=['AG.LND.ARBL.HA','AG.LND.FRST.K2']
country=['GBR','USA','IND']
my_dataframe=wb.data.DataFrame(indicator,country, mrv=5)
```

In [18]:

```
my_dataframe
```

Out[18]:

economy	series	YR2016	YR2017	YR2018	YR2019	YR2020
GBR	AG.LND.ARBL.HA	6026010.5	6082879.4	6037787.4	NaN	NaN
	AG.LND.FRST.K2	31590.0	31640.0	31730.0	31820.0	31900.0
IND	AG.LND.ARBL.HA	156416000.0	156416000.0	156416000.0	NaN	NaN
	AG.LND.FRST.K2	710944.0	713608.0	716272.0	718936.0	721600.0
USA	AG.LND.ARBL.HA	157191000.0	157736800.0	157736800.0	NaN	NaN
	AG.LND.FRST.K2	3100950.0	3097950.0	3097950.0	3097950.0	3097950.0

In [19]:

```
De = wb.data.DataFrame('AG.LND.FRST.K2',
                       ['GBR','USA','IND'],
                       time=range(2011,2020,1))
```

In [20]:

```
De
```

Out[20]:

economy	YR2011	YR2012	YR2013	YR2014	YR2015	YR2016	YR2017	YR2018	YR2019
GBR	30782.0	30974.0	31166.0	31358.0	31550.0	31590.0	31640.0	31730.0	31820.0
IND	697624.0	700288.0	702952.0	705616.0	708280.0	710944.0	713608.0	716272.0	718936.0
USA	3089950.0	3092700.0	3095450.0	3098200.0	3100950.0	3100950.0	3097950.0	3097950.0	3097950.0

In [21]:

```
De.describe()
```

Out[21]:

	YR2011	YR2012	YR2013	YR2014	YR2015	YR2016	YR2017	YR2018
count	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.0
mean	1.272785e+06	1.274654e+06	1.276523e+06	1.278391e+06	1.280260e+06	1.281161e+06	1.281066e+06	1.281984e+06	1.
std	1.608644e+06	1.609647e+06	1.610651e+06	1.611656e+06	1.612661e+06	1.612174e+06	1.609991e+06	1.609487e+06	1.
min	3.078200e+04	3.097400e+04	3.116600e+04	3.135800e+04	3.155000e+04	3.159000e+04	3.164000e+04	3.173000e+04	3.
25%	3.642030e+05	3.656310e+05	3.670590e+05	3.684870e+05	3.699150e+05	3.712670e+05	3.726240e+05	3.740010e+05	3.
50%	6.976240e+05	7.002880e+05	7.029520e+05	7.056160e+05	7.082800e+05	7.109440e+05	7.136080e+05	7.162720e+05	7.
75%	1.893787e+06	1.896494e+06	1.899201e+06	1.901908e+06	1.904615e+06	1.905947e+06	1.905779e+06	1.907111e+06	1.
max	3.089950e+06	3.092700e+06	3.095450e+06	3.098200e+06	3.100950e+06	3.100950e+06	3.097950e+06	3.097950e+06	3.

In [27]:

```
ar= wb.data.DataFrame('AG.LND.ARBL.HA',
                      ['GBR','USA','IND'],
                      time=range(2011,2020,1))
```

In [28]:

```
ar
```

Out[28]:

	YR2011	YR2012	YR2013	YR2014	YR2015	YR2016	YR2017	YR2018	YR2019

economy	YR2011	YR2012	YR2013	YR2014	YR2015	YR2016	YR2017	YR2018	YR2019
economyGBR	6062000.0	6212000.0	6265400.0	6233500.0	6011000.0	6026010.5	6082879.4	6037787.4	NaN
IND	156979000.0	156546000.0	156442000.0	156463000.0	156416000.0	156416000.0	156416000.0	156416000.0	NaN
USA	156362000.0	155007500.0	155553400.0	156099300.0	156645100.0	157191000.0	157736800.0	157736800.0	NaN

In [29]:

