Problem statement

National statistical systems are facing significant challenges. These challenges arise from increasing demands for high quality and trustworthy data to guide decision making, coupled with the rapidly changing landscape of the data revolution. To help create a mechanism for learning amongst national statistical systems, the World Bank has developed improved Statistical Performance Indicators (SPI) to monitor the statistical performance of countries.

Importing Libraries

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing Dataset

In [2]:

```
df=pd.read_csv(r"C:\Users\user\Downloads\spi index.csv")
df
```

Out[2]:

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
0	Norway	NOR	2019	100.0	92.233333	77.56875	80.666667	
1	Italy	ITA	2019	100.0	91.866667	75.28750	81.825000	
2	Austria	AUT	2019	100.0	91.300000	74.55000	79.750000	
3	Poland	POL	2019	100.0	95.100000	70.53750	79.716667	
4	Slovenia	SVN	2019	100.0	96.933333	76.28125	71.441667	
...	...	...	...	...	...	...	...	
3483	Virgin Islands (U.S.)	VIR	2004	20.0	NaN	NaN	NaN	
3484	West Bank and Gaza	PSE	2004	20.0	NaN	NaN	NaN	
3485	Yemen, Rep.	YEM	2004	20.0	NaN	NaN	NaN	

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
3486	Zambia	ZMB	2004	40.0	NaN	NaN	NaN	
3487	Zimbabwe	ZWE	2004	20.0	NaN	NaN	NaN	

3488 rows × 79 columns

Data Cleaning and Data Preprocessing

head-To display the specified data from first

In [3]:

```
dat=df.head(50)
dat
```

Out[3]:

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
0	Norway	NOR	2019	100.0	92.233333	77.56875	80.666667	
1	Italy	ITA	2019	100.0	91.866667	75.28750	81.825000	
2	Austria	AUT	2019	100.0	91.300000	74.55000	79.750000	
3	Poland	POL	2019	100.0	95.100000	70.53750	79.716667	
4	Slovenia	SVN	2019	100.0	96.933333	76.28125	71.441667	
5	United States	USA	2019	100.0	94.000000	63.11875	87.500000	
6	Spain	ESP	2019	100.0	90.866667	75.53750	77.866667	
7	Sweden	SWE	2019	100.0	94.866667	75.23750	72.516667	
8	Finland	FIN	2019	100.0	94.933333	75.23750	72.216667	
9	Korea, Rep.	KOR	2019	100.0	93.400000	75.61875	82.400000	

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
10	Australia	AUS	2019	100.0	92.666667	74.07500	74.466667	
11	Netherlands	NLD	2019	100.0	98.500000	71.36250	69.916667	
12	Mexico	MEX	2019	100.0	92.933333	89.31875	80.283333	
13	Germany	DEU	2019	100.0	96.466667	71.10625	74.916667	
14	Canada	CAN	2019	100.0	93.200000	60.01250	84.116667	
15	Ireland	IRL	2019	100.0	94.733333	74.11250	66.341667	
16	Switzerland	CHE	2019	100.0	87.666667	76.55000	80.858333	
17	France	FRA	2019	100.0	90.800000	74.30000	66.641667	
18	Denmark	DNK	2019	90.0	98.700000	68.01875	73.866667	
19	Estonia	EST	2019	100.0	93.933333	67.49375	68.941667	
20	Japan	JPN	2019	90.0	90.533333	73.51250	80.000000	
21	Slovak Republic	SVK	2019	90.0	94.933333	69.95000	73.091667	
22	Portugal	PRT	2019	100.0	90.766667	71.03750	65.791667	
23	Greece	GRC	2019	100.0	87.500000	68.53750	70.766667	
24	New Zealand	NZL	2019	100.0	91.633333	71.51875	63.166667	
25	Czech Republic	CZE	2019	90.0	90.100000	74.58750	75.616667	
26	Lithuania	LTU	2019	100.0	91.833333	60.96875	71.866667	
27	Hungary	HUN	2019	100.0	86.866667	69.03750	68.266667	

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
28	Turkey	TUR	2019	100.0	84.000000	86.08125	53.108333	
29	Latvia	LVA	2019	100.0	89.166667	61.38750	68.041667	
30	United Kingdom	GBR	2019	100.0	86.800000	74.02500	64.941667	
31	Chile	CHL	2019	100.0	76.966667	80.69375	59.525000	
32	Belgium	BEL	2019	100.0	83.933333	62.22500	65.916667	
33	Bulgaria	BGR	2019	100.0	90.100000	60.67500	70.841667	
34	Armenia	ARM	2019	100.0	84.966667	81.22500	59.941667	
35	Cyprus	CYP	2019	100.0	92.266667	53.33125	73.308333	
36	Georgia	GEO	2019	100.0	86.466667	73.71875	60.116667	
37	Costa Rica	CRI	2019	100.0	86.466667	75.95625	66.466667	
38	Moldova	MDA	2019	100.0	94.233333	53.56250	58.816667	
39	Kyrgyz Republic	KGZ	2019	100.0	81.466667	73.63750	53.058333	
40	Kazakhstan	KAZ	2019	100.0	82.133333	78.31250	62.350000	
41	Luxembourg	LUX	2019	80.0	90.433333	61.28125	59.291667	
42	Russian Federation	RUS	2019	93.4	83.666667	58.51875	65.366667	
43	Israel	ISR	2019	100.0	85.733333	58.46250	46.483333	

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
44	Iceland	ISL	2019	80.0	87.800000	59.70000	61.666667	
45	Romania	ROU	2019	90.0	87.166667	56.73125	73.641667	
46	Belarus	BLR	2019	100.0	79.233333	67.88125	53.558333	
47	Brazil	BRA	2019	90.0	83.300000	73.09375	62.375000	
48	Mongolia	MNG	2019	100.0	80.000000	77.47500	63.941667	
49	Thailand	THA	2019	100.0	76.466667	76.23750	57.850000	

50 rows × 79 columns

tail-To display specified no of data from last

In [4]:

```
df.tail(50)
df
```

Out[4]:

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
0	Norway	NOR	2019	100.0	92.233333	77.56875	80.666667	
1	Italy	ITA	2019	100.0	91.866667	75.28750	81.825000	
2	Austria	AUT	2019	100.0	91.300000	74.55000	79.750000	
3	Poland	POL	2019	100.0	95.100000	70.53750	79.716667	
4	Slovenia	SVN	2019	100.0	96.933333	76.28125	71.441667	
...	...	...	...	...	...	...	...	
3483	Virgin Islands (U.S.)	VIR	2004	20.0	NaN	NaN	NaN	

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
3484	West Bank and Gaza	PSE	2004	20.0	NaN	NaN	NaN	
3485	Yemen, Rep.	YEM	2004	20.0	NaN	NaN	NaN	
3486	Zambia	ZMB	2004	40.0	NaN	NaN	NaN	
3487	Zimbabwe	ZWE	2004	20.0	NaN	NaN	NaN	

3488 rows × 79 columns

shape

In [5]:

```
data=np.shape(df)
data
```

Out[5]: (3488, 79)

size

In [6]:

```
print(np.size(df))
```

275552

In [7]:

```
df=df.fillna(value=0)
df
```

Out[7]:

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
0	Norway	NOR	2019	100.0	92.233333	77.56875	80.666667	
1	Italy	ITA	2019	100.0	91.866667	75.28750	81.825000	
2	Austria	AUT	2019	100.0	91.300000	74.55000	79.750000	
3	Poland	POL	2019	100.0	95.100000	70.53750	79.716667	

	country	iso3c	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.IN
4	Slovenia	SVN	2019	100.0	96.933333	76.28125	71.441667	
...	...	...	...	...	...	...	...	
3483	Virgin Islands (U.S.)	VIR	2004	20.0	0.000000	0.00000	0.000000	
3484	West Bank and Gaza	PSE	2004	20.0	0.000000	0.00000	0.000000	
3485	Yemen, Rep.	YEM	2004	20.0	0.000000	0.00000	0.000000	
3486	Zambia	ZMB	2004	40.0	0.000000	0.00000	0.000000	
3487	Zimbabwe	ZWE	2004	20.0	0.000000	0.00000	0.000000	

3488 rows × 79 columns

columns

In [8]:

```
df.columns
```

Out[8]:

```
Index(['country', 'iso3c', 'date', 'SPI.INDEX.PIL1', 'SPI.INDEX.PIL2',
       'SPI.INDEX.PIL3', 'SPI.INDEX.PIL4', 'SPI.INDEX.PIL5', 'SPI.INDEX',
       'SPI.DIM1.5.INDEX', 'SPI.DIM2.1.INDEX', 'SPI.DIM2.2.INDEX',
       'SPI.DIM2.4.INDEX', 'SPI.DIM3.1.INDEX', 'SPI.DIM3.2.INDEX',
       'SPI.DIM3.3.INDEX', 'SPI.DIM3.4.INDEX', 'SPI.DIM4.1.CEN.INDEX',
       'SPI.DIM4.1.SVY.INDEX', 'SPI.DIM4.2.INDEX', 'SPI.DIM4.3.INDEX',
       'SPI.DIM5.1.INDEX', 'SPI.DIM5.2.INDEX', 'SPI.DIM5.5.INDEX',
       'SPI.D1.5.POV', 'SPI.D1.5.CHLD.MORT', 'SPI.D1.5.DT.TDS.DPPF.XP.ZS',
       'SPI.D1.5.SAFE.MAN.WATER', 'SPI.D1.5.LFP', 'SPI.D2.1.GDDS',
       'SPI.D2.2.Machine.readable', 'SPI.D2.2.Non.proprietary',
       'SPI.D2.2.Download.options', 'SPI.D2.2.Metadata.available',
       'SPI.D2.2.Terms.of.use', 'SPI.D2.2.Openness.subscore', 'SPI.D2.4.NADA',
       'SPI.D3.1.POV', 'SPI.D3.2.HNGR', 'SPI.D3.3.HLTH', 'SPI.D3.4.EDUC',
       'SPI.D3.5.GEND', 'SPI.D3.6.WTRS', 'SPI.D3.7.ENRG', 'SPI.D3.8.WORK',
       'SPI.D3.9.INDY', 'SPI.D3.10.NEQL', 'SPI.D3.11.CITY', 'SPI.D3.12.CNSP',
       'SPI.D3.15.LAND', 'SPI.D3.16.INST', 'SPI.D3.17.PTNS', 'SPI.D3.13.CLMT',
       'SPI.D4.1.1.POPU', 'SPI.D4.1.2.AGRI', 'SPI.D4.1.3.BIZZ',
       'SPI.D4.1.4.HOUS', 'SPI.D4.1.5.AGSVY', 'SPI.D4.1.6.LABR',
       'SPI.D4.1.7.HLTH', 'SPI.D4.1.8.BZSVY', 'SPI.D4.2.3.CRVS',
       'SPI.D4.3.GEO.first.admin.level', 'SPI.D5.1.DILG', 'SPI.D5.2.1.SNAU',
       'SPI.D5.2.2.NABY', 'SPI.D5.2.3.CNIN', 'SPI.D5.2.4.CPIBY',
       'SPI.D5.2.5.HOUS', 'SPI.D5.2.6.EMPL', 'SPI.D5.2.7.CGOV',
       'SPI.D5.2.8.FINA', 'SPI.D5.2.9.MONY', 'SPI.D5.2.10.GSBP',
       'SPI.D5.5.DIFI', 'income', 'region', 'weights', 'population'],
      dtype='object')
```

info

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3488 entries, 0 to 3487
Data columns (total 79 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   country                       3488 non-null   object 
 1   iso3c                         3488 non-null   object 
 2   date                          3488 non-null   int64  
 3   SPI.INDEX.PIL1                3488 non-null   float64
 4   SPI.INDEX.PIL2                3488 non-null   float64
 5   SPI.INDEX.PIL3                3488 non-null   float64
 6   SPI.INDEX.PIL4                3488 non-null   float64
 7   SPI.INDEX.PIL5                3488 non-null   float64
 8   SPI.INDEX                     3488 non-null   float64
 9   SPI.DIM1.5.INDEX              3488 non-null   float64
 10  SPI.DIM2.1.INDEX              3488 non-null   float64
 11  SPI.DIM2.2.INDEX              3488 non-null   float64
 12  SPI.DIM2.4.INDEX              3488 non-null   float64
 13  SPI.DIM3.1.INDEX              3488 non-null   float64
 14  SPI.DIM3.2.INDEX              3488 non-null   float64
 15  SPI.DIM3.3.INDEX              3488 non-null   float64
 16  SPI.DIM3.4.INDEX              3488 non-null   float64
 17  SPI.DIM4.1.CEN.INDEX          3488 non-null   float64
 18  SPI.DIM4.1.SVY.INDEX          3488 non-null   float64
 19  SPI.DIM4.2.INDEX              3488 non-null   float64
 20  SPI.DIM4.3.INDEX              3488 non-null   float64
 21  SPI.DIM5.1.INDEX              3488 non-null   int64  
 22  SPI.DIM5.2.INDEX              3488 non-null   float64
 23  SPI.DIM5.5.INDEX              3488 non-null   int64  
 24  SPI.D1.5.POV                  3488 non-null   float64
 25  SPI.D1.5.CHLD.MORT            3488 non-null   int64  
 26  SPI.D1.5.DT.TDS.DPPF.XP.ZS    3488 non-null   float64
 27  SPI.D1.5.SAFE.MAN.WATER       3488 non-null   float64
 28  SPI.D1.5.LFP                  3488 non-null   float64
 29  SPI.D2.1.GDDS                 3488 non-null   float64
 30  SPI.D2.2.Machine.readable     3488 non-null   float64
 31  SPI.D2.2.Non.proprietary      3488 non-null   float64
 32  SPI.D2.2.Download.options     3488 non-null   float64
 33  SPI.D2.2.Metadata.available   3488 non-null   float64
 34  SPI.D2.2.Terms.of.use         3488 non-null   float64
 35  SPI.D2.2.Openness.subscore    3488 non-null   float64
 36  SPI.D2.4.NADA                 3488 non-null   float64
 37  SPI.D3.1.POV                  3488 non-null   float64
 38  SPI.D3.2.HNGR                 3488 non-null   float64
 39  SPI.D3.3.HLTH                 3488 non-null   float64
 40  SPI.D3.4.EDUC                 3488 non-null   float64
 41  SPI.D3.5.GEND                 3488 non-null   float64
 42  SPI.D3.6.WTRS                 3488 non-null   float64
 43  SPI.D3.7.ENRG                 3488 non-null   float64
 44  SPI.D3.8.WORK                 3488 non-null   float64
 45  SPI.D3.9.INDY                 3488 non-null   float64
 46  SPI.D3.10.NEQL                3488 non-null   float64
 47  SPI.D3.11.CITY                3488 non-null   float64
 48  SPI.D3.12.CNSP                3488 non-null   float64
 49  SPI.D3.15.LAND                3488 non-null   float64
 50  SPI.D3.16.INST                3488 non-null   float64
 51  SPI.D3.17.PTNS                3488 non-null   float64
 52  SPI.D3.13.CLMT                3488 non-null   float64
 53  SPI.D4.1.1.POPU               3488 non-null   float64
 54  SPI.D4.1.2.AGRI               3488 non-null   float64
 55  SPI.D4.1.3.BIZZ               3488 non-null   float64
 56  SPI.D4.1.4.HOUS               3488 non-null   float64
```

```
 57  SPI.D4.1.5.AGSVY               3488 non-null   float64
 58  SPI.D4.1.6.LABR                3488 non-null   float64
 59  SPI.D4.1.7.HLTH                3488 non-null   float64
 60  SPI.D4.1.8.BZSVY               3488 non-null   float64
 61  SPI.D4.2.3.CRVS                3488 non-null   float64
 62  SPI.D4.3.GEO.first.admin.level 3488 non-null   float64
 63  SPI.D5.1.DILG                  3488 non-null   float64
 64  SPI.D5.2.1.SNAU                3488 non-null   float64
 65  SPI.D5.2.2.NABY                3488 non-null   float64
 66  SPI.D5.2.3.CNIN                3488 non-null   float64
 67  SPI.D5.2.4.CPIBY               3488 non-null   float64
 68  SPI.D5.2.5.HOUS                3488 non-null   float64
 69  SPI.D5.2.6.EMPL                3488 non-null   float64
 70  SPI.D5.2.7.CGOV                3488 non-null   float64
 71  SPI.D5.2.8.FINA                3488 non-null   float64
 72  SPI.D5.2.9.MONY                3488 non-null   float64
 73  SPI.D5.2.10.GSBP               3488 non-null   float64
 74  SPI.D5.5.DIFI                  3488 non-null   float64
 75  income                         3488 non-null   object
 76  region                         3488 non-null   object
 77  weights                        3488 non-null   int64
 78  population                     3488 non-null   float64
dtypes: float64(70), int64(5), object(4)
memory usage: 2.1+ MB
```

In [17]:

```
data=df[['population' ,'income','SPI.INDEX.PIL1', 'SPI.INDEX.PIL2',
        'SPI.INDEX.PIL3', 'SPI.INDEX.PIL4', 'SPI.INDEX.PIL5', 'SPI.INDEX']]
data
```

Out[17]:

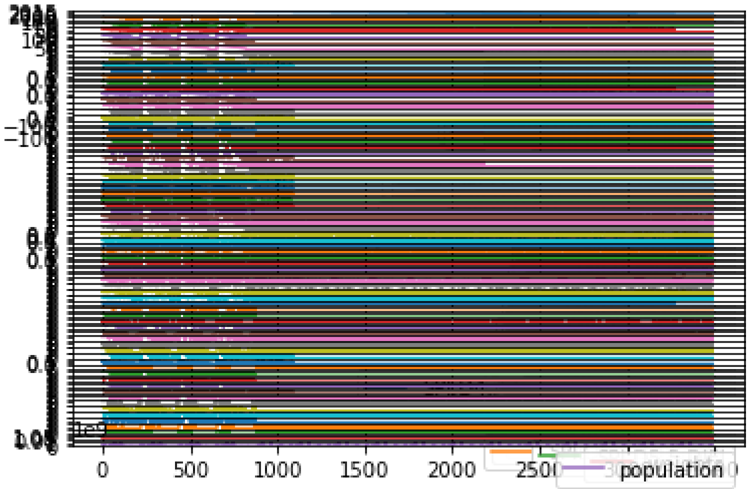
	population	income	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.INDE
0	5347896.0	High income	100.0	92.233333	77.56875	80.666667	
1	60297396.0	High income	100.0	91.866667	75.28750	81.825000	
2	8877067.0	High income	100.0	91.300000	74.55000	79.750000	
3	37970874.0	High income	100.0	95.100000	70.53750	79.716667	
4	2087946.0	High income	100.0	96.933333	76.28125	71.441667	
...	...	...	...	...	...	...	
3483	108467.0	High income	20.0	0.000000	0.00000	0.000000	
3484	3236626.0	Lower middle income	20.0	0.000000	0.00000	0.000000	
3485	19540098.0	Low income	20.0	0.000000	0.00000	0.000000	
3486	11550642.0	Lower middle income	40.0	0.000000	0.00000	0.000000	
3487	12019912.0	Lower middle income	20.0	0.000000	0.00000	0.000000	

3488 rows × 8 columns

Line chart

In [11]:
```
df.plot.line(subplots=True)
```

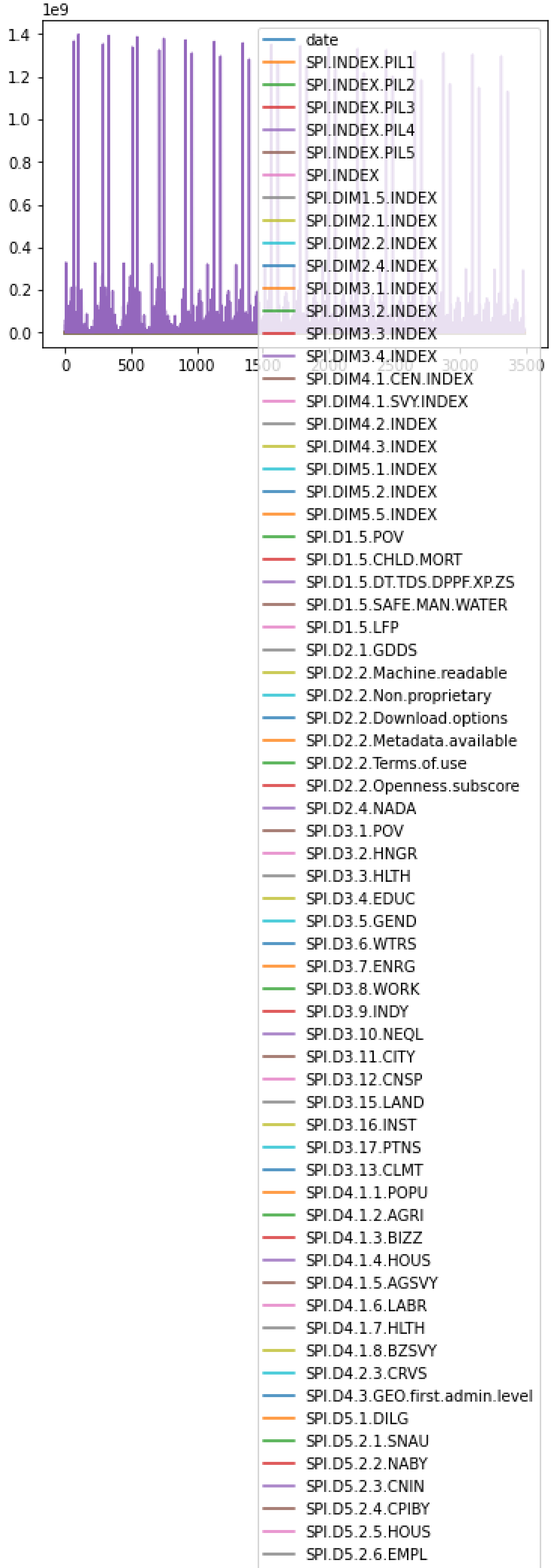
Out[11]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)
```

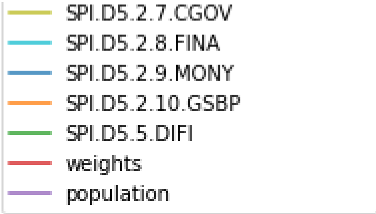


Line chart

In [12]:
```
df.plot.line()
```

Out[12]:
```
<AxesSubplot:>
```

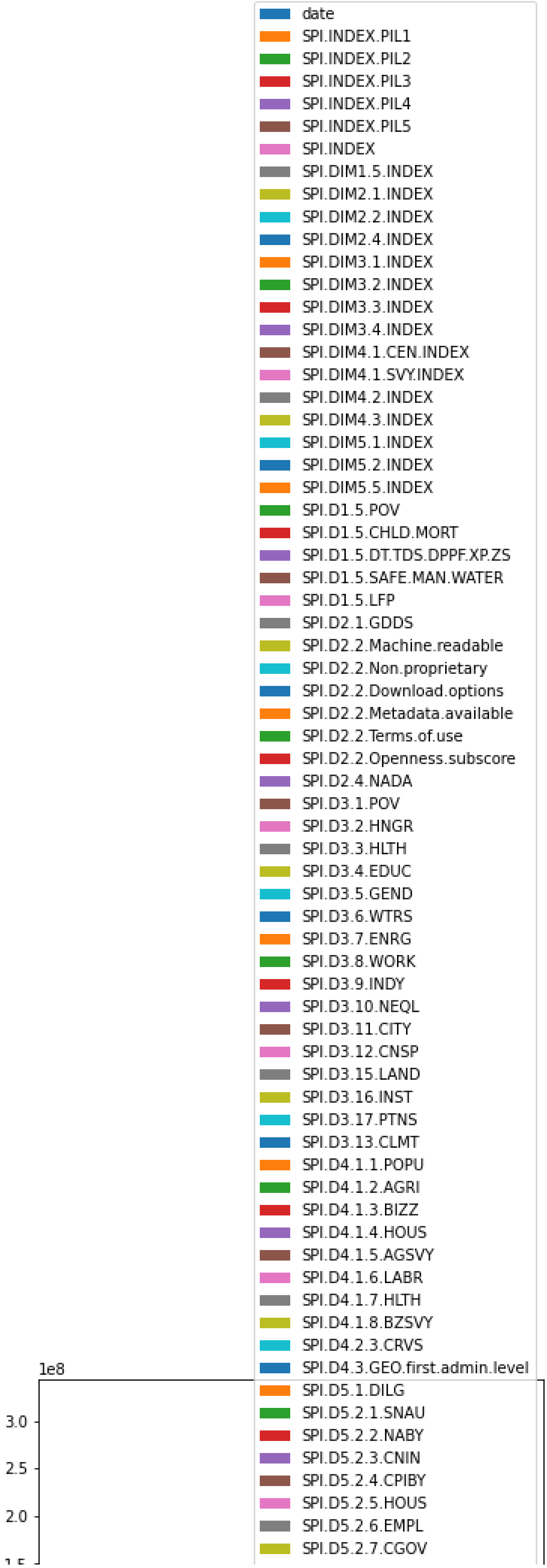

1e9
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
3000
3500
date
SPI.INDEX.PIL1
SPI.INDEX.PIL2
SPI.INDEX.PIL3
SPI.INDEX.PIL4
SPI.INDEX.PIL5
SPI.INDEX
SPI.DIM1.5.INDEX
SPI.DIM2.1.INDEX
SPI.DIM2.2.INDEX
SPI.DIM2.4.INDEX
SPI.DIM3.1.INDEX
SPI.DIM3.2.INDEX
SPI.DIM3.3.INDEX
SPI.DIM3.4.INDEX
SPI.DIM4.1.CEN.INDEX
SPI.DIM4.1.SVY.INDEX
SPI.DIM4.2.INDEX
SPI.DIM4.3.INDEX
SPI.DIM5.1.INDEX
SPI.DIM5.2.INDEX
SPI.DIM5.5.INDEX
SPI.D1.5.POV
SPI.D1.5.CHLD.MORT
SPI.D1.5.DT.TDS.DPPF.XP.ZS
SPI.D1.5.SAFE.MAN.WATER
SPI.D1.5.LFP
SPI.D2.1.GDDS
SPI.D2.2.Machine.readable
SPI.D2.2.Non.proprietary
SPI.D2.2.Download.options
SPI.D2.2.Metadata.available
SPI.D2.2.Terms.of.use
SPI.D2.2.Openness.subscore
SPI.D2.4.NADA
SPI.D3.1.POV
SPI.D3.2.HNGR
SPI.D3.3.HLTH
SPI.D3.4.EDUC
SPI.D3.5.GEND
SPI.D3.6.WTRS
SPI.D3.7.ENRG
SPI.D3.8.WORK
SPI.D3.9.INDY
SPI.D3.10.NEQL
SPI.D3.11.CITY
SPI.D3.12.CNSP
SPI.D3.15.LAND
SPI.D3.16.INST
SPI.D3.17.PTNS
SPI.D3.13.CLMT
SPI.D4.1.1.POPU
SPI.D4.1.2.AGRI
SPI.D4.1.3.BIZZ
SPI.D4.1.4.HOUS
SPI.D4.1.5.AGSVY
SPI.D4.1.6.LABR
SPI.D4.1.7.HLTH
SPI.D4.1.8.BZSVY
SPI.D4.2.3.CRVS
SPI.D4.3.GEO.first.admin.level
SPI.D5.1.DILG
SPI.D5.2.1.SNAU
SPI.D5.2.2.NABY
SPI.D5.2.3.CNIN
SPI.D5.2.4.CPIBY
SPI.D5.2.5.HOUS
SPI.D5.2.6.EMPL

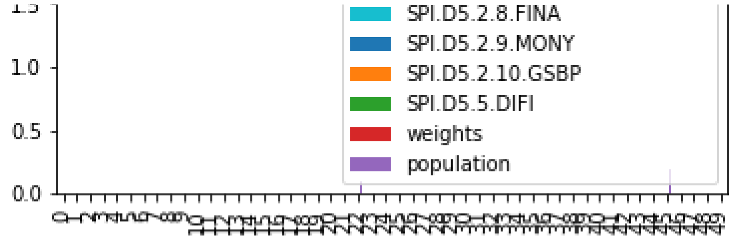


Bar chart

In [13]:
```
dat.plot.bar()
```

Out[13]: <AxesSubplot:>


date
SPI.INDEX.PIL1
SPI.INDEX.PIL2
SPI.INDEX.PIL3
SPI.INDEX.PIL4
SPI.INDEX.PIL5
SPI.INDEX
SPI.DIM1.5.INDEX
SPI.DIM2.1.INDEX
SPI.DIM2.2.INDEX
SPI.DIM2.4.INDEX
SPI.DIM3.1.INDEX
SPI.DIM3.2.INDEX
SPI.DIM3.3.INDEX
SPI.DIM3.4.INDEX
SPI.DIM4.1.CEN.INDEX
SPI.DIM4.1.SVY.INDEX
SPI.DIM4.2.INDEX
SPI.DIM4.3.INDEX
SPI.DIM5.1.INDEX
SPI.DIM5.2.INDEX
SPI.DIM5.5.INDEX
SPI.D1.5.POV
SPI.D1.5.CHLD.MORT
SPI.D1.5.DT.TDS.DPPF.XP.ZS
SPI.D1.5.SAFE.MAN.WATER
SPI.D1.5.LFP
SPI.D2.1.GDDS
SPI.D2.2.Machine.readable
SPI.D2.2.Non.proprietary
SPI.D2.2.Download.options
SPI.D2.2.Metadata.available
SPI.D2.2.Terms.of.use
SPI.D2.2.Openness.subscore
SPI.D2.4.NADA
SPI.D3.1.POV
SPI.D3.2.HNGR
SPI.D3.3.HLTH
SPI.D3.4.EDUC
SPI.D3.5.GEND
SPI.D3.6.WTRS
SPI.D3.7.ENRG
SPI.D3.8.WORK
SPI.D3.9.INDY
SPI.D3.10.NEQL
SPI.D3.11.CITY
SPI.D3.12.CNSP
SPI.D3.15.LAND
SPI.D3.16.INST
SPI.D3.17.PTNS
SPI.D3.13.CLMT
SPI.D4.1.1.POPU
SPI.D4.1.2.AGRI
SPI.D4.1.3.BIZZ
SPI.D4.1.4.HOUS
SPI.D4.1.5.AGSVY
SPI.D4.1.6.LABR
SPI.D4.1.7.HLTH
SPI.D4.1.8.BZSVY
SPI.D4.2.3.CRVS
SPI.D4.3.GEO.first.admin.level
SPI.D5.1.DILG
SPI.D5.2.1.SNAU
SPI.D5.2.2.NABY
SPI.D5.2.3.CNIN
SPI.D5.2.4.CPIBY
SPI.D5.2.5.HOUS
SPI.D5.2.6.EMPL
SPI.D5.2.7.CGOV
1e8
3.0
2.5
2.0



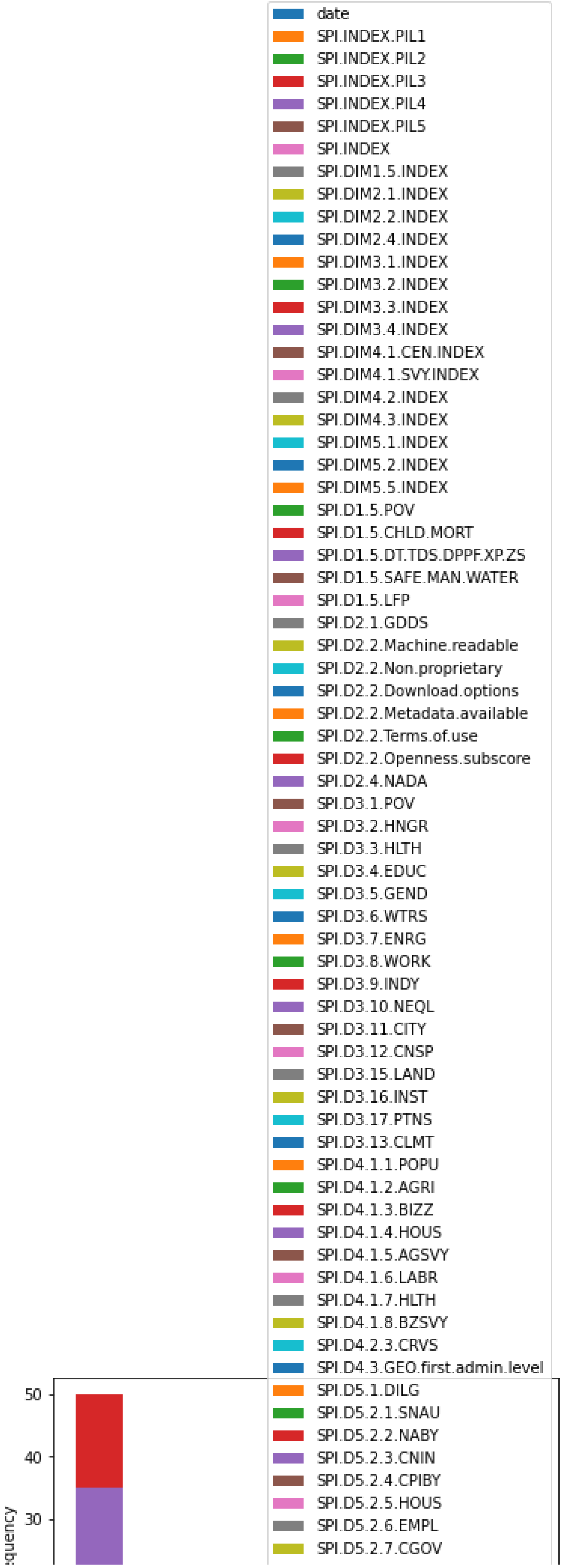
Histogram

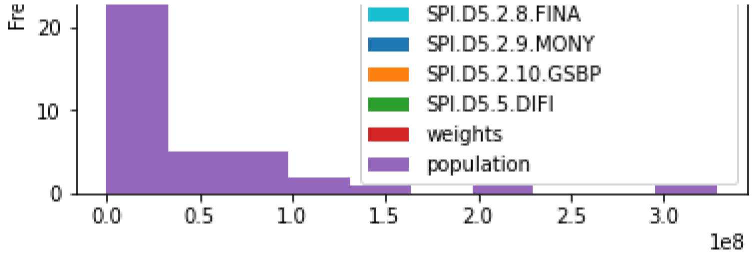
In [14]:

```
dat.plot.hist()
```

Out[14]:

```
<AxesSubplot:ylabel='Frequency'>
```

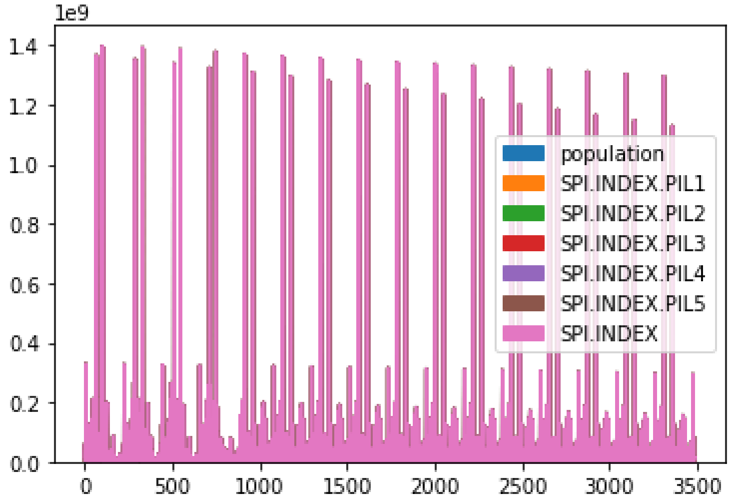

date
SPI.INDEX.PIL1
SPI.INDEX.PIL2
SPI.INDEX.PIL3
SPI.INDEX.PIL4
SPI.INDEX.PIL5
SPI.INDEX
SPI.DIM1.5.INDEX
SPI.DIM2.1.INDEX
SPI.DIM2.2.INDEX
SPI.DIM2.4.INDEX
SPI.DIM3.1.INDEX
SPI.DIM3.2.INDEX
SPI.DIM3.3.INDEX
SPI.DIM3.4.INDEX
SPI.DIM4.1.CEN.INDEX
SPI.DIM4.1.SVY.INDEX
SPI.DIM4.2.INDEX
SPI.DIM4.3.INDEX
SPI.DIM5.1.INDEX
SPI.DIM5.2.INDEX
SPI.DIM5.5.INDEX
SPI.D1.5.POV
SPI.D1.5.CHLD.MORT
SPI.D1.5.DT.TDS.DPPF.XP.ZS
SPI.D1.5.SAFE.MAN.WATER
SPI.D1.5.LFP
SPI.D2.1.GDDS
SPI.D2.2.Machine.readable
SPI.D2.2.Non.proprietary
SPI.D2.2.Download.options
SPI.D2.2.Metadata.available
SPI.D2.2.Terms.of.use
SPI.D2.2.Openness.subscore
SPI.D2.4.NADA
SPI.D3.1.POV
SPI.D3.2.HNGR
SPI.D3.3.HLTH
SPI.D3.4.EDUC
SPI.D3.5.GEND
SPI.D3.6.WTRS
SPI.D3.7.ENRG
SPI.D3.8.WORK
SPI.D3.9.INDY
SPI.D3.10.NEQL
SPI.D3.11.CITY
SPI.D3.12.CNSP
SPI.D3.15.LAND
SPI.D3.16.INST
SPI.D3.17.PTNS
SPI.D3.13.CLMT
SPI.D4.1.1.POPU
SPI.D4.1.2.AGRI
SPI.D4.1.3.BIZZ
SPI.D4.1.4.HOUS
SPI.D4.1.5.AGSVY
SPI.D4.1.6.LABR
SPI.D4.1.7.HLTH
SPI.D4.1.8.BZSVY
SPI.D4.2.3.CRVS
SPI.D4.3.GEO.first.admin.level
SPI.D5.1.DILG
SPI.D5.2.1.SNAU
SPI.D5.2.2.NABY
SPI.D5.2.3.CNIN
SPI.D5.2.4.CPIBY
SPI.D5.2.5.HOUS
SPI.D5.2.6.EMPL
SPI.D5.2.7.CGOV
50
40
30



Area chart

In [18]:
```
data.plot.area()
```

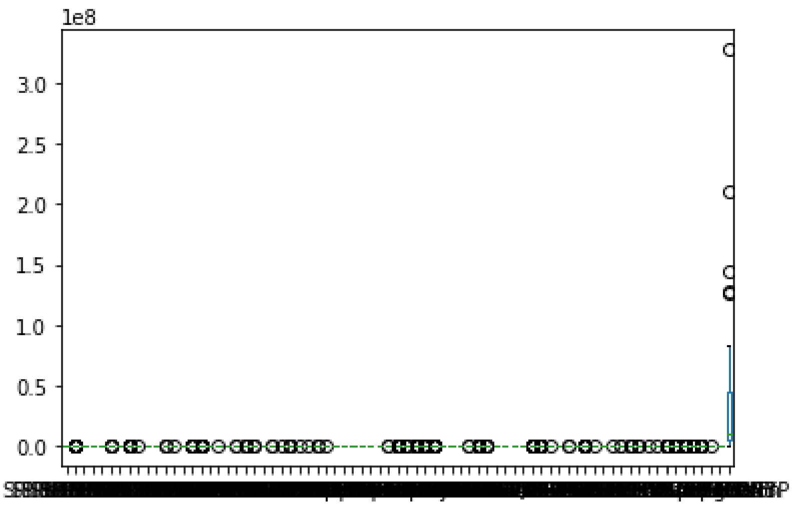
Out[18]: <AxesSubplot:>



Box chart

In [19]:
```
dat.plot.box()
```

Out[19]: <AxesSubplot:>



Pie chart

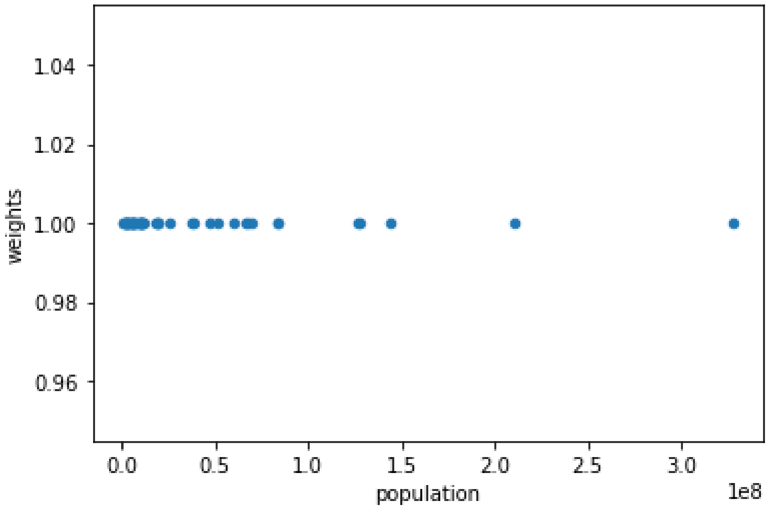
In [21]:
```
dat.plot.pie(y='population' )
```

Out[21]: <AxesSubplot:ylabel='population'>

Scatter chart

In [23]:
```
dat.plot.scatter(x='population' ,y='weights')
```

Out[23]: <AxesSubplot:xlabel='population', ylabel='weights'>



In [24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3488 entries, 0 to 3487
Data columns (total 79 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   country                         3488 non-null   object 
 1   iso3c                           3488 non-null   object 
 2   date                            3488 non-null   int64  
 3   SPI.INDEX.PIL1                  3488 non-null   float64
 4   SPI.INDEX.PIL2                  3488 non-null   float64
 5   SPI.INDEX.PIL3                  3488 non-null   float64
 6   SPI.INDEX.PIL4                  3488 non-null   float64
 7   SPI.INDEX.PIL5                  3488 non-null   float64
 8   SPI.INDEX                       3488 non-null   float64
 9   SPI.DIM1.5.INDEX                3488 non-null   float64
 10  SPI.DIM2.1.INDEX                3488 non-null   float64
 11  SPI.DIM2.2.INDEX                3488 non-null   float64
 12  SPI.DIM2.4.INDEX                3488 non-null   float64
 13  SPI.DIM3.1.INDEX                3488 non-null   float64
 14  SPI.DIM3.2.INDEX                3488 non-null   float64
 15  SPI.DIM3.3.INDEX                3488 non-null   float64
 16  SPI.DIM3.4.INDEX                3488 non-null   float64
 17  SPI.DIM4.1.CEN.INDEX            3488 non-null   float64
 18  SPI.DIM4.1.SVY.INDEX            3488 non-null   float64
 19  SPI.DIM4.2.INDEX                3488 non-null   float64
 20  SPI.DIM4.3.INDEX                3488 non-null   float64
 21  SPI.DIM5.1.INDEX                3488 non-null   int64  
 22  SPI.DIM5.2.INDEX                3488 non-null   float64
 23  SPI.DIM5.5.INDEX                3488 non-null   int64  
 24  SPI.D1.5.POV                    3488 non-null   float64
 25  SPI.D1.5.CHLD.MORT              3488 non-null   int64  
 26  SPI.D1.5.DT.TDS.DPPF.XP.ZS      3488 non-null   float64
 27  SPI.D1.5.SAFE.MAN.WATER         3488 non-null   float64
 28  SPI.D1.5.LFP                    3488 non-null   float64
 29  SPI.D2.1.GDDS                   3488 non-null   float64
 30  SPI.D2.2.Machine.readable       3488 non-null   float64
 31  SPI.D2.2.Non.proprietary        3488 non-null   float64
 32  SPI.D2.2.Download.options       3488 non-null   float64
 33  SPI.D2.2.Metadata.available     3488 non-null   float64
 34  SPI.D2.2.Terms.of.use           3488 non-null   float64
 35  SPI.D2.2.Openness.subscore      3488 non-null   float64
 36  SPI.D2.4.NADA                   3488 non-null   float64
 37  SPI.D3.1.POV                    3488 non-null   float64
 38  SPI.D3.2.HNGR                   3488 non-null   float64
 39  SPI.D3.3.HLTH                   3488 non-null   float64
 40  SPI.D3.4.EDUC                   3488 non-null   float64
 41  SPI.D3.5.GEND                   3488 non-null   float64
 42  SPI.D3.6.WTRS                   3488 non-null   float64
```

```
 43  SPI.D3.7.ENRG                   3488 non-null   float64
 44  SPI.D3.8.WORK                   3488 non-null   float64
 45  SPI.D3.9.INDY                   3488 non-null   float64
 46  SPI.D3.10.NEQL                  3488 non-null   float64
 47  SPI.D3.11.CITY                  3488 non-null   float64
 48  SPI.D3.12.CNSP                  3488 non-null   float64
 49  SPI.D3.15.LAND                  3488 non-null   float64
 50  SPI.D3.16.INST                  3488 non-null   float64
 51  SPI.D3.17.PTNS                  3488 non-null   float64
 52  SPI.D3.13.CLMT                  3488 non-null   float64
 53  SPI.D4.1.1.POPU                 3488 non-null   float64
 54  SPI.D4.1.2.AGRI                 3488 non-null   float64
 55  SPI.D4.1.3.BIZZ                 3488 non-null   float64
 56  SPI.D4.1.4.HOUS                 3488 non-null   float64
 57  SPI.D4.1.5.AGSVY                3488 non-null   float64
 58  SPI.D4.1.6.LABR                 3488 non-null   float64
 59  SPI.D4.1.7.HLTH                 3488 non-null   float64
 60  SPI.D4.1.8.BZSVY                3488 non-null   float64
 61  SPI.D4.2.3.CRVS                 3488 non-null   float64
 62  SPI.D4.3.GEO.first.admin.level  3488 non-null   float64
 63  SPI.D5.1.DILG                   3488 non-null   float64
 64  SPI.D5.2.1.SNAU                 3488 non-null   float64
 65  SPI.D5.2.2.NABY                 3488 non-null   float64
 66  SPI.D5.2.3.CNIN                 3488 non-null   float64
 67  SPI.D5.2.4.CPIBY                3488 non-null   float64
 68  SPI.D5.2.5.HOUS                 3488 non-null   float64
 69  SPI.D5.2.6.EMPL                 3488 non-null   float64
 70  SPI.D5.2.7.CGOV                 3488 non-null   float64
 71  SPI.D5.2.8.FINA                 3488 non-null   float64
 72  SPI.D5.2.9.MONY                 3488 non-null   float64
 73  SPI.D5.2.10.GSBP                3488 non-null   float64
 74  SPI.D5.5.DIFI                   3488 non-null   float64
 75  income                          3488 non-null   object
 76  region                          3488 non-null   object
 77  weights                         3488 non-null   int64
 78  population                      3488 non-null   float64
dtypes: float64(70), int64(5), object(4)
memory usage: 2.1+ MB
```

In [25]:

```
df.columns
```

Out[25]:

```
Index(['country', 'iso3c', 'date', 'SPI.INDEX.PIL1', 'SPI.INDEX.PIL2',
       'SPI.INDEX.PIL3', 'SPI.INDEX.PIL4', 'SPI.INDEX.PIL5', 'SPI.INDEX',
       'SPI.DIM1.5.INDEX', 'SPI.DIM2.1.INDEX', 'SPI.DIM2.2.INDEX',
       'SPI.DIM2.4.INDEX', 'SPI.DIM3.1.INDEX', 'SPI.DIM3.2.INDEX',
       'SPI.DIM3.3.INDEX', 'SPI.DIM3.4.INDEX', 'SPI.DIM4.1.CEN.INDEX',
       'SPI.DIM4.1.SVY.INDEX', 'SPI.DIM4.2.INDEX', 'SPI.DIM4.3.INDEX',
       'SPI.DIM5.1.INDEX', 'SPI.DIM5.2.INDEX', 'SPI.DIM5.5.INDEX',
       'SPI.D1.5.POV', 'SPI.D1.5.CHLD.MORT', 'SPI.D1.5.DT.TDS.DPPF.XP.ZS',
       'SPI.D1.5.SAFE.MAN.WATER', 'SPI.D1.5.LFP', 'SPI.D2.1.GDDS',
       'SPI.D2.2.Machine.readable', 'SPI.D2.2.Non.proprietary',
       'SPI.D2.2.Download.options', 'SPI.D2.2.Metadata.available',
       'SPI.D2.2.Terms.of.use', 'SPI.D2.2.Openness.subscore', 'SPI.D2.4.NADA',
       'SPI.D3.1.POV', 'SPI.D3.2.HNGR', 'SPI.D3.3.HLTH', 'SPI.D3.4.EDUC',
       'SPI.D3.5.GEND', 'SPI.D3.6.WTRS', 'SPI.D3.7.ENRG', 'SPI.D3.8.WORK',
       'SPI.D3.9.INDY', 'SPI.D3.10.NEQL', 'SPI.D3.11.CITY', 'SPI.D3.12.CNSP',
       'SPI.D3.15.LAND', 'SPI.D3.16.INST', 'SPI.D3.17.PTNS', 'SPI.D3.13.CLMT',
       'SPI.D4.1.1.POPU', 'SPI.D4.1.2.AGRI', 'SPI.D4.1.3.BIZZ',
       'SPI.D4.1.4.HOUS', 'SPI.D4.1.5.AGSVY', 'SPI.D4.1.6.LABR',
       'SPI.D4.1.7.HLTH', 'SPI.D4.1.8.BZSVY', 'SPI.D4.2.3.CRVS',
       'SPI.D4.3.GEO.first.admin.level', 'SPI.D5.1.DILG', 'SPI.D5.2.1.SNAU',
       'SPI.D5.2.2.NABY', 'SPI.D5.2.3.CNIN', 'SPI.D5.2.4.CPIBY',
       'SPI.D5.2.5.HOUS', 'SPI.D5.2.6.EMPL', 'SPI.D5.2.7.CGOV',
       'SPI.D5.2.8.FINA', 'SPI.D5.2.9.MONY', 'SPI.D5.2.10.GSBP',
       'SPI.D5.5.DIFI', 'income', 'region', 'weights', 'population'],
      dtype='object')
```

In [26]:
```
df.describe()
```

Out[26]:

	date	SPI.INDEX.PIL1	SPI.INDEX.PIL2	SPI.INDEX.PIL3	SPI.INDEX.PIL4	SPI.INDEX.PIL5
count	3488.000000	3488.000000	3488.000000	3488.000000	3488.000000	3488.000000
mean	2011.500000	45.928440	11.868014	47.938459	9.662335	10.688073
std	4.610433	26.206034	26.404537	21.481061	21.334042	24.446254
min	2004.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2007.750000	20.000000	0.000000	39.950000	0.000000	0.000000
50%	2011.500000	40.000000	0.000000	53.078125	0.000000	0.000000
75%	2015.250000	60.000000	0.000000	63.629688	0.000000	0.000000
max	2019.000000	100.000000	100.000000	90.937500	92.600000	100.000000

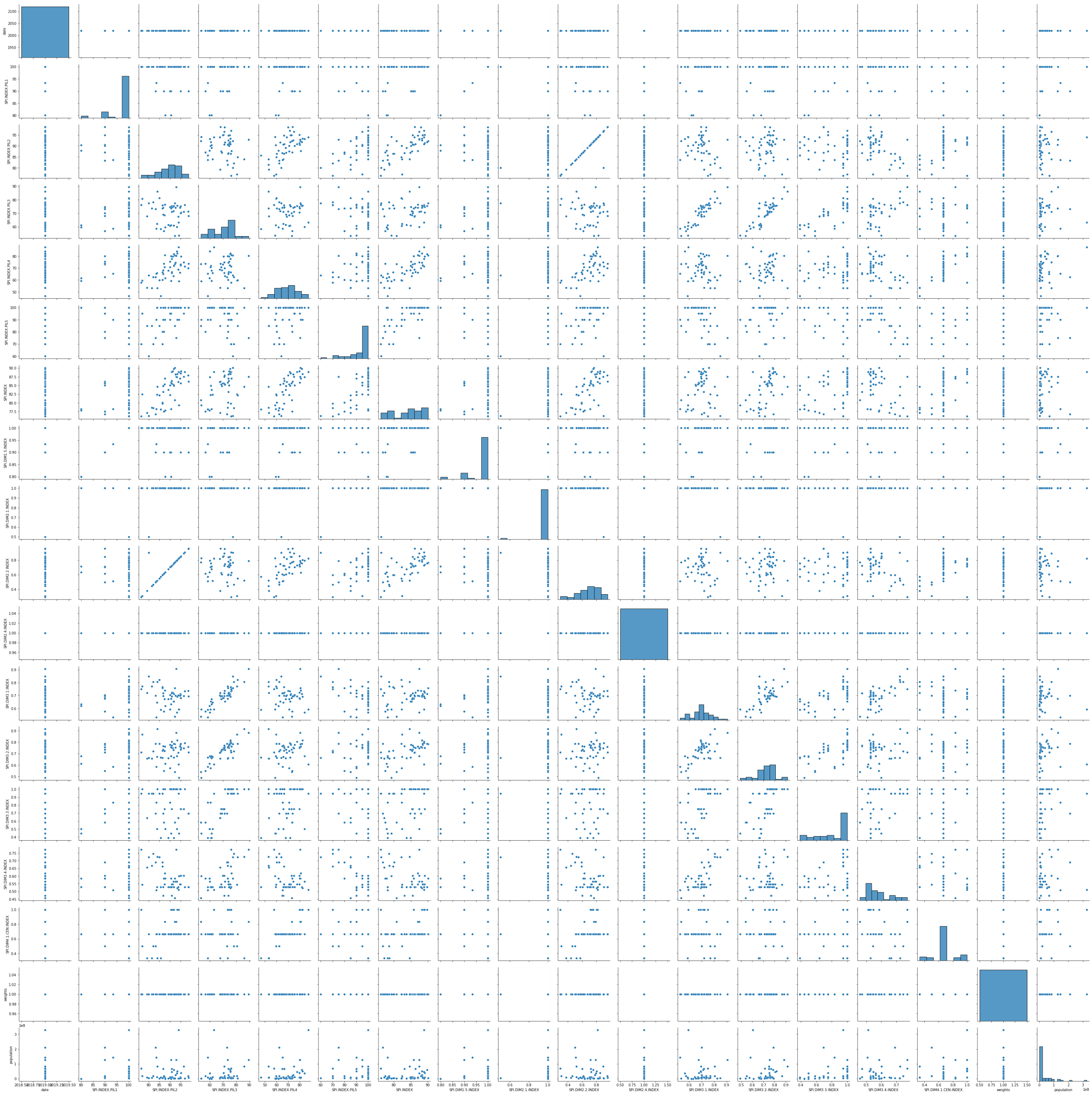
8 rows × 75 columns

In [29]:
```
df1=df[[ 'date', 'SPI.INDEX.PIL1', 'SPI.INDEX.PIL2',
        'SPI.INDEX.PIL3', 'SPI.INDEX.PIL4', 'SPI.INDEX.PIL5', 'SPI.INDEX',
        'SPI.DIM1.5.INDEX', 'SPI.DIM2.1.INDEX', 'SPI.DIM2.2.INDEX',
        'SPI.DIM2.4.INDEX', 'SPI.DIM3.1.INDEX', 'SPI.DIM3.2.INDEX',
        'SPI.DIM3.3.INDEX', 'SPI.DIM3.4.INDEX', 'SPI.DIM4.1.CEN.INDEX', 'income', 'we
```

EDA AND VISUALIZATION

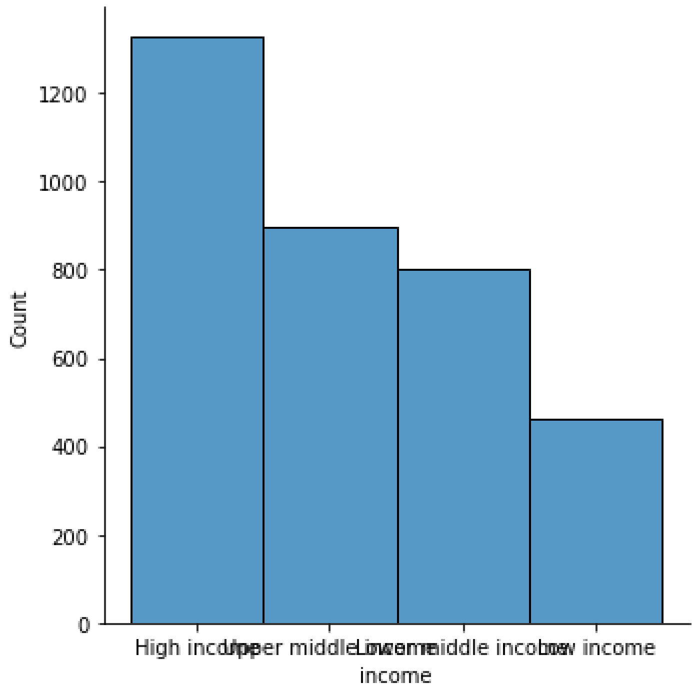
In [30]:
```
sns.pairplot(df1[0:50])
```

Out[30]: <seaborn.axisgrid.PairGrid at 0x1d8ff336f10>



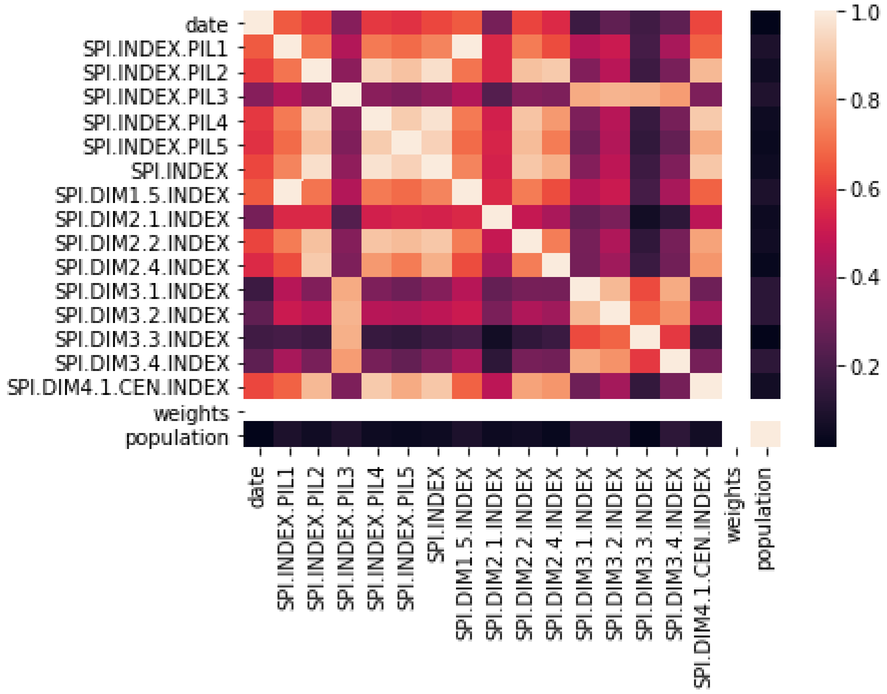
In [31]:
```
sns.displot(df['income'])
```

Out[31]: <seaborn.axisgrid.FacetGrid at 0x1d8c3fc2fd0>



In [33]:
```
sns.heatmap(df1.corr())
```

Out[33]: <AxesSubplot:>



TO TRAIN THE MODEL AND MODEL BULDING

In [183…
```
x=df[['weights','SPI.DIM5.1.INDEX']]
y=df['population']
```

In [184…
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

Linear Regression

In [185…

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[185…

```
LinearRegression()
```

In [186…

```
lr.intercept_
```

Out[186…

```
31943906.726010595
```

In [187…

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[187…

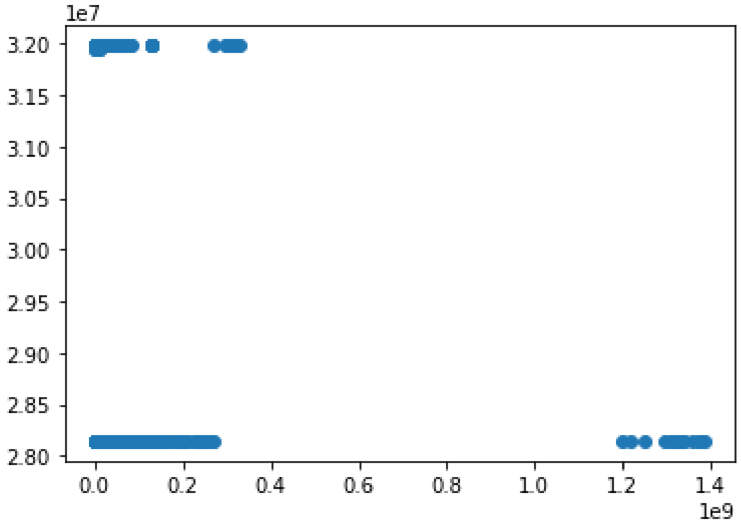
	Co-efficient
weights	0.000000
SPI.DIM5.1.INDEX	38385.429968

In [188…

```
prediction =lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[188…

```
<matplotlib.collections.PathCollection at 0x1d90aceca30>
```



ACCURACY

In [189…

```
lr.score(x_test,y_test)
```

Out[189…

```
-0.005454678054893858
```

In [190…

```
lr.score(x_train,y_train)
```

Out[190…
```
0.00017360799921994907
```

Ridge and Lasso

In [191…
```
from sklearn.linear_model import Ridge,Lasso
```

In [192…
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[192…
```
Ridge(alpha=10)
```

Accuracy(Ridge)

In [193…
```
rr.score(x_test,y_test)
```

Out[193…
```
-0.005454676808237746
```

In [194…
```
rr.score(x_train,y_train)
```

Out[194…
```
0.00017360799921894987
```

In [195…
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[195…
```
Lasso(alpha=10)
```

In [196…
```
la.score(x_train,y_train)
```

Out[196…
```
0.0001736079992200601
```

Accuracy(Lasso)

In [197…
```
la.score(x_test,y_test)
```

Out[197…
```
-0.005454677975616384
```

Elastic Net regression

In [198…
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[198…
```
ElasticNet()
```

In [199…
```
en.coef_
```

Out[199…
```
array([    0.        , 38373.21575689])
```

In [200…
```
en.intercept_
```

Out[200…
```
31942936.094027862
```

In [201…
```
prediction=en.predict(x_test)
```

In [202…
```
en.score(x_test,y_test)
```

Out[202…
```
-0.005454525954040834
```

Evaluation Metrics

In [203…
```
from sklearn import metrics
print(metrics.mean_absolute_error(y_test,prediction))
print(metrics.mean_squared_error(y_test,prediction))
print(np.sqrt(metrics.mean_squared_error(y_test,prediction)))
```

```
46923236.340679206
2.434104758717166e+16
156016177.32521093
```

Logistic Regression

In [204…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3488 entries, 0 to 3487
Data columns (total 79 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   country                         3488 non-null   object
 1   iso3c                           3488 non-null   object
 2   date                            3488 non-null   int64
 3   SPI.INDEX.PIL1                  3488 non-null   float64
 4   SPI.INDEX.PIL2                  3488 non-null   float64
 5   SPI.INDEX.PIL3                  3488 non-null   float64
 6   SPI.INDEX.PIL4                  3488 non-null   float64
 7   SPI.INDEX.PIL5                  3488 non-null   float64
 8   SPI.INDEX                       3488 non-null   float64
 9   SPI.DIM1.5.INDEX                3488 non-null   float64
 10  SPI.DIM2.1.INDEX                3488 non-null   float64
 11  SPI.DIM2.2.INDEX                3488 non-null   float64
 12  SPI.DIM2.4.INDEX                3488 non-null   float64
 13  SPI.DIM3.1.INDEX                3488 non-null   float64
 14  SPI.DIM3.2.INDEX                3488 non-null   float64
 15  SPI.DIM3.3.INDEX                3488 non-null   float64
 16  SPI.DIM3.4.INDEX                3488 non-null   float64
 17  SPI.DIM4.1.CEN.INDEX            3488 non-null   float64
 18  SPI.DIM4.1.SVY.INDEX            3488 non-null   float64
 19  SPI.DIM4.2.INDEX                3488 non-null   float64
 20  SPI.DIM4.3.INDEX                3488 non-null   float64
 21  SPI.DIM5.1.INDEX                3488 non-null   int64
 22  SPI.DIM5.2.INDEX                3488 non-null   float64
 23  SPI.DIM5.5.INDEX                3488 non-null   int64
```

```
 24  SPI.D1.5.POV                   3488 non-null   float64
 25  SPI.D1.5.CHLD.MORT             3488 non-null   int64
 26  SPI.D1.5.DT.TDS.DPPF.XP.ZS     3488 non-null   float64
 27  SPI.D1.5.SAFE.MAN.WATER        3488 non-null   float64
 28  SPI.D1.5.LFP                   3488 non-null   float64
 29  SPI.D2.1.GDDS                  3488 non-null   float64
 30  SPI.D2.2.Machine.readable      3488 non-null   float64
 31  SPI.D2.2.Non.proprietary       3488 non-null   float64
 32  SPI.D2.2.Download.options      3488 non-null   float64
 33  SPI.D2.2.Metadata.available    3488 non-null   float64
 34  SPI.D2.2.Terms.of.use          3488 non-null   float64
 35  SPI.D2.2.Openness.subscore     3488 non-null   float64
 36  SPI.D2.4.NADA                  3488 non-null   float64
 37  SPI.D3.1.POV                   3488 non-null   float64
 38  SPI.D3.2.HNGR                  3488 non-null   float64
 39  SPI.D3.3.HLTH                  3488 non-null   float64
 40  SPI.D3.4.EDUC                  3488 non-null   float64
 41  SPI.D3.5.GEND                  3488 non-null   float64
 42  SPI.D3.6.WTRS                  3488 non-null   float64
 43  SPI.D3.7.ENRG                  3488 non-null   float64
 44  SPI.D3.8.WORK                  3488 non-null   float64
 45  SPI.D3.9.INDY                  3488 non-null   float64
 46  SPI.D3.10.NEQL                 3488 non-null   float64
 47  SPI.D3.11.CITY                 3488 non-null   float64
 48  SPI.D3.12.CNSP                 3488 non-null   float64
 49  SPI.D3.15.LAND                 3488 non-null   float64
 50  SPI.D3.16.INST                 3488 non-null   float64
 51  SPI.D3.17.PTNS                 3488 non-null   float64
 52  SPI.D3.13.CLMT                 3488 non-null   float64
 53  SPI.D4.1.1.POPU                3488 non-null   float64
 54  SPI.D4.1.2.AGRI                3488 non-null   float64
 55  SPI.D4.1.3.BIZZ                3488 non-null   float64
 56  SPI.D4.1.4.HOUS                3488 non-null   float64
 57  SPI.D4.1.5.AGSVY               3488 non-null   float64
 58  SPI.D4.1.6.LABR                3488 non-null   float64
 59  SPI.D4.1.7.HLTH                3488 non-null   float64
 60  SPI.D4.1.8.BZSVY               3488 non-null   float64
 61  SPI.D4.2.3.CRVS                3488 non-null   float64
 62  SPI.D4.3.GEO.first.admin.level 3488 non-null   float64
 63  SPI.D5.1.DILG                  3488 non-null   float64
 64  SPI.D5.2.1.SNAU                3488 non-null   float64
 65  SPI.D5.2.2.NABY                3488 non-null   float64
 66  SPI.D5.2.3.CNIN                3488 non-null   float64
 67  SPI.D5.2.4.CPIBY               3488 non-null   float64
 68  SPI.D5.2.5.HOUS                3488 non-null   float64
 69  SPI.D5.2.6.EMPL                3488 non-null   float64
 70  SPI.D5.2.7.CGOV                3488 non-null   float64
 71  SPI.D5.2.8.FINA                3488 non-null   float64
 72  SPI.D5.2.9.MONY                3488 non-null   float64
 73  SPI.D5.2.10.GSBP               3488 non-null   float64
 74  SPI.D5.5.DIFI                  3488 non-null   float64
 75  income                         3488 non-null   object
 76  region                         3488 non-null   object
 77  weights                        3488 non-null   int64
 78  population                     3488 non-null   float64
dtypes: float64(70), int64(5), object(4)
memory usage: 2.1+ MB
```

In [205…

```
from sklearn.linear_model import LogisticRegression
```

In [206…

```
feature_matrix=df[['SPI.DIM5.5.INDEX']]
target_vector=df['SPI.DIM5.1.INDEX']
```

In [207…

```
feature_matrix.shape
```

Out[207...
```
(3488, 1)
```

In [208...
```
target_vector.shape
```

Out[208...
```
(3488,)
```

In [209...
```
from sklearn.preprocessing import StandardScaler
```

In [210...
```
fs=StandardScaler().fit_transform(feature_matrix)
```

In [211...
```
logr=LogisticRegression()
logr.fit(fs,target_vector)
```

Out[211...
```
LogisticRegression()
```

In [212...
```
observation=[[1]]
```

In [213...
```
prediction=logr.predict(observation)
print(prediction)
```

```
[1]
```

In [214...
```
logr.classes_
```

Out[214...
```
array([-99,   0,   1], dtype=int64)
```

In [215...
```
logr.score(fs,target_vector)
```

Out[215...
```
0.9905389908256881
```

In [216...
```
logr.predict_proba(observation)[0][0]
```

Out[216...
```
0.3502335092476515
```

In [217...
```
logr.predict_proba(observation)
```

Out[217...
```
array([[0.35023351, 0.07283107, 0.57693542]])
```

Random Forest

In [229...
```
df['SPI.DIM5.1.INDEX'].value_counts()
```

Out[229...
```
-99    2811
 1      650
 0       27
Name: SPI.DIM5.1.INDEX, dtype: int64
```

In [253…

```
x=df[['SPI.DIM5.5.INDEX']]
y=df['SPI.DIM5.1.INDEX']
```

In [268…

```
g1={"SPI.DIM5.1.INDEX":{'-99':5,'1':6,'0':7}}
df=df.replace(g1)
print(df)
```

```
                   country iso3c  date  SPI.INDEX.PIL1  SPI.INDEX.PIL2  \
0                   Norway   NOR  2019           100.0       92.233333
1                    Italy   ITA  2019           100.0       91.866667
2                  Austria   AUT  2019           100.0       91.300000
3                   Poland   POL  2019           100.0       95.100000
4                 Slovenia   SVN  2019           100.0       96.933333
...                    ...   ...   ...             ...             ...
3483  Virgin Islands (U.S.)   VIR  2004            20.0        0.000000
3484     West Bank and Gaza   PSE  2004            20.0        0.000000
3485          Yemen, Rep.   YEM  2004            20.0        0.000000
3486               Zambia   ZMB  2004            40.0        0.000000
3487             Zimbabwe   ZWE  2004            20.0        0.000000

      SPI.INDEX.PIL3  SPI.INDEX.PIL4  SPI.INDEX.PIL5  SPI.INDEX  \
0           77.56875       80.666667           100.0  90.093750
1           75.28750       81.825000           100.0  89.795833
2           74.55000       79.750000           100.0  89.120000
3           70.53750       79.716667           100.0  89.070833
4           76.28125       71.441667           100.0  88.931250
...              ...             ...             ...        ...
3483         0.00000        0.000000             0.0   0.000000
3484         0.00000        0.000000             0.0   0.000000
3485         0.00000        0.000000             0.0   0.000000
3486         0.00000        0.000000             0.0   0.000000
3487         0.00000        0.000000             0.0   0.000000

      SPI.DIM1.5.INDEX  ...  SPI.D5.2.6.EMPL  SPI.D5.2.7.CGOV  \
0                  1.0  ...              1.0              1.0
1                  1.0  ...              1.0              1.0
2                  1.0  ...              1.0              1.0
3                  1.0  ...              1.0              1.0
4                  1.0  ...              1.0              1.0
...                ...  ...              ...              ...
3483               0.2  ...              0.0              0.0
3484               0.2  ...              0.0              0.0
3485               0.2  ...              0.0              0.0
3486               0.4  ...              0.0              0.0
3487               0.2  ...              0.0              0.0

      SPI.D5.2.8.FINA  SPI.D5.2.9.MONY  SPI.D5.2.10.GSBP  SPI.D5.5.DIFI  \
0                 1.0              1.0               1.0            1.0
1                 1.0              1.0               1.0            1.0
2                 1.0              1.0               1.0            1.0
3                 1.0              1.0               1.0            1.0
4                 1.0              1.0               1.0            1.0
...               ...              ...               ...            ...
3483              0.0              0.0               0.0            0.0
3484              0.0              0.0               0.0            0.0
3485              0.0              0.0               0.0            0.0
3486              0.0              0.0               0.0            0.0
3487              0.0              0.0               0.0            0.0

                   income                 region  weights  population
0             High income  Europe & Central Asia        1   5347896.0
1             High income  Europe & Central Asia        1  60297396.0
2             High income  Europe & Central Asia        1   8877067.0
3             High income  Europe & Central Asia        1  37970874.0
4             High income  Europe & Central Asia        1   2087946.0
...                   ...                    ...      ...         ...
```

```
3483          High income   Latin America & Caribbean        1     108467.0
3484  Lower middle income  Middle East & North Africa        1    3236626.0
3485           Low income  Middle East & North Africa        1   19540098.0
3486  Lower middle income          Sub-Saharan Africa        1   11550642.0
3487  Lower middle income          Sub-Saharan Africa        1   12019912.0

[3488 rows x 79 columns]
```

In [269…
```
from sklearn.ensemble import RandomForestClassifier
```

In [270…
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[270…
```
RandomForestClassifier()
```

In [271…
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]
}
```

In [273…
```
from sklearn.model_selection import GridSearchCV
grid_search =GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy
grid_search.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:666: Us
erWarning: The least populated class in y has only 1 members, which is less than n_s
plits=2.
  warnings.warn(("The least populated class in y has only %d"
```

Out[273…
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [275…
```
grid_search.best_score_
```

Out[275…
```
0.0081933647507418
```

In [277…
```
rfc_best=grid_search.best_estimator_
```

Conclusion

Accuracy

linear regression

In [278…
```
lr.score(x_test,y_test)
```

Out[278…
```
-0.005454678054893858
```

Ridge regression

In [279…

```
rr.score(x_test,y_test)
```

Out[279…

```
-0.005454676808237746
```

Lasso regression

In [280…

```
la.score(x_test,y_test)
```

Out[280…

```
-0.005454677975616384
```

Elastic net regression

In [281…

```
en.score(x_test,y_test)
```

Out[281…

```
-0.005454525954040834
```

Logistic regression

In [282…

```
logr.score(fs,target_vector)
```

Out[282…

```
0.9905389908256881
```

Random forest

In [283…

```
grid_search.best_score_
```

Out[283…

```
0.0081933647507418
```

Accuracy for logistic regression is higher so it is the best fit model