Hamoye Stage D

It is often said that the number if trees in an area increases life expectancy, I will like to look into the statement to see if it is a fact or myth. therefore statement of hypothesis

H1: Number of trees in an area increases life expectancy.

H0: Number of trees in an area does not increases life expectancy.

What is life expectancy: Life Expectancy is a statistical measure of the average (see below) time an organism is expected to live, based on the year of its birth, its current age, and other demographic factors including gender. The most commonly used measure is life expectancy at birth (LEB). (wikipedia)

What is forest area: Forest Area generally refers to all the geographic areas recorded as forest in government records. Recorded forest areas largely comprises Reserved Forests (RF) and Protected Forests (PF). Besides RFs and PFs, the recorded forest area may include all such areas, which have been recorded as forests in the revenue records or have been constituted so under any State Act or local laws.

Trees have quite a number of importance in human existence, some of them include:

1. Production of oxygen
2. Prevention of erosion
3. phytoremediation
4. Controlling noise pollution
5. Absorption of carbon dioxide
6. Provision of shade
7. Increase of lifespan

Let's get into scraping our data from wikipedia

1. https://en.wikipedia.org/wiki/List_of_countries_by_life_expectancy (https://en.wikipedia.org/wiki/List_of_countries_by_life_expectancy) for life expectancy data
2. https://en.wikipedia.org/wiki/List_of_countries_by_forest_area (https://en.wikipedia.org/wiki/List_of_countries_by_forest_area) for forest area

In [1]:
```
#importing required libraries
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup as bs
import requests
import urllib.request
import time
from urllib.request import urlopen
```

In [2]:
```
#parsing the web page
url="https://en.wikipedia.org/wiki/List_of_countries_by_life_expectancy"
html=urlopen(url)
soup=bs(html, "html.parser")
```

In [3]:
```
#extracting the table
tables=soup.find_all("table", id="CIA2017")
```

In [4]:
```
#declaring a function to remove unwanted attributes/text
import re
def change_type(text):
    return float(re.sub(r'[^\w\s.]','', text))
```

In [5]:
```
countries=[]
males=[]
females=[]
averages=[]

#parsing the columns to list
for table in tables:
    rows=table.find_all('tr')

    for row in rows:
        cells = row.find_all('td')

        if len(cells) > 1:

            country =cells[0]
            countries.append(country.text.strip())

            male = cells[1]
            males.append(change_type(male.text.strip()))

            try:
                female = cells[2]
                females.append(change_type(female.text.strip()))
            except ValueError:
                females.append(float(69.4))

            average = cells[3]
            averages.append(change_type(average.text.strip()))
```

In [6]:
```
ranks = list(map(int, range(1,225)))
```

In [7]:
```
#converting the lists to dictionary
data={
    "Country Name": countries,
    "Rank": ranks,
    "Male": males,
    "Female": females,
    "Average": averages
}
```

In [8]:
```
#converting the dictionary to dataframe
df = pd.DataFrame(data)
```

In [9]:
```
df.head(10)
```

Out[9]:

	Country Name	Rank	Male	Female	Average
0	Monaco	1	85.6	93.5	89.4
1	Japan	2	81.9	88.8	85.3
2	Singapore	3	82.6	88.1	85.2
3	Macau	4	81.6	87.7	84.6
4	San Marino	5	80.8	86.1	83.3
5	Iceland	6	80.9	85.4	83.1
6	Hong Kong	7	80.4	85.9	83.0
7	Andorra	8	80.7	85.2	82.9
8	Guernsey	9	79.9	85.4	82.6
9	Switzerland	10	80.3	85.1	82.6

In [10]:
```
url1="https://en.m.wikipedia.org/wiki/List_of_countries_by_forest_area"
html1=urlopen(url1)
soup1=bs(html1, "html.parser")
```

In [11]:
```
tables1=soup1.find_all("table", class_="wikitable")[1]
```

In [12]:
```
countries1=[]
xx20s=[]

for table in tables1:
    rows=table.find_all('tr')

    for row in rows:
        cells = row.find_all('td')

        if len(cells) > 1:

            country =cells[0]
            countries1.append(country.text.strip())

            xx20 = cells[4]
            xx20s.append(change_type(xx20.text.strip()))
```