```
ar.describe()
```

Out[29]:

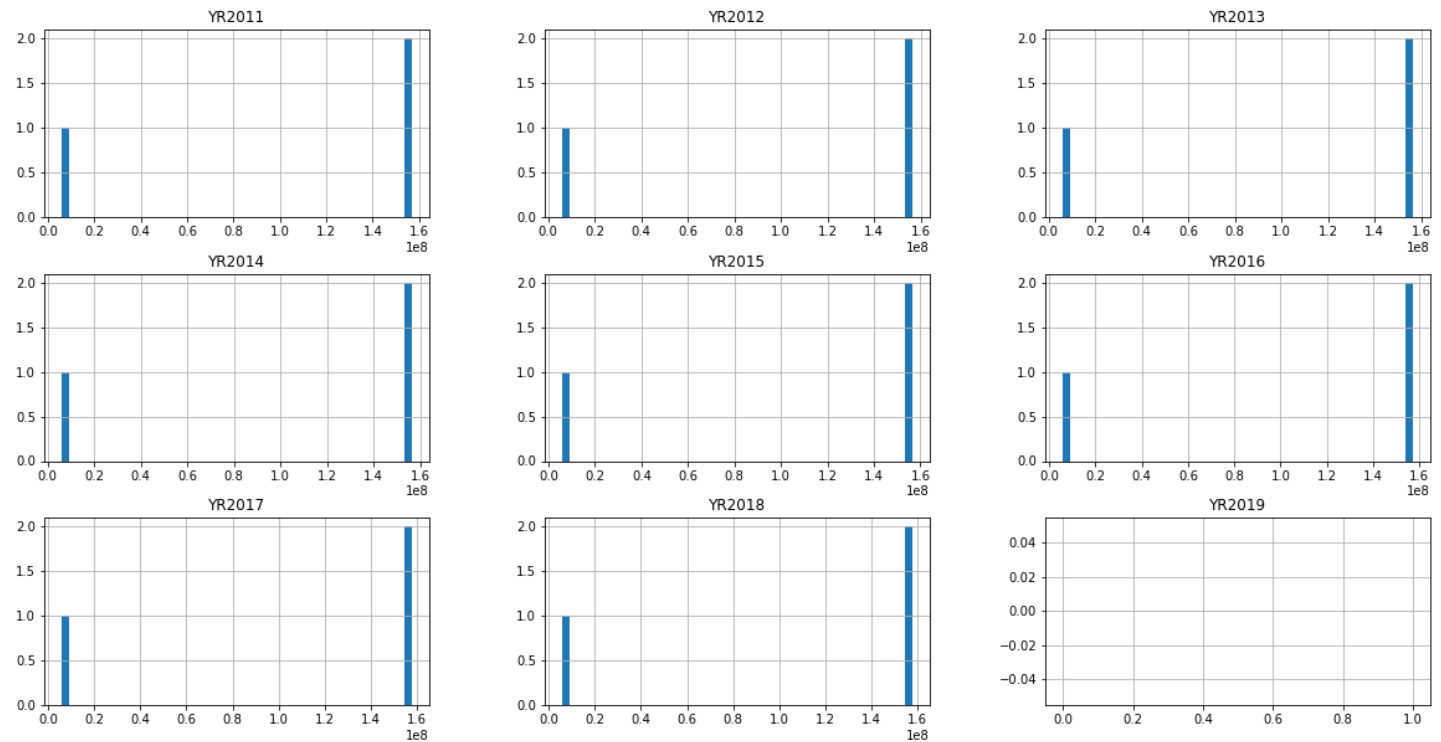|	YR2011	YR2012	YR2013	YR2014	YR2015	YR2016	YR2017	YR2018	YI
count	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	3.000000e+00	
mean	1.064677e+08	1.059218e+08	1.060869e+08	1.062653e+08	1.063574e+08	1.065443e+08	1.067452e+08	1.067302e+08	
std	8.695441e+07	8.635467e+07	8.644913e+07	8.663024e+07	8.690258e+07	8.705229e+07	8.717865e+07	8.720468e+07	
min	6.062000e+06	6.212000e+06	6.265400e+06	6.233500e+06	6.011000e+06	6.026010e+06	6.082879e+06	6.037787e+06	
25%	8.121200e+07	8.060975e+07	8.090940e+07	8.116640e+07	8.121350e+07	8.122101e+07	8.124944e+07	8.122689e+07	
50%	1.563620e+08	1.550075e+08	1.555534e+08	1.560993e+08	1.564160e+08	1.564160e+08	1.564160e+08	1.564160e+08	
75%	1.566705e+08	1.557768e+08	1.559977e+08	1.562812e+08	1.565306e+08	1.568035e+08	1.570764e+08	1.570764e+08	
max	1.569790e+08	1.565460e+08	1.564420e+08	1.564630e+08	1.566451e+08	1.571910e+08	1.577368e+08	1.577368e+08	

In [36]:

```
De.hist(bins=50, figsize=(20, 15))
plt.savefig('numeric_attributes.png')
plt.show()
```
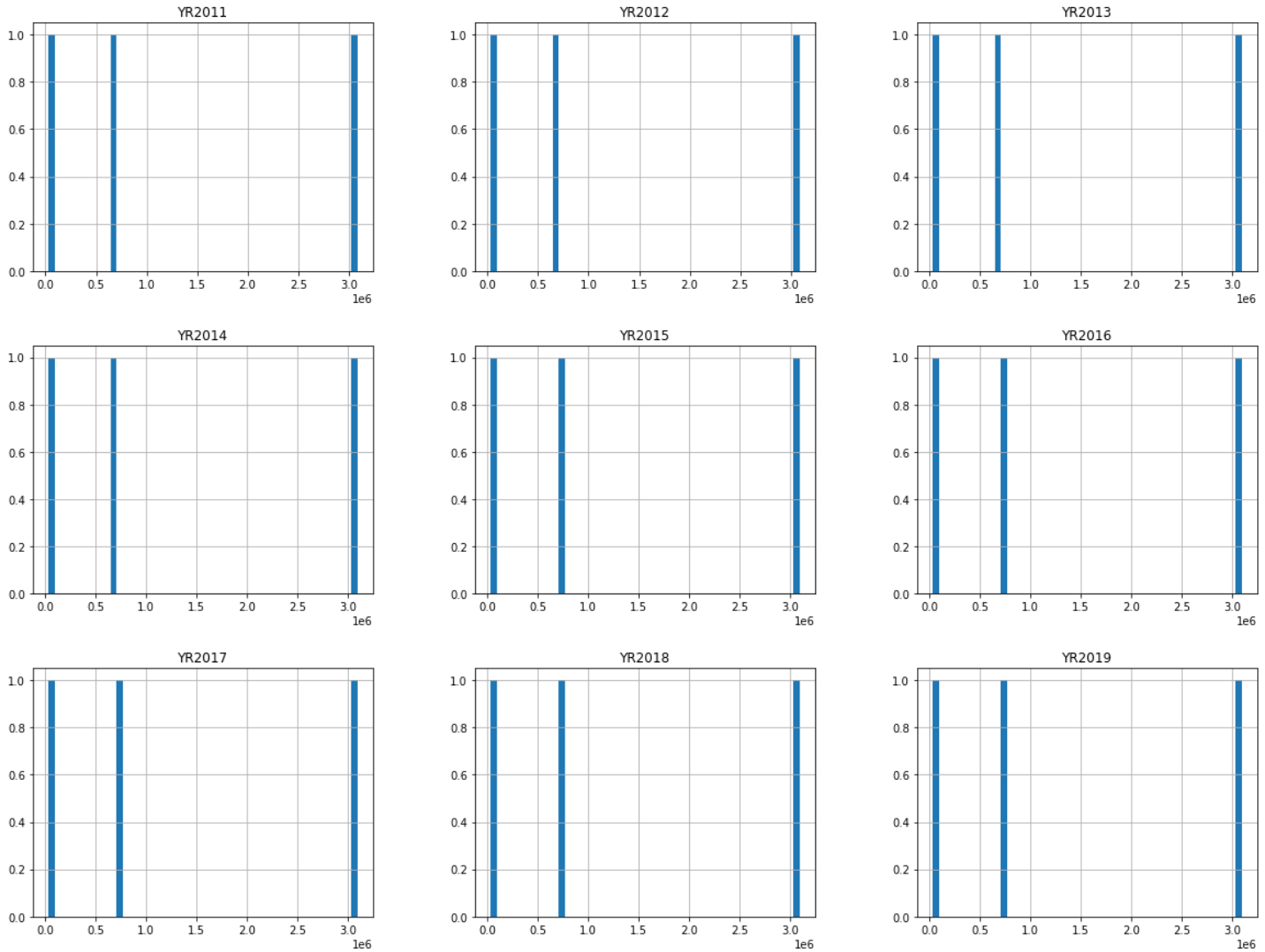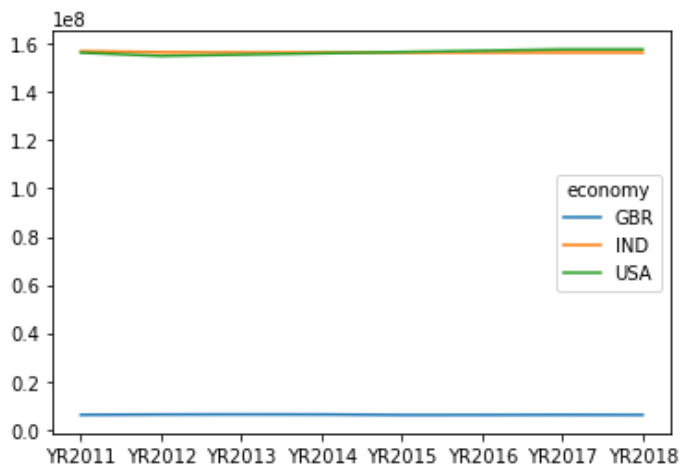
```
ar.hist(bins=50, figsize=(20, 10))
plt.savefig('numeric_attributes.png')
plt.show()
```
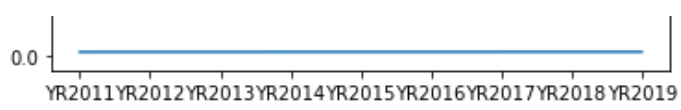
```
wb.data.DataFrame('AG.LND.ARBL.HA',
                  ['GBR','USA','IND'],
                  time=range(2011,2020,1)).T.plot();
```

```
wb.data.DataFrame('AG.LND.FRST.K2',
                  ['GBR','USA','IND'],
                  time=range(2011,2020,1)).T.plot();
```

```
0.0
     YR2011 YR2012 YR2013 YR2014 YR2015 YR2016 YR2017 YR2018 YR2019
```

In [ ]: