PRODUCTION AND USAGE OF RENEWABLE ENERGY

ES1101: COMPUTATIONAL DATA ANALYSIS

SUBMITTED BY:

Aditya Shukla (2020BTechCSE006)
Arushi Singh (2020BTechCSE012)
Garv Baheti (2020BTechCSE031)
Sparsh Goud (2020BTechCSE077)

FACULTY GUIDE:

Dr. Umesh Gupta
Dr. Sonal Jain

Institute of Engineering and Technology (IET)
JK LAKSHMIPAT UNIVERSITY, JAIPUR

13 February 2021

TABLE OF CONTENTS

ABSTRACT

With the increasing population and advancement of technology the demand of energy to be put in use is incrementing on a daily basis. The limitation of fossil fuels, natural resources on earth puts forth a serious issue and challenge in the upcoming future. To cope up with the current needs and the needs of the future generation, consumption & production of renewable energy must increase. Advantage of using renewable energy is that it does not affect or harm the mother earth in anyway and it is practically limitless in nature. In 2020 global scenario of renewable energy has increased by 1.5% relative to 2019 and production has increased by 3%. The share of renewables in global electricity generation had a sudden increment from 26% in 2019 to 28% in 2020.

Introduction

The 17 Sustainable Development Goals (SDGs), with their 169 targets, structure the centre of the 2030 Agenda. They balance the monetary, social, and environmental components of manageable turn of events, and spot the battle against destitution and reasonable advancement on a similar plan interestingly.

Here is a brief introduction of each one of them-

Goal 1. "End poverty in all its forms everywhere"

The focus of this goal is eradicating poverty in all forms for each people everywhere in the world. Current estimation of poverty is considered regarding individuals living under $1.25 every day. The objective likewise centres around diminishing extent of men, ladies and kids living under neediness considerably.

Goal 2. "End hunger, achieve food security and improved nutrition and promote sustainable agriculture"

Focus of this goal is to eliminate the hunger and ensuring that nutritious food is open to all individuals all through the globe, ailing health of all structures by 2025, increasing profitability of horticultural and food maker's pay by multiple times.

Goal 3. "Ensure healthy lives and promote wellbeing for all at all ages"

Centre of point of this is the declining the maternal mortality, eliminating all preventable /reparable reasons for death of kids under 5 years old, battling the transferable illness and non-transmittable infection and to advance emotional well-being.

Goal 4. "Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all"

It ensures equivalent learning freedoms to the two people in instructive level, and offer admittance to quality youth advancement, moderate professional, tertiary schooling and quality specialized to the two people and taking out sexual orientation incongruities.

Goal 5. "GENDER EQUALITY"

This objective spotlights on destroy victimization ladies and young ladies around globe, savagery against them including dealing, kid marriage or constrained marriage and guaranteeing equivalent freedoms is given.

Goal 6. "Clean WATER AND SANITATION"

The mission focuses on ensuring that safe, accessible and clean drinking water is available to people, providing sufficient sanitation and hygiene and banishing open defecation.

Goal 7. "Affordable and Clean energy"

The aim is to have access to efficient, commercially viable, integrated energy resources and to maximize the overall share of renewable energy in the globe.

Goal 8. "Decent work and economic growth"

It focuses on achieving balanced economic growth per capita, on growing economic competitiveness through technological innovations, and on implementing development-oriented policies.

Goal 9. "Industry innovation and infrastructure"

It focuses on the growth of quality, safe and durable infrastructure, the promotion of sustainable industrialization and the ease of creating small-scale businesses.

Goal 10. "REDUCE INEQUALITY"

The goal focuses on accomplish and sustaining income growth of last most 40% of population at rate higher than the national average, promoting inclusivity for all.

Goal 11. "Sustainable cities and communities"

It focuses on providing sufficient housing and transit infrastructure, essential amenities, to improve sustainable urbanization, to conserve and protect the cultural and natural resources of the world.

Goal 12. "Ensure sustainable consumption and production patterns"

It reflects on the introduction of the 10-year Sustainable Development & Use Trend Policy Plan, the sustainability of natural resource use management, the 50 percent elimination of global waste at the retail & customer level, and the environmentally responsible management of chemicals and other waste going through the life cycle.

Goal 13. "**Take urgent action to combat climate change and its impacts"**

A broad variety of topics related to climate action are addressed. Strengthening climate-related catastrophe adaptation capability, sustainability and incorporation of climate change initiatives into policies.

In addition, it is interested in the settlement of means for meeting goals and the adoption of the UN Framework Convention on Climate Change.

Goal 14. "**Conserve and sustainably use the oceans, seas and marine resources for sustainable development"**

It works on the prevention of sea and underwater life and the elimination of marine waste related to land-based activities and pollution. This task also centers on the minimization of ocean acidification in an organisation of increased scientific collaboration.

Goal 15. "Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss"

It focuses on preservation, the regeneration of terrestrial and freshwater habitats, the end of deforestation, the recovery of degraded forests and the end of desertification. Preventing invasive alien species on land thus prevents the poaching and exploitation of endangered species.

Goal 16."Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels"

It works mainly on preventing crime and associated deaths. It also insists on ensuring that, by upholding the national and international rule of law, fair treatment is provided to all people, reducing bribery and corruption in all ways.

Goal 17. "Strengthen the means of implementation and revitalize the Global Partnership for Sustainable Development"

It focuses on reinforcing the mobilization/flow of domestic capital, supplying developing countries with financial resources from different sources, and integrating development assistance from developed countries.

We have chosen SDG -7 i.e.

"Affordable and Clean energy"

Our fundamental objective is to guarantee admittance to dependable, moderate, maintainable and present day energy for all.

Our standard day by day presences rely upon strong and sensible energy organizations to work effectively and to develop fairly. A grounded energy system supports all regions: from associations, drug and preparing to cultivating, establishment, correspondences and high-advancement. Then again, nonappearance of admittance to energy supplies and change systems is a restriction to human and financial unforeseen development

For quite a while, fossil empowers, for instance, coal, oil or gas have been huge wellsprings of force creation, regardless, burning-through carbon invigorates conveys a ton of ozone draining substances which cause ecological change and harmfully influence people's success and the environment. This impacts everyone, not just a couple. Likewise, overall force use is rising rapidly. Pretty much, without a consistent force supply, countries will not be able to control their economies.

Countries can revive the change

Survey of Literature

Leontiy Eder et al.(2018)[1] explained the review headway considering the degree of monetary advancement of various areas of energy power of the world economy. Study has uncovered that Major/Large scale locales of the world have just started decreasing the energy power of their financial matters. Monetary energy force has been portrayed by the dramatic pattern, obviously showing decrease in the energy effectiveness in the economies. Estimating the Beta combination in time arrangement of energy force, validating accordingly assembly of economy approach has been utilized. Relative investigation exhibited that the dramatic pattern is an extraordinary method of depicting the energy power of an economy in lasting and variable costs both for created and creating areas of the world.

Anca Mehedintu et al.(2018)[2] Contemplated the necessity of expanding the sustainable power utilization and decrementing last energy utilization. Expansion in the sustainable power utilization should arrive at the objective of 20% in the last energy. European Association Order 2009/28/EC expresses that the measure of sustainable power in the last energy utilization should reach upto 20% by 2030 in EU nations.

The last energy utilization is the energy utilized by purchasers just as misfortunes and influence plants utilization. In Sustainable power Order utilization of gross last energy is the energy conveyed agribusiness, family units, ranger service, transport and utilization of power utilization of gross last energy from all assets incorporates atomic warmth, strong fills, electrical energy complete oil items and environmentally friendly power.

Michael Cary et al.(2019)[3] examined the monetary parts of simplicity of having clean fills and clean innovations at family unit level for cooking purposes. Using the contingent assembly model, simplicity of having clean powers and clean innovation is in the end going to ascend around the planet, reason being monetary development, alongside concerns in regard to spatial heterogeneity and autocorrelation of residuals. Comparative development on getting to clean energizes and innovations among all the nations has not been found. Comparable development speculation is dismissed based on the club intermingling approach, shifting standard of conduct across various gatherings is noticed while tending to the worries of autocorrelation of residuals and spatial heterogeneity.

Nicole Vandaele et.al(2015)[4] Proposed the universal framework outlining the capacity installation per year until 2040 required in producing 100% of the demand of electricity through five major renewable resources.

The framework could be applied to all countries, areas, region without concerning about their current state of development, socioeconomic prosperity.

The framework is applied on 4 different countries according to their development index.

The installation in developing countries results in sudden socioeconomic progress resulting in development in all major areas of human needs.

Objectives:

Objective 1: To study the energy intensity measured in terms of primary energy and GDP to know dependency on primary energy among countries.

Objective 1.1: To verify the claim of UN that energy intensity from 2001 to 2017 has decreased by 1.48 (Megajoules per Constant 2011 purchasing power parity GDP)

Objective 1.2: To observe the energy intensity measured in terms of primary energy and GDP data from year 2000 to 2017 and then predict the energy intensity measured in terms of primary energy and GDP from 2018 to 2030.

Objective 1.3: To analyze the developmental status of all continents of energy intensity measured in terms of primary energy and GDP across the world.

Objective 2: To investigate the usage of renewable energy in total energy consumption to know progress and development of countries.

Objective 2.1: To verify the claim of UN that the Contribution of Renewable Energy in Total Energy is Increased.

Objective 2.2: To observe the usage of renewable energy in total energy consumption from year 2000 to 2017 and then predict the usage of renewable energy in total energy consumption from 2018 to 2030.

Objective 2.3: To analyze the developmental status of all continents of usage of renewable energy on total energy consumption.

Objective 3: To examine the proportion of population with relying on clean fuels and technology to enhance perception about awareness of people of different regions.

Objective 3.1: To verify the claim of UN that the proportion of population relying on clean fuel is 50% in 2001 and 62% in 2017.

Objective 3.2: To observe the proportion of population with relying on clean fuels and technology from year 2000 to 2018 and then predict the proportion of population with relying on clean fuels and technology from 2019 to 2030.

Objective 3.3: To analyze the progress of different regions of Africa on the basis of reliance of population on clean energy and technology from year 2014-2018

Objective 4: To analyze the installed renewable energy generating capacity in developing countries showing the advancements of technologies.

Objective 4.1: To verify the claim of UN that installed renewable electricity generating capacity (watts per capita) is 65.202 in 2001 and 188.01 in 2017.

Objective 4.2: To observe the installed renewable energy generating capacity in developing countries from year 2000 to 2018 and then predict the analyze the installed renewable energy generating capacity in developing countries from 2019 to 2030.

Objective 4.3: To analyze the progress of different regions of Asia on the basis of Installed renewable electricity-generating capacity (watts per capita) from year 2011-2018

Data Collection

We have taken data from United Nations authentic database. We have considered data of the year 2001 and 2017 of different countries.

Dataset for Objective 1-

(Click to view all Objectives)

GeoAreaN	2001	2017				
Afghanista	11	34				
Albania	41	78				
Algeria	95	95				
Andorra	95	95				
Angola	44	48				
Antigua an	95	95				
Argentina	95	95				
Armenia	83	95				
Australia	95	95				

Dataset for Objective 2-

(Click to view all Objectives)

GeoAreaN	2001	2017				
Afghanista	54.06	24.65				
Africa	59.79	54.38				
Albania	39.13	37.19				
Algeria	0.43	0.14				
American	0	1.75				
Americas	11.11	16.47				
Andorra	15.39	19.14				
Angola	72.34	56.16				
Anguilla	0.18	0.18				

Dataset for Objective 3-

(Click to view all Objectives)

GeoAreaN	2001	2017				
Afghanista	1.86	1.93				
Africa	6.98	5.65				
Albania	4.2	2.91				
Algeria	3.45	4.05				
Americas	6.37	4.79				
Angola	4.44	3.41				
Antigua ar	2.93	3.24				
Argentina	4.65	4.28				
Armenia	8.54	5.18				

Dataset for Objective 4-

(Click to view all Objectives)

GeoAreaN	2001	2017				
Afghanista	8.863	9.792				
Algeria	8.794	16.024				
American S	0	75.085				
Angola	13.904	81.472				
Anguilla	0	146.764				
Antigua an	0	40.87				
Argentina	234.094	244.819				
Armenia	331.893	452.367				
Aruba	0	361.614				

METHODOLOGY

MEAN: Arithmetic mean (average) is the sum of complete perceptions partitioned by the quantity of perceptions.

Mean Calculating method:

- First calculate the sum of all the observations.
- Then divide the sum by total number of observations.

$$M = \sum_{i=1}^{n} x$$

M = Mean (Average)

x = observations

n = total no. of observations

i = 1,2, 3, 4, 5……n

STANDARD DEVIATION: It is a quantity which states that by how much the observations differ from the mean value.

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ = Standard Deviation

xi = observations

N = no. of observations

μ = mean value of the observations

Standard Deviation

Find mean then for each observations , find the square of its difference to mean value of observation. Then add all the square values. After that divide it by the number of observations. Then take square root of the value obtained.

HYPOTHESIS TESTING

Whenever we are trying to derive a certain conclusion of the complete data set by taking into account only a sample of the whole population , then we use hypothesis testing.

Two attributes of hypothesis testing are:

1. Null Hypothesis (H_0)
2. Alternate Hypothesis (H_a)

We validate our assumption as statistically effective or not, using the null hypothesis and alternate hypothesis.

So , basically in building hypothesis we will first take the mean of our sample and claimed mean , then claimed mean will be taken as null hypothesis(H_0) and the opposite case will go in alternate hypothesis(H_a) then after the test we will see whether our hypothesis is rejected or accepted.

There are two types of test:

1. Two Tailed Test
2. One Tailed Test: These are of two type right tailed tests & left tailed tests.

Test of Hypothesis concerning single mean (single Population)

Building Hypothesis

Two tailed tests:

1. Null Hypothesis will be (H_0) : $\mu=\mu_0$
2. Alternate Hypothesis will be (H_a): $\mu \neq \mu_0$

One tailed Tests:

1. Left-tailed :

   a. Null Hypothesis will be $H_0 : \mu \geq \mu_0$

   b. Alternate Hypothesis will be $H_a : \mu < \mu_0$
2. Right tailed :

a. Null Hypothesis will be (H_0): $\mu \leq \mu_0$

b. Alternate Hypothesis will be (H_a): $\mu > \mu_0$

Test applied after building hypothesis

Case 1 : **Z-test** (applied when sample size (i.e n) n>30)

If population variance is known

$$Z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n}}$$

Case 2 : **t-test** (applied when sample size (i.e n) n<30)

If population variance is unknown

$$: t = \frac{\bar{x}-\mu}{s/\sqrt{n}}, \text{ where } s^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

After this , according to the sample size z and t test will be performed then we will see whether the hypothesis is accepted or rejected. Value of z or t according to the sample size will be given in the table then we have to compare the calculated z or t with the value(of z or t) given in the table , so after the comparison our hypothesis will be rejected or failed to be rejected.

Criteria of Rejection:

Rejection of Null Hypothesis if :

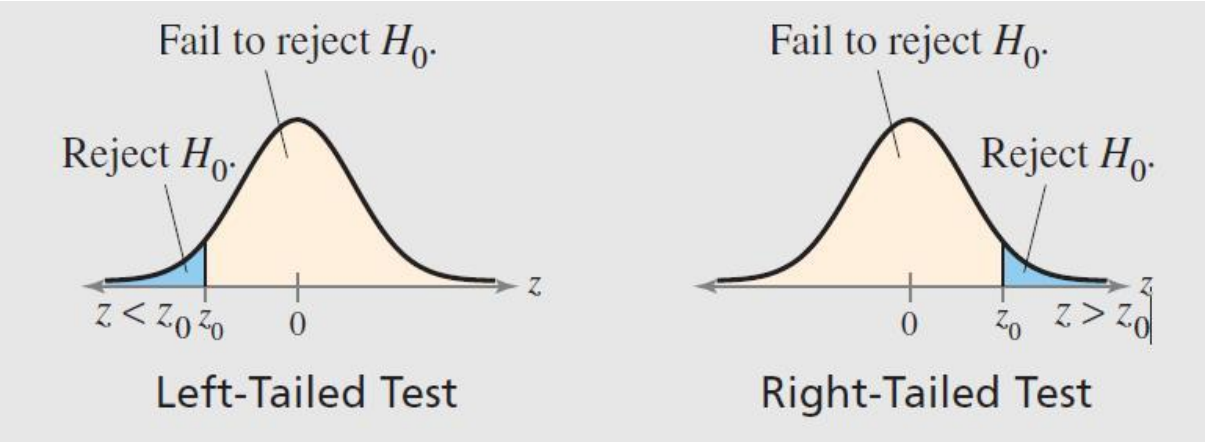
n : Sample size

α : Significance Level

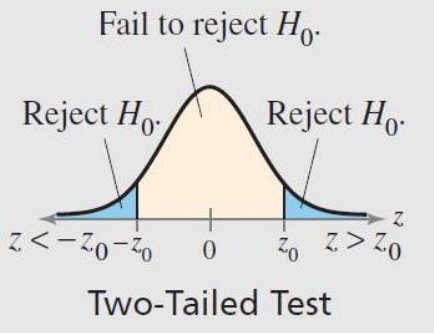
Case 1:

1. $z > z_\alpha$ (Right Tailed)
2. $z < -z_\alpha$ (Left Tailed)

} One-Tailed



3. $|z| > z_{\alpha/2}$



Case 2:

1. $t > t_{\alpha, n-1}$ (Right Tailed)
2. $t < -t_{\alpha, n-1}$ (Left Tailed)

} One-Tailed

3. $|t| > t_{\alpha/2, n-1}$ (Two Tailed)

TEST of Hypothesis [difference of means (Two Population)]

Building Of Hypothesis

Null Hypothesis will be (H_0): $\mu_1 - \mu_2 = \delta$

Alternate Hypothesis will be (H_a): $\mu_1 - \mu_2 \neq \delta$ (Two Tailed)

$\mu_1 - \mu_2 < \delta$ (Left Tailed)

$\mu_1 - \mu_2 > \delta$ (Right Tailed)

Significance Level: α

Sample size : Sample1 = n_1

Sample2 = n_2

Test applied after building Hypothesis

Case 1: If population variance is known :

$$z = \frac{\overline{x_1} - \overline{x_2} - \delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Where, σ_1^2 and σ_2^2 are population variances of sample1 and sample 2 respectively.

Case 2: If population variance is unknown and n < 30 :

$$t = \frac{\overline{x_1} - \overline{x_2} - \delta}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Where,

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

s_1^2 and s_2^2 are sample variances of sample1 and sample2 respectively.