In [13]:
```
data1={
    "Country Name": countries1,
    "2020": xx20s
}
```

In [14]:
```
df1 = pd.DataFrame(data1)
```

In [15]:
```
df1.head()
```

Out[15]:

	Country Name	2020
0	Afghanistan	1208.0
1	Albania	789.0
2	Algeria	1949.0
3	American Samoa	17.0
4	Andorra	16.0

In [16]:
```
#merging both dataframes
final_data = pd.merge(
                df, df1,
                on=["Country Name"]
             )
final_data.rename(columns={"2020":"Forest area (1000 ha)"}, inplace=True)
```

In [17]:
```
final_data.head()
```

Out[17]:

	Country Name	Rank	Male	Female	Average	Forest area (1000 ha)
0	Monaco	1	85.6	93.5	89.4	0.0
1	Japan	2	81.9	88.8	85.3	24935.0
2	Singapore	3	82.6	88.1	85.2	16.0
3	San Marino	5	80.8	86.1	83.3	1.0
4	Iceland	6	80.9	85.4	83.1	51.0

In [18]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 179 entries, 0 to 178
Data columns (total 6 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Country Name           179 non-null    object 
 1   Rank                   179 non-null    int64  
 2   Male                   179 non-null    float64
 3   Female                 179 non-null    float64
 4   Average                179 non-null    float64
 5   Forest area (1000 ha)  179 non-null    float64
dtypes: float64(4), int64(1), object(1)
memory usage: 9.8+ KB
```

In [19]:
```
#saving the csv
final_data.to_csv("Life expectancy and forest area data.csv", index=False)
```

In [20]:
```
#importing the population data
pop=pd.read_csv("Population data.csv")
pop.head()
```

Out[20]:

	Country Name	Country Code	Population
0	Aruba	ABW	106314.0
1	Afghanistan	AFG	38041754.0
2	Angola	AGO	31825295.0
3	Albania	ALB	2854191.0
4	Andorra	AND	77142.0

In [21]:
```
#merging the population dataframe
df3= pd.merge(final_data, pop, on="Country Name")
df3.head()
```

Out[21]:

	Country Name	Rank	Male	Female	Average	Forest area (1000 ha)	Country Code	Population
0	Monaco	1	85.6	93.5	89.4	0.0	MCO	38964.0
1	Japan	2	81.9	88.8	85.3	24935.0	JPN	126264931.0
2	Singapore	3	82.6	88.1	85.2	16.0	SGP	5703569.0
3	San Marino	5	80.8	86.1	83.3	1.0	SMR	33860.0
4	Iceland	6	80.9	85.4	83.1	51.0	ISL	361313.0

In [22]:
```
df3.to_csv("Countries,tree and life data.csv", index=False)
```

In [25]:
```
#reducing the population by 1000 to avoid ambiguity
df3["Population"]=df3["Population"]/1000
```

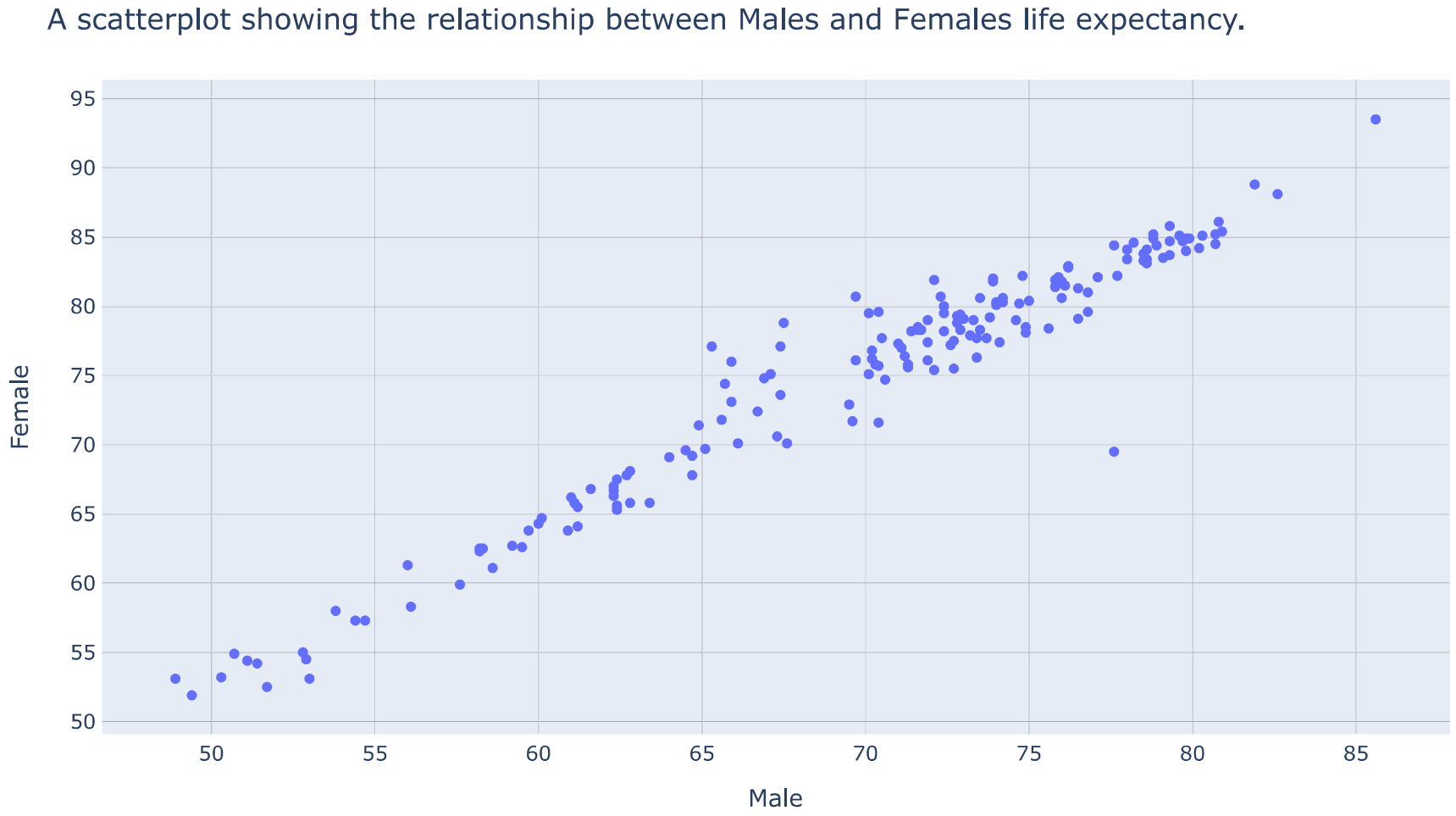
In [26]:
```
#renaming columns
df3.rename(columns={"Population": "Population (1000)"}, inplace=True)
df3.head()
```

Out[26]:

	Country Name	Rank	Male	Female	Average	Forest area (1000 ha)	Country Code	Population (1000)
0	Monaco	1	85.6	93.5	89.4	0.0	MCO	38.964
1	Japan	2	81.9	88.8	85.3	24935.0	JPN	126264.931
2	Singapore	3	82.6	88.1	85.2	16.0	SGP	5703.569
3	San Marino	5	80.8	86.1	83.3	1.0	SMR	33.860
4	Iceland	6	80.9	85.4	83.1	51.0	ISL	361.313

In [27]:
```
#importing the library for plotting
import plotly.express as px
```

In [28]:
```
#showing ratio of male to femaleslife expectancy
fig = px.scatter(df3, x="Male", y="Female", hover_data=["Country Name", "Rank"])
fig.update_layout(title="A scatterplot showing the relationship between Males and Females life expectancy.")
fig.show()
```

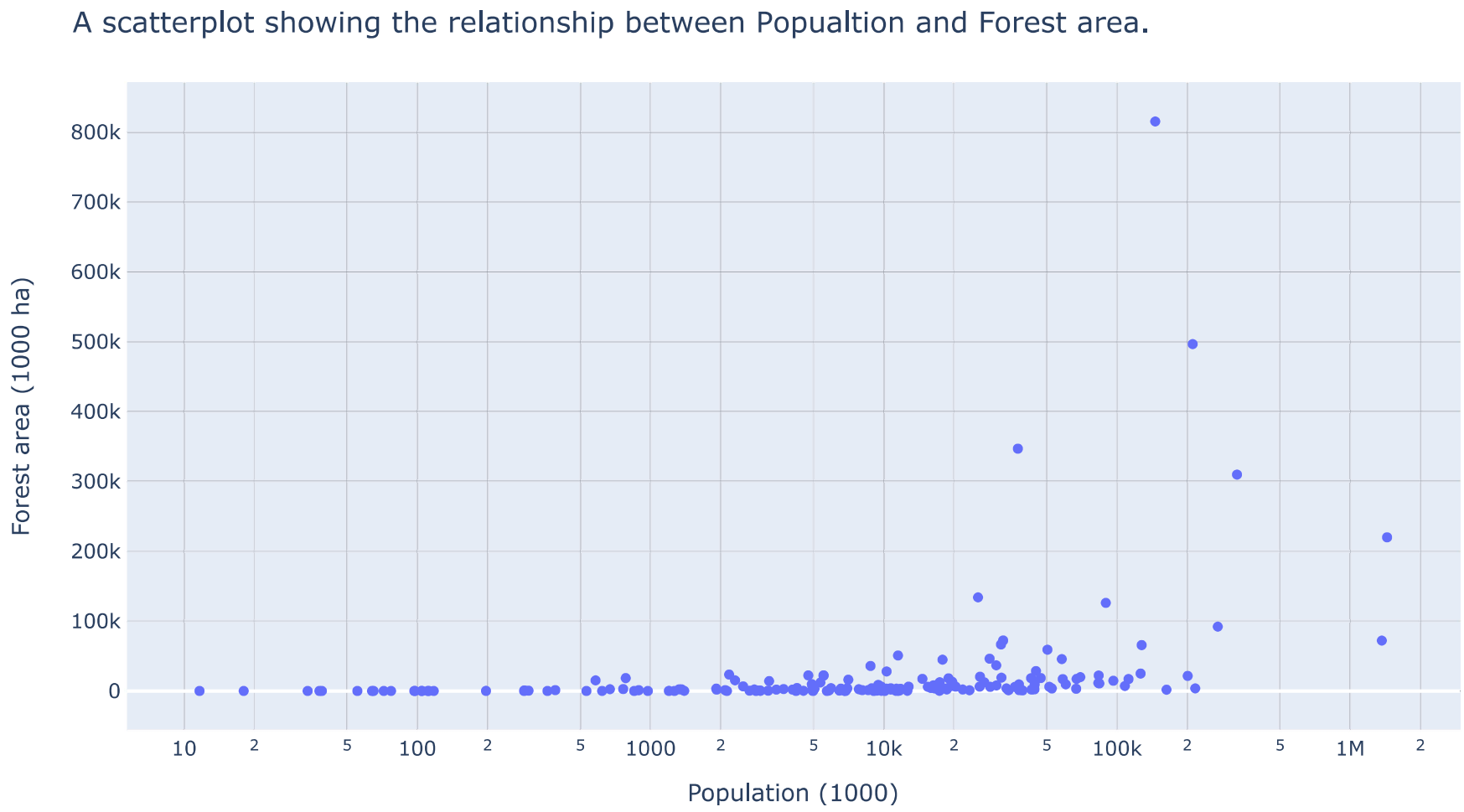


In [29]:
```
#checking the correlation coefficient of males and females
from scipy import stats
a=df3["Female"]
b=df3["Male"]
corr = stats.pearsonr(a, b)
print("Correlation coefficient:", corr[0])
```

```
Correlation coefficient: 0.9723735323147487
```

The scatterplot shows that the relationship between the females and males is strong and the correlation coefficient of 0.97 also shows that St.Vincent and the Grenadines appears to be an outlier with males having 77.6% life expectancy and females having 69.4% with rank of 104 and it appears to be so due to the large distant between the percentage of both sex.

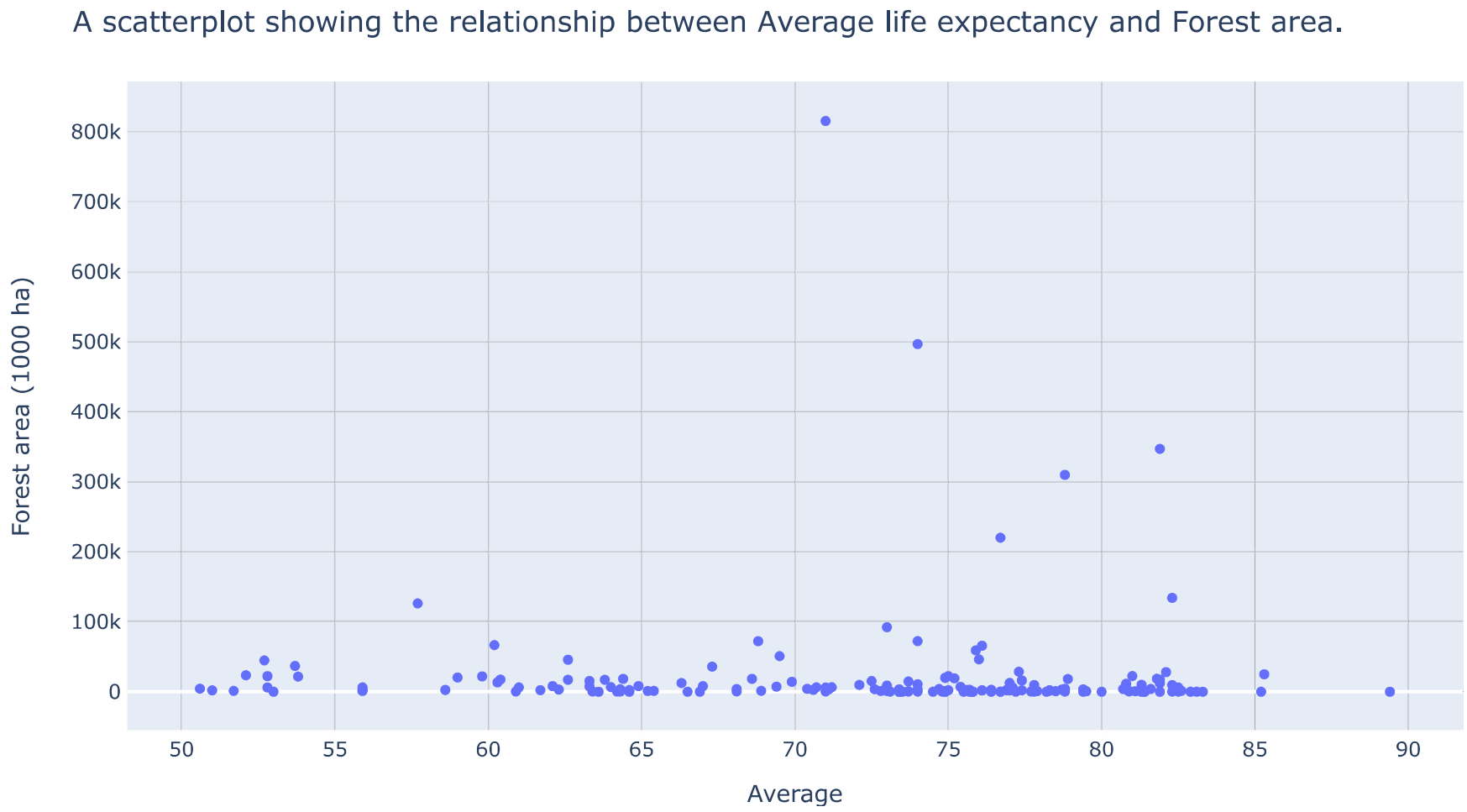
In [30]:
```
#showing population ratio tree planted
fig=px.scatter(df3, x="Population (1000)", y="Forest area (1000 ha)", hover_data=["Country Name", "Average","Rank"],
fig.update_layout(title="A scatterplot showing the relationship between Popualtion and Forest area.")
fig.show()
```



China has the largest population in the world with about 1.4 billion persons and 219,978 hectares of forest area but still makes it to the top 100 countries in terms of life expectancy with an average of 76.7% but in comparison to the number 1 country (Russia) in terms of forest area has a

population of 144.5 million and still ranks 155 despite the small population in comparison to forest area. The second leading country in terms of forest area ranks even higher than Russia with rank of 126.

In [31]:
```
#showing life expectancy ratio trees
fig=px.scatter(df3, x="Average", y="Forest area (1000 ha)", hover_data=["Country Name", "Rank"])
fig.update_layout(title="A scatterplot showing the relationship between Average life expectancy and Forest area.")
fig.show()
```



Most countries forest area appears to be below 100,000 and surprisingly the forest area for Monaco which ranks 1 with an average of 89.4% in terms of life expectancy is 0 because it has no forest area while Russia which ranks 155 with 75% average in terms of life expectancy has 815,312 hectares of forest area.

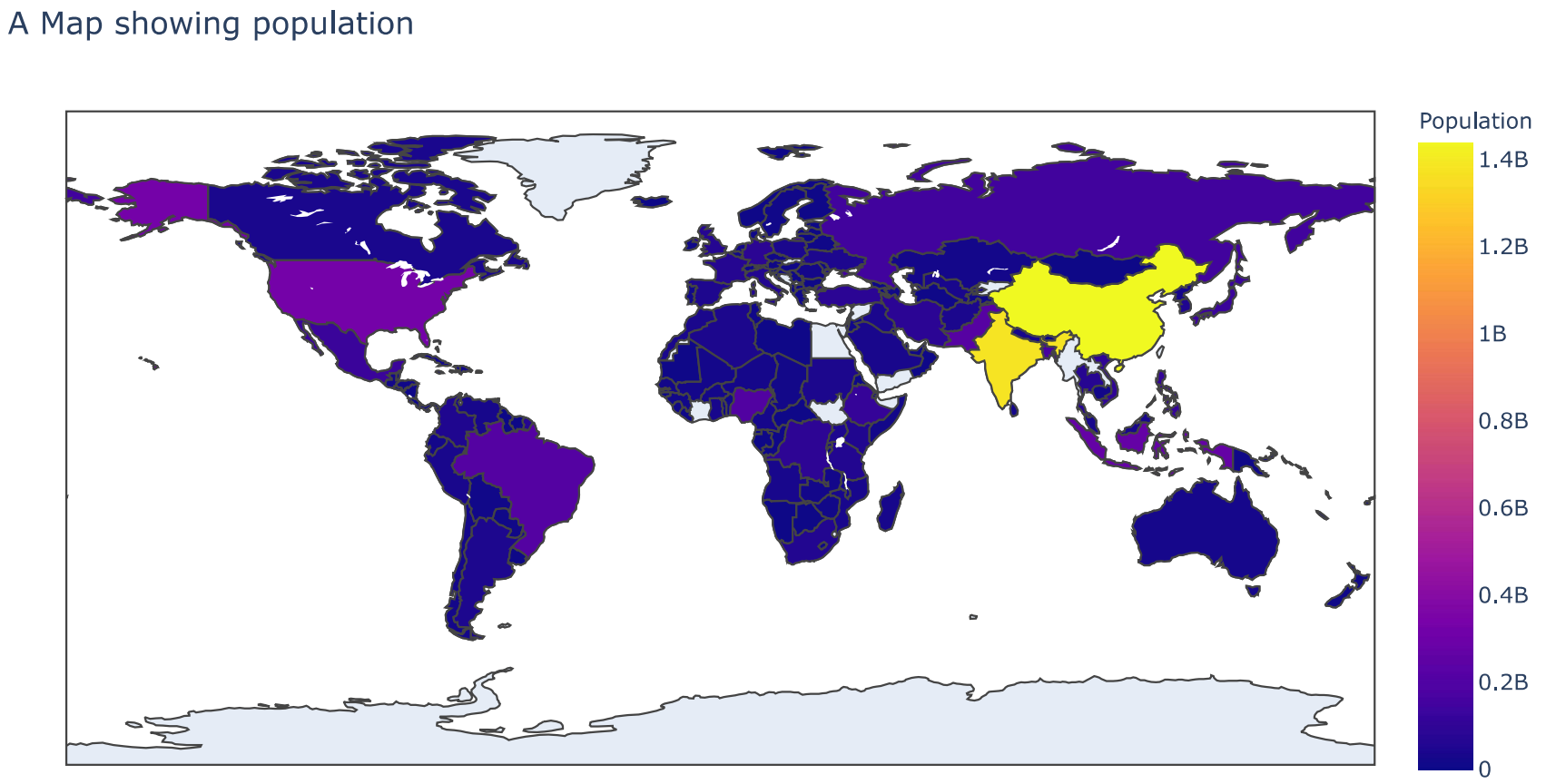
In [34]:
```
#checking the correlation coefficient
a=df3["Average"]
b=df3["Forest area (1000 ha)"]
corr = stats.pearsonr(a, b)
print("Correlation coefficient:", corr[0])
```

```
Correlation coefficient: 0.015616438009609133
```

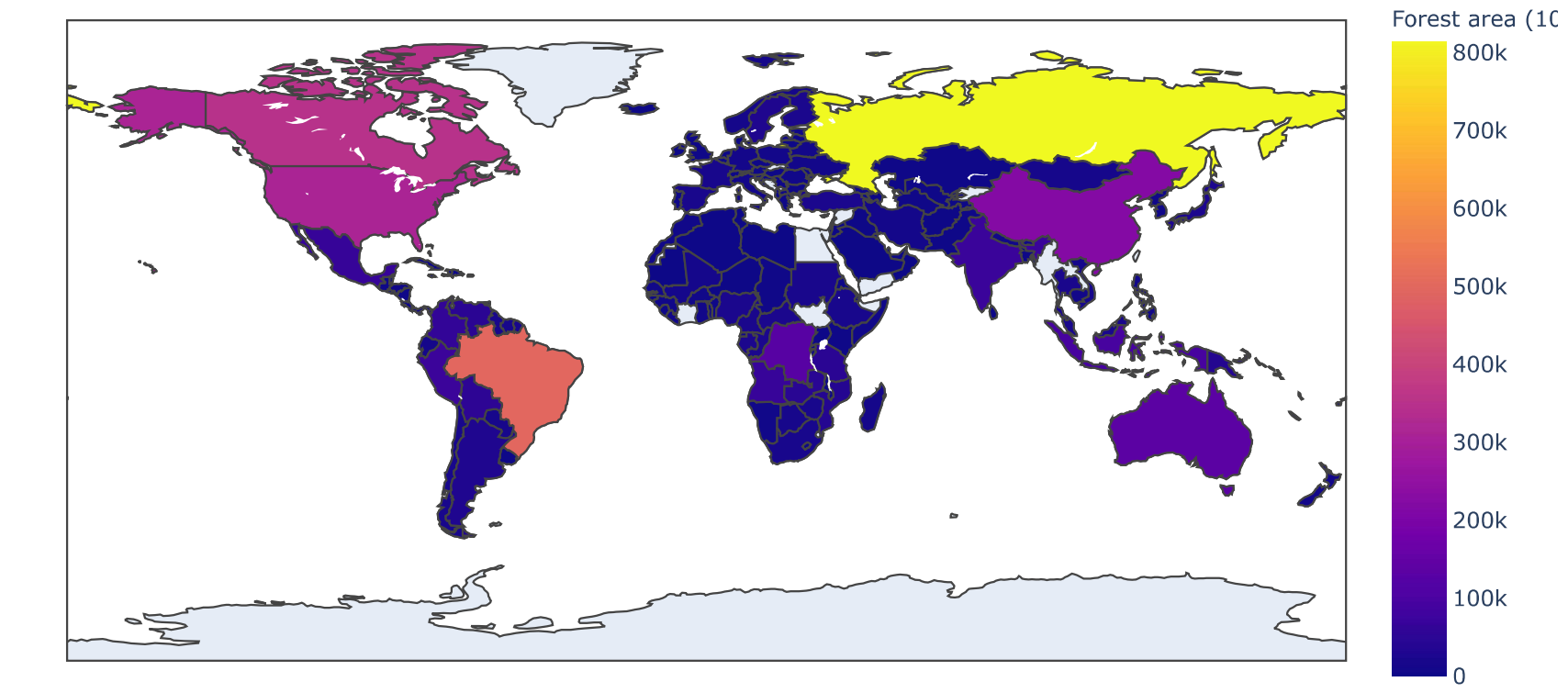
The correlation coefficient shows that there is a very low and therefore implies that there is little or no relationship between number of trees and life expectancy and we therefore reject the null hypothesis and accpet the alternative hypothesis which says "Number of trees in an area does not increase life expectancy".

Showing the geospatial data.

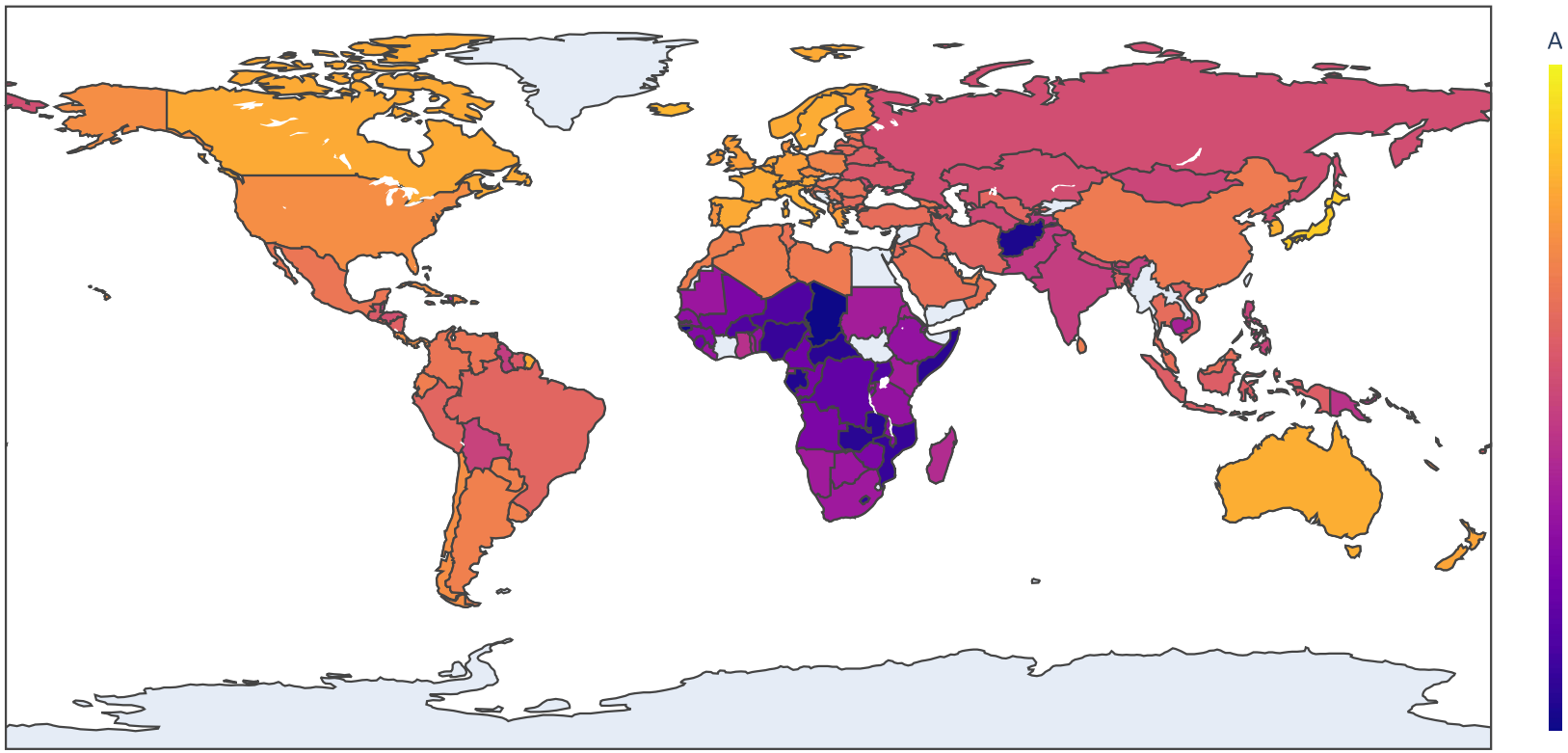
In [35]:
```
#showing map view of population
fig = px.choropleth(df3, locations="Country Code", color="Population (1000)", hover_name="Country Name",
                    color_continuous_scale="Plasma")
fig.update_layout(title="A Map showing population")
fig.show()
```



In [36]:
```
#showing map view of forest area
fig = px.choropleth(df3, locations="Country Code", color="Forest area (1000 ha)",
                    hover_data=["Country Name", "Average","Rank"],
                    color_continuous_scale="Plasma")
fig.update_layout(title="A Map showing Forest area (1000 ha)")
fig.show()
```



In [37]:
```
#showing map view of forest area
fig = px.choropleth(df3, locations="Country Code", color="Average",
                    hover_data=["Country Name", "Male", "Female","Rank", "Population (1000)"],
                    color_continuous_scale="Plasma")
fig.update_layout(title="A Map showing Life expectancy")
fig.show()
```



In [ ]: