Table of Contents

Goals

This project consist in analyzing social and macroeconomic development in Brazil and correlate it to sustainability variables such as CO2 emission and deforestation. The goal for this project was to do the following:

- Get acquainted with the data
- Clean the data so it is ready for analysis
- Develop some questions for analysis
- Analyze variables within the data to gain patterns and insights on these questions

Data

The data for this project was downloaded from the World Bank:

https://databank.worldbank.org/source/world-development-indicators

Loading the Data

First, the necessary libraries are loaded into the notebook. The pandas library is used to import data from wb_data_latam.csv and preview the first five rows of the DataFrame.

In [1]:
```
# sets up matplotlib with interactive features
%matplotlib inline
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import matplotlib.dates as mdates
from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import re
```

In [2]:
```
latam_data = pd.read_csv('wb_data_latam.csv')

# see all columns
pd.set_option('display.max_columns', None)
latam_data.head()
```

Out[2]:

	Country Name	Country Code	Series Name	Series Code	1999 [YR1999]	200
0	Latin America & Caribbean	LCN	Access to electricity (% of population)	EG.ELC.ACCS.ZS	91.2475876868212	91.722
1	Latin America & Caribbean	LCN	Adjusted net national income (constant 2015 US$)	NY.ADJ.NNTY.KD	2764811152598.09	284819
2	Latin America & Caribbean	LCN	Adjusted net national income per capita (const...	NY.ADJ.NNTY.PC.KD	5382.00038538551	5463.84
3	Latin America & Caribbean	LCN	Agricultural land (% of land area)	AG.LND.AGRI.ZS	33.6754480992417	33.648
4	Latin America & Caribbean	LCN	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS	..	

Explatory Data Analysis

Data Information

Some immediate insights are:

- There are 29 columns and 1897 rows.
- The name and datatype of each column -- all values are set as object. We need to to change values from column 5 onward to float.
- It says all the columns have just 5 lines of missing data, but when investigating it using Data Wrangler its possible to see these empty lines should be deleted and

there are many '..' values that should be set as Null.

- The column names could be renamed considering just the year (not [YRxxxx]) for simplicity.

In [3]:
```
# describe columns data
latam_data.describe()
```

Out[3]:

	Country Name	Country Code	Series Name	Series Code	1999 [YR1999]	2000 [YR2000]	[YR
count	1894	1892	1892	1892	1892	1892	
unique	45	43	44	44	1119	1211	
top	Latin America & Caribbean	LCN	Access to electricity (% of population)	EG.ELC.ACCS.ZS	..	..	
freq	44	44	43	43	745	649	

In [4]:
```
latam_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1897 entries, 0 to 1896
Data columns (total 29 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Country Name   1894 non-null   object
 1   Country Code   1892 non-null   object
 2   Series Name    1892 non-null   object
 3   Series Code    1892 non-null   object
 4   1999 [YR1999]  1892 non-null   object
 5   2000 [YR2000]  1892 non-null   object
 6   2001 [YR2001]  1892 non-null   object
 7   2002 [YR2002]  1892 non-null   object
 8   2003 [YR2003]  1892 non-null   object
 9   2004 [YR2004]  1892 non-null   object
 10  2005 [YR2005]  1892 non-null   object
 11  2006 [YR2006]  1892 non-null   object
 12  2007 [YR2007]  1892 non-null   object
 13  2008 [YR2008]  1892 non-null   object
 14  2009 [YR2009]  1892 non-null   object
 15  2010 [YR2010]  1892 non-null   object
 16  2011 [YR2011]  1892 non-null   object
 17  2012 [YR2012]  1892 non-null   object
 18  2013 [YR2013]  1892 non-null   object
 19  2014 [YR2014]  1892 non-null   object
 20  2015 [YR2015]  1892 non-null   object
 21  2016 [YR2016]  1892 non-null   object
 22  2017 [YR2017]  1892 non-null   object
 23  2018 [YR2018]  1892 non-null   object
 24  2019 [YR2019]  1892 non-null   object
 25  2020 [YR2020]  1892 non-null   object
 26  2021 [YR2021]  1892 non-null   object
 27  2022 [YR2022]  1892 non-null   object
 28  2023 [YR2023]  1892 non-null   object
dtypes: object(29)
memory usage: 429.9+ KB
```

In [5]:
```python
# replace '..' values with 'null'
latam_data = latam_data.replace('..', np.nan)

# change data type of column 5 onwards from object to float
for col in latam_data.columns[4:]:
    latam_data[col] = latam_data[col].astype(float)

latam_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1897 entries, 0 to 1896
Data columns (total 29 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Country Name   1894 non-null   object 
 1   Country Code   1892 non-null   object 
 2   Series Name    1892 non-null   object 
 3   Series Code    1892 non-null   object 
 4   1999 [YR1999]  1147 non-null   float64
 5   2000 [YR2000]  1243 non-null   float64
 6   2001 [YR2001]  1226 non-null   float64
 7   2002 [YR2002]  1238 non-null   float64
 8   2003 [YR2003]  1238 non-null   float64
 9   2004 [YR2004]  1245 non-null   float64
 10  2005 [YR2005]  1294 non-null   float64
 11  2006 [YR2006]  1278 non-null   float64
 12  2007 [YR2007]  1286 non-null   float64
 13  2008 [YR2008]  1290 non-null   float64
 14  2009 [YR2009]  1301 non-null   float64
 15  2010 [YR2010]  1336 non-null   float64
 16  2011 [YR2011]  1319 non-null   float64
 17  2012 [YR2012]  1329 non-null   float64
 18  2013 [YR2013]  1322 non-null   float64
 19  2014 [YR2014]  1346 non-null   float64
 20  2015 [YR2015]  1390 non-null   float64
 21  2016 [YR2016]  1297 non-null   float64
 22  2017 [YR2017]  1287 non-null   float64
 23  2018 [YR2018]  1281 non-null   float64
 24  2019 [YR2019]  1297 non-null   float64
 25  2020 [YR2020]  1212 non-null   float64
 26  2021 [YR2021]  1115 non-null   float64
 27  2022 [YR2022]  875 non-null    float64
 28  2023 [YR2023]  517 non-null    float64
dtypes: float64(25), object(4)
memory usage: 429.9+ KB
```

In [6]:
```
# see what columns have missing data
```

In [7]:
```
# delete the 5 rows of the dataset
latam_data = latam_data.drop(latam_data.tail(5).index)

latam_data.isnull().sum()
```

Out[7]:
```
Country Name        0
Country Code        0
Series Name         0
Series Code         0
1999 [YR1999]     745
2000 [YR2000]     649
2001 [YR2001]     666
2002 [YR2002]     654
2003 [YR2003]     654
2004 [YR2004]     647
2005 [YR2005]     598
2006 [YR2006]     614
2007 [YR2007]     606
2008 [YR2008]     602
2009 [YR2009]     591
2010 [YR2010]     556
2011 [YR2011]     573
2012 [YR2012]     563
2013 [YR2013]     570
2014 [YR2014]     546
2015 [YR2015]     502
2016 [YR2016]     595
2017 [YR2017]     605
2018 [YR2018]     611
2019 [YR2019]     595
2020 [YR2020]     680
2021 [YR2021]     777
2022 [YR2022]    1017
2023 [YR2023]    1375
dtype: int64
```

In [8]:
```
# rename column names from column 5 onwards
latam_data.columns = [col.split('[YR')[1].split(']')[0] if '[YR' in col e

latam_data.head()
```

Out[8]:

	Country Name	Country Code	Series Name	Series Code	1999	20(
0	Latin America & Caribbean	LCN	Access to electricity (% of population)	EG.ELC.ACCS.ZS	9.124759e+01	9.172250e+
1	Latin America & Caribbean	LCN	Adjusted net national income (constant 2015 US$)	NY.ADJ.NNTY.KD	2.764811e+12	2.848200e+
2	Latin America & Caribbean	LCN	Adjusted net national income per capita (const...	NY.ADJ.NNTY.PC.KD	5.382000e+03	5.463846e+
3	Latin America & Caribbean	LCN	Agricultural land (% of land area)	AG.LND.AGRI.ZS	3.367545e+01	3.364881e+
4	Latin America & Caribbean	LCN	Central government debt, total (% of GDP)	GC.DOD.TOTL.GD.ZS	NaN	N

Data Cleaning

- The dataset contains the data for all Latam countries but we just want to analyze Brazil. Let's filter it and keep just Brazil.
- The column "Series Code" don't tell us much, so let's take it off.
- The data is in the wide format (years in columns and variables in rows). Let's change it to the long format (variables in columns and years in rows), since it is more efficient considering the purpose of the project and the libraries we will be using.
- We don't need all 44 variables. Let's keep just those that might be interesting for our analysis.
- Finally, let's deal with the missing values. Since the variable we are looking at tend not to vary much and to make it simpler for this project, let's use the value from the previous period in the cases where we miss values.

In [9]:
```
# filter just Brazil
br_data = latam_data[latam_data['Country Name'] == 'Brazil']

# drop the 'Series Code' column
br_data = br_data.drop('Series Code', axis=1)
```

```
br_data.head()
```

Out[9]:

	Country Name	Country Code	Series Name	1999	2000	2001	
352	Brazil	BRA	Access to electricity (% of population)	9.476296e+01	9.440000e+01	9.600000e+01	9
353	Brazil	BRA	Adjusted net national income (constant 2015 US$)	9.306510e+11	9.653745e+11	9.668836e+11	
354	Brazil	BRA	Adjusted net national income per capita (const...	5.364407e+03	5.489021e+03	5.425472e+03	5
355	Brazil	BRA	Agricultural land (% of land area)	2.730509e+01	2.731750e+01	2.732991e+01	2
356	Brazil	BRA	Central government debt, total (% of GDP)	NaN	NaN	NaN	

In [10]:

```
# Transforming from wide to long
br_data_long = pd.melt(br_data, id_vars=['Country Name', 'Country Code',

# Ordering by Series Name and Year
br_data_long = br_data_long.sort_values(['Series Name', 'Year'])

br_data_long.head()
```

Out[10]:

	Country Name	Country Code	Series Name	Year	Value
0	Brazil	BRA	Access to electricity (% of population)	1999	94.76296
44	Brazil	BRA	Access to electricity (% of population)	2000	94.40000
88	Brazil	BRA	Access to electricity (% of population)	2001	96.00000
132	Brazil	BRA	Access to electricity (% of population)	2002	96.70000
176	Brazil	BRA	Access to electricity (% of population)	2003	97.00000

In [11]:

```
# Filtering the DataFrame to keep only the relevant variables
variables_to_keep = [
    'Agricultural land (% of land area)',
    'CO2 emissions (metric tons per capita)',
    'Current health expenditure (% of GDP)',
    'Forest area (% of land area)',
    'GDP (constant 2015 US$)',
    'GDP per capita (constant 2015 US$)',
    'Gini index',
    'Gross capital formation (% of GDP)',
    'Individuals using the Internet (% of population)',
    'Intentional homicides (per 100,000 people)',
    'Life expectancy at birth, total (years)',
    'Population, total',
    'Renewable electricity output (% of total electricity output)',
    'Rural population (% of total population)',
    'Unemployment, total (% of total labor force) (national estimate)'
]

br_data_long = br_data_long[br_data_long['Series Name'].isin(variables_to

br_data_long.head()
```

Out[11]:

	Country Name	Country Code	Series Name	Year	Value
3	Brazil	BRA	Agricultural land (% of land area)	1999	27.305094
47	Brazil	BRA	Agricultural land (% of land area)	2000	27.317501
91	Brazil	BRA	Agricultural land (% of land area)	2001	27.329908
135	Brazil	BRA	Agricultural land (% of land area)	2002	27.342315
179	Brazil	BRA	Agricultural land (% of land area)	2003	27.354722

In [12]:
```
# Pivot the DataFrame to have each unique variable within Series Name set
br_data_pivoted = br_data_long.pivot(index=['Country Name', 'Country Code

br_data_pivoted.head(50)
```

Out[12]:

	Series Name	Country Name	Country Code	Year	Agricultural land (% of land area)	CO2 emissions (metric tons per capita)	Current health expenditure (% of GDP)	Forest area (% of land area)
0		Brazil	BRA	1999	27.305094	1.734769	NaN	66.386725
1		Brazil	BRA	2000	27.317501	1.783500	8.334572	65.934359
2		Brazil	BRA	2001	27.329908	1.792112	8.549633	65.461671
3		Brazil	BRA	2002	27.342315	1.760651	8.696870	64.988983
4		Brazil	BRA	2003	27.354722	1.701853	8.188994	64.516295
5		Brazil	BRA	2004	27.367130	1.778441	8.124920	64.043608
6		Brazil	BRA	2005	27.379537	1.775663	8.035396	63.570920
7		Brazil	BRA	2006	27.392578	1.777478	8.249525	63.098232
8		Brazil	BRA	2007	27.478817	1.847976	8.201830	62.625545
9		Brazil	BRA	2008	27.565056	1.939215	8.011285	62.152857
10		Brazil	BRA	2009	27.651296	1.799328	8.394805	61.680169
11		Brazil	BRA	2010	27.737535	2.026606	7.945166	61.207482
12		Brazil	BRA	2011	27.823774	2.110628	7.788191	61.023328
13		Brazil	BRA	2012	27.910013	2.271418	7.736377	60.839175
14		Brazil	BRA	2013	27.996253	2.413447	7.976602	60.655021
15		Brazil	BRA	2014	28.082492	2.514592	8.396441	60.470868
16		Brazil	BRA	2015	28.168731	2.365361	8.909300	60.286715
17		Brazil	BRA	2016	28.254971	2.161260	9.169015	60.071033
18		Brazil	BRA	2017	28.341066	2.185487	9.471249	59.832881
19		Brazil	BRA	2018	28.415628	2.064261	9.464750	59.708428
20		Brazil	BRA	2019	28.490130	2.050770	9.614491	59.558526
21		Brazil	BRA	2020	28.564619	1.942523	10.182350	59.417478
22		Brazil	BRA	2021	28.639094	NaN	9.890723	59.270527
23		Brazil	BRA	2022	NaN	NaN	NaN	NaN
24		Brazil	BRA	2023	NaN	NaN	NaN	NaN

In [13]:
```
# Fill missing values with the value from the previous row
br_data_pivoted = br_data_pivoted.set_index(['Country Name', 'Country Cod
```

```
br_data_pivoted = br_data_pivoted.fillna(method='ffill')

# Fill missing values from the year 1999 with the value from the first ro
br_data_pivoted = br_data_pivoted.fillna(method='bfill')

# Reset the index to get back to the original format
br_data_pivoted = br_data_pivoted.reset_index()

br_data_pivoted.head(50)
```

```
/var/folders/7z/t0nvsc6s1810pdgw0lygqrjc0000gn/T/ipykernel_1911/162138812
8.py:3: FutureWarning: DataFrame.fillna with 'method' is deprecated and wi
ll raise in a future version. Use obj.ffill() or obj.bfill() instead.
  br_data_pivoted = br_data_pivoted.fillna(method='ffill')
/var/folders/7z/t0nvsc6s1810pdgw0lygqrjc0000gn/T/ipykernel_1911/162138812
8.py:6: FutureWarning: DataFrame.fillna with 'method' is deprecated and wi
ll raise in a future version. Use obj.ffill() or obj.bfill() instead.
  br_data_pivoted = br_data_pivoted.fillna(method='bfill')
```

Out[13]:

	Series Name	Country Name	Country Code	Year	Agricultural land (% of land area)	CO2 emissions (metric tons per capita)	Current health expenditure (% of GDP)	Forest area (% of land area)
0	Brazil	BRA	1999	27.305094	1.734769	8.334572	66.386725
1	Brazil	BRA	2000	27.317501	1.783500	8.334572	65.934359
2	Brazil	BRA	2001	27.329908	1.792112	8.549633	65.461671
3	Brazil	BRA	2002	27.342315	1.760651	8.696870	64.988983
4	Brazil	BRA	2003	27.354722	1.701853	8.188994	64.516295
5	Brazil	BRA	2004	27.367130	1.778441	8.124920	64.043608
6	Brazil	BRA	2005	27.379537	1.775663	8.035396	63.570920
7	Brazil	BRA	2006	27.392578	1.777478	8.249525	63.098232
8	Brazil	BRA	2007	27.478817	1.847976	8.201830	62.625545
9	Brazil	BRA	2008	27.565056	1.939215	8.011285	62.152857
10	Brazil	BRA	2009	27.651296	1.799328	8.394805	61.680169
11	Brazil	BRA	2010	27.737535	2.026606	7.945166	61.207482
12	Brazil	BRA	2011	27.823774	2.110628	7.788191	61.023328
13	Brazil	BRA	2012	27.910013	2.271418	7.736377	60.839175
14	Brazil	BRA	2013	27.996253	2.413447	7.976602	60.655021
15	Brazil	BRA	2014	28.082492	2.514592	8.396441	60.470868
16	Brazil	BRA	2015	28.168731	2.365361	8.909300	60.286715
17	Brazil	BRA	2016	28.254971	2.161260	9.169015	60.071033
18	Brazil	BRA	2017	28.341066	2.185487	9.471249	59.832881
19	Brazil	BRA	2018	28.415628	2.064261	9.464750	59.708428
20	Brazil	BRA	2019	28.490130	2.050770	9.614491	59.558526
21	Brazil	BRA	2020	28.564619	1.942523	10.182350	59.417478
22	Brazil	BRA	2021	28.639094	1.942523	9.890723	59.270527
23	Brazil	BRA	2022	28.639094	1.942523	9.890723	59.270527
24	Brazil	BRA	2023	28.639094	1.942523	9.890723	59.270527

It's ok not to have the data for the last 2Y since we would be just copying it from the previous one and keep with the same value.

- Let's now make each unique variable within Series Name as one column with its own values.

In [14]:
```
# perform summary statistics
br_data_pivoted.describe()
```

Out[14]:

Series Name	Agricultural land (% of land area)	CO2 emissions (metric tons per capita)	Current health expenditure (% of GDP)	Forest area (% of land area)	GDP (constant 2015 US$)	GDP pe capit (constar 2015 US$
count	25.000000	25.000000	25.000000	25.000000	2.500000e+01	25.00000
mean	27.887458	1.984996	8.697940	61.813675	1.608357e+12	8107.09409
std	0.497855	0.228507	0.764239	2.310589	2.628681e+11	867.18828
min	27.305094	1.701853	7.736377	59.270527	1.136548e+12	6551.22667
25%	27.379537	1.783500	8.124920	59.832881	1.368459e+12	7325.90355
50%	27.823774	1.942523	8.394805	61.023328	1.743173e+12	8426.84322
75%	28.341066	2.110628	9.464750	63.570920	1.804862e+12	8783.21542
max	28.639094	2.514592	10.182350	66.386725	1.954752e+12	9216.13228

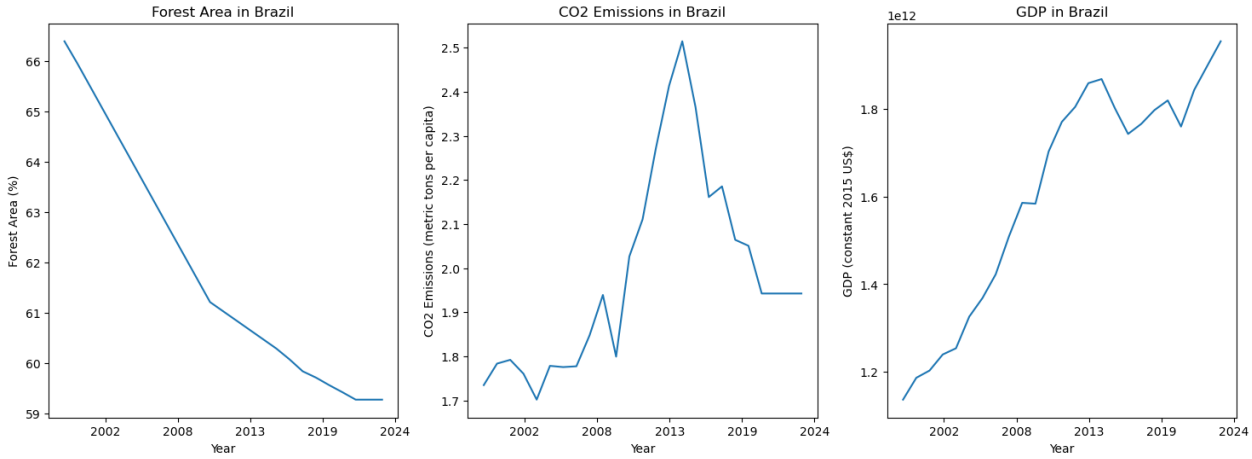
Data Exploration

- Now let's start exploring the date and search for insights with a focus on deforestation and CO2 emission.
    - 1. Evolution of % Forest Area and CO2 Emissions along the years: There is a clear process of deforestation ocurring in Brazil. The forest area as percentage of land area has reduced from 66.4% in 1999 to 59.7% in 2023. It happened together with an increase in CO2 emissions, which peaked in 2014 and eased in more recent periods and an increase of agricultural area (from 27.31% in 1999 to 28.64% in 2023, or 0.055% per year on average).
    - 2. Correlation between % Forest Area vs % Rural Population: There is a strong correlation between deforestation and urbanization.
    - 3. Correlation between % Forest Area vs % Agricultural Land: There is a strong correlation between deforestation and an increase in agricultural land.
    - 4. Correlation berween CO2 Emissions vs GDP: There is a moderate correlation between increased CO2 emissions and GDP growth.

- 5. Linear Regression between % Forest Area (dependent variable) vs % Agricultural Area (independent variable): Significant negative relationship between the variables, with a good degree of explanatory power (R2 = 76% confidence).
- 6. Forecast of % Forest Area if % Agricultural Area continues to increase by 0.055% per year: Considering the actual trend of increase in Agricultural Area as percentage of land area consinues, Brazil will reach the mark of less than 50% of Forest Area as percentage of land area by 2060.

1. Evolution of % Forest Area and CO2 Emissions along the years

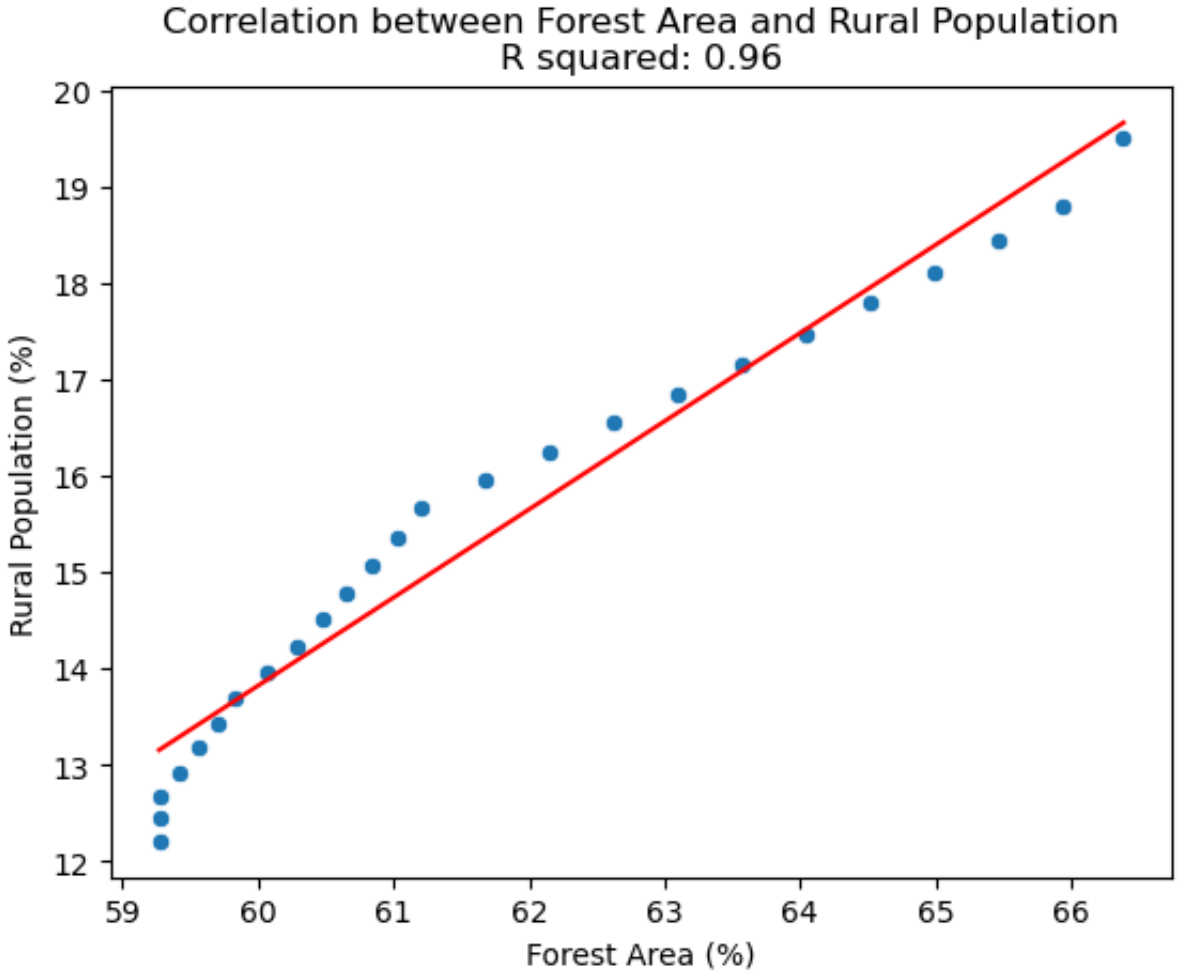
In [15]:
```
# Plot Forest area %, CO2 emissions, and GDP along the years side by side

# Create a figure and three axes
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(18, 6))

# Set the x-axis to Year for all plots
x = mdates.datestr2num(br_data_pivoted['Year'])

# Plot Forest area % on the first axis
ax1.plot(x, br_data_pivoted['Forest area (% of land area)'], label='Fores
ax1.set_xlabel('Year')
ax1.set_ylabel('Forest Area (%)')
ax1.set_title('Forest Area in Brazil')
ax1.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
plt.xticks(rotation=45)

# Plot CO2 emissions on the second axis
ax2.plot(x, br_data_pivoted['CO2 emissions (metric tons per capita)'], la
ax2.set_xlabel('Year')
ax2.set_ylabel('CO2 Emissions (metric tons per capita)')
ax2.set_title('CO2 Emissions in Brazil')
ax2.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
plt.xticks(rotation=45)

# Plot GDP on the third axis
ax3.plot(x, br_data_pivoted['GDP (constant 2015 US$)'], label='GDP')
ax3.set_xlabel('Year')
ax3.set_ylabel('GDP (constant 2015 US$)')
ax3.set_title('GDP in Brazil')
ax3.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
plt.xticks(rotation=0)

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



- 2. Correlation between % Forest Area vs % Rural Population

In [16]:
```
# Correlation between Forest area and Rural population
forest_area_rural_pop = br_data_pivoted['Forest area (% of land area)'].c
print("Correlation between Forest area and Rural population:", forest_are

# Calculate the trend line
x = br_data_pivoted['Forest area (% of land area)']
y = br_data_pivoted['Rural population (% of total population)']
x = sm.add_constant(x)
model = sm.OLS(y, x).fit()
predictions = model.predict(x)

# Calculate the R squared
r_squared = model.rsquared

# Plot the scatter plot with the trend line and R squared
sns.scatterplot(x='Forest area (% of land area)', y='Rural population (%
plt.plot(x['Forest area (% of land area)'], predictions, color='red')
plt.title(f"Correlation between Forest Area and Rural Population\nR squar
plt.xlabel("Forest Area (%)")
plt.ylabel("Rural Population (%)")
plt.show()
```

```
Correlation between Forest area and Rural population: 0.980161458932757
```

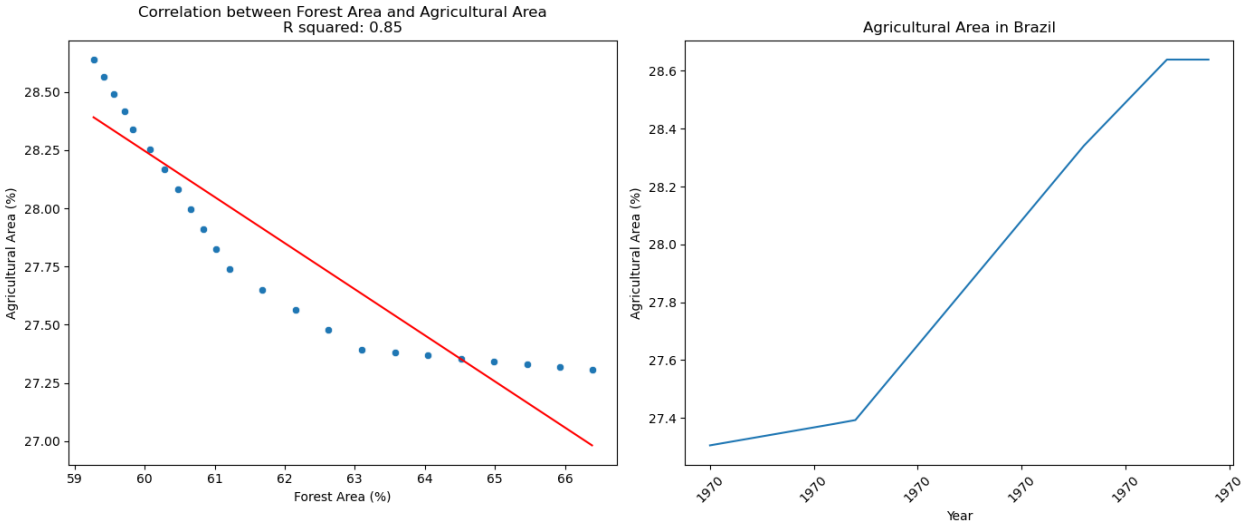


- 3. Correlation between % Forest Area vs % Agricultural Area

In [17]:
```
# Calculate the correlation between Forest area and Agricultural area
forest_area_agri_area = br_data_pivoted['Forest area (% of land area)'].c
print("Correlation between Forest area and Agricultural area:", forest_ar

# Calculate the trend line
x = br_data_pivoted['Forest area (% of land area)']
y = br_data_pivoted['Agricultural land (% of land area)']
x_const = sm.add_constant(x)
model = sm.OLS(y, x_const).fit()
predictions = model.predict(x_const)

# Calculate the R squared
r_squared = model.rsquared

# Set up the figure for side-by-side plots
fig, axs = plt.subplots(1, 2, figsize=(14, 6))

# Subplot 1: Correlation Chart
sns.scatterplot(x='Forest area (% of land area)', y='Agricultural land (%
axs[0].plot(x, predictions, color='red')
axs[0].set_title(f"Correlation between Forest Area and Agricultural Area\
axs[0].set_xlabel("Forest Area (%)")
```

```
axs[0].set_ylabel("Agricultural Area (%)")

# Subplot 2: Agricultural Area Line Chart
axs[1].plot(br_data_pivoted.index, br_data_pivoted['Agricultural land (%
axs[1].set_xlabel('Year')
axs[1].set_ylabel('Agricultural Area (%)')
axs[1].set_title('Agricultural Area in Brazil')
axs[1].xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
plt.xticks(rotation=45)

# Show the plots
plt.tight_layout()
plt.show()
```

```
Correlation between Forest area and Agricultural area: -0.9192949778906869
```

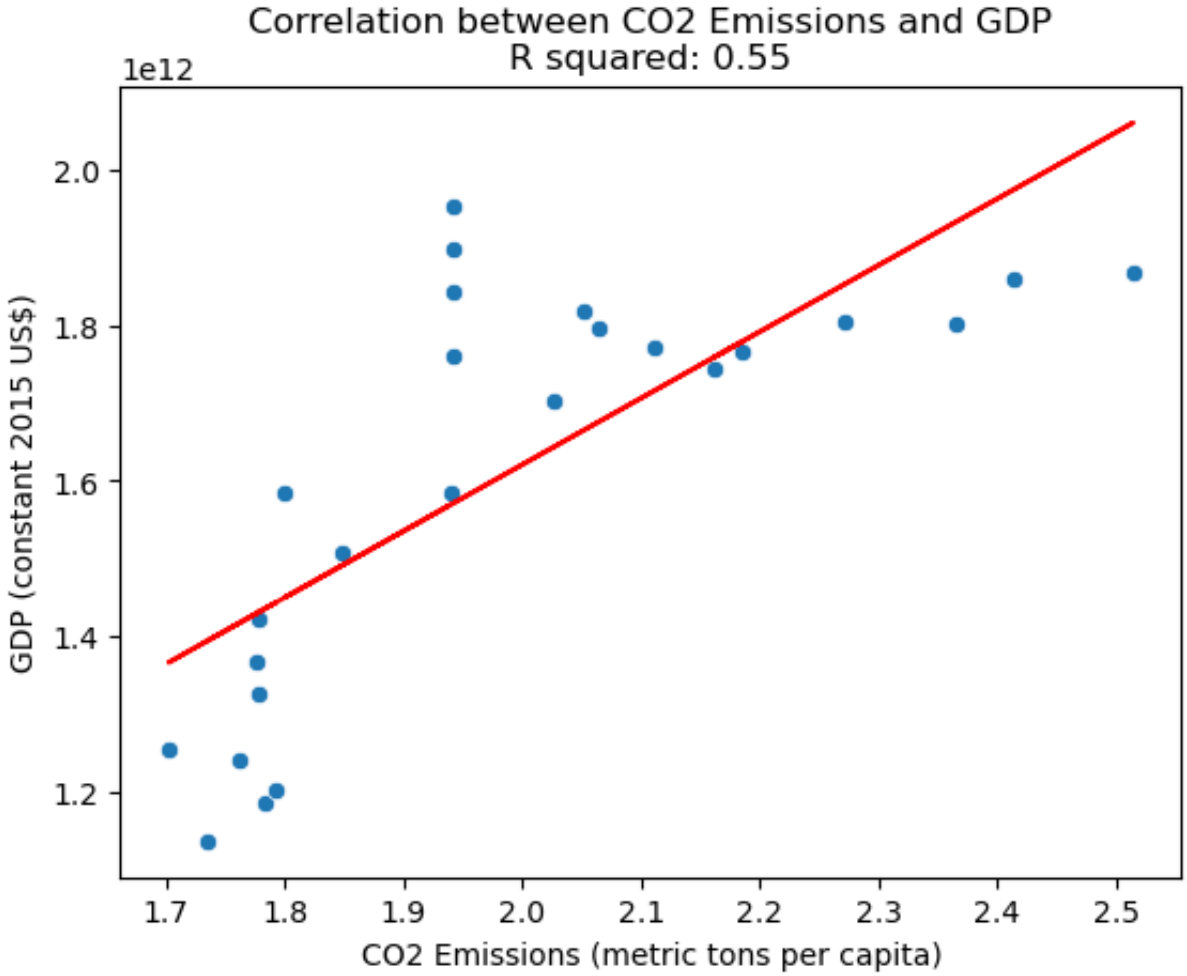


- 4. Correlation between CO2 Emissions vs GDP Growth

In [18]:
```
# Correlation between CO2 emissions and GDP
co2_emissions_gdp = br_data_pivoted['CO2 emissions (metric tons per capit
print("Correlation between CO2 emissions and GDP:", co2_emissions_gdp)

# Calculate the trend line
x = br_data_pivoted['CO2 emissions (metric tons per capita)']
y = br_data_pivoted['GDP (constant 2015 US$)']
x = sm.add_constant(x)
model = sm.OLS(y, x).fit()
predictions = model.predict(x)

# Calculate the R squared
r_squared = model.rsquared

# Plot the scatter plot with the trend line and R squared
sns.scatterplot(x='CO2 emissions (metric tons per capita)', y='GDP (const
plt.plot(x['CO2 emissions (metric tons per capita)'], predictions, color=
plt.title(f"Correlation between CO2 Emissions and GDP\nR squared: {r_squa
plt.xlabel("CO2 Emissions (metric tons per capita)")
plt.ylabel("GDP (constant 2015 US$)")
```

```
plt.show()
```

```
Correlation between CO2 emissions and GDP: 0.742921411651695
```

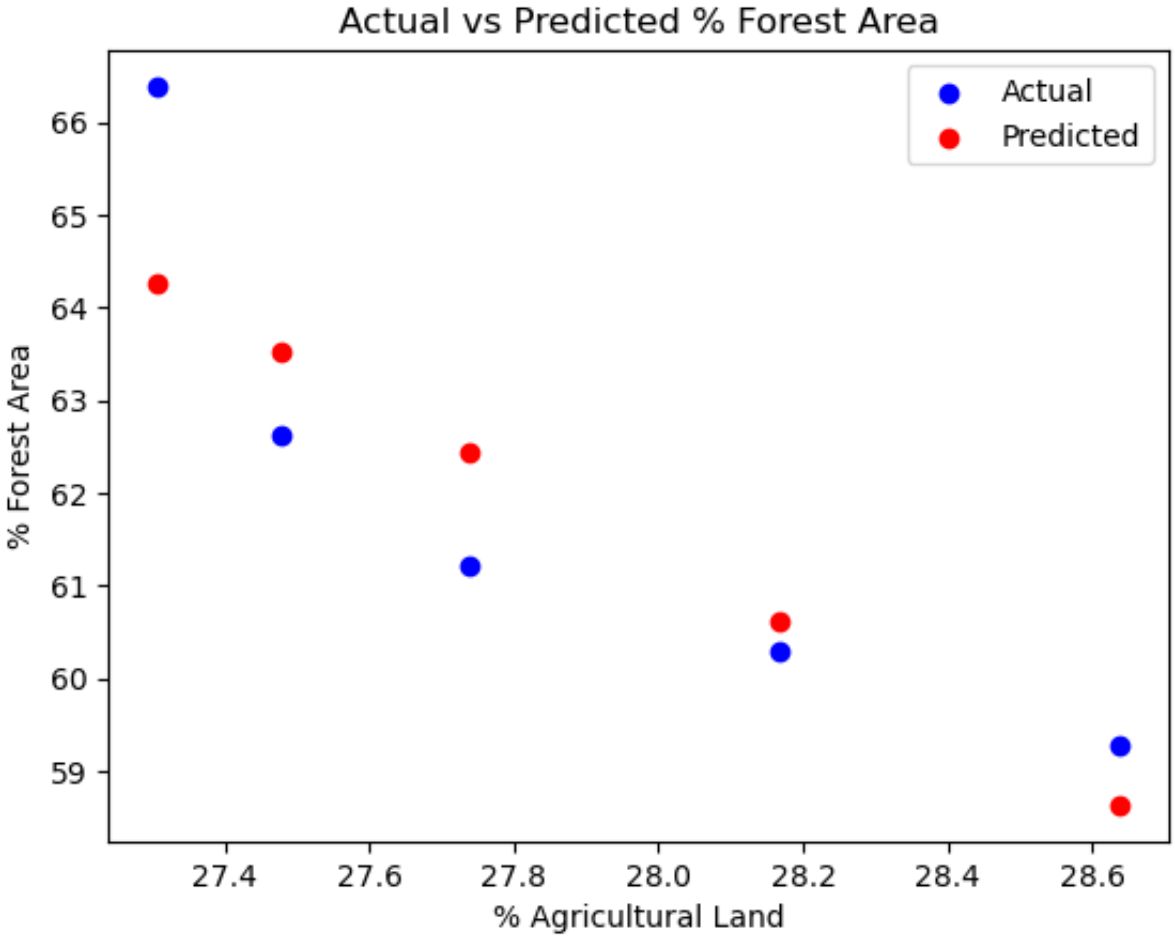


- 5. Linear Regression between % Forest Area (dependent variable) vs % Agricultural Area (independent variable)

In [19]:
```
data = pd.DataFrame({
    'forest_area': br_data_pivoted['Forest area (% of land area)'],
    'agricultural_land': br_data_pivoted['Agricultural land (% of land ar
})

# Check for missing values
print(data.isnull().sum())

# Drop rows with missing values (if any)
data = data.dropna()

# Define features (independent variable) and target (dependent variable)
X = data[['agricultural_land']]
y = data['forest_area']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

```
# Create a linear regression model
model = LinearRegression()

# Fit the model on the training data
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')

# Print coefficients
print(f'Intercept: {model.intercept_}')
print(f'Coefficient for % Agricultural Land: {model.coef_[0]}')

# Plot the results
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.scatter(X_test, y_pred, color='red', label='Predicted')
plt.xlabel('% Agricultural Land')
plt.ylabel('% Forest Area')
plt.title('Actual vs Predicted % Forest Area')
plt.legend()
plt.show()
```

```
forest_area          0
agricultural_land    0
dtype: int64
Mean Squared Error: 1.4727247837819941
R-squared: 0.7596584701176169
Intercept: 179.36378737547085
Coefficient for % Agricultural Land: -4.2156145901900794
```

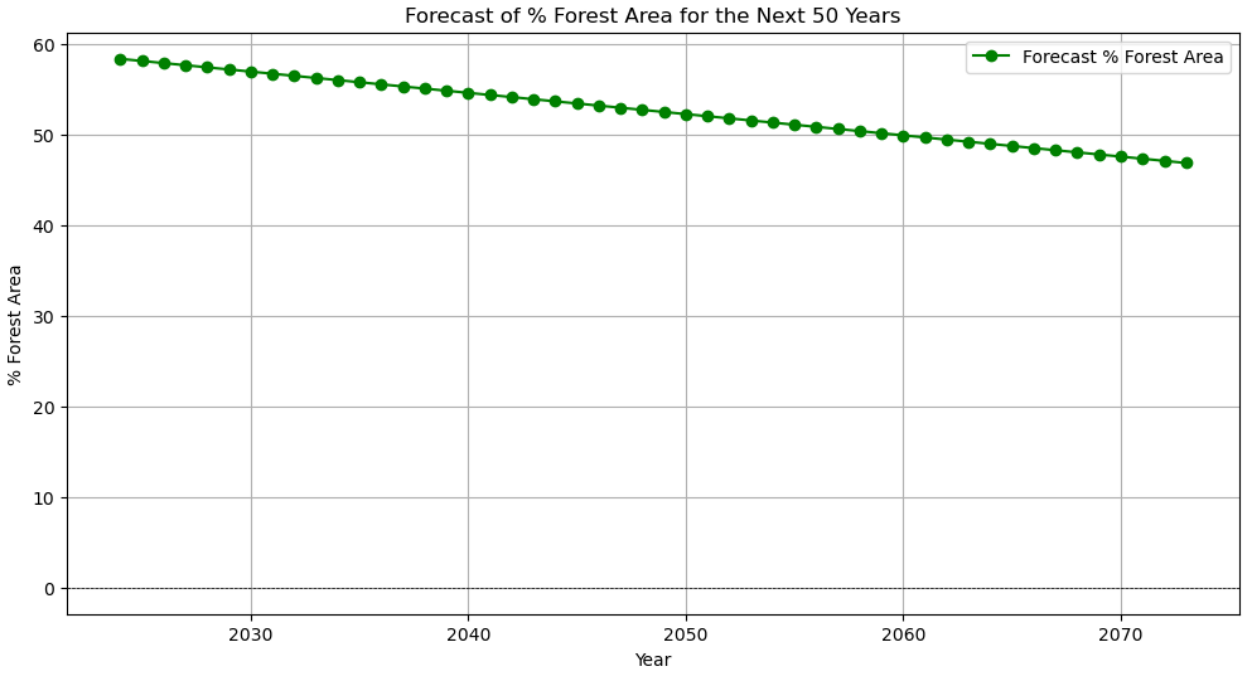


- 6. Forecast of % Forest Area if % Agricultural Area continues to increase by 0.055% per year

In [20]:
```
# Prepare your data again (just as a reference)
data = pd.DataFrame({
    'forest_area': br_data_pivoted['Forest area (% of land area)'],
    'agricultural_land': br_data_pivoted['Agricultural land (% of land ar
})

# Create a linear regression model
X = data[['agricultural_land']]
y = data['forest_area']
model = LinearRegression()
model.fit(X, y)

# Create future data for % Agricultural Land
# Assume you have a starting value for % Agricultural Land, e.g., the las
last_agricultural_land = data['agricultural_land'].iloc[-1]
years = np.arange(1, 51)  # Next 50 years

# Generate future % Agricultural Land values
# Example: assuming an increase of +0.5% each year
change_per_year = 0.055
future_agricultural_land = last_agricultural_land + (years * change_per_y
```

```
# Reshape for prediction
future_agricultural_land = future_agricultural_land.reshape(-1, 1)

# Make predictions
future_forest_area = model.predict(future_agricultural_land)

# Create a DataFrame for future predictions
forecast_years = pd.DataFrame({
    'Year': np.arange(2024, 2024 + 50),
    'Predicted % Forest Area': future_forest_area
})

# Plotting the forecast
plt.figure(figsize=(12, 6))
plt.plot(forecast_years['Year'], forecast_years['Predicted % Forest Area'
plt.xlabel('Year')
plt.ylabel('% Forest Area')
plt.title('Forecast of % Forest Area for the Next 50 Years')
plt.axhline(0, color='black', linewidth=0.5, ls='--')  # Adding a horizon
plt.legend()
plt.grid()
plt.show()
```

```
/opt/anaconda3/lib/python3.11/site-packages/sklearn/base.py:439: UserWarni
ng: X does not have valid feature names, but LinearRegression was fitted w
ith feature names
  warnings.warn(
```



Conclusion

- The climate change is a burning topic and the increasing CO2 emissions have a key role in this process. Brazilian forests - Amazonia as the most important one for decarbonization purposes - are know as the breath of the Earth and help

filtering the CO2 emited not just in Brazil but across the world.

- The analyzed data from 1999 to 2023 suggests there is a gradual deforestation process in Brazil. During this period, the forest area as percentage of land area reduced by 0.28 percentage points per year, resulting in less 6.7% of forest areas by 2023 compared to 1999.
- Simultaneously, Brazil's CO2 emissions increased during the first half of the period, peaking in 2014 and easing since then.
- The urbanization process is highly correlated with the deforestationization process, which might suggest that the presence of rural families helps preserve the forest.
- The expansion of agricultural activities across Brazilian land affects negatively the forest area. On average, for every 1% increase in agricultural land, the percentage of forest area decreases by approximately 4.22% according to the model used (R2 = 76% confidence).
- The percentage of agricultural land in Brazil increases at a rate of 0.055% per year. Considering this trend remains for the next years, forest areas will go from more than 66% of total land area in 1999 to less than 50% in 2060.