```
import pandas as pd

df = pd.read_csv("/content/API_AG.LND.FRST.ZS_DS2_en_csv_v2_5358376.csv")
df.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	1990	1991	1992	1993	1994	1995	..
0	Aruba	ABW	Forest area (% of land area)	AG.LND.FRST.ZS	2.333333	2.333333	2.333333	2.333333	2.333333	2.333333	
1	Africa Eastern and Southern	AFE	Forest area (% of land area)	AG.LND.FRST.ZS	36.230006	36.078752	35.927499	35.776246	35.624993	35.473739	
2	Afghanistan	AFG	Forest area (% of land area)	AG.LND.FRST.ZS	1.852782	1.852782	1.852782	1.852782	1.852782	1.852782	
3	Africa Western and Central	AFW	Forest area (% of land area)	AG.LND.FRST.ZS	22.776908	22.658746	22.540583	22.422421	22.304258	22.186096	
4	Angola	AGO	Forest area (% of land area)	AG.LND.FRST.ZS	63.578070	63.453407	63.328745	63.204082	63.079419	62.954757	

5 rows × 36 columns

```
import pandas as pd

df2 = pd.read_csv("/content/ArableLand.csv")
df2.head()
```

	Country	Country	Indicator	Indicator Code	1960	1961	1962	1963	1964	1965

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	...	1
0	Aruba	ABW	Arable land (% of land area)	AG.LND.ARBL.ZS	NaN	11.111111	11.111111	11.111111	11.111111	11.111111	...	1
1	Africa Eastern and Southern	AFE	Arable land (% of land area)	AG.LND.ARBL.ZS	NaN	4.702843	4.754588	4.866723	4.918674	4.972683	...	7
2	Afghanistan	AFG	Arable land (% of land area)	AG.LND.ARBL.ZS	NaN	11.728991	11.805651	11.882311	11.958972	11.958972	...	11
3	Africa Western and Central	AFW	Arable land (% of land area)	AG.LND.ARBL.ZS	NaN	6.984022	7.076486	7.331711	7.445694	7.698172	...	10
4	Angola	AGO	Arable land (% of land area)	AG.LND.ARBL.ZS	NaN	2.141654	2.165717	2.181760	2.205823	2.221866	...	3

5 rows × 66 columns

```
def col_convert(df):
    id_vars = ['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code']
    value_vars = df.columns.difference(id_vars).tolist()
    df = pd.melt(df, id_vars=id_vars, value_vars=value_vars, var_name='Year', value_name='Value')
    df['Year'] = pd.to_datetime(df['Year'], format='%Y')
    return df['Year']

def datafram_convert(df):
  Year = col_convert(df)

  return df['Country Name'] , Year
```

```
path = "/content/API_AG.LND.FRST.ZS_DS2_en_csv_v2_5358376.csv"
df = pd.read_csv(path)
Country_Name, Year = datafram_convert(df)

print(Country_Name)
print(Year)
```

```
0                            Aruba
1      Africa Eastern and Southern
2                      Afghanistan
3       Africa Western and Central
4                           Angola
                  ...
261                         Kosovo
262                  Yemen, Rep.
263                 South Africa
264                       Zambia
265                     Zimbabwe
Name: Country Name, Length: 266, dtype: object
0      1990-01-01
1      1990-01-01
2      1990-01-01
3      1990-01-01
4      1990-01-01
          ...
8507   2021-01-01
8508   2021-01-01
8509   2021-01-01
8510   2021-01-01
8511   2021-01-01
Name: Year, Length: 8512, dtype: datetime64[ns]
```

statistical exploration of datasets done by utilizing the "describe" method to analyze their columns and overall structure.

```
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv("/content/API_AG.LND.FRST.ZS_DS2_en_csv_v2_5358376.csv")
df.columns
```

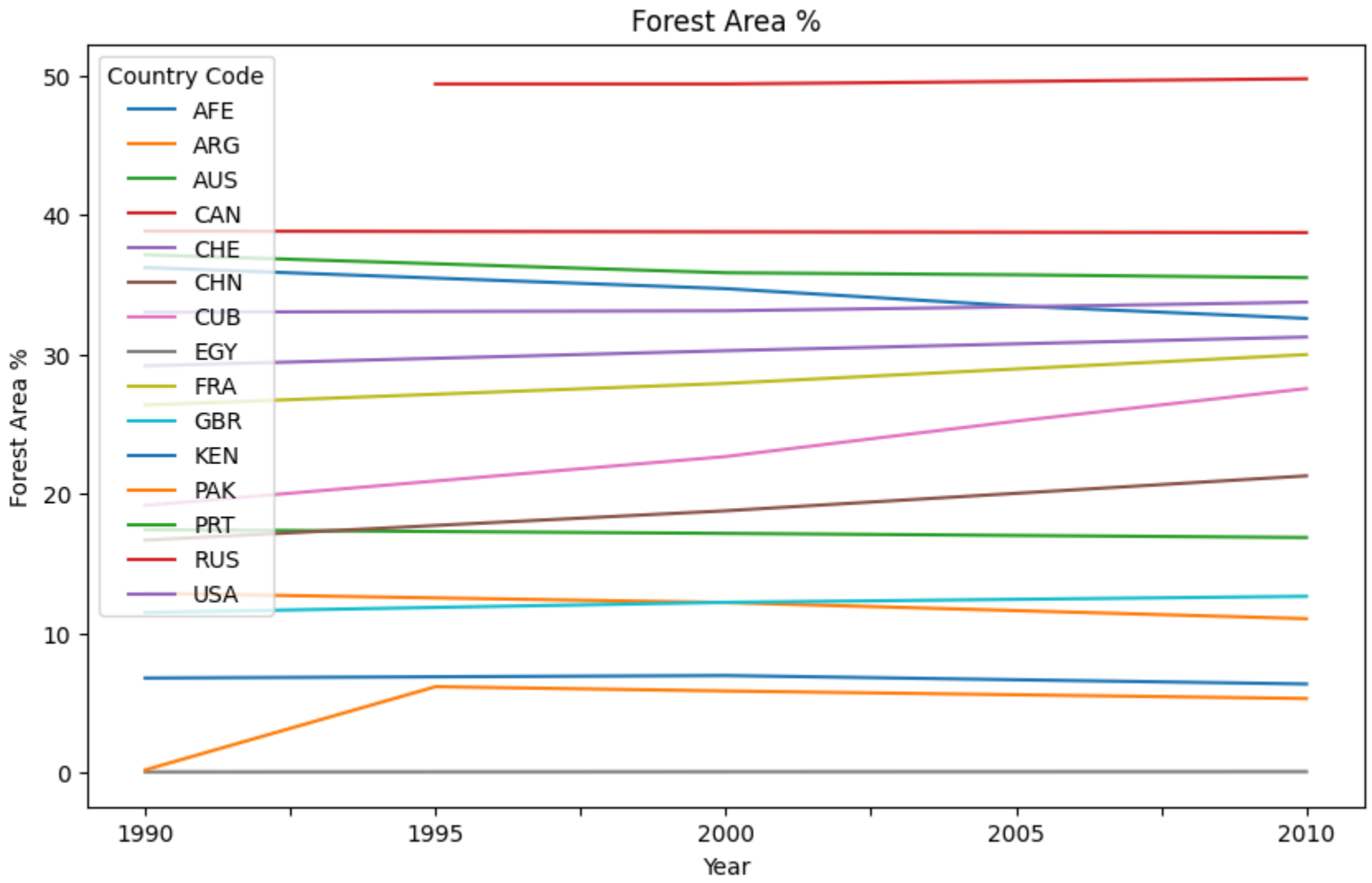
```
Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
       '1990', '1991', '1992', '1993', '1994', '1995', '1996', '1997', '1998',
       '1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007',
       '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016',
       '2017', '2018', '2019', '2020', '2021'],
      dtype='object')
```

```
# Calculate summary statistics
print(df.describe())
```

```
              1990        1991        1992        1993        1994        1995  \
count  222.000000  226.000000  251.000000  254.000000  254.000000  254.000000
mean    32.927665   33.654898   33.184709   33.083255   33.072996   33.024235
std     24.319267   24.690332   24.015348   23.849543   23.828786   23.769685
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%     13.223427   13.450064   12.871721   12.742997   12.657357   12.571716
50%     30.730555   30.906027   31.218556   31.123515   31.311720   31.245346
75%     48.580105   51.993871   48.894391   48.263310   48.173347   48.107819
max     98.574551   98.550987   98.527423   98.503859   98.480295   98.456731

              1996        1997        1998        1999  ...        2012  \
count  254.000000  254.000000  254.000000  254.000000  ...  261.000000
mean    32.975222   32.925691   32.900866   32.851716  ...   32.093941
std     23.712863   23.658048   23.625678   23.574716  ...   23.057537
min      0.000000    0.000000    0.000000    0.000000  ...    0.000000
25%     12.474370   12.421529   12.366474   12.357720  ...   12.430556
50%     31.469788   31.434611   31.374077   31.261292  ...   30.348111
75%     48.076107   47.714775   47.417713   47.484942  ...   46.902521
max     98.433167   98.409603   98.386038   98.362474  ...   97.952462

              2013        2014        2015        2016        2017        2018  \
count  261.000000  261.000000  261.000000  261.000000  260.000000  260.000000
mean    32.050349   32.009783   31.970608   31.904915   31.866522   31.817643
```

```
std     23.016365   22.983945   22.953139   22.899125   22.925268   22.901671
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%     12.430556   12.430556   12.430556   12.430556   12.145109   11.991243
50%     30.442858   30.464260   30.585975   30.716421   30.545952   30.300493
75%     46.129997   45.925926   45.924603   45.878355   46.214932   46.364728
max     97.890679   97.828897   97.767115   97.694359   97.647564   97.569103

             2019        2020  2021
count  260.000000  260.000000   0.0
mean    31.764707   31.714485   NaN
std     22.876692   22.854770   NaN
min      0.000000    0.000000   NaN
25%     11.836865   11.729713   NaN
50%     30.324426   30.273383   NaN
75%     46.505122   46.010410   NaN
max     97.490577   97.412115   NaN

[8 rows x 32 columns]
```

```
# Select countries of interest
countries = ['AFE', 'AUS','PRT','RUS','ARG', 'PAK','CAN','CHE','CHN','USA','CUB','EGY', 'GBR','FRA','KEN']

# Subset the data for these countries
subset = df[df["Country Code"].isin(countries)]

# Set the index to be the country names
subset.set_index("Country Code", inplace=True)

# Select columns of interest
cols = [  '1990','1995', '2000','2005', '2010']
subset = subset[cols]

# Plot the data
subset.T.plot(kind='line', figsize=(10,6))
plt.title('Forest Area %')
plt.xlabel('Year')
plt.ylabel('Forest Area %')
plt.show()
```



Statistical exploration of Arble land datasets done by utilizing the "describe" method to analyze their columns and overall structure.

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

# Load dataset
df = pd.read_csv("/content/API_AG.LND.ARBL.ZS_DS2_en_csv_v2_5362201.csv")
df.columns
```

```
Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
       '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968',
       '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977',
       '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985', '1986',
       '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995',
       '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
       '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021'],
      dtype='object')
```

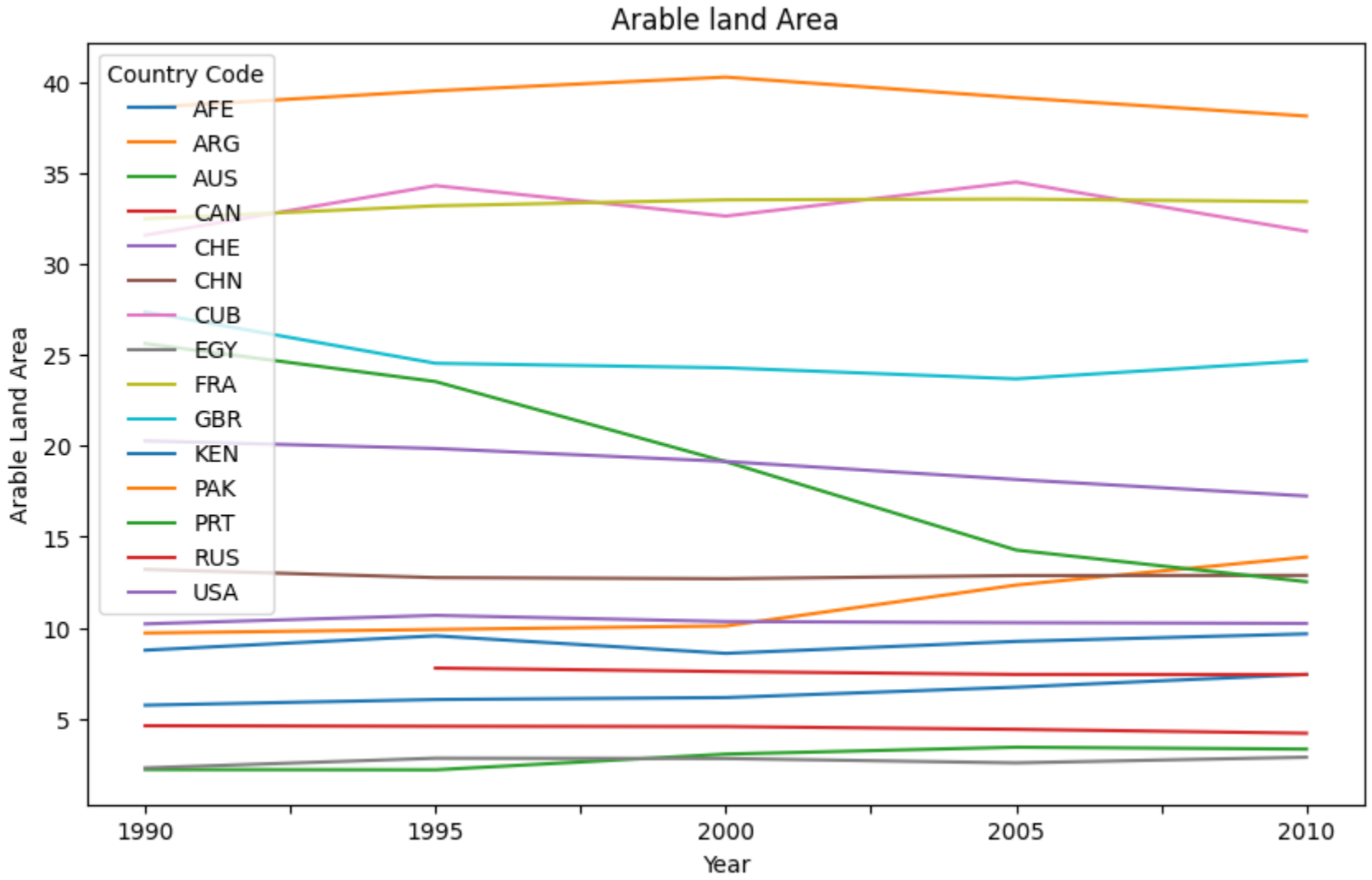
```
# Calculate summary statistics
print(df.describe())
```

```
             1960        1961        1962        1963        1964        1965  \
count   0.0  219.000000  219.000000  219.000000  219.000000  219.000000
mean    NaN   12.511610   12.526574   12.619511   12.637377   12.697057
std     NaN   13.490237   13.427044   13.387671   13.340765   13.349528
min     NaN    0.043141    0.043141    0.043141    0.043141    0.043141
25%     NaN    3.083875    3.099022    3.107868    3.122380    3.134404
50%     NaN    8.357041    8.357041    8.357041    8.357041    8.385791
75%     NaN   15.765700   15.910750   16.219994   16.316706   16.638981
max     NaN   70.175000   69.325000   68.600000   68.150000   67.427211

             1966        1967        1968        1969  ...        2012  \
count  219.000000  219.000000  219.000000  219.000000  ...  254.000000
mean    12.683259   12.684125   12.710481   12.782845  ...   13.428193
std     13.251589   13.263634   13.284544   13.327050  ...   12.660516
min      0.043141    0.043141    0.043141    0.043141  ...    0.045845
25%      3.148280    3.192969    3.243886    3.267237  ...    4.191660
50%      8.662791    8.895349    8.895349    8.712660  ...   10.021763
```

```
75%     16.722038   16.553365   16.722038   16.823811  ...   17.910355
max     67.588538   67.934240   67.826688   67.680725  ...   60.450000

              2013        2014        2015        2016        2017        2018  \
count  254.000000  254.000000  254.000000  254.000000  254.000000  254.000000
mean    13.451108   13.428380   13.485382   13.495486   13.541176   13.513289
std     12.763818   12.726172   12.769022   12.786653   12.773576   12.768481
min      0.045845    0.045845    0.045845    0.047278    0.047278    0.047278
25%      4.161301    4.137267    4.138871    4.128607    4.161130    4.143301
50%     10.192786   10.182616   10.182616   10.131937   10.199844   10.191341
75%     17.983974   17.045342   17.054012   17.119531   17.723626   17.506412
max     60.190000   60.800000   59.401091   59.646693   59.593839   59.710000

              2019        2020  2021
count  254.000000  254.000000   0.0
mean    13.478947   13.519868   NaN
std     12.837590   12.791902   NaN
min      0.047278    0.051245   NaN
25%      4.153325    4.182287   NaN
50%     10.133192   10.233163   NaN
75%     17.167395   17.481020   NaN
max     61.204579   61.458093   NaN

[8 rows x 62 columns]
```

```
# Select countries of interest
countries = ['AFE', 'AUS','PRT','RUS','ARG', 'PAK', 'BTH', 'CAN','CHE','CHN','USA','CUB','EGY', 'GBR','FRA','KEN

# Subset the data for these countries
subset = df[df["Country Code"].isin(countries)]

# Set the index to be the country names
subset.set_index("Country Code", inplace=True)

# Select columns of interest
cols = [  '1990','1995', '2000','2005', '2010']
subset = subset[cols]

# Plot the data
```

```
subset.T.plot(kind='line', figsize=(10,6))
plt.title('Arable land Area')
plt.xlabel('Year')
plt.ylabel('Arable Land Area')
plt.show()
```

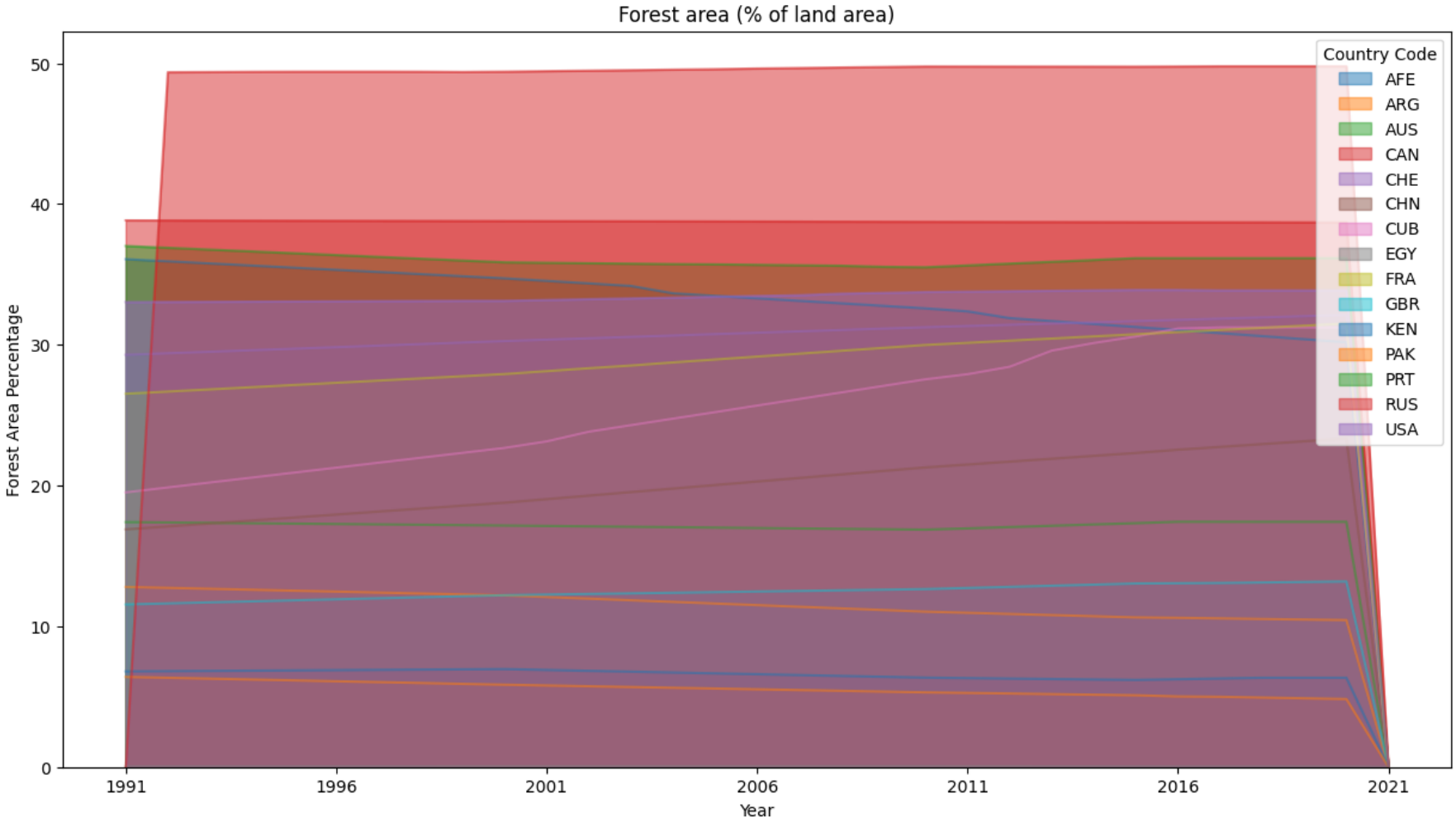


Analyze the potential correlations between various countries and indicators, examining any potential differences in these

relationships across countries and over time. To facilitate this analysis, area charts, bar charts, correlation matrices, and box plots have been utilized and are displayed below

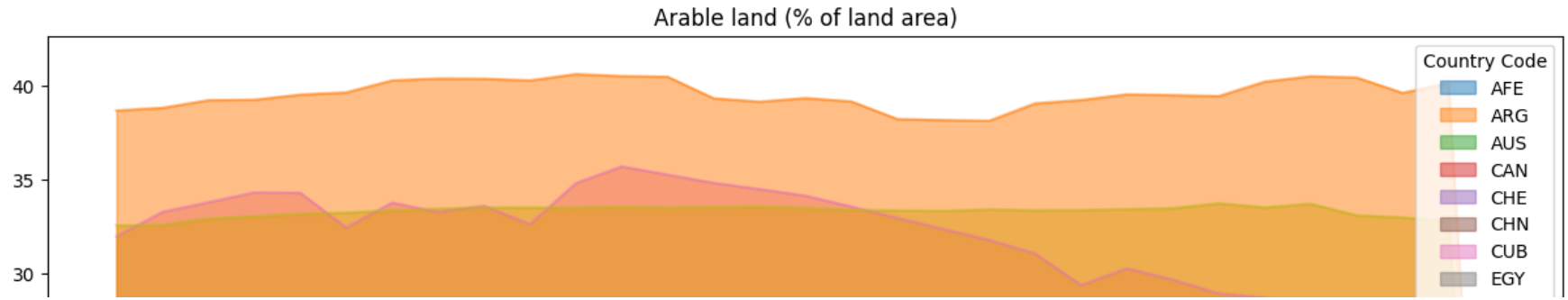
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data into a Pandas DataFrame
data = pd.read_csv('/content/API_AG.LND.FRST.ZS_DS2_en_csv_v2_5358376.csv')
# create a list of countries to select
countries_to_select = ['AFE', 'AUS','PRT','RUS','ARG', 'PAK', 'BTH', 'CAN','CHE','CHN','USA','CUB','EGY', 'GBR',

# filter the dataset by the selected countries
selected_df = data[data['Country Code'].isin(countries_to_select)]


df = selected_df[['Country Code', '1991', '1992', '1993', '1994', '1995', '1996',
                  '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
                  '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012',
                  '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021']]



# Set the index of the DataFrame to 'Country Name' column
df.set_index('Country Code', inplace=True)

# Transpose the DataFrame to have the years as the index and the countries as columns
df = df.T

# Plot the area chart
ax = df.plot(kind='area', figsize=(15, 8), stacked=False)

# Customize the plot
ax.set_xlabel('Year')
ax.set_ylabel('Forest Area Percentage')
ax.set_title('Forest area (% of land area)')

# Display the plot
```

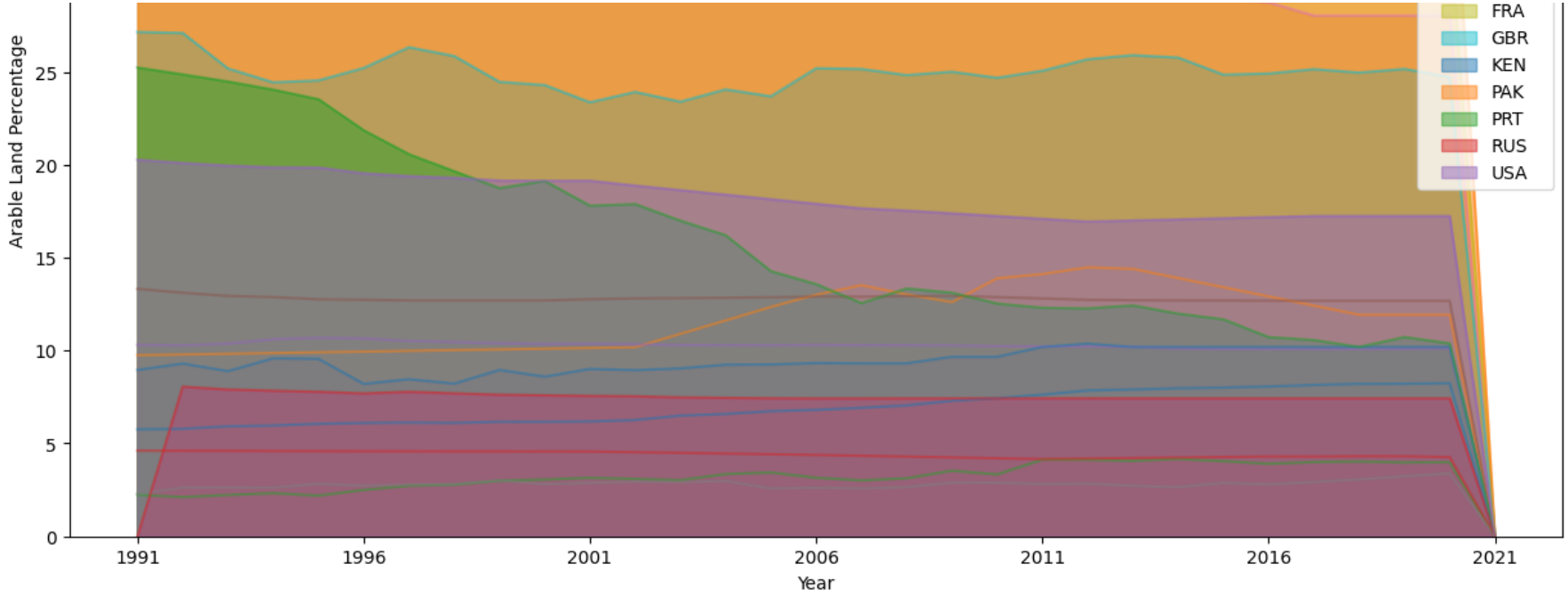
```
plt.show()
```




```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data into a Pandas DataFrame
data = pd.read_csv('/content/API_AG.LND.ARBL.ZS_DS2_en_csv_v2_5362201.csv')
# create a list of countries to select
countries_to_select = ['AFE', 'AUS','PRT','RUS','ARG', 'PAK', 'BTH', 'CAN','CHE','CHN','USA','CUB','EGY', 'GBR',
```

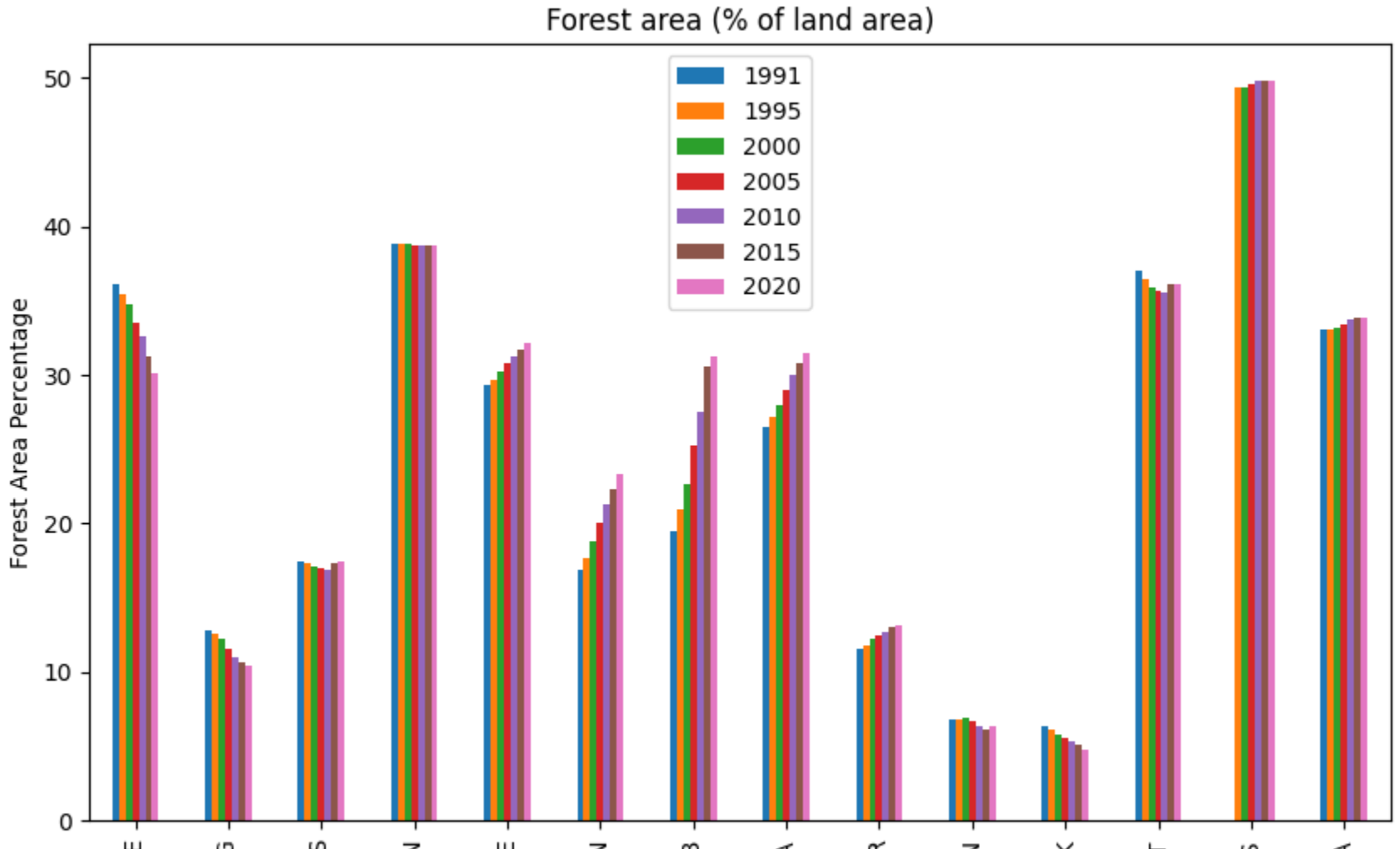
```
# filter the dataset by the selected countries
selected_df = data[data['Country Code'].isin(countries_to_select)]


df = selected_df[['Country Code', '1991', '1992', '1993', '1994', '1995', '1996',
                  '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
                  '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012',
                  '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021']]



# Set the index of the DataFrame to 'Country Name' column
df.set_index('Country Code', inplace=True)

# Transpose the DataFrame to have the years as the index and the countries as columns
df = df.T

# Plot the area chart
ax = df.plot(kind='area', figsize=(15, 8), stacked=False)

# Customize the plot
ax.set_xlabel('Year')
ax.set_ylabel('Arable Land Percentage')
ax.set_title('Arable land (% of land area)')

# Display the plot
plt.show()
```



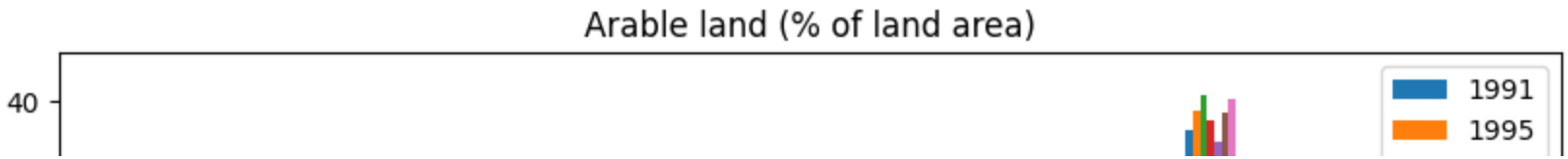


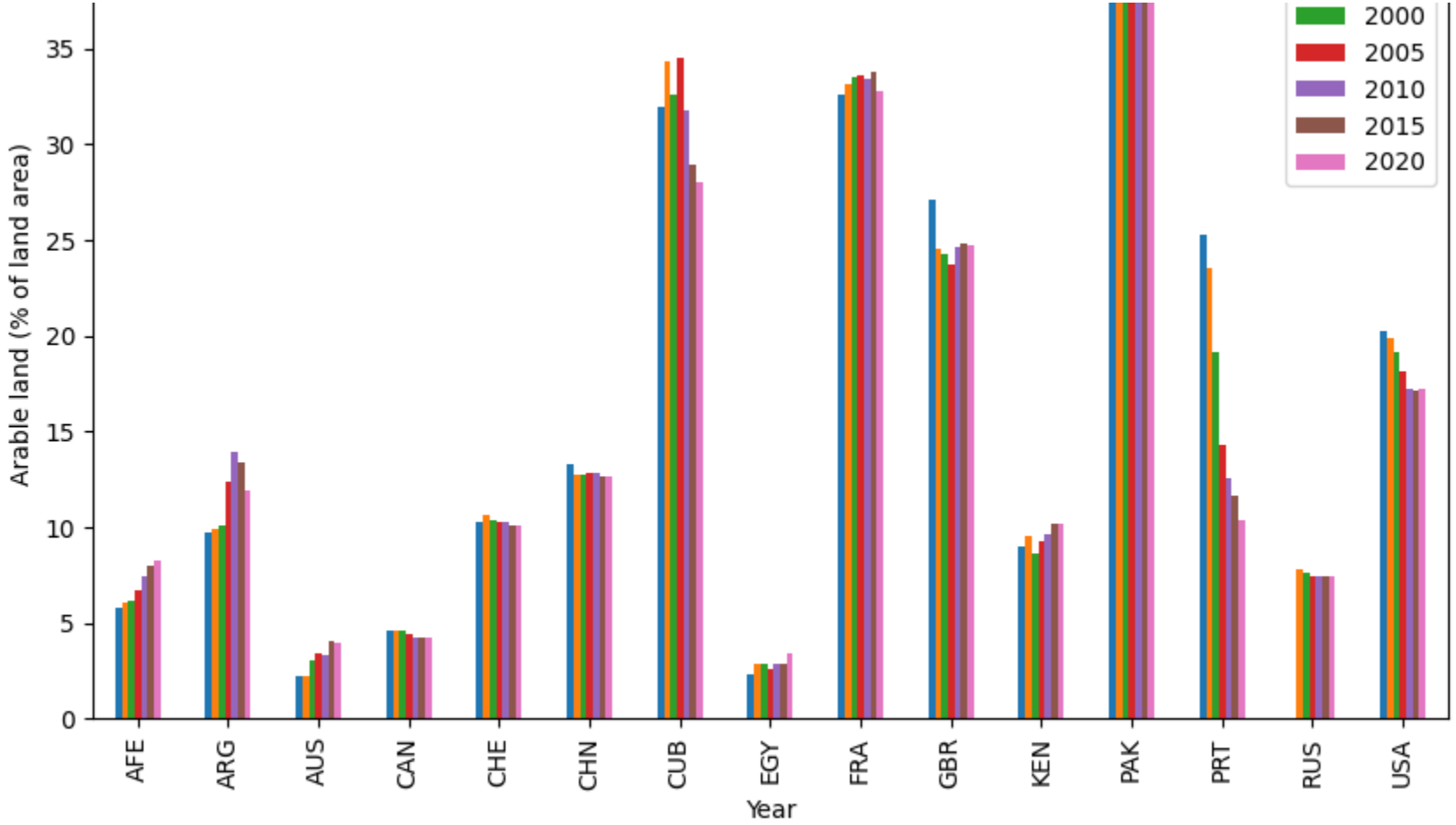

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('/content/API_AG.LND.FRST.ZS_DS2_en_csv_v2_5358376.csv')

# Select countries of interest
countries = ['AFE', 'AUS','PRT','RUS','ARG', 'PAK', 'CAN','CHE','CHN','USA','CUB', 'GBR','FRA','KEN']

# Filter the dataset for the selected countries
subset = data[data['Country Code'].isin(countries)]

# Set the index to be the country names
subset.set_index('Country Code', inplace=True)

# Select the columns to use for the chart
columns = ['1991', '1995','2000', '2005', '2010', '2015', '2020']

# Create the bar chart
```

```
subset[columns].plot(kind='bar', figsize=(10, 6))

# Add a title and axis labels
plt.title('Forest area (% of land area)')
plt.xlabel('Year')
plt.ylabel('Forest Area Percentage')

# Show the chart
plt.show()
```



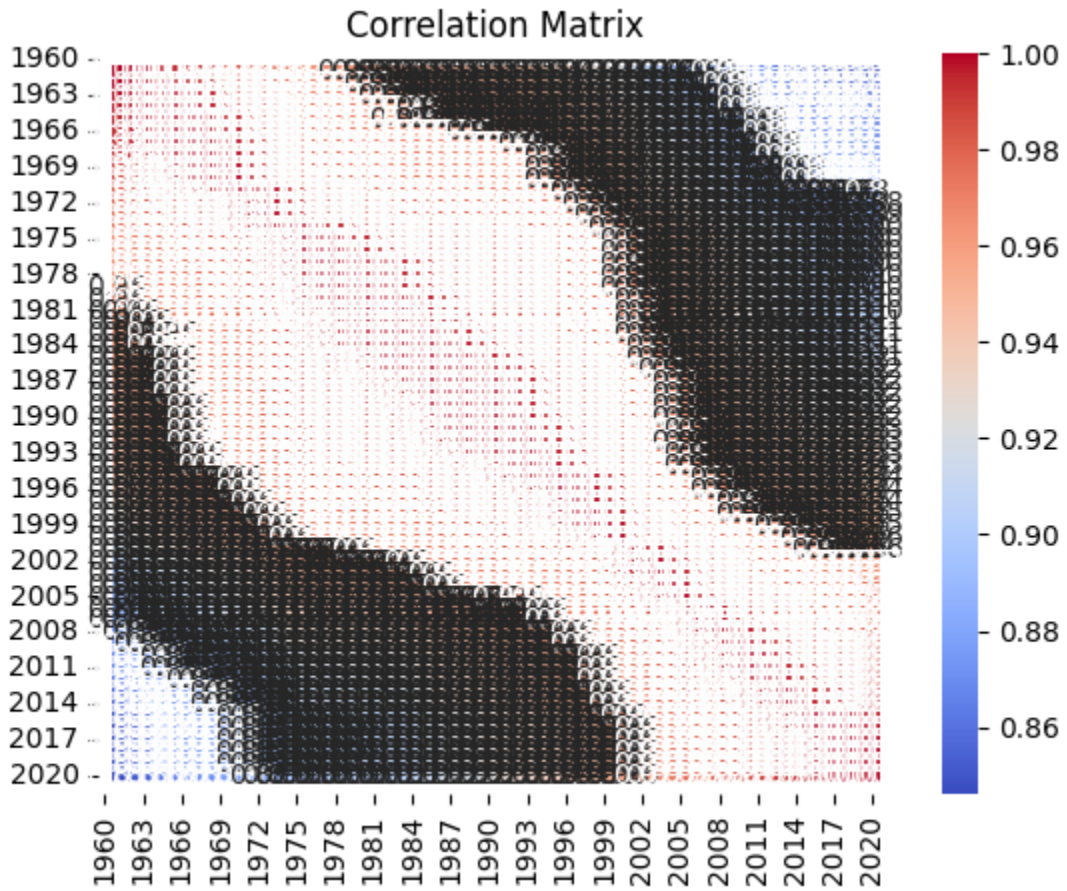

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('/content/API_AG.LND.ARBL.ZS_DS2_en_csv_v2_5362201.csv')

# Select countries of interest
countries = ['AFE', 'AUS','PRT','RUS','ARG', 'PAK', 'CAN','CHE','CHN','USA','CUB','EGY', 'GBR','FRA','KEN']

# Filter the dataset for the selected countries
subset = data[data['Country Code'].isin(countries)]

# Set the index to be the country names
subset.set_index('Country Code', inplace=True)

# Select the columns to use for the chart
columns = ['1991', '1995','2000', '2005', '2010', '2015', '2020']

# Create the bar chart
subset[columns].plot(kind='bar', figsize=(10, 6))

# Add a title and axis labels
plt.title('Arable land (% of land area)')
plt.xlabel('Year')
plt.ylabel('Arable land (% of land area)')

# Show the chart
plt.show()
```





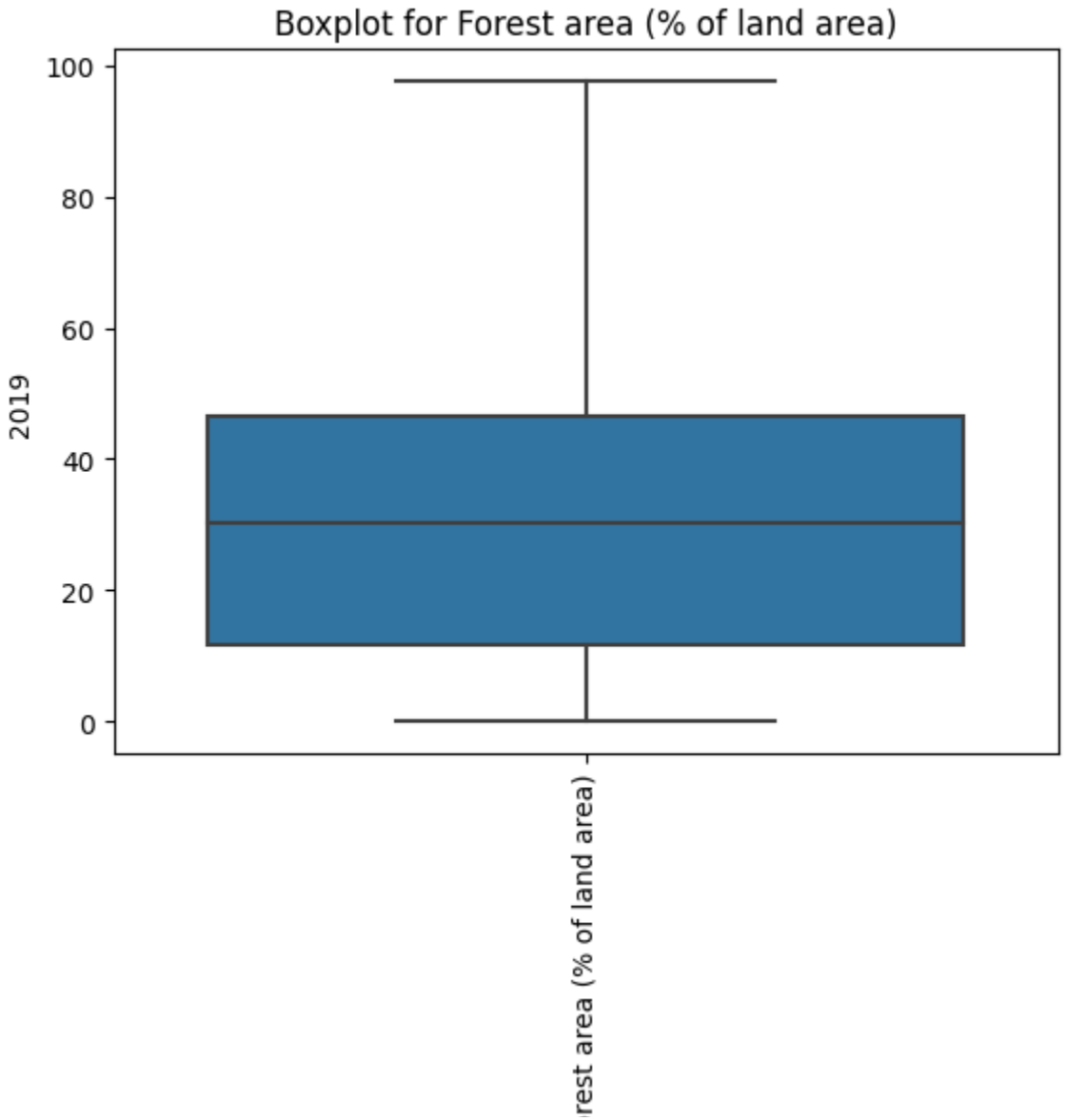

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Read the data
df = pd.read_csv('/content/API_AG.LND.ARBL.ZS_DS2_en_csv_v2_5362201.csv')

# Calculate correlation matrix
corr_matrix = df.corr()

# Plot correlation matrix using heatmap
```

```
sns.heatmap(corr_matrix, cmap='coolwarm', annot=True, fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```



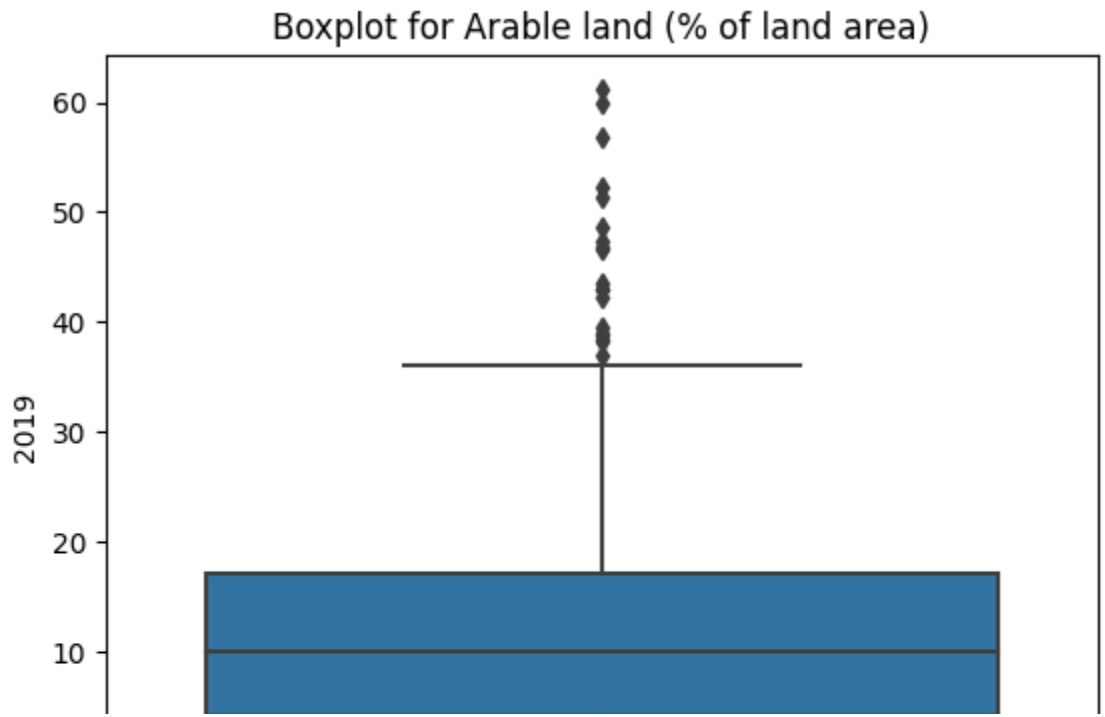

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Read the data
df = pd.read_csv('/content/API_AG.LND.FRST.ZS_DS2_en_csv_v2_5358376.csv')
```

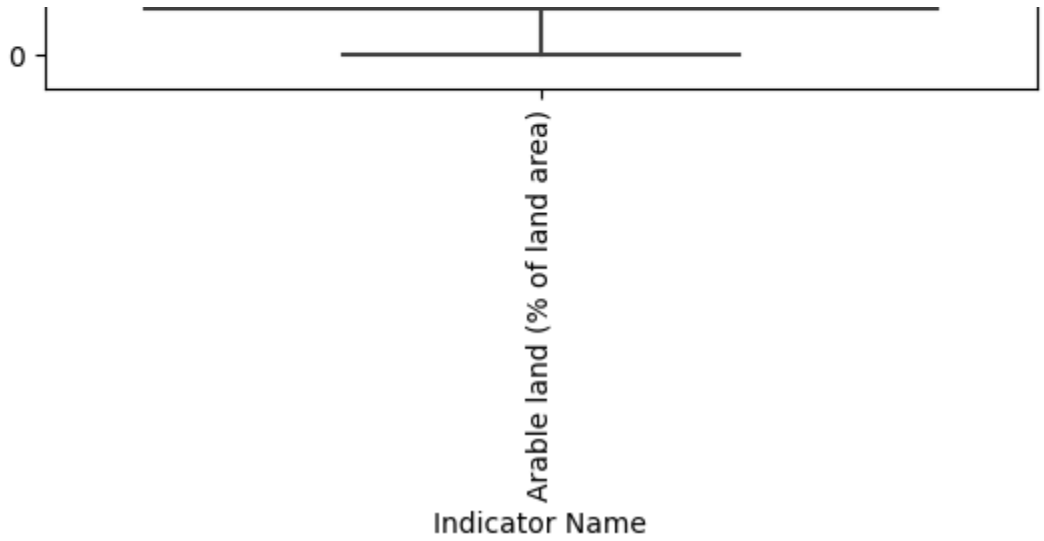
```
# Plot boxplot for value added in agriculture, forestry, and fishing
sns.boxplot(x='Indicator Name', y='2019', data=df)
plt.title('Boxplot for Forest area (% of land area)')
plt.xticks(rotation=90)
plt.show()
```



Fo
Indicator Name

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

# Read the data
df = pd.read_csv('/content/API_AG.LND.ARBL.ZS_DS2_en_csv_v2_5362201.csv')


# Plot boxplot for value added in agriculture, forestry, and fishing
sns.boxplot(x='Indicator Name', y='2019', data=df)
plt.title('Boxplot for Arable land (% of land area)')
plt.xticks(rotation=90)
plt.show()
```




0
Arable land (% of land area)
Indicator Name

Colab paid products - Cancel contracts here