Here, if $n_1 = n_2$

$$s_p^2 = \frac{s_1^2 + s_2^2}{2}$$

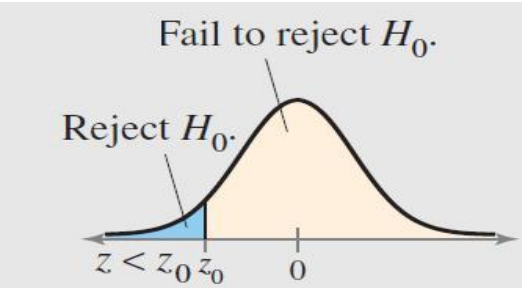
Criteria of Rejection :

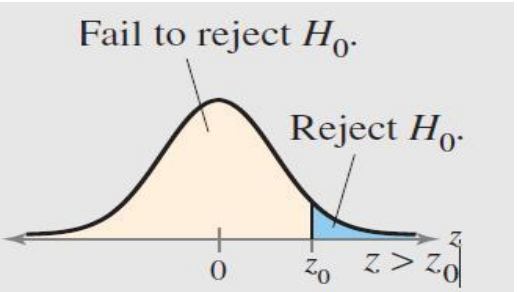
Null hypothesis would be rejected if :

Case 1:

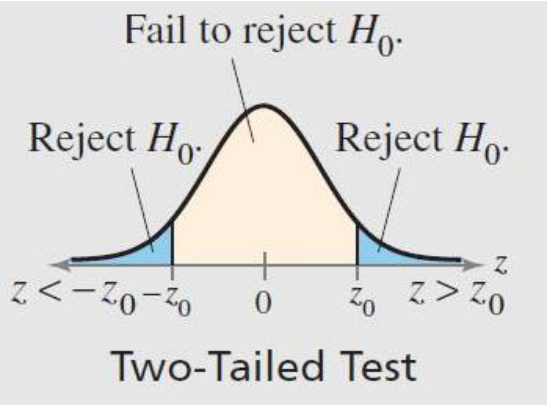
1. $z < -z_\alpha$ (Left Tailed)
2. $z > z_\alpha$ (Right Tailed)

} One-Tailed





3. $|z| > z_{\alpha/2}$ (Two Tailed)



Case 2:

1. $t < -t_{\alpha, n1+n2-2}$ (Left Tailed)
2. $t > t_{\alpha, n1+n2-2}$ (Right Tailed)

} One-Tailed

3. $|t| > t_{\alpha/2, n1+n2-2}$ (Two Tailed)

TEST OF HYPOTHESIS CONCERNING PROPORTION (SINGLE PROPERTION)

Building Hypothesis

Null Hypothesis will be (H_0): $p = p_0$

Alternate Hypothesis will be (H_a): $p \neq p_0$ (Two Tailed)

$p < p_0$ (Left Tailed)

$p > p_0$ (Right Tailed)

} One-Tailed

Sample size : n

Significance Level: α

Test applied after building hypothesis

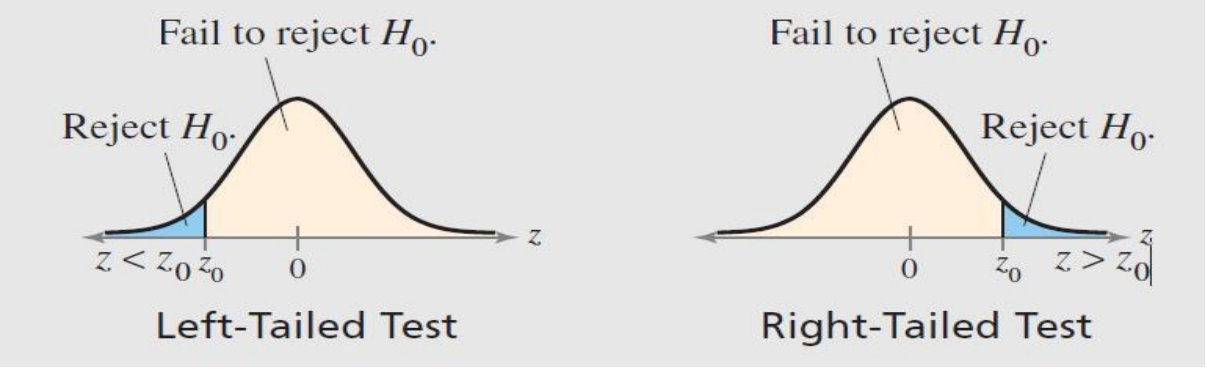
$$z = \frac{x - np_0}{\sqrt{np_0(1-p_0)}}$$

Criteria Of Rejection

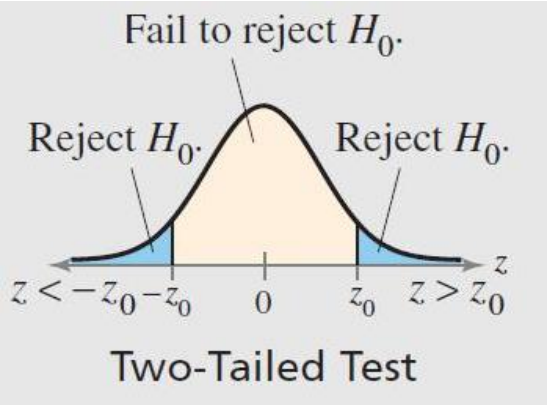
Reject the null hypothesis if :

1. $z > z_\alpha$ (Right Tailed)
2. $z < -z_\alpha$ (Left Tailed)

} One-Tailed



3. $|z| > z_{\alpha/2}$ (Two Tailed)



TEST OF HYPOTHESIS CONCERNING DIFFERENCE OF PROPORTIONS (TWO POPULATIONS)

Building Hypothesis

Null Hypothesis $H_0 : p_1 = p_2$

Alternative Hypothesis $H_a : p_1 > p_2$ (Right Tailed)

$p_1 < p_2$ (Left Tailed) } One-Tailed

$p_1 \neq p_2$ (Two Tailed)

Level Of Significance : α

Sample Size : sample1 = n_1

sample2 = n_2

Test applied after building hypothesis

$$z = \frac{p_1 - p_2}{\sqrt{pq\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}}$$

Where,

$$p = \frac{n_1 p_1 + n_2 p_2}{n_1 + n_2}$$

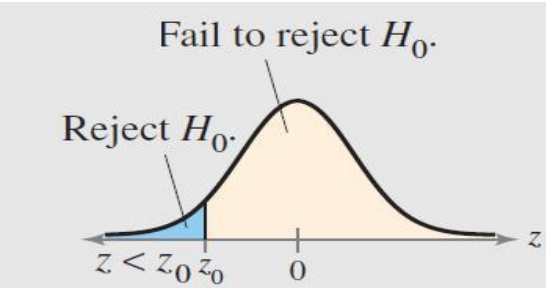
p_1 and p_2 are proportions of sample1 and sample2 respectively , and q=1-p

Criteria Of Rejection

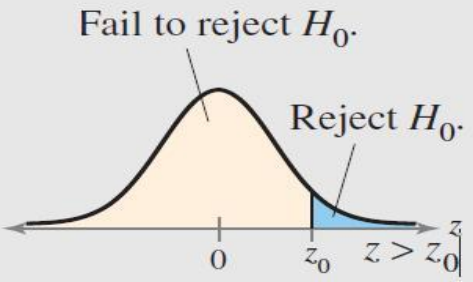
Null hypothesis would be rejected if:

1. $z < -z_\alpha$ (Left Tailed)
2. $z > z_\alpha$ (Right Tailed)

} One-Tailed

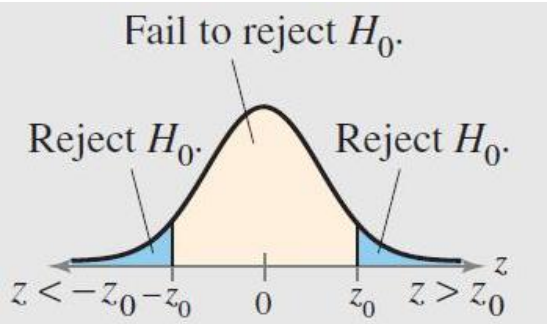


Left-Tailed Test



Right-Tailed Test

3. $|z| > z_{\alpha/2}$ (Two Tailed)



Two-Tailed Test

Regression

Regression is the analysis of the essence of the variable's relationship such that the uncertain value of one variable can be estimated for another variable's known value.

One variable is treated as an independent variable in regression, and another variable is treated as a dependent variable.

With the help of regression, possible values of the dependent variable are estimated on the basis of the values of the independent variable.

$$\hat{y} = a + bX$$

Equation of the line

Here, b is the slope

x coordinate will be year just for an example for 2000 : x =1 , for : 2001 x=2 , for : 2002 x=3 ……. for : 2020 x=21

y coordinate will be the mean per year

example:

for year 2000 :

x=1 y=mean of 2000

$\overline{y}$ · Is the mean of y

$\overline{x}$ Is the mean of x

$$b = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sum x^2 - \frac{(\sum x)^2}{n}}$$

n: total no. of observations(i.e year)

$$\hat{y} = \bar{y} + b(x - \bar{x})$$

Here, for x=1 plot y(cap) in the graph then we have to do the same thing for each year then after plotting the observed data then in the graph we will plot for the upcoming years till 2030 in which prediction will be set and criteria for each upcoming year will be set so that we can achieve the criteria set every year to reach the target by 2030 as per claimed by UN

Correlation

Correlational assessment is such a non-test research system in which an expert appraisals two factors, understands, and reviews the real association between them with no effect from any pointless variable.

Calculating the Correlation

Now we're ready to compute the correlation value. The formula for the correlation is:

$$r = \frac{N\sum xy - \sum x \sum y}{\sqrt{(N\sum x^2 - (\sum x)^2)(N\sum y^2 - (\sum y)^2)}}$$

where:

- N is the number of pairs of scores,
- Σxy is the sum of the products of paired scores,
- Σx is the sum of x scores,
- Σy is the sum of y scores,
- Σx^2 is the sum of squared x scores,
- Σy^2 is the sum of squared y scores.

EIGEN VECTOR:

A patented or trademark vector with a direct change is a nonzero vector in straight polynomial math that changes by a scalar factor as that direct change is added to it. The element by which the eigenvector is scaled is the related eigenvalue that [lambda] often shows.

Mathematically, an eigenvector depends on a direction along which the transition is expanded, corresponding to a genuine nonzero eigenvalue, and the eigenvalue is the element by which it is extended. The heading is turned around in the event that the eigenvalue is negative.

Power Iteration Method

Explanation of Power Method

First step: The user-provided square matrix A is multiplied by the original guess matrix X0 (i.e column matrix)

Second Step: The maximum value is then extracted from AX0 and X will be the matrix obtained after extraction.

Third step: Then X will be multiplied again with A and step 1 and step 2 will be repeated until after comparing the previous and present eigen value and the dominant eigen value of the two, the final eigen value will be equal to the previous and present eigen vector.

Result and Discussion

Objective 1: (Click to view all Objectives)

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import scipy
from collections import OrderedDict
import math
import numpy as np
import io
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
data=pd.read_csv('3 EG_EGY_PRIM.csv',index_col='S.No.')
data
```

Out[4]:

S.No.	GeoAreaName	2001	2017
1	Afghanistan	1.86	1.93
2	Albania	4.20	2.91
3	Algeria	3.45	4.05
4	Americas	6.37	4.79
5	Angola	4.44	3.41
...	...	...	...
201	Vanuatu	4.22	3.65
202	Venezuela (Bolivarian Republic of)	6.08	6.02
203	Yemen	3.05	2.04
204	Zambia	11.68	8.05

205	Zimbabwe	11.77	12.95

205 rows × 3 columns

In [5]:
```
#for calculating mean
def cal_mean(data,column):
  column=str(column)
  count=len(data[column])
  sum=0
  for index,row in data.iterrows():
      sum=sum+row[column]
  u=round(sum/count,2)
  return u
```

In [6]:
```
#for calculating standard deviation
def cal_dev(data,column):
  sum=0
  column=str(column)
  count=len(data[column])
  for index,row in data.iterrows():
      diff=round((row[column]-cal_mean(data,column))**2,2)
      sum=sum+diff
  sd=round(math.sqrt(sum/count),2)
  return sd
```


For 2001

UN Claim is 6.49

In [7]:
```
mean_1=cal_mean(data,2001)
dev_1=cal_dev(data,2001)
print('Mean: ',mean_1)
print('Standard Deviation: ',dev_1)
```

```
Mean:  6.86
Standard Deviation:  4.88
```

For 2017

UN Claim is 5.01

In [8]:
```
mean_2=cal_mean(data,2017)
dev_2=cal_dev(data,2017)
print('Mean: ',mean_2)
print('Standard Deviation: ',dev_2)
```

```
Mean:  5.14
Standard Deviation:  3.02
```

Test of Hypothesis concerning differences of means (Two Poulation)

In [9]:
```
#Null Hypothesis H0:
print("Null Hypothesis H0:\nμ1-μ2=δ")
print("Alternate Hypothesis H1:\nμ1-μ2≠δ")
```

```
Null Hypothesis H0:
```

μ1-μ2=δ
Alternate Hypothesis H1:
μ1-μ2≠δ

In [10]:
```
x1=mean_1
x2=mean_2
μ1=6.49
μ2=5.01
δ=round(μ1-μ2,2)
print("x1:",x1)
print("x2:",x2)
print("μ1:",μ1)
print("μ2:",μ2)
print("δ:",δ)
```

```
x2: 5.14
μ1: 6.49
μ2: 5.01
x1: 6.86
δ: 1.48
```

[11]:
```
def z_test(x1,x2,δ,dev1,dev2,n1,n2):
  a=x1-x2-δ
  b=((dev1)**2)/n1
  c=((dev2)**2)/n2
  d=math.sqrt(b+c)
  z=a/d
  if -1.69<z<1.69:
    print("Hypothesis Accepted")
  else:
    print("Hypothesis Rejected")
  return z
```

In [12]:
```
z_test(x1,x2,δ,dev_1,dev_2,len(data['2001']),len(data['2017']))
```

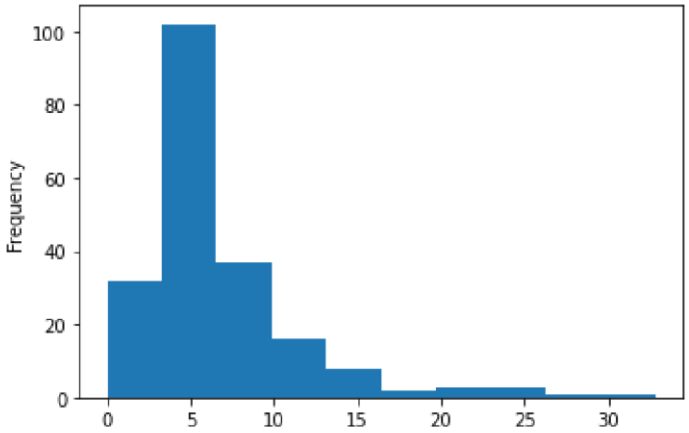
```
Hypothesis Accepted
```

Out[12]:
```
0.5987708604736225
```

Graphs for Year 2001

In [13]:
```
data['2001'].plot.hist(bins=10)
```

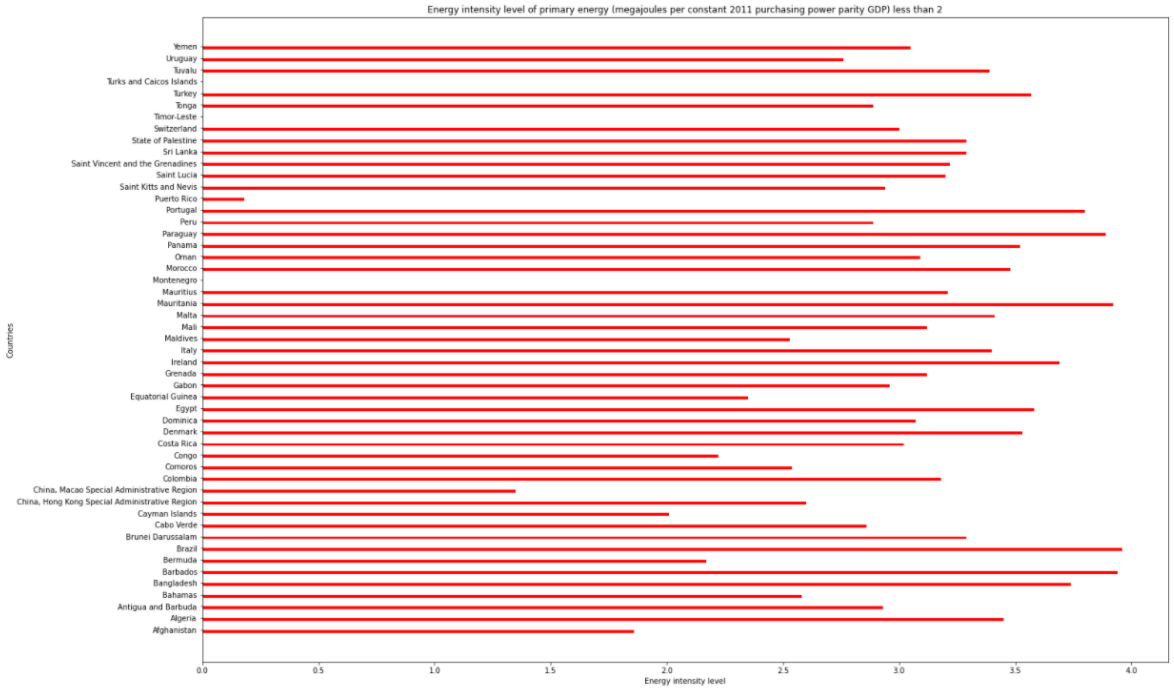
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x29bad2e79a0>



In [14]:
```
el=data.loc[(data['2001']<4)].drop(columns=['2017'])
el3=data.loc[(data['2001']>=4)&(data['2001']<6)].drop(columns=['2017'])
el4=data.loc[(data['2001']>=6)&(data['2001']<8)].drop(columns=['2017'])
el5=data.loc[(data['2001']>=8)&(data['2001']<15)].drop(columns=['2017'])
el6=data.loc[(data['2001']>=15)].drop(columns=['2017'])
```

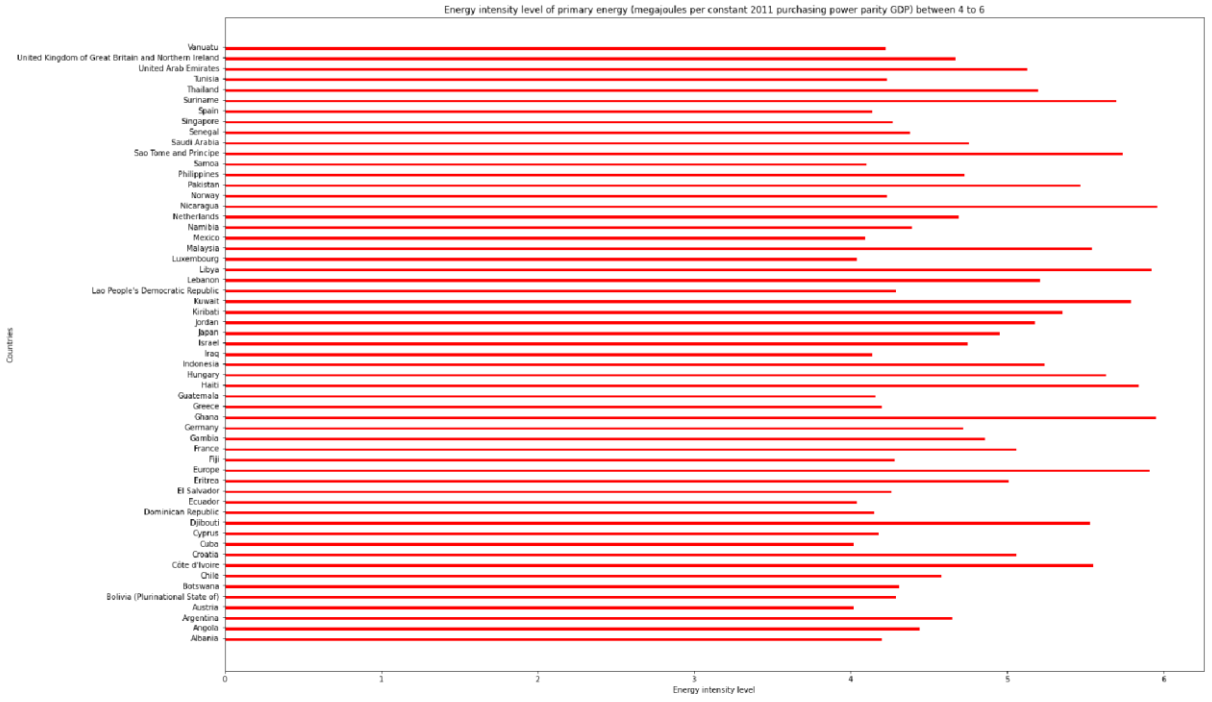
In [15]:
```
x = el['GeoAreaName']
y = el['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purchasing power parity GDP) less than 2')
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()
```

Out[15]:
```
Text(0, 0.5, 'Countries')
```



[17]:
```
x = el4['GeoAreaName']
y = el4['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()AC
```

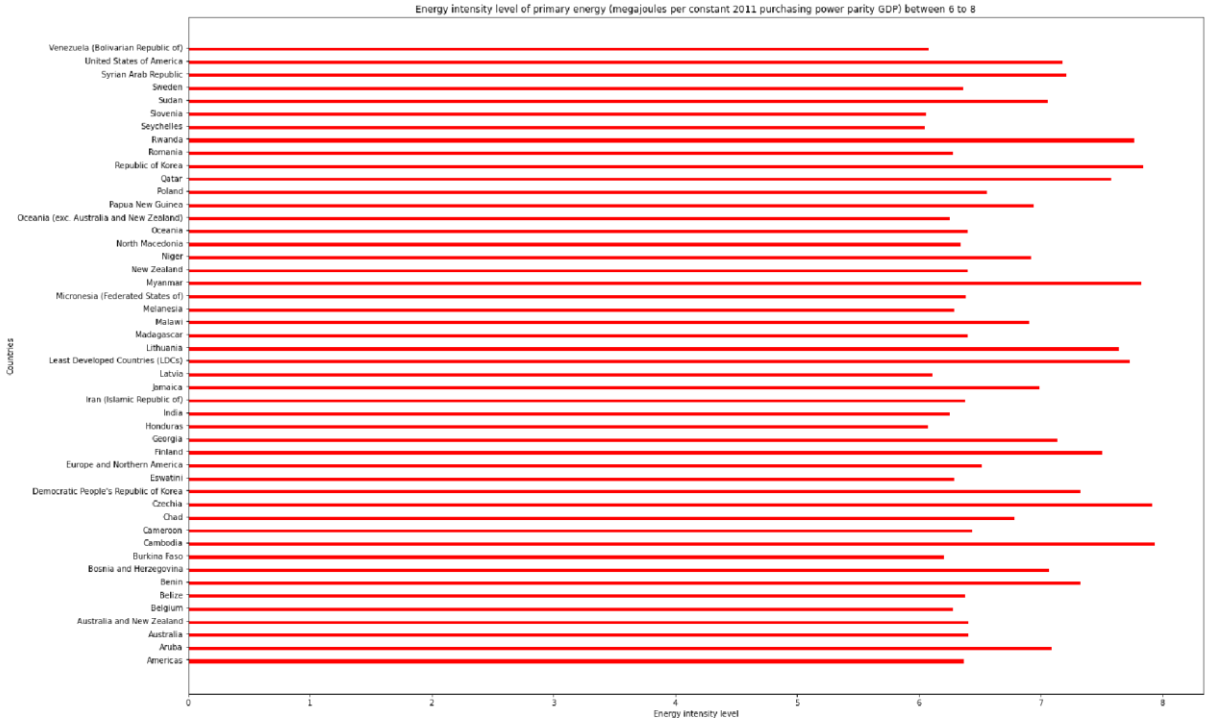
Out[17]: Text(0, 0.5, 'Countries')



In [18]:

```
x = el5['GeoAreaName']
y = el5['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()
```

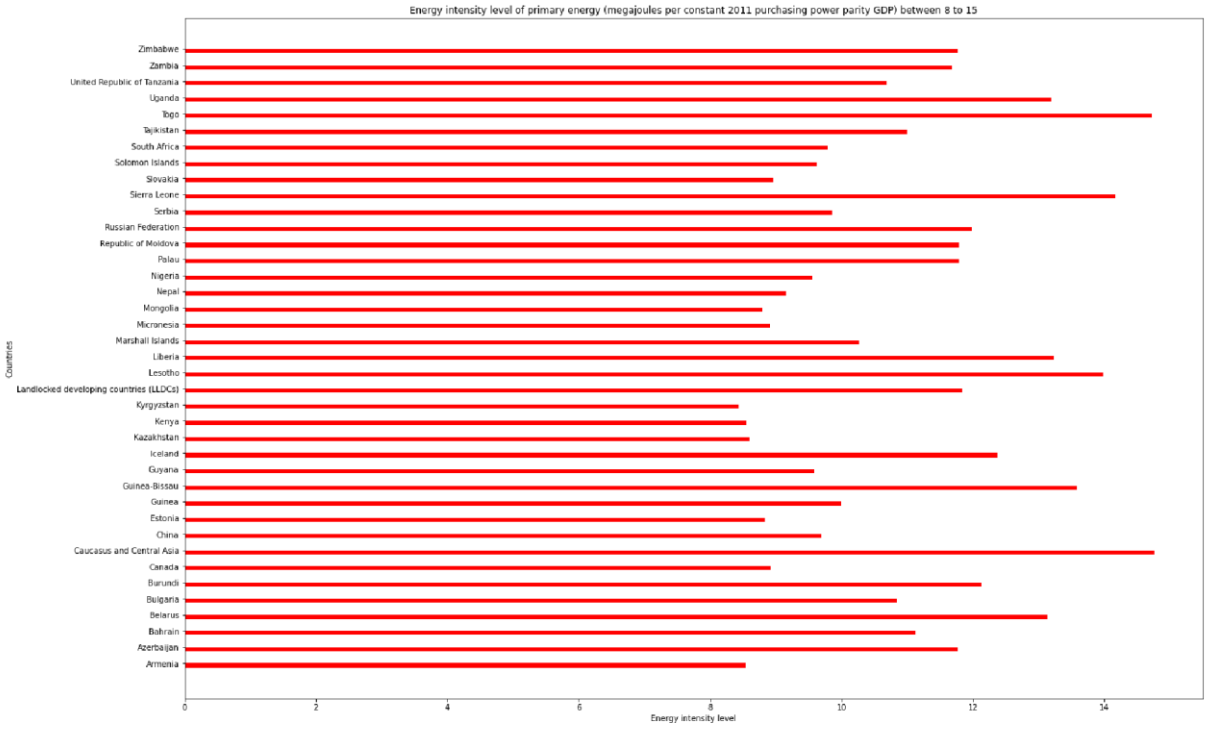
Out[18]: Text(0, 0.5, 'Countries')



In [19]:

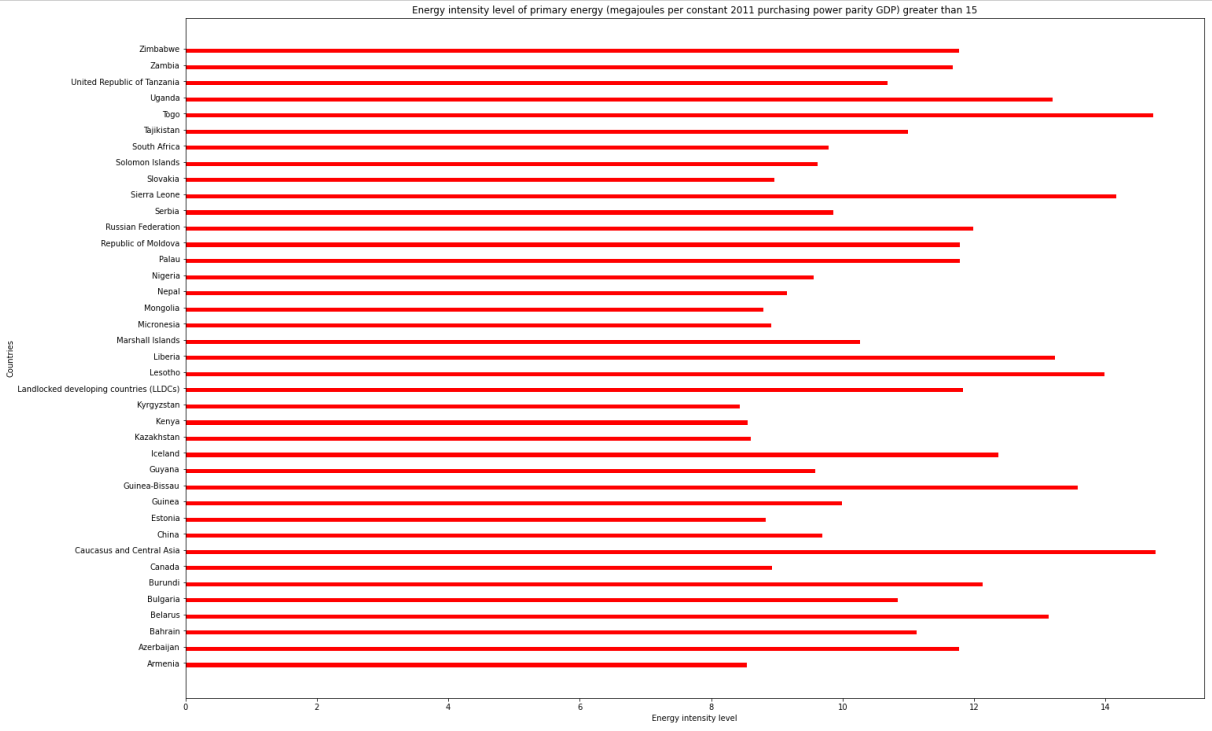
```
x = el5['GeoAreaName']
y = el5['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()
```

Out[19]: Text(0, 0.5, 'Countries')

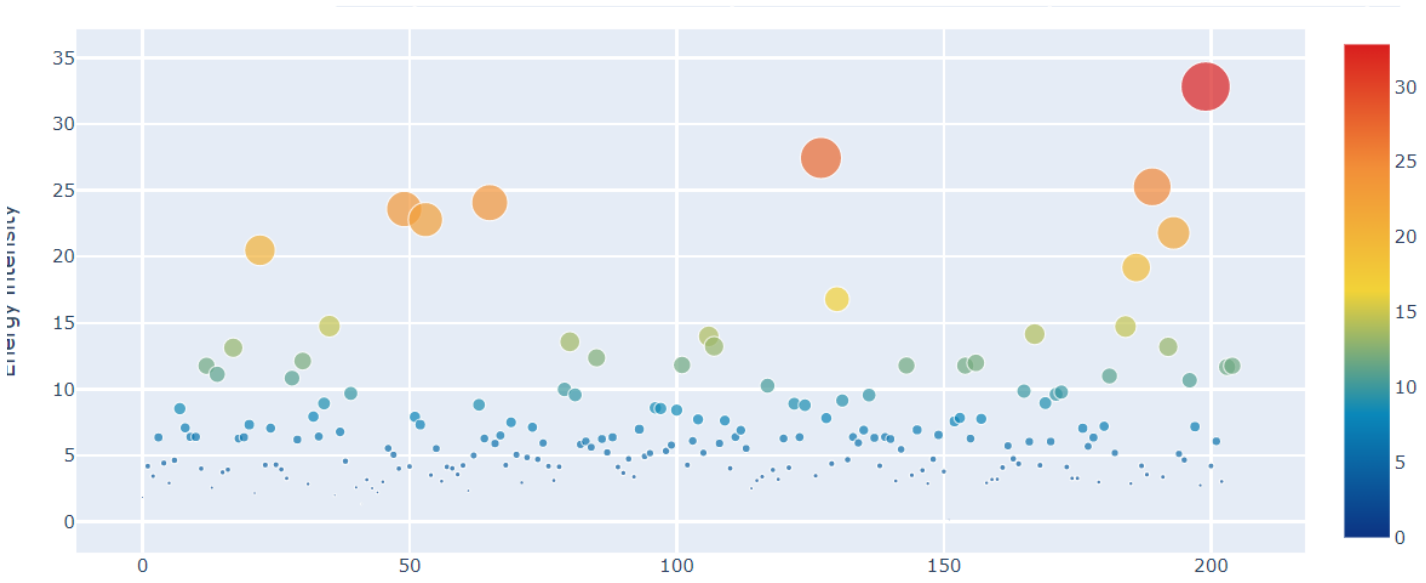


In [20]:

```
l=[]
trace0= go.Scatter(
        y= data['2001'],
        mode= 'markers',
        name='Energy intensity',
        marker= dict(size= data['2001'].values,
                     line= dict(width=1),
                     color= data['2001'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of Energy intensity level of primary energy in 2001',
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Energy intensity',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```



In

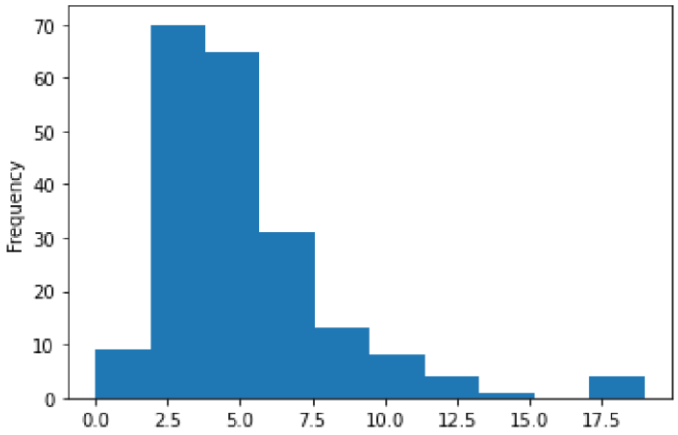

Scatter plot of Energy intensity level of primary energy in 2001
Energy intensity
0
5
10
15
20
25
30
35
0
50
100
150
200
0
5
10
15
20
25
30

In

Graphs For Year 2017

In [21]:
```
data['2017'].plot.hist(bins=10)
```

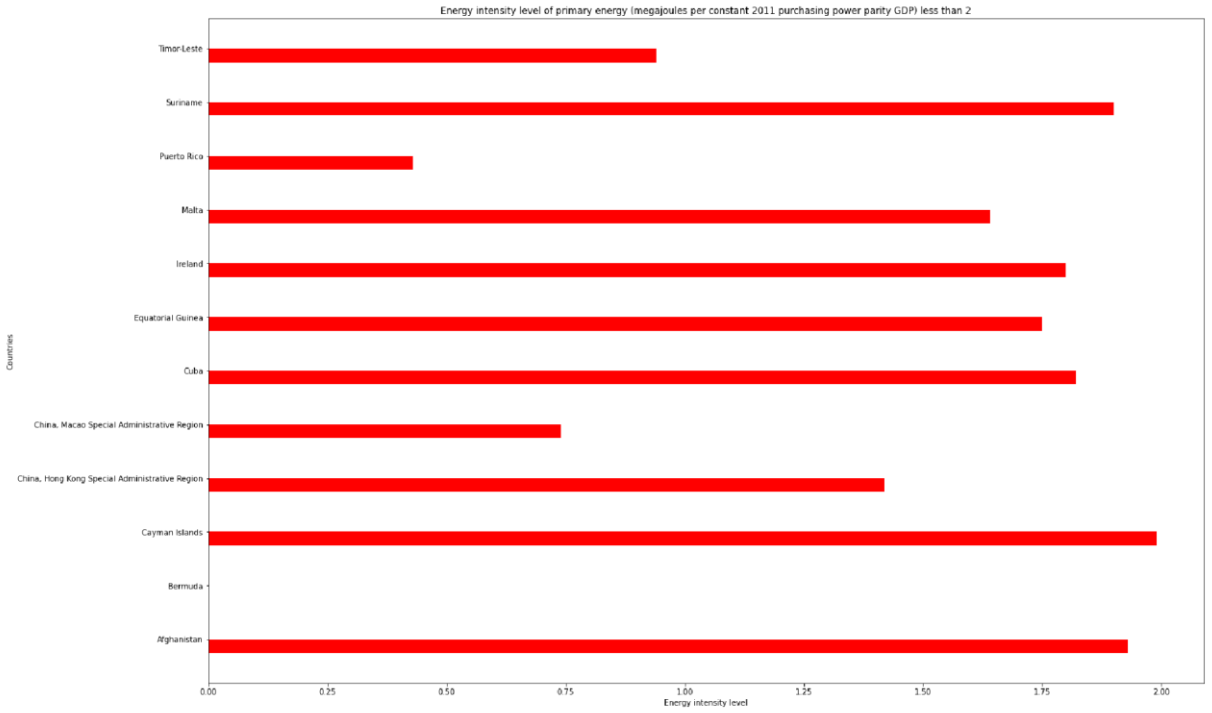
Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x29baef9f9a0>



In [22]:
```
em=data.loc[(data['2017']<2)].drop(columns=['2001'])
em2=data.loc[(data['2017']>=2)&(data['2017']<4)].drop(columns=['2001'])
em3=data.loc[(data['2017']>=4)&(data['2017']<6)].drop(columns=['2001'])
em4=data.loc[(data['2017']>=6)&(data['2017']<8)].drop(columns=['2001'])
em5=data.loc[(data['2017']>=8)&(data['2017']<15)].drop(columns=['2001'])
em6=data.loc[(data['2017']>=15)].drop(columns=['2001'])
```

In [23]:
```
x = em['GeoAreaName']
y = em['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()
```

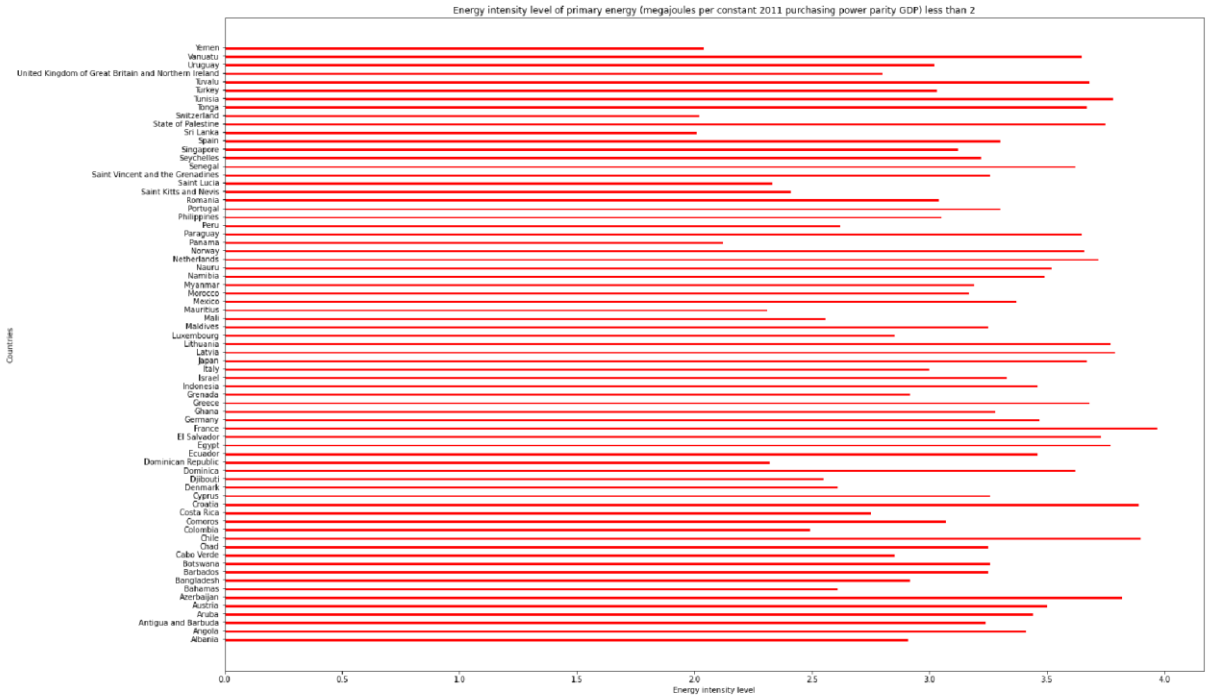
Out[23]: Text(0, 0.5, 'Countries')



In [24]:
```
x = em2['GeoAreaName']
y = em2['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()
```

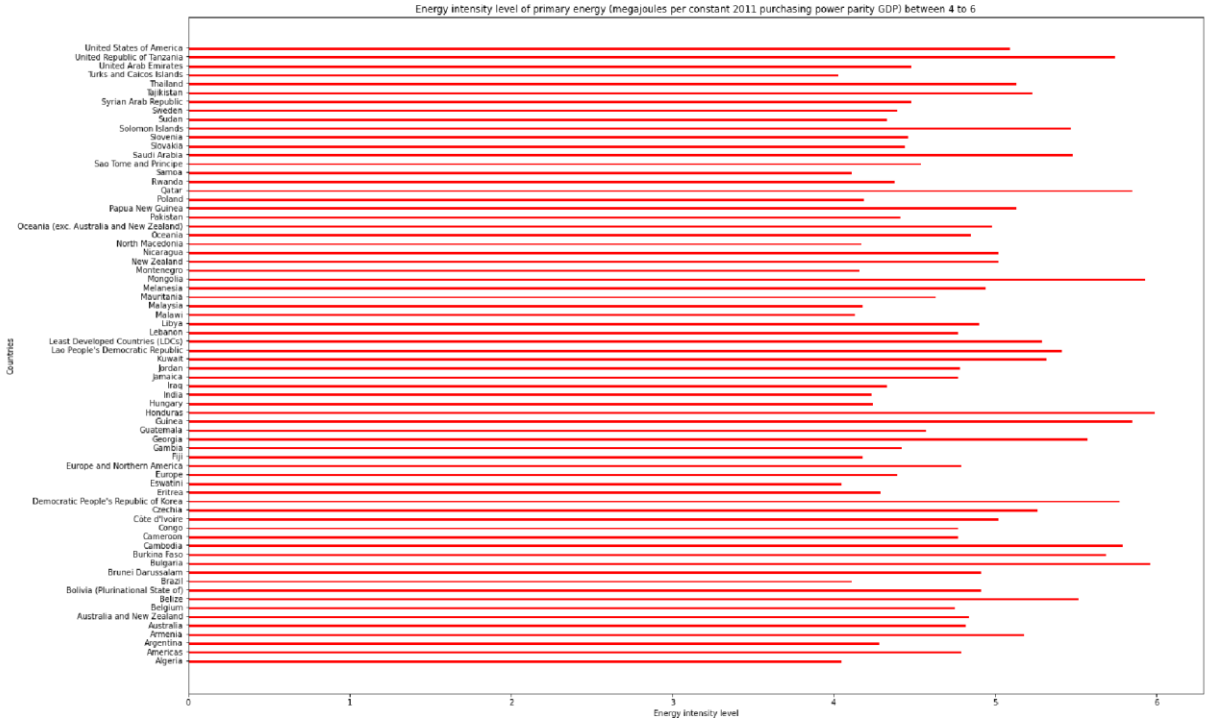
Out[24]: Text(0, 0.5, 'Countries')

In



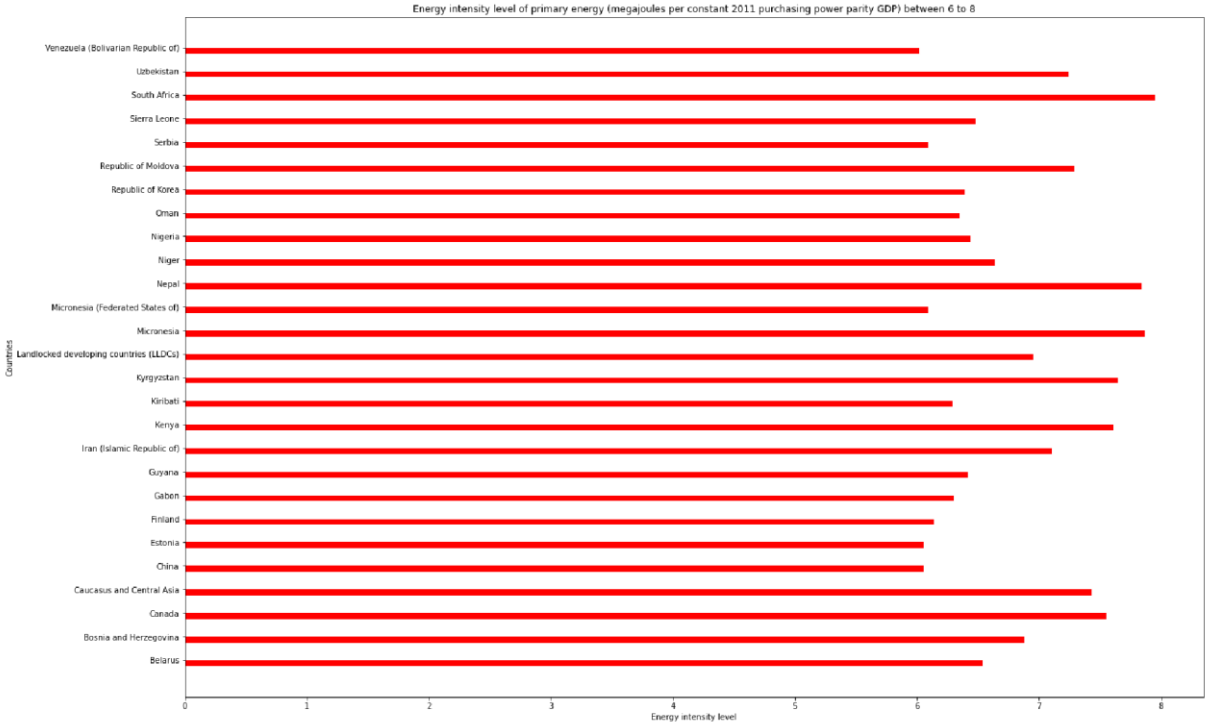
[25]:
```
x = em3['GeoAreaName']
y = em3['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()
```

Out[25]: Text(0, 0.5, 'Countries')



In [26]:
```
x = em4['GeoAreaName']
y = em4['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()AC
```

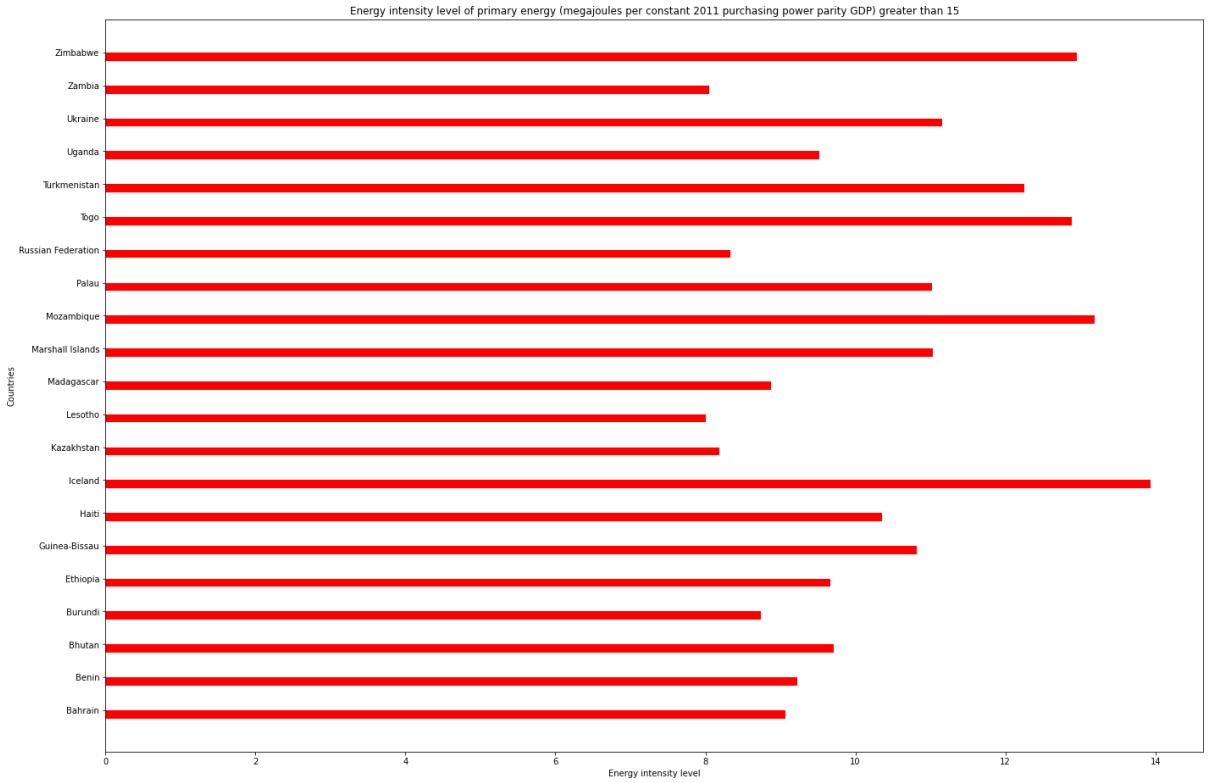
Out[26]: Text(0, 0.5, 'Countries')



[27]:
```
x = em5['GeoAreaName']
y = em5['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Energy intensity level of primary energy (megajoules per constant 2011 purcha
plt.xlabel('Energy intensity level')
plt.ylabel('Countries')
#plt.show()
```

Out[27]: Text(0, 0.5, 'Countries')

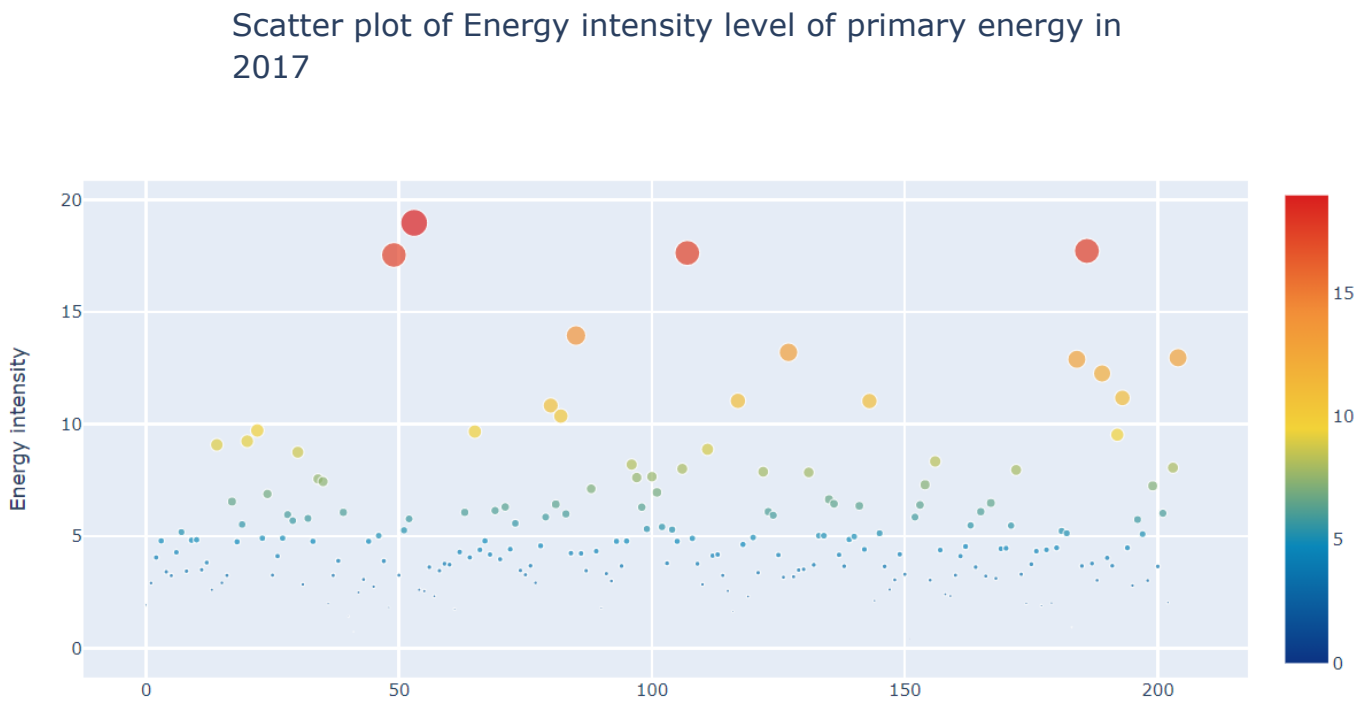
In



[28]:

```
l=[]
trace0= go.Scatter(
        y= data['2017'],
        mode= 'markers',
        name='Energy intensity',
        marker= dict(size= data['2017'].values,
                     line= dict(width=1),
                     color= data['2017'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of Energy intensity level of primary energy in 2017',
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Energy intensity',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```

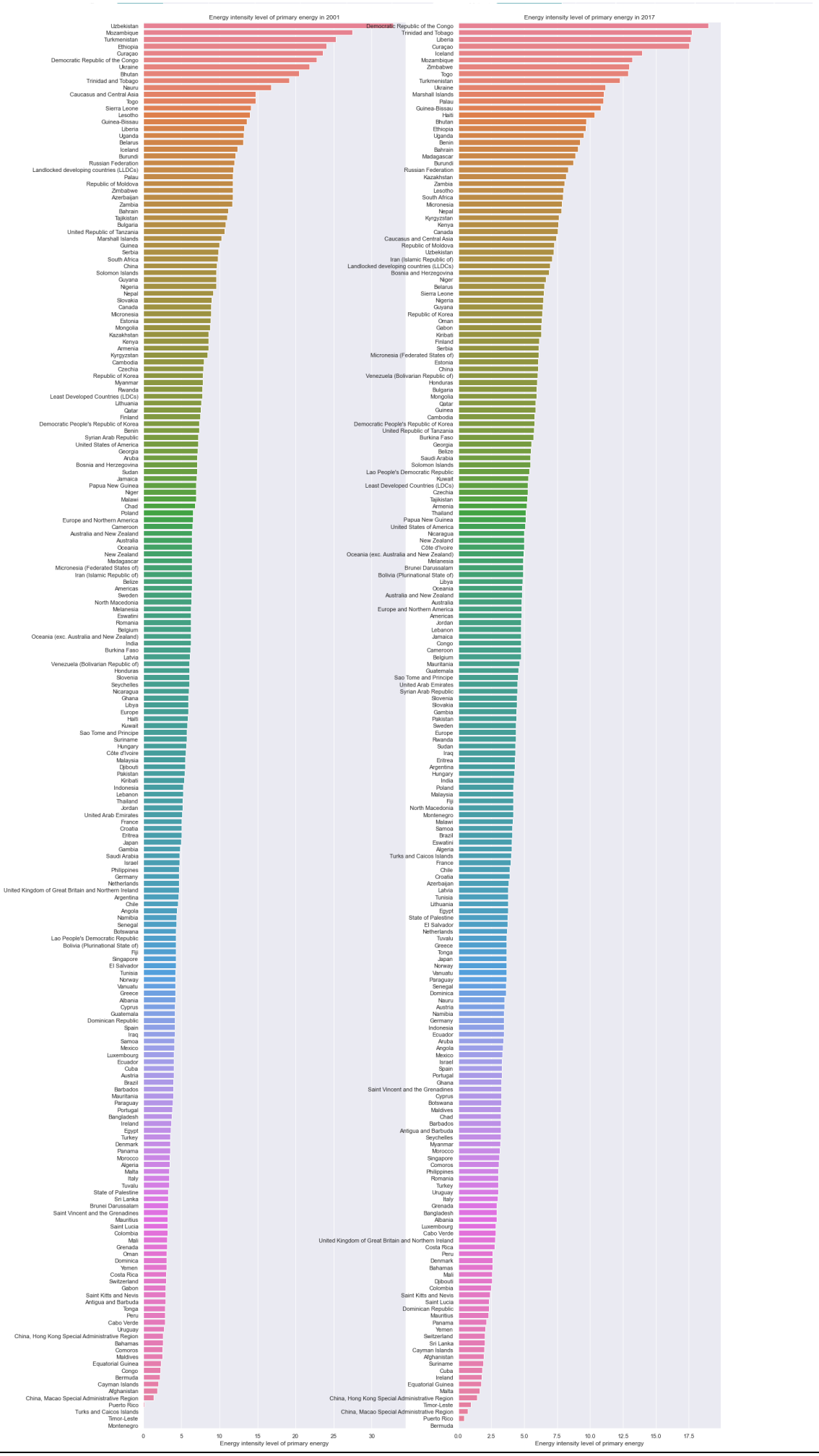
In


Scatter plot of Energy intensity level of primary energy in 2017
Energy intensity
0
5
10
15
20
0
50
100
150
200
0
5
10
15

In [29]:

```
# Tried with Plotly now going with seaborn
twoyearchange1_bar, countries_bar1 = (list(x) for x in zip(*sorted(zip(data['2001'], dat
                                                                   reverse = True)))

twoyearchange2_bar, countries_bar2 = (list(x) for x in zip(*sorted(zip(data['2017'], dat
                                                                   reverse = True)))

# Another direct way of sorting according to values is creating distinct sorted datafram
# passing their values directly as in below mentioned code to achieve the same effect as

# df_country_sorted=df_country.sort(columns='2014-2012 change',ascending=False)
# df_country_sorted.head()


sns.set(font_scale=1)
fig, axes = plt.subplots(1,2,figsize=(20, 50))
colorspal = sns.color_palette('husl', len(data['2001']))
sns.barplot(twoyearchange1_bar, countries_bar1, palette = colorspal,ax=axes[0])
sns.barplot(twoyearchange2_bar, countries_bar2, palette = colorspal,ax=axes[1])
axes[0].set(xlabel='Energy intensity level of primary energy', title='Energy intensity l
axes[1].set(xlabel='Energy intensity level of primary energy', title='Energy intensity l
fig.savefig('output.png')
```

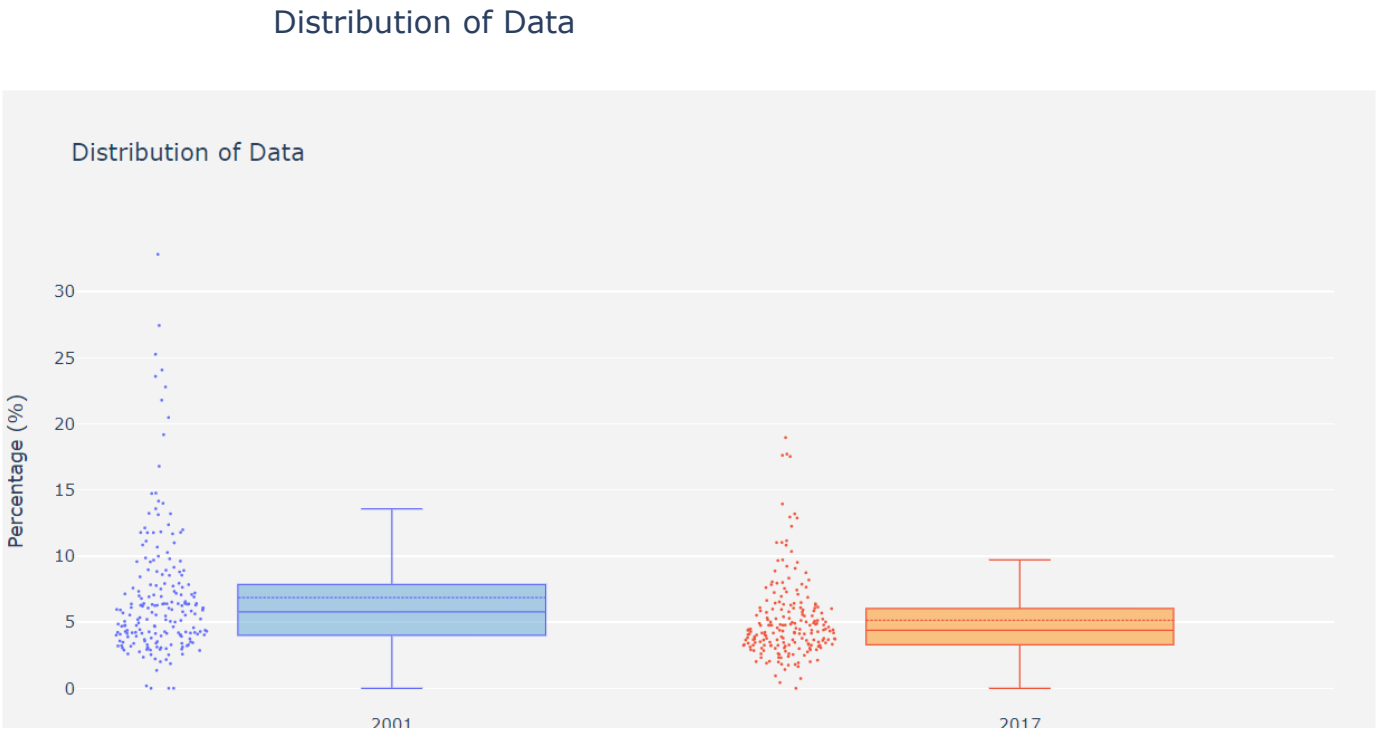
In


Energy intensity level of primary energy in 2001
Uzbekistan
Mozambique
Turkmenistan
Ethiopia
Curaçao
Democratic Republic of the Congo
Ukraine
Bhutan
Trinidad and Tobago
Nauru
Caucasus and Central Asia
Togo
Sierra Leone
Lesotho
Guinea-Bissau
Liberia
Uganda
Belarus
Iceland
Burundi
Russian Federation
Landlocked developing countries (LLDCs)
Palau
Republic of Moldova
Zimbabwe
Azerbaijan
Zambia
Bahrain
Tajikistan
Bulgaria
United Republic of Tanzania
Marshall Islands
Guinea
Serbia
South Africa
China
Solomon Islands
Guyana
Nigeria
Nepal
Slovakia
Canada
Micronesia
Estonia
Mongolia
Kazakhstan
Kenya
Armenia
Kyrgyzstan
Cambodia
Czechia
Republic of Korea
Myanmar
Rwanda
Least Developed Countries (LDCs)
Lithuania
Qatar
Finland
Democratic People's Republic of Korea
Benin
Syrian Arab Republic
United States of America
Georgia
Aruba
Bosnia and Herzegovina
Sudan
Jamaica
Papua New Guinea
Niger
Malawi
Chad
Poland
Europe and Northern America
Cameroon
Australia and New Zealand
Australia
Oceania
New Zealand
Madagascar
Micronesia (Federated States of)
Iran (Islamic Republic of)
Belize
Americas
Sweden
North Macedonia
Melanesia
Eswatini
Romania
Belgium
Oceania (exc. Australia and New Zealand)
India
Burkina Faso
Latvia
Venezuela (Bolivarian Republic of)
Honduras
Slovenia
Seychelles
Nicaragua
Ghana
Libya
Europe
Haiti
Kuwait
Sao Tome and Principe
Suriname
Hungary
Côte d'Ivoire
Malaysia
Djibouti
Pakistan
Kiribati
Indonesia
Lebanon
Thailand
Jordan
United Arab Emirates
France
Croatia
Eritrea
Japan
Gambia
Saudi Arabia
Israel
Philippines
Germany
Netherlands
United Kingdom of Great Britain and Northern Ireland
Argentina
Chile
Angola
Namibia
Senegal
Botswana
Lao People's Democratic Republic
Bolivia (Plurinational State of)
Fiji
Singapore
El Salvador
Tunisia
Norway
Vanuatu
Greece
Albania
Cyprus
Guatemala
Dominican Republic
Spain
Iraq
Samoa
Mexico
Luxembourg
Ecuador
Cuba
Austria
Brazil
Barbados
Mauritania
Paraguay
Portugal
Bangladesh
Ireland
Egypt
Turkey
Denmark
Panama
Morocco
Algeria
Malta
Italy
Tuvalu
State of Palestine
Sri Lanka
Brunei Darussalam
Saint Vincent and the Grenadines
Mauritius
Saint Lucia
Colombia
Mali
Grenada
Oman
Dominica
Yemen
Costa Rica
Switzerland
Gabon
Saint Kitts and Nevis
Antigua and Barbuda
Tonga
Peru
Cabo Verde
Uruguay
China, Hong Kong Special Administrative Region
Bahamas
Comoros
Maldives
Equatorial Guinea
Congo
Bermuda
Cayman Islands
Afghanistan
China, Macao Special Administrative Region
Puerto Rico
Turks and Caicos Islands
Timor-Leste
Montenegro
0
5
10
15
20
25
30
Energy intensity level of primary energy
Energy intensity level of primary energy in 2017
Democratic Republic of the Congo
Trinidad and Tobago
Liberia
Curaçao
Iceland
Mozambique
Zimbabwe
Togo
Turkmenistan
Ukraine
Marshall Islands
Palau
Guinea-Bissau
Haiti
Bhutan
Ethiopia
Uganda
Benin
Bahrain
Madagascar
Burundi
Russian Federation
Kazakhstan
Zambia
Lesotho
South Africa
Micronesia
Nepal
Kyrgyzstan
Kenya
Canada
Caucasus and Central Asia
Republic of Moldova
Uzbekistan
Iran (Islamic Republic of)
Landlocked developing countries (LLDCs)
Bosnia and Herzegovina
Niger
Belarus
Sierra Leone
Nigeria
Guyana
Republic of Korea
Oman
Gabon
Kiribati
Finland
Serbia
Micronesia (Federated States of)
Estonia
China
Venezuela (Bolivarian Republic of)
Honduras
Bulgaria
Mongolia
Qatar
Guinea
Cambodia
Democratic People's Republic of Korea
United Republic of Tanzania
Burkina Faso
Georgia
Belize
Saudi Arabia
Solomon Islands
Lao People's Democratic Republic
Kuwait
Least Developed Countries (LDCs)
Czechia
Tajikistan
Armenia
Thailand
Papua New Guinea
United States of America
Nicaragua
New Zealand
Côte d'Ivoire
Oceania (exc. Australia and New Zealand)
Melanesia
Brunei Darussalam
Bolivia (Plurinational State of)
Libya
Oceania
Australia and New Zealand
Australia
Europe and Northern America
Americas
Jordan
Lebanon
Jamaica
Congo
Cameroon
Belgium
Mauritania
Guatemala
Sao Tome and Principe
United Arab Emirates
Syrian Arab Republic
Slovenia
Slovakia
Gambia
Pakistan
Sweden
Europe
Rwanda
Sudan
Iraq
Eritrea
Argentina
Hungary
India
Poland
Malaysia
Fiji
North Macedonia
Montenegro
Malawi
Samoa
Brazil
Eswatini
Algeria
Turks and Caicos Islands
France
Chile
Croatia
Azerbaijan
Latvia
Tunisia
Lithuania
Egypt
State of Palestine
El Salvador
Netherlands
Tuvalu
Greece
Tonga
Japan
Norway
Vanuatu
Paraguay
Senegal
Dominica
Nauru
Austria
Namibia
Germany
Indonesia
Ecuador
Aruba
Angola
Mexico
Israel
Spain
Portugal
Ghana
Saint Vincent and the Grenadines
Cyprus
Botswana
Maldives
Chad
Barbados
Antigua and Barbuda
Seychelles
Myanmar
Morocco
Singapore
Comoros
Philippines
Romania
Turkey
Uruguay
Italy
Grenada
Bangladesh
Albania
Luxembourg
Cabo Verde
United Kingdom of Great Britain and Northern Ireland
Costa Rica
Peru
Denmark
Bahamas
Mali
Djibouti
Colombia
Saint Kitts and Nevis
Saint Lucia
Dominican Republic
Mauritius
Panama
Yemen
Switzerland
Sri Lanka
Cayman Islands
Afghanistan
Suriname
Cuba
Ireland
Equatorial Guinea
Malta
China, Hong Kong Special Administrative Region
Timor-Leste
China, Macao Special Administrative Region
Puerto Rico
Bermuda
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Energy intensity level of primary energy

In [30]:

```
x_data = ['2001','2017']

y0 = data['2001']
y1 = data['2017']


y_data = [y0,y1]

colors = ['rgba(93, 164, 214, 0.5)', 'rgba(255, 144, 14, 0.5)']

traces = []

for xd, yd, color in zip(x_data, y_data, colors):
        traces.append(go.Box(
            y=yd,
            name=xd,
            boxpoints='all',
            whiskerwidth=0.2,
            fillcolor=color,
            marker=dict(
                size=2,
            ),
            boxmean=True,
            line=dict(width=1),
        ))

layout = go.Layout(
    title='Distribution of Data',
    xaxis=dict(
        title='Year'
    ),
    yaxis=dict(
        title='Percentage (%)',
        autorange=True,
        showgrid=True,
        zeroline=False,
        dtick=5,
        gridcolor='rgb(255, 255, 255)',
        gridwidth=1,
#         zerolinecolor='rgb(255, 255, 255)',
#         zerolinewidth=2,
    ),
    margin=dict(
        l=40,
        r=30,
        b=80,
        t=100,
    ),
    paper_bgcolor='rgb(243, 243, 243)',
    plot_bgcolor='rgb(243, 243, 243)',
    showlegend=False
)

fig = go.Figure(data=traces, layout=layout)
py.iplot(fig)
```

In


Distribution of Data
Distribution of Data
Percentage (%)
30
25
20
15
10
5
0
2001
2017

Regression Equation

In [1]:

```
import pandas as pd
data = pd.read_csv('data3.csv')
data.index=range(1,242)
data
```

Out[1]:

	Countries	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
1	Afghanistan	1.72	1.86	1.38	1.35	1.15	1.28	1.43	1.48	2.17	2.63	2.79
2	Africa	7.07	6.98	6.73	6.73	6.71	6.55	6.38	6.23	6.17	6.07	5.93
3	Albania	4.47	4.20	4.43	4.11	4.28	4.06	3.74	3.37	3.25	3.22	3.08
4	Algeria	3.55	3.45	3.50	3.47	3.37	3.31	3.49	3.58	3.54	3.81	3.61
5	Americas	6.51	6.37	6.36	6.27	6.16	6.00	5.81	5.74	5.60	5.51	5.48
...	...	...	...	...	...	...	...	...	...	...	...	...
237	Western Europe	4.72	4.78	4.70	4.77	4.72	4.65	4.53	4.26	4.25	4.22	4.29
238	World	6.59	6.49	6.44	6.43	6.38	6.26	6.12	5.97	5.87	5.85	5.87
239	Yemen	2.86	3.05	2.84	3.05	3.20	3.22	3.35	3.17	3.22	3.44	3.09
240	Zambia	11.88	11.68	11.60	11.26	10.88	10.53	10.04	9.33	9.00	8.49	8.04
241	Zimbabwe	11.97	11.77	12.69	14.62	15.17	16.88	18.02	18.47	21.03	19.37	16.56

241 rows × 19 columns

In [2]:

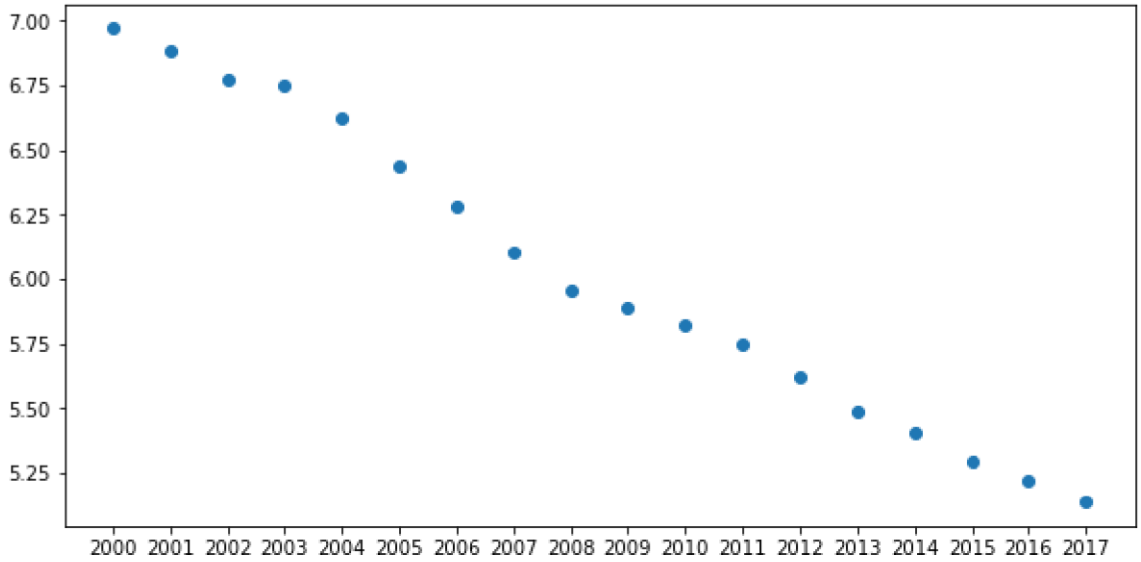
```
mean_year=pd.Series(data[1:].mean())
mean_year
```

Out[2]:

```
2000    6.970802
2001    6.886371
2002    6.776513
2003    6.748319
2004    6.623613
2005    6.436778
2006    6.279916
2007    6.105063
2008    5.952887
2009    5.886067
2010    5.818619
2011    5.745272
2012    5.619833
2013    5.485083
2014    5.406833
2015    5.294792
```

2016 5.218750 2017 5.135607 dtype: float64 In [14]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,5))
plt.scatter(mean_year.index,mean_year)
plt.show()
```



In [4]:

```
table=pd.DataFrame(mean_year)
table.columns=['Mean (Y)']
```

In [5]:

```
table [ 'X' ] = range(  1 , 19 )
table [ 'XY' ] = table[  'Mean (Y)' ] *
table [ 'X' ]  table [ 'X^2' ] = table [
'X' ] ** 2   table = pd . DataFrame (
table)
```

In [6]:

```
sum=pd.DataFrame(table.sum())
sum.columns=['Sigma']
sum

mean=pd.DataFrame(table.mean())
mean.columns=['Mean']
mean
```

Out[6]:

	Mean
Mean (Y)	6.021729
X	9.500000
XY	54.156532
X^2	117.166667

In [7]:

```
table=pd.concat([table,sum.T,mean.T])
```

table

Out[7]:

	Mean (Y)	X	XY	X^2
2000	6.970802	1.0	6.970802	1.000000
2001	6.886371	2.0	13.772743	4.000000
2002	6.776513	3.0	20.329538	9.000000
2003	6.748319	4.0	26.993277	16.000000
2004	6.623613	5.0	33.118067	25.000000
2005	6.436778	6.0	38.620669	36.000000
2006	6.279916	7.0	43.959414	49.000000
2007	6.105063	8.0	48.840502	64.000000
2008	5.952887	9.0	53.575983	81.000000
2009	5.886067	10.0	58.860669	100.000000
2010	5.818619	11.0	64.004812	121.000000
2011	5.745272	12.0	68.943264	144.000000
2012	5.619833	13.0	73.057833	169.000000
2013	5.485083	14.0	76.791167	196.000000
2014	5.406833	15.0	81.102500	225.000000
2015	5.294792	16.0	84.716667	256.000000
2016	5.218750	17.0	88.718750	289.000000
2017	5.135607	18.0	92.440921	324.000000
Sigma	108.391119	171.0	974.817578	2109.000000
Mean	6.021729	9.5	54.156532	117.166667

In [8]:

```
numerator = table ['XY'  ] . loc [ 'Sigma' ] - table [ 'X' ].  loc [ 'Sigma' ] * table
['Mean (Y)'  ] . loc [ 'Sigm a' ] / 20   numerator
```

Out[8]:

48.07350808298247

In [9]:

```
denominator = table [ 'X^2']  . loc [ 'Sigma' ]-  table[  'X' ] . loc [ 'Sigma'
] ** 2 /20   denominator
```

Out[9]:

646.95

In [10]:

```
b=numerator/denominator
b
```

Out[10]:

0.07430791882368416

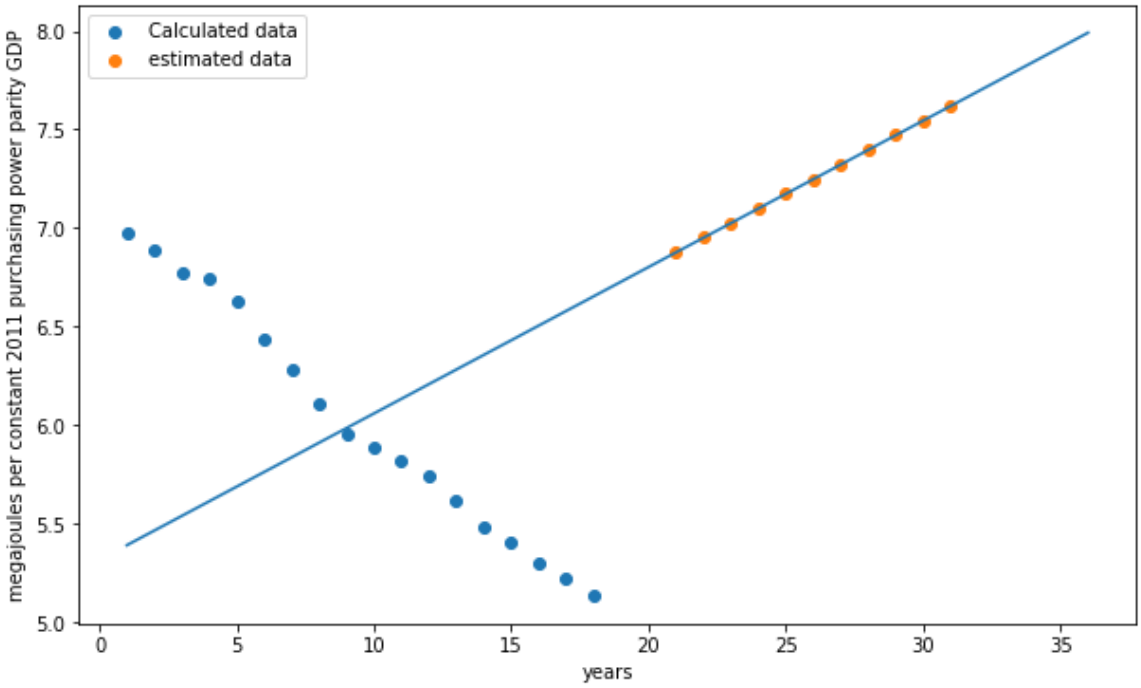
In [11]:

```
def  eq_regression ( x ):
     y = table[  'Mean (Y)' ] . loc [ 'Mean']  + b*  (x  - table[  'X' ] .
 loc [ 'Mean' ])      return ( y )
```

In [19]:

```
plt.figure(figsize=(10,6))
x_values=[]
y_values=[]
for x in range(21,32):
    x_values.append(x)
    y_values.append(eq_regression(x))

print("estimated megajoules per constant 2011 purchasing power parity GDP next 11 year
s",y_values)

aabbccdd=[]
for i in mean_year.index:
    aabbccdd.append(int(i)-1999)
plt.scatter(aabbccdd,mean_year,label='Calculated data')
plt.scatter([range(21,32)],y_values,label='estimated data')
plt.legend()

y_line=[]
x_line=[]
for x in range(1,37):
    x_line.append(x)
    y_line.append(eq_regression(x))
plt.plot(x_line,y_line)
plt.xlabel("years")
plt.ylabel("megajoules per constant 2011 purchasing power parity GDP")
plt.show()
```

```
estimated for next 11 years [6.876269913451225, 6.950577832274909,
7.024885751098593, 7.09919366
9922277, 7.173501588745962, 7.247809507569646, 7.32211742639333, 7.3964253
45217015, 7.470733264040699, 7.545041182864383, 7.619349101688067]
```



Progress Report of Different Regions

In [19]:
```
#reading data
import pandas as pd
df=pd.read_csv('primary_energy_intensity.csv')
df.index=range(1,7)
df
```

Out[19]:

	Continent	2017
1	Africa	5.65
2	Asia	5.03
3	Australia	4.82
4	Europe	4.39
5	Northern America	5.30
6	South America	3.93

In [20]:
```
#converting data into required matrix
import pandas as pd
import numpy as np
en_rank = pd.read_csv("Primary_intensity_ranking.csv")
df = pd.DataFrame(en_rank)
df.index=range(1,7)
df
```

Out[20]:

	Unnamed: 0	Africa	Asia	Australia	Europe	Northen America	South America
1	Africa	0	2	2	2	2	2
2	Asia	1	0	2	2	1	2
3	Australia	1	1	0	2	1	2
4	Europe	1	1	1	0	1	2
5	Northern America	1	2	2	2	0	2
6	South America	1	1	1	1	1	0

In [21]:
```
#converting data frame into matrix
a=df.iloc[:,1:].to_numpy()
print("Matrix:")
print(a)
```

```
Matrix:
[[0 2 2 2 2 2]
 [1 0 2 2 1 2]
 [1 1 0 2 1 2]
 [1 1 1 0 1 2]
 [1 2 2 2 0 2]
 [1 1 1 1 1 0]]
```

In [27]:
```
#calculating eigen value and eigen vector
import numpy as np
x0=np.array([[1],[1],[1],[1],[1],[1]])
j=1
#first iteration
x1=np.matmul(a,x0)
previous_eigen_value=np.round(max([abs(i) for i in x1]),4)
x1=x1/previous_eigen_value
print("Step",j,":",'\n',x1,'\n',"Eigen Value : ",previous_eigen_value)
j+=1
#second iteration
x3=np.matmul(a,x1)
present_eigen_value=np.round(max([abs(i) for i in x3]),4)
x3=np.round(x3/present_eigen_value,4)
print("Step",j,":",'\n',x3,'\n',"Eigen Value : ",present_eigen_value)
#while loop
while(previous_eigen_value!=present_eigen_value):#will calculate until previous and present eigen value are same
    j+=1
    previous_eigen_value=present_eigen_value
    x1=np.matmul(a,x3)
    present_eigen_value=np.round(max([abs(i) for i in x1]),4)
    x1=x1/present_eigen_value
    print("Step",j,"\n",x1,'\n',"Eigen Value : ",present_eigen_value)
    x3=x1
```

```
Step 1 :
 [[1. ]
 [0.8]
 [0.7]
 [0.6]
 [0.9]
 [0.5]]
 Eigen Value :  [10]
Step 2 :
 [[1.    ]
 [0.7857]
 [0.7   ]
 [0.6286]
 [0.8857]
 [0.5714]]
 Eigen Value :  [7.]
Step 3
 [[1.        ]
 [0.79600437]
 [0.71000168]
 [0.63199306]
 [0.89200314]
 [0.56000448]]
 Eigen Value :  [7.1428]
Step 4
 [[1.00000187]
 [0.79331498]
 [0.70640704]
 [0.62925044]
 [0.8908088 ]
 [0.56128165]]
 Eigen Value :  [7.18]
Step 5
 [[1.00000361]
 [0.79371817]
 [0.70722132]
 [0.63013585]
 [0.89087141]
 [0.56125761]]
 Eigen Value :  [7.1621]
Step 6
 [[1.00000122]
 [0.79371855]
 [0.70710261]
 [0.62992991]
 [0.89091727]
 [0.56122326]]
 Eigen Value :  [7.1664]
Step 7
 [[0.99999765]
 [0.79369087]
 [0.70710086]
 [0.62996262]
 [0.89089144]
 [0.56123106]]
 Eigen Value :  [7.1658]
Step 8
 [[0.99999354]
 [0.79369759]
 [0.70710421]
 [0.62995659]
 [0.89089404]
 [0.56122742]]
 Eigen Value :  [7.1658]
```

Ranking is determined on the basis of the elements present in the most dominant Eigen Vector

1. Africa (0.999)
2. North America (0.890)
3. Asia (0.794)
4. Australia (0.707)
5. Europe (0.630)
6. South America (0.561)

Objective 2: (Click to view all Objectives)

In [52]:
```
import pandas as pd
import matplotlib.pyplot as plt
import scipy
from collections import OrderedDict
import math
import numpy as np
import io
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [53]:
```
data=pd.read_csv('2 EG_FEC_RNEW.csv',index_col='S.No.')
data
```

Out[53]:

S.No.	GeoAreaName	2001	2017
1	Afghanistan	54.06	24.65
2	Albania	39.13	37.19
3	Algeria	0.43	0.14
4	American Samoa	0.00	1.75
5	Americas	11.11	16.47
...	...	...	...
230	Viet Nam	56.35	31.98
231	Wallis and Futuna Islands	1.23	0.67
232	Yemen	1.07	4.85
233	Zambia	89.78	84.53
234	Zimbabwe	71.54	83.29

234 rows × 3 columns

[54]:
```
#for calculating mean
def cal_mean(data,column):
  column=str(column)
  count=len(data[column])
  sum=0
  for index,row in data.iterrows():
      sum=sum+row[column]
  u=round(sum/count,2)
  return u
#for calculating standard deviation
def cal_dev(data,column):
  sum=0
  column=str(column)
  count=len(data[column])
  for index,row in data.iterrows():
      diff=round((row[column]-cal_mean(data,column))**2,2)
      sum=sum+diff
  sd=round(math.sqrt(sum/count),2)
  return sd
```

In

For 2001

In [55]:
```
mean_1=cal_mean(data,2001)
dev_1=cal_dev(data,2001)
print('Mean: ',mean_1)
print('Standard Deviation: ',dev_1)
```

```
Mean:  28.8
Standard Deviation:  30.65
```

For 2017

In [56]:
```
mean_2=cal_mean(data,2017)
dev_2=cal_dev(data,2017)
print('Mean: ',mean_2)
print('Standard Deviation: ',dev_2)
```

```
Mean:  26.94
Standard Deviation:  26.96
```

Forming Hypothesis

It is claimed by UN that the Contribution of Renewable Energy in Total Energy is Increased

Null Hypothesis:

$p1-p2 \leq 0$

Alternate Hypothesis:

$p1-p2>0$

Left Tail

In [57]:
```
p1=(mean_1)/100
p2=(mean_2)/100
n1=len(data['2001'])
n2=len(data['2017'])
```

[58]:
```
def z_test_propdiff(p1,p2,n1,n2):
  p1=float(p1)
  p2=float(p2)
  n1=float(n1)
  n2=float(n2)
  p=(p1*n1 + p2*n2)/(n1+n2)
  q=1-p1
  a=p1-p2
  c=p*q
  b=(p*q*((1/n1)+(1/n2)))**(1/2)
  z=a/b
  if z<1.645:
    print("Null Hypothesis Accepted")
  else:
    print("Null Hypothesis Rejected")
  return z
```

In [59]:
```
z_test_propdiff(p1,p2,n1,n2)
```

```
Null Hypothesis Accepted
```

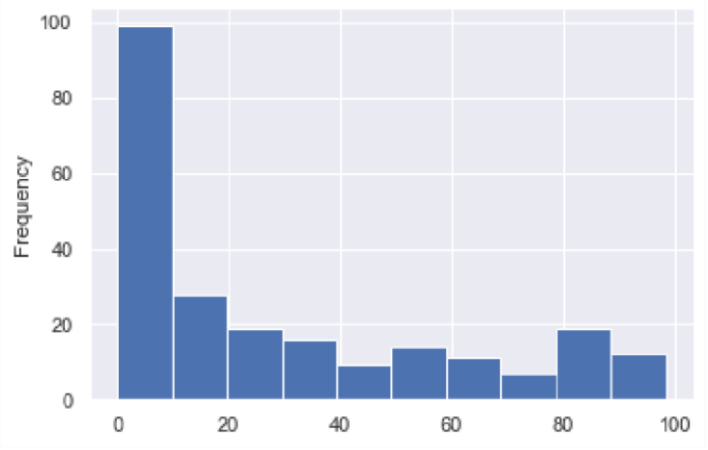
Out[59]:
```
0.4516451947199482
```

In

Graphs For Year 2001

In [60]:
```
data['2001'].plot.hist(bins=10)
```

Out[60]: <matplotlib.axes._subplots.AxesSubplot at 0x1d394b7cca0>



In [61]:
```
el=data.loc[(data['2001']<5)&(data['2001']>0)].drop(columns=['2017'])
el1=data.loc[(data['2001']<10)&(data['2001']>=5)].drop(columns=['2017'])
el1=data.loc[(data['2001']>=10)&(data['2001']<20)].drop(columns=['2017'])
el2=data.loc[(data['2001']>=20)&(data['2001']<40)].drop(columns=['2017'])
el3=data.loc[(data['2001']>=40)&(data['2001']<60)].drop(columns=['2017'])
el4=data.loc[(data['2001']>=60)&(data['2001']<80)].drop(columns=['2017'])
el5=data.loc[(data['2001']>=80)&(data['2001']<90)].drop(columns=['2017'])
el6=data.loc[(data['2001']>=90)&(data['2001']<100)].drop(columns=['2017'])
```

[62]:
```
x = el['GeoAreaName']
y = el['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption 1
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
```

Out[62]: Text(0, 0.5, 'Countries')

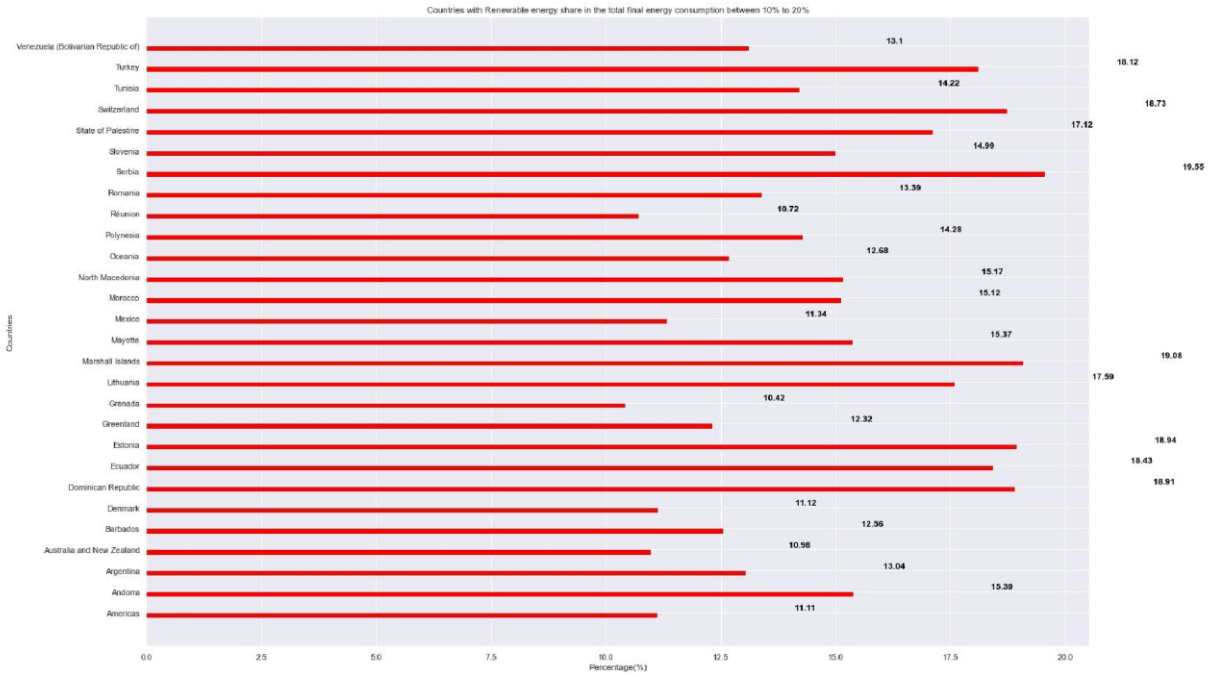
In

[63]:
```
x = ell['GeoAreaName']
y = ell['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
```

Out[63]: Text(0, 0.5, 'Countries')

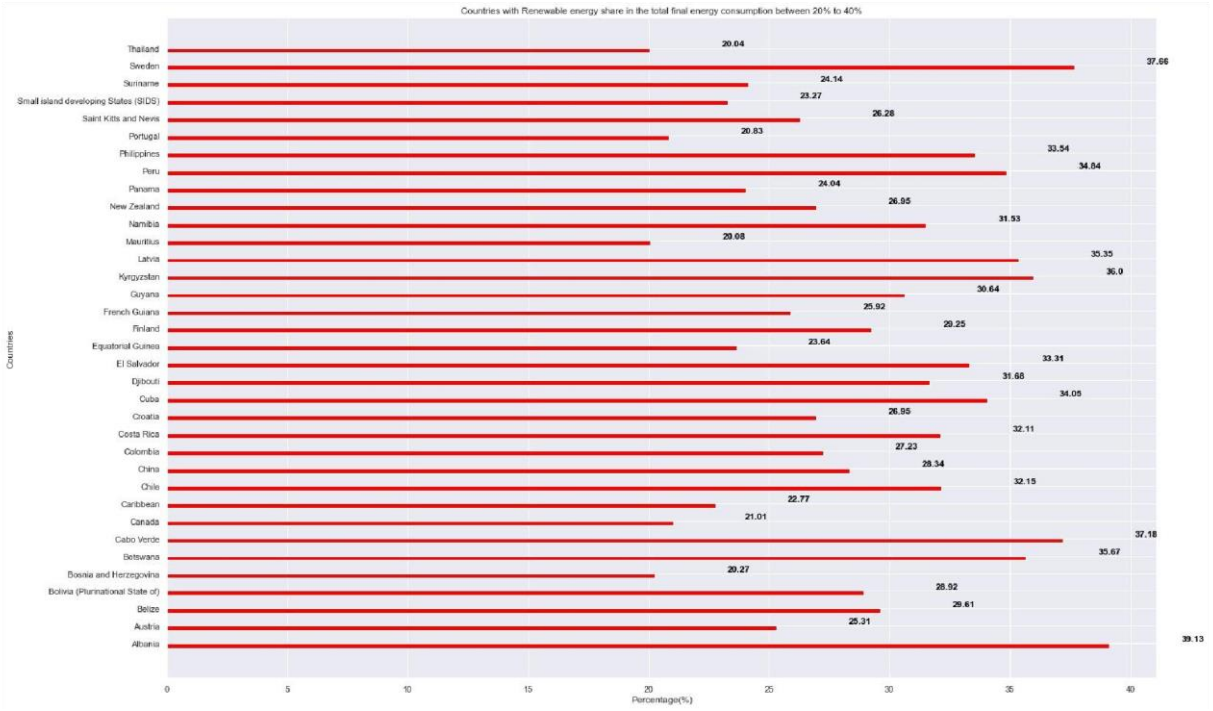
In

[64]:
```
x = el1['GeoAreaName']
y = el1['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



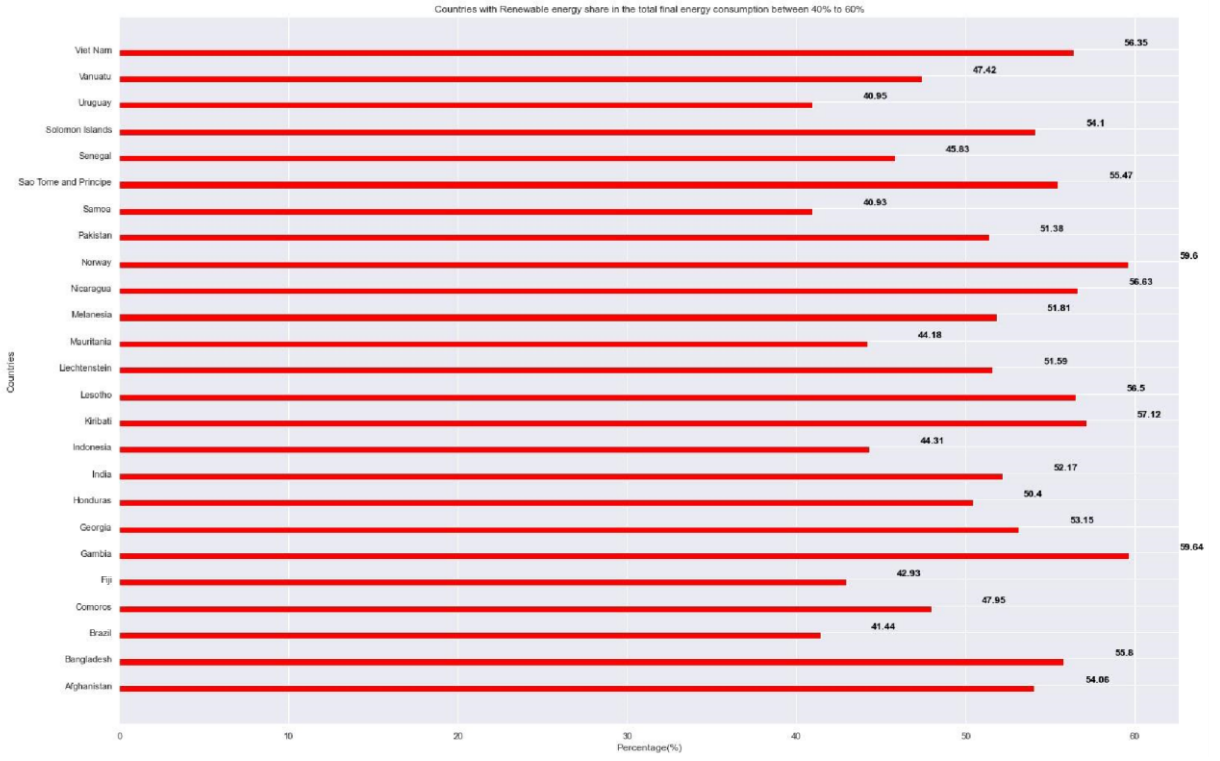
In

[65]:
```
x = el2['GeoAreaName']
y = el2['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



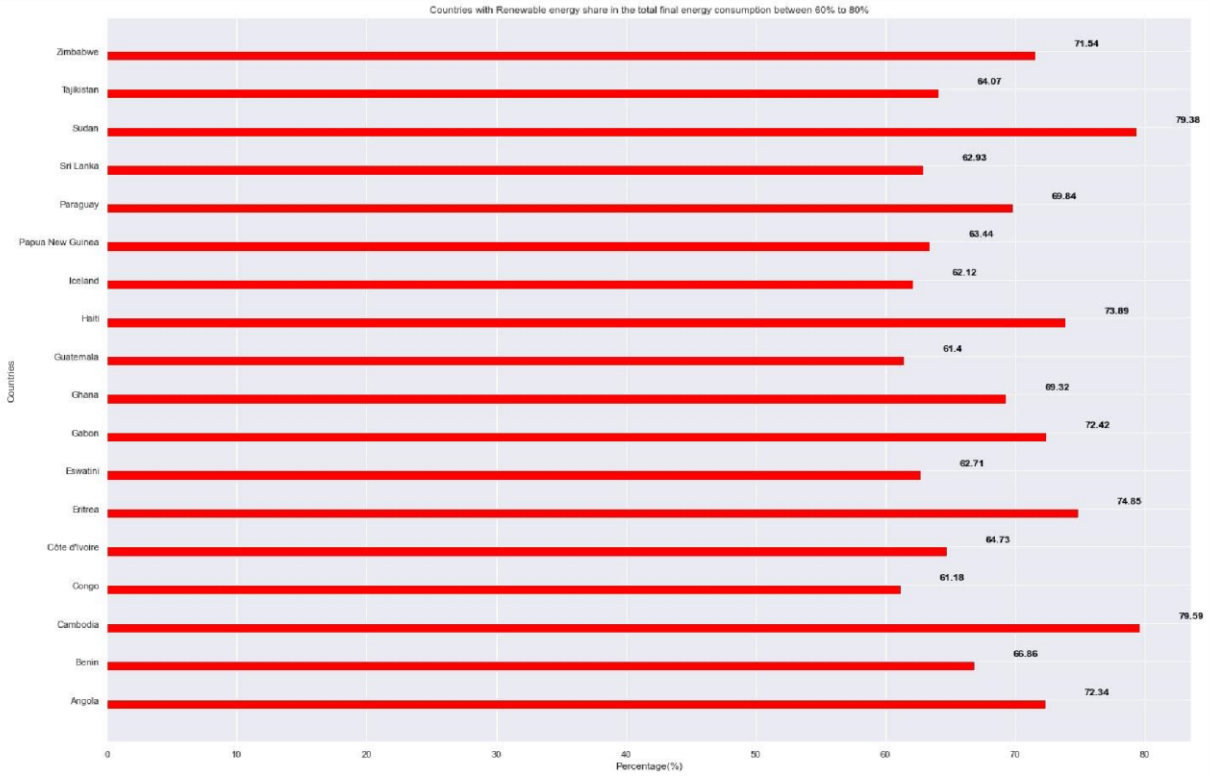
In

[66]:
```
x = el3['GeoAreaName']
y = el3['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

[67]:
```
x = el4['GeoAreaName']
y = el4['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

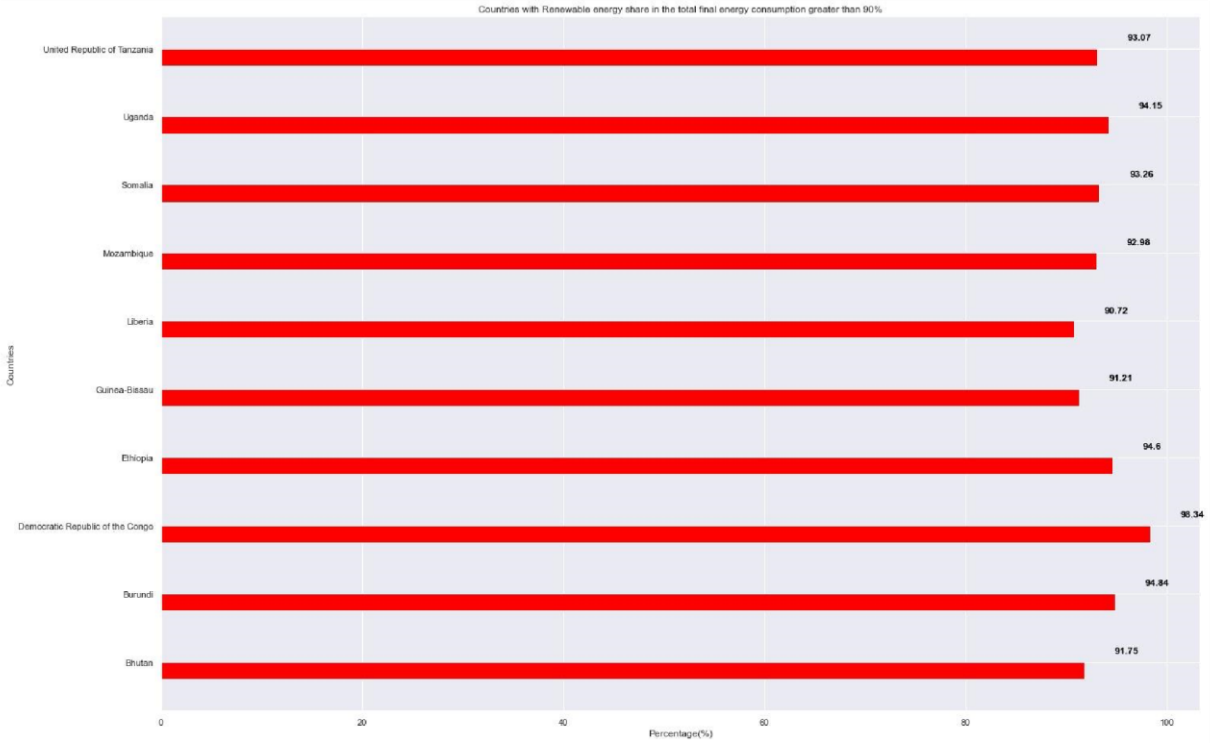


In

[68]:
```
x = el5['GeoAreaName']
y = el5['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

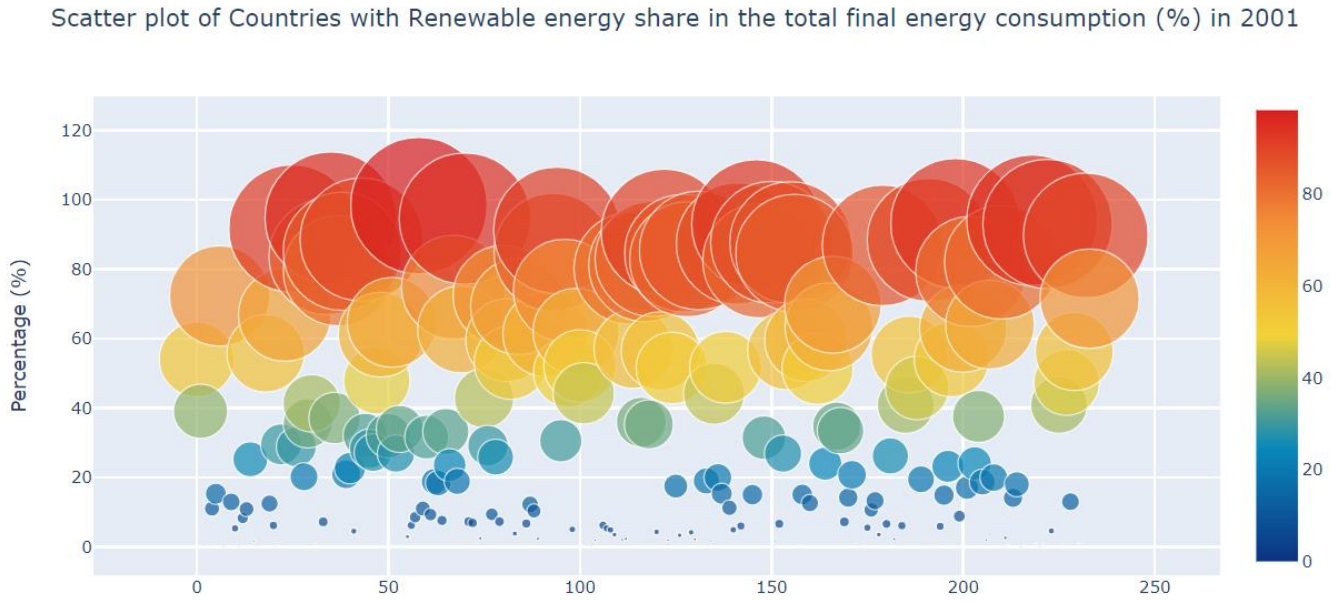
In

[69]:
```
x = el6['GeoAreaName']
y = el6['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption g
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

[70]:
```
l=[]
trace0= go.Scatter(
        y= data['2001'],
        mode= 'markers',
        name='Percentage (%)',
        marker= dict(size= data['2001'].values,
                     line= dict(width=1),
                     color= data['2001'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of Countries with Renewable energy share in the total final ene
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Percentage (%)',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```

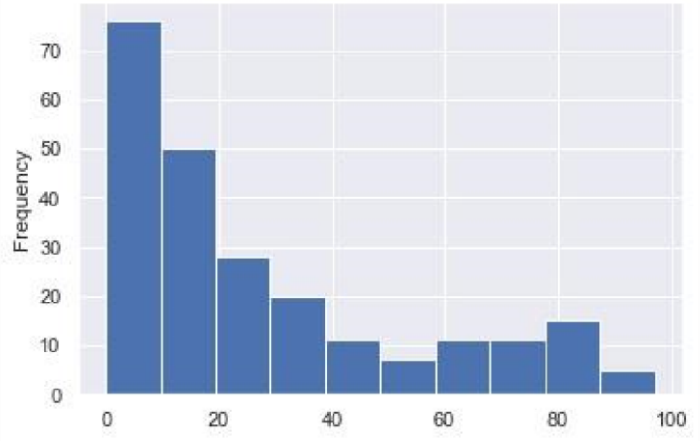


In

In [71]:
```
data['2017'].plot.hist(bins=10)
```

Graphs For Year 2017

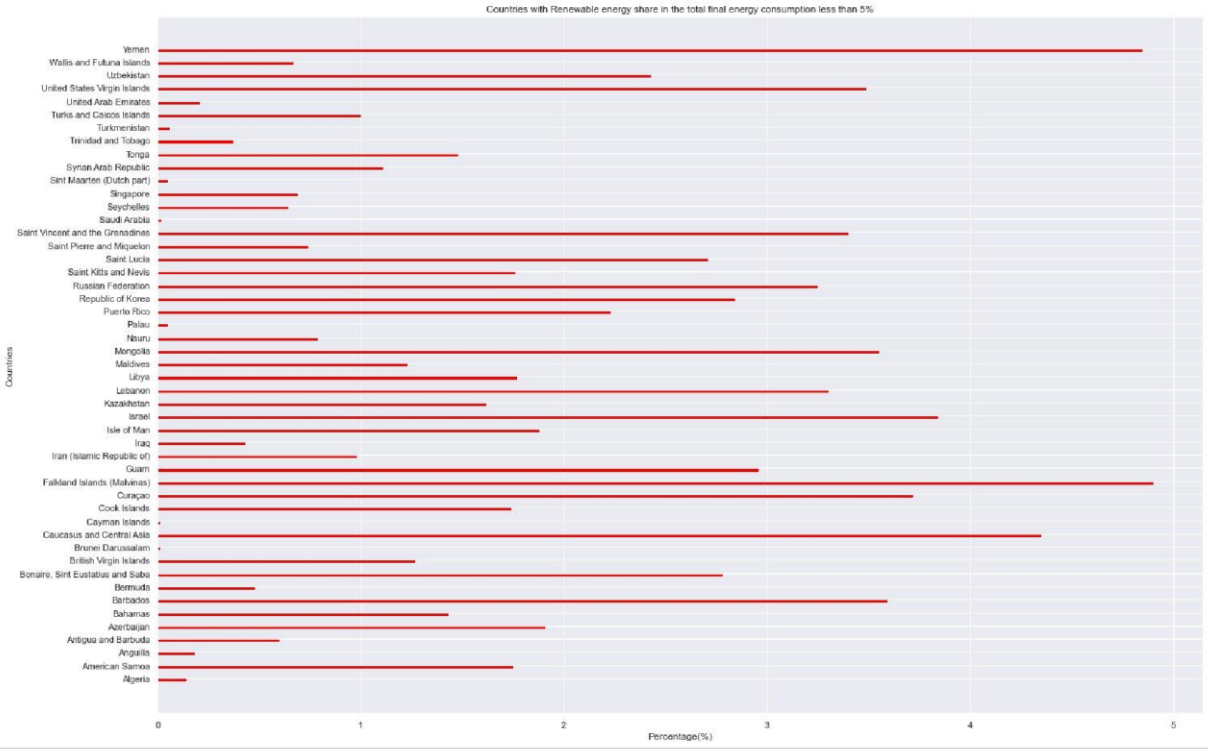
Out[71]: <matplotlib.axes._subplots.AxesSubplot at 0x1d3989c6eb0>



In [72]:
```
em=data.loc[(data['2017']<5)&(data['2017']>0)].drop(columns=['2001'])
em1=data.loc[(data['2017']<10)&(data['2017']>=5)].drop(columns=['2001'])
em1=data.loc[(data['2017']>=10)&(data['2017']<20)].drop(columns=['2001'])
em2=data.loc[(data['2017']>=20)&(data['2017']<40)].drop(columns=['2001'])
em3=data.loc[(data['2017']>=40)&(data['2017']<60)].drop(columns=['2001'])
em4=data.loc[(data['2017']>=60)&(data['2017']<80)].drop(columns=['2001'])
em5=data.loc[(data['2017']>=80)].drop(columns=['2001'])
```

[73]:
```
x = em['GeoAreaName']
y = em['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption l
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
```

Out[73]: Text(0, 0.5, 'Countries')

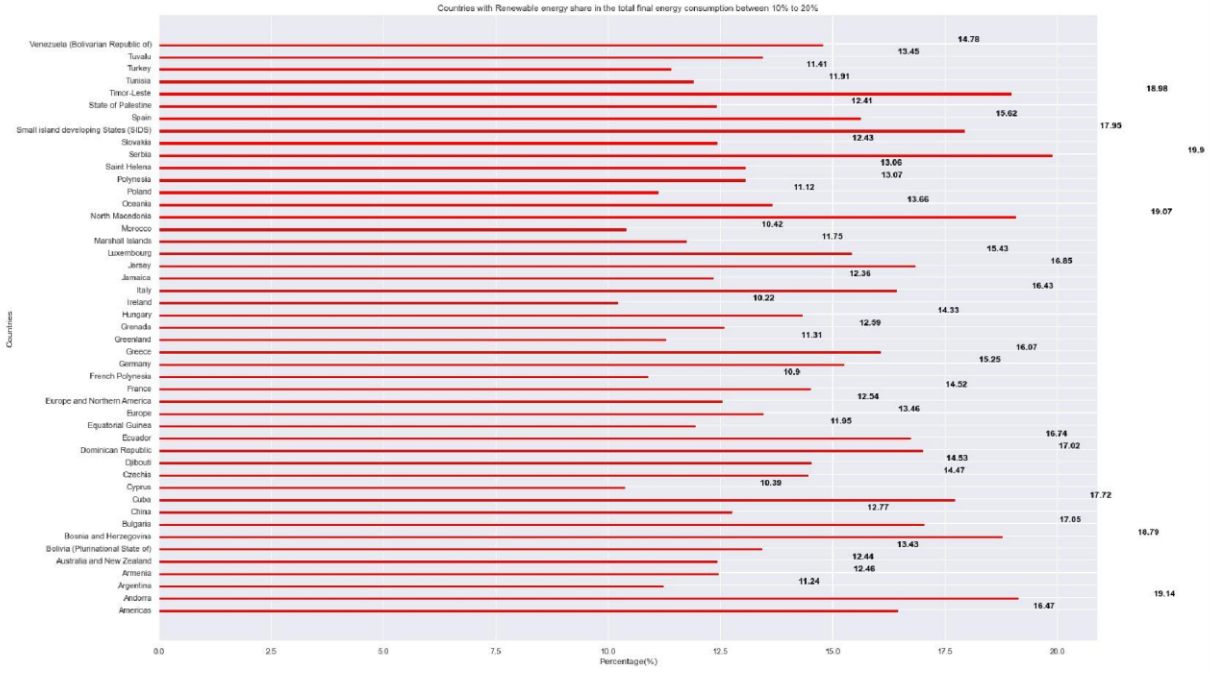


[74]:
```
x = eml['GeoAreaName']
y = eml['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
```

Out[74]: Text(0, 0.5, 'Countries')

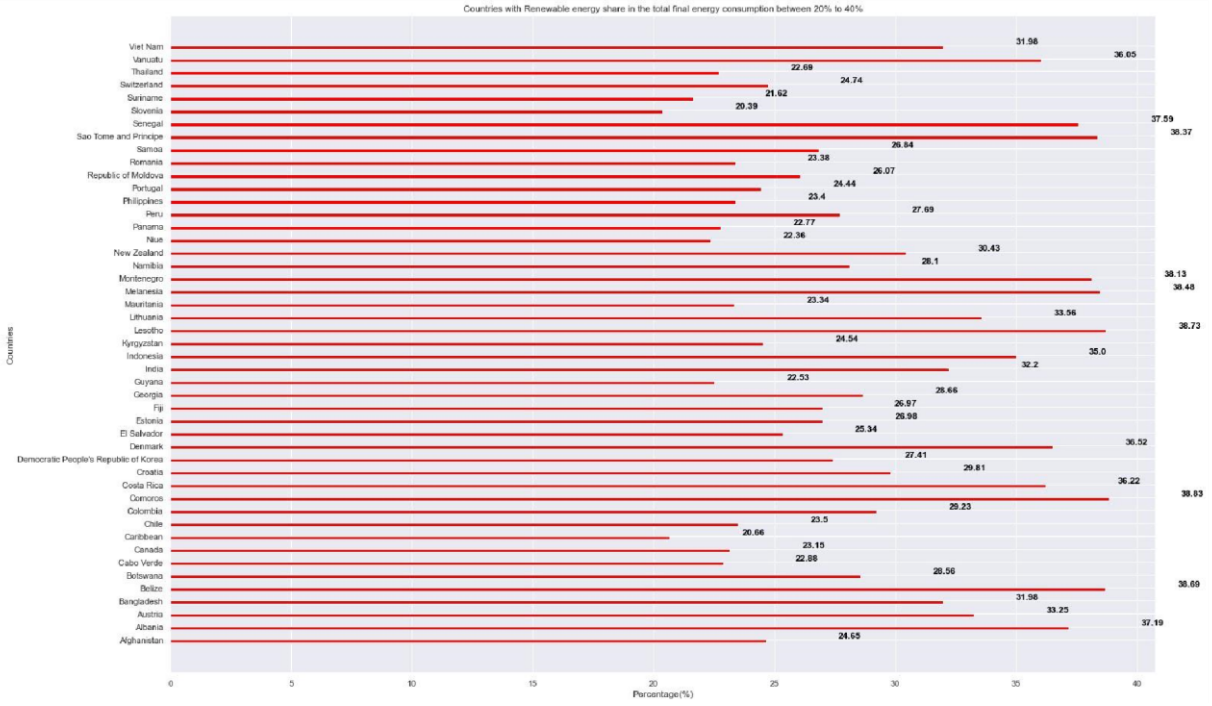
In

[75]:
```
x = em1['GeoAreaName']
y = em1['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

[76]:
```
x = em2['GeoAreaName']
y = em2['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

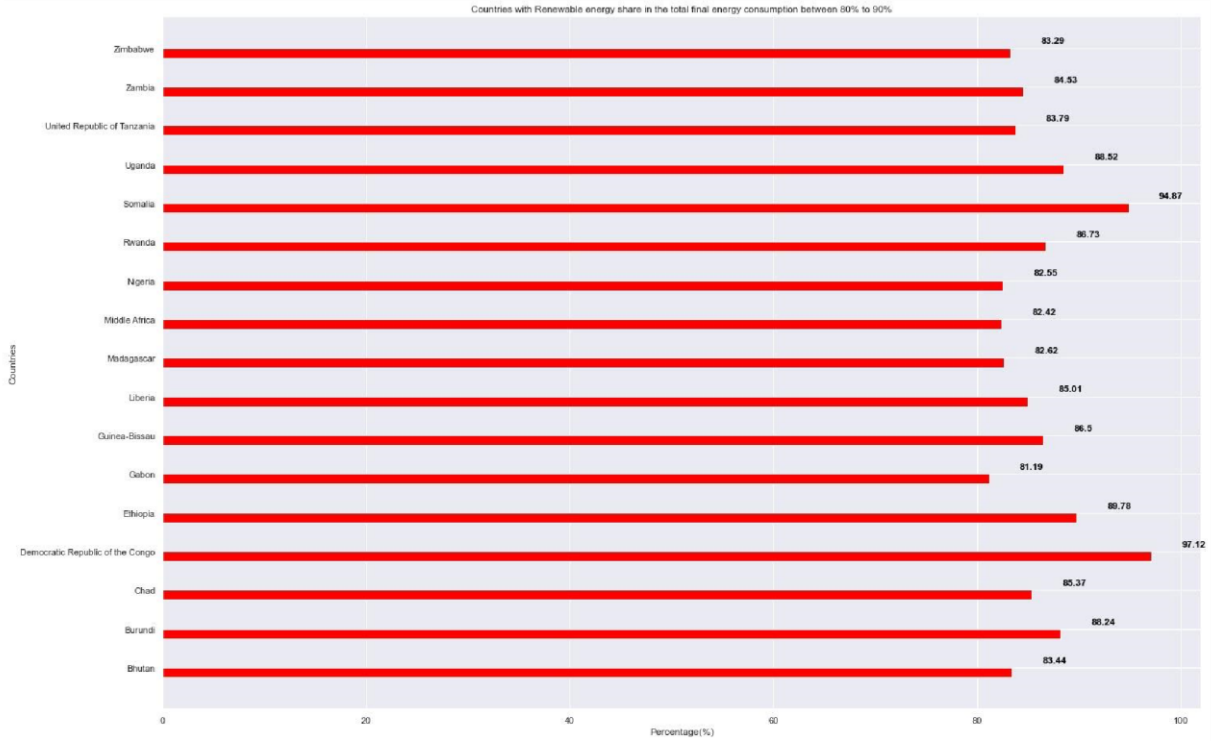
[77]:
```
x = em3['GeoAreaName']
y = em3['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

In

[78]:
```
x = em4['GeoAreaName']
y = em4['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

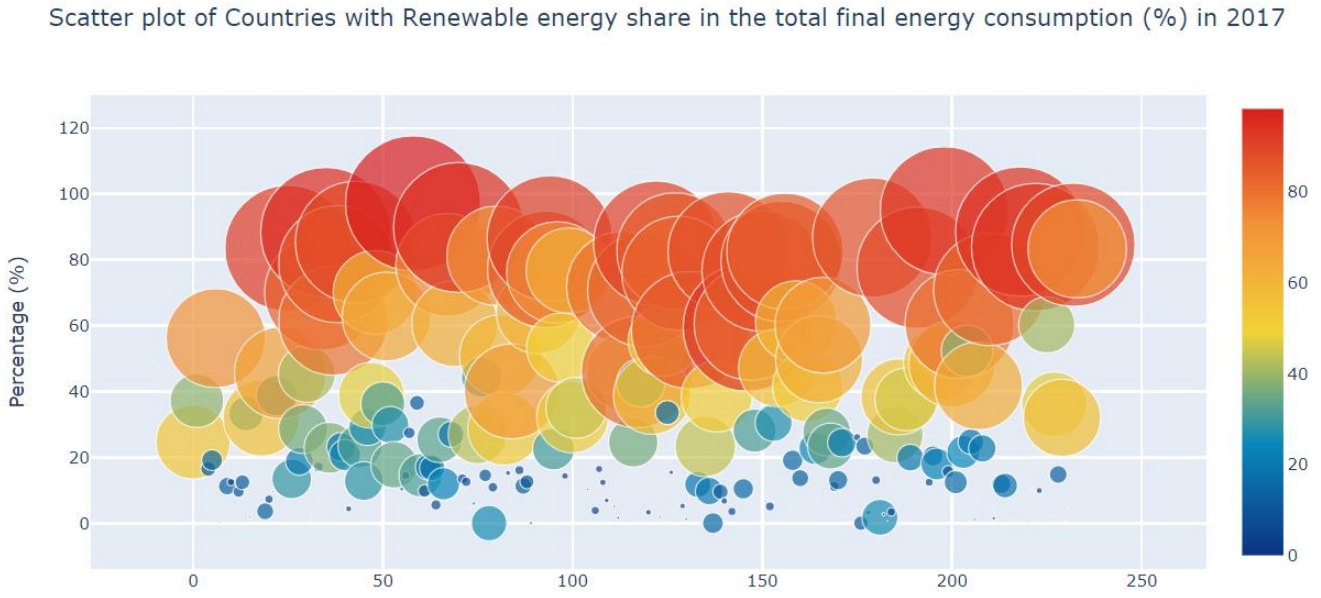
In

[79]:
```
x = em5['GeoAreaName']
y = em5['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Countries with Renewable energy share in the total final energy consumption b
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

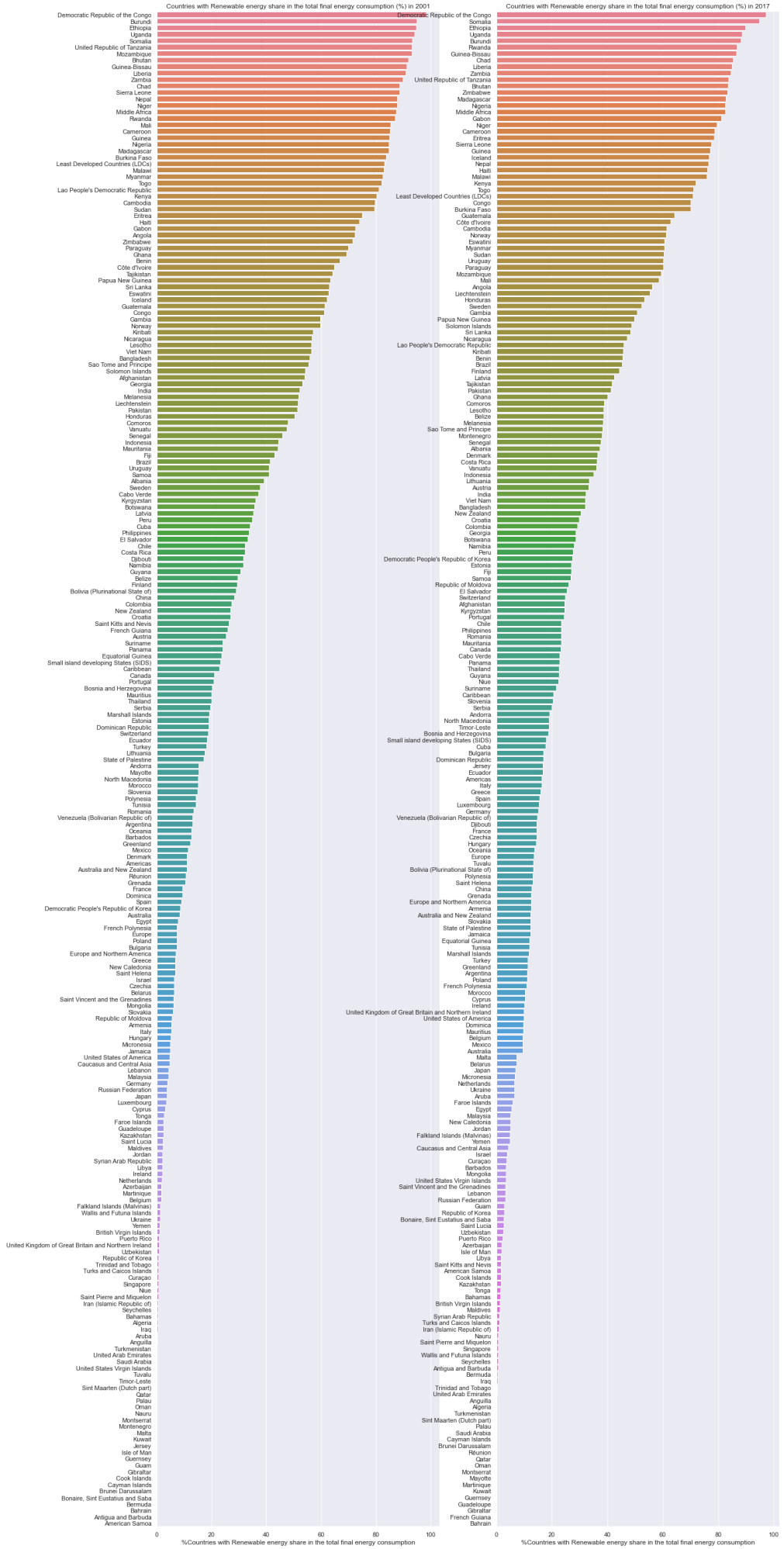
[80]:
```
l=[]
trace0= go.Scatter(
        y= data['2017'],
        mode= 'markers',
        name='Percentage (%)',
        marker= dict(size= data['2001'].values,
                     line= dict(width=1),
                     color= data['2001'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of Countries with Renewable energy share in the total final ene
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Percentage (%)',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```



In

[85]:
```
# Tried with Plotly now going with seaborn
twoyearchange1_bar, countries_bar1 = (list(x) for x in zip(*sorted(zip(data['2001'], dat
                                                                   reverse = True)))

twoyearchange2_bar, countries_bar2 = (list(x) for x in zip(*sorted(zip(data['2017'], dat
                                                                   reverse = True)))

# Another direct way of sorting according to values is creating distinct sorted datafram
# passing their values directly as in below mentioned code to achieve the same effect as

# df_country_sorted=df_country.sort(columns='2014-2012 change',ascending=False)
# df_country_sorted.head()


sns.set(font_scale=1)
fig, axes = plt.subplots(1,2,figsize=(20, 50))
colorspal = sns.color_palette('husl', len(data['2001']))
sns.barplot(twoyearchange1_bar, countries_bar1, palette = colorspal,ax=axes[0])
sns.barplot(twoyearchange2_bar, countries_bar2, palette = colorspal,ax=axes[1])
axes[0].set(xlabel='%Countries with Renewable energy share in the total final energy con
axes[1].set(xlabel='%Countries with Renewable energy share in the total final energy con
fig.savefig('output.png')
```

[86]:


Countries with Renewable energy share in the total final energy consumption (%) in 2001
Democratic Republic of the Congo
Burundi
Ethiopia
Uganda
Somalia
United Republic of Tanzania
Mozambique
Bhutan
Guinea-Bissau
Liberia
Zambia
Chad
Sierra Leone
Nepal
Niger
Middle Africa
Rwanda
Mali
Cameroon
Guinea
Nigeria
Madagascar
Burkina Faso
Least Developed Countries (LDCs)
Malawi
Myanmar
Togo
Lao People's Democratic Republic
Kenya
Cambodia
Sudan
Eritrea
Haiti
Gabon
Angola
Zimbabwe
Paraguay
Ghana
Benin
Côte d'Ivoire
Tajikistan
Papua New Guinea
Sri Lanka
Eswatini
Iceland
Guatemala
Congo
Gambia
Norway
Kiribati
Nicaragua
Lesotho
Viet Nam
Bangladesh
Sao Tome and Principe
Solomon Islands
Afghanistan
Georgia
India
Melanesia
Liechtenstein
Pakistan
Honduras
Comoros
Vanuatu
Senegal
Indonesia
Mauritania
Fiji
Brazil
Uruguay
Samoa
Albania
Sweden
Cabo Verde
Kyrgyzstan
Botswana
Latvia
Peru
Cuba
Philippines
El Salvador
Chile
Costa Rica
Djibouti
Namibia
Guyana
Belize
Finland
Bolivia (Plurinational State of)
China
Colombia
New Zealand
Croatia
Saint Kitts and Nevis
French Guiana
Austria
Suriname
Panama
Equatorial Guinea
Small island developing States (SIDS)
Caribbean
Canada
Portugal
Bosnia and Herzegovina
Mauritius
Thailand
Serbia
Marshall Islands
Estonia
Dominican Republic
Switzerland
Ecuador
Turkey
Lithuania
State of Palestine
Andorra
Mayotte
North Macedonia
Morocco
Slovenia
Polynesia
Tunisia
Romania
Venezuela (Bolivarian Republic of)
Argentina
Oceania
Barbados
Greenland
Mexico
Denmark
Americas
Australia and New Zealand
Réunion
Grenada
France
Dominica
Spain
Democratic People's Republic of Korea
Australia
Egypt
French Polynesia
Europe
Poland
Bulgaria
Europe and Northern America
Greece
New Caledonia
Saint Helena
Israel
Czechia
Belarus
Saint Vincent and the Grenadines
Mongolia
Slovakia
Republic of Moldova
Armenia
Italy
Hungary
Micronesia
Jamaica
United States of America
Caucasus and Central Asia
Lebanon
Malaysia
Germany
Russian Federation
Japan
Luxembourg
Cyprus
Tonga
Faroe Islands
Guadeloupe
Kazakhstan
Saint Lucia
Maldives
Jordan
Syrian Arab Republic
Libya
Ireland
Netherlands
Azerbaijan
Martinique
Belgium
Falkland Islands (Malvinas)
Wallis and Futuna Islands
Ukraine
Yemen
British Virgin Islands
Puerto Rico
United Kingdom of Great Britain and Northern Ireland
Uzbekistan
Republic of Korea
Trinidad and Tobago
Turks and Caicos Islands
Curaçao
Singapore
Niue
Saint Pierre and Miquelon
Iran (Islamic Republic of)
Seychelles
Bahamas
Algeria
Iraq
Aruba
Anguilla
Turkmenistan
United Arab Emirates
Saudi Arabia
United States Virgin Islands
Tuvalu
Timor-Leste
Sint Maarten (Dutch part)
Qatar
Palau
Oman
Nauru
Montserrat
Montenegro
Malta
Kuwait
Jersey
Isle of Man
Guernsey
Guam
Gibraltar
Cook Islands
Cayman Islands
Brunei Darussalam
Bonaire, Sint Eustatius and Saba
Bermuda
Bahrain
Antigua and Barbuda
American Samoa
0
20
40
60
80
100
%Countries with Renewable energy share in the total final energy consumption
Countries with Renewable energy share in the total final energy consumption (%) in 2017
Democratic Republic of the Congo
Somalia
Ethiopia
Uganda
Burundi
Rwanda
Guinea-Bissau
Chad
Liberia
Zambia
United Republic of Tanzania
Bhutan
Zimbabwe
Madagascar
Nigeria
Middle Africa
Gabon
Niger
Cameroon
Eritrea
Sierra Leone
Guinea
Iceland
Nepal
Haiti
Malawi
Kenya
Togo
Least Developed Countries (LDCs)
Congo
Burkina Faso
Guatemala
Côte d'Ivoire
Cambodia
Norway
Eswatini
Myanmar
Sudan
Uruguay
Paraguay
Mozambique
Mali
Angola
Liechtenstein
Honduras
Sweden
Gambia
Papua New Guinea
Solomon Islands
Sri Lanka
Nicaragua
Lao People's Democratic Republic
Kiribati
Benin
Brazil
Finland
Latvia
Tajikistan
Pakistan
Ghana
Comoros
Lesotho
Belize
Melanesia
Sao Tome and Principe
Montenegro
Senegal
Albania
Denmark
Costa Rica
Vanuatu
Indonesia
Lithuania
Austria
India
Viet Nam
Bangladesh
New Zealand
Croatia
Colombia
Georgia
Botswana
Namibia
Peru
Democratic People's Republic of Korea
Estonia
Fiji
Samoa
Republic of Moldova
El Salvador
Switzerland
Afghanistan
Kyrgyzstan
Portugal
Chile
Philippines
Romania
Mauritania
Canada
Cabo Verde
Panama
Thailand
Guyana
Niue
Suriname
Caribbean
Slovenia
Serbia
Andorra
North Macedonia
Timor-Leste
Bosnia and Herzegovina
Small island developing States (SIDS)
Cuba
Bulgaria
Dominican Republic
Jersey
Ecuador
Americas
Italy
Greece
Spain
Luxembourg
Germany
Venezuela (Bolivarian Republic of)
Djibouti
France
Czechia
Hungary
Oceania
Europe
Tuvalu
Bolivia (Plurinational State of)
Polynesia
Saint Helena
China
Grenada
Europe and Northern America
Armenia
Australia and New Zealand
Slovakia
State of Palestine
Jamaica
Equatorial Guinea
Tunisia
Marshall Islands
Turkey
Greenland
Argentina
Poland
French Polynesia
Morocco
Cyprus
Ireland
United Kingdom of Great Britain and Northern Ireland
United States of America
Dominica
Mauritius
Belgium
Mexico
Australia
Malta
Belarus
Japan
Micronesia
Netherlands
Ukraine
Aruba
Faroe Islands
Egypt
Malaysia
New Caledonia
Jordan
Falkland Islands (Malvinas)
Yemen
Caucasus and Central Asia
Israel
Curaçao
Barbados
Mongolia
United States Virgin Islands
Saint Vincent and the Grenadines
Lebanon
Russian Federation
Guam
Republic of Korea
Bonaire, Sint Eustatius and Saba
Saint Lucia
Uzbekistan
Puerto Rico
Azerbaijan
Isle of Man
Libya
Saint Kitts and Nevis
American Samoa
Cook Islands
Kazakhstan
Tonga
Bahamas
British Virgin Islands
Maldives
Syrian Arab Republic
Turks and Caicos Islands
Iran (Islamic Republic of)
Nauru
Saint Pierre and Miquelon
Singapore
Wallis and Futuna Islands
Seychelles
Antigua and Barbuda
Bermuda
Iraq
Trinidad and Tobago
United Arab Emirates
Anguilla
Algeria
Turkmenistan
Sint Maarten (Dutch part)
Palau
Saudi Arabia
Cayman Islands
Brunei Darussalam
Réunion
Qatar
Oman
Montserrat
Mayotte
Martinique
Kuwait
Guernsey
Guadeloupe
Gibraltar
French Guiana
Bahrain
0
20
40
60
80
100
%Countries with Renewable energy share in the total final energy consumption

```
x_data = ['2001','2017']

y0 = data['2001']
y1 = data['2017']


y_data = [y0,y1]

colors = ['rgba(93, 164, 214, 0.5)', 'rgba(255, 144, 14, 0.5)']

traces = []

for xd, yd, color in zip(x_data, y_data, colors):
        traces.append(go.Box(
            y=yd,
            name=xd,
            boxpoints='all',
            whiskerwidth=0.2,
            fillcolor=color,
            marker=dict(
                size=2,
            ),
            boxmean=True,
            line=dict(width=1),
        ))

layout = go.Layout(
    title='Distribution of Data',
    xaxis=dict(
        title='Year'
    ),
    yaxis=dict(
        title='Percentage (%)',
        autorange=True,
        showgrid=True,
        zeroline=False,
        dtick=5,
        gridcolor='rgb(255, 255, 255)',
        gridwidth=1,
#         zerolinecolor='rgb(255, 255, 255)',
#         zerolinewidth=2,
    ),
    margin=dict(
        l=40,
        r=30,
        b=80,
        t=100,
    ),
    paper_bgcolor='rgb(243, 243, 243)',
    plot_bgcolor='rgb(243, 243, 243)',
    showlegend=False
)

fig = go.Figure(data=traces, layout=layout)
py.iplot(fig)
```


Distribution of Data
Percentage (%)
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
−5
2001
2017
Year

Regression Equation

In [2]:

```
import pandas as pd
data=pd.read_csv('renewable.csv')
```

data

Out[2]:

	GeoAreaName	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	20
0	Afghanistan	54.24	54.06	43.77	42.28	49.84	40.86	37.14	33.86	21.34	17.85	14.
1	Africa	60.46	59.79	59.30	58.61	58.58	57.42	57.22	56.24	55.99	55.91	56.
2	Albania	41.45	39.13	35.90	33.75	35.93	36.87	31.71	32.10	35.91	37.22	37.
3	Algeria	0.43	0.43	0.51	0.47	0.44	0.58	0.41	0.41	0.30	0.31	0.
4	American Samoa	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
...	...	...	...	...	...	...	...	...	...	...	...	...
267	Western Asia (exc. Armenia,Azerbaijan, Cyprus...	5.70	5.31	5.31	5.22	5.34	4.90	4.65	4.05	3.67	3.67	3.
268	WesternEurope	6.71	6.91	6.82	7.03	7.45	7.89	8.40	9.69	9.79	10.70	11.
269	Yemen	1.15	1.07	1.06	0.93	0.91	0.89	0.93	0.94	0.90	0.86	0.
270	Zambia	89.93	89.78	89.67	89.44	89.63	89.18	90.05	93.13	92.40	92.33	92.
271	Zimbabwe	69.27	71.54	74.30	77.84	81.59	80.16	78.65	78.09	82.05	82.35	82.

272 rows × 19 columns

In [3]:

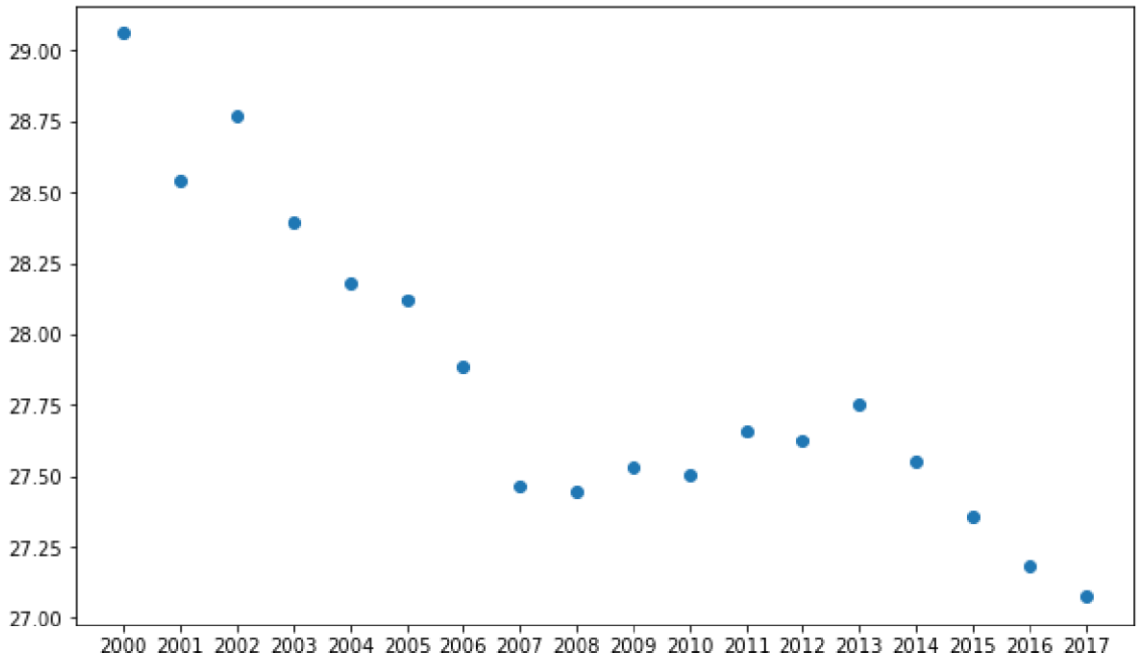
```
mean_year=pd.Series(data[1:].mean())
mean_year
```

Out[3]:

```
2000   29.059145
2001   28.542639
2002   28.769926
2003   28.396171
2004   28.180929
2005   28.116815
2006   27.883148
2007   27.463370
2008   27.444037
2009   27.533222
2010   27.503704
2011   27.660755
2012   27.624060
```

```
2013   27.750940
2014   27.549436
2015   27.354774
2016   27.186842 2017    27.076955 dtype: float64 In [15]:
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
plt.scatter(mean_year.index,mean_year)
plt.show()
```



In [5]:

```
table=pd.DataFrame(mean_year)
table.columns=['Mean (Y)']
```

In [6]:

```
table['X']=range(1,19)
table['XY']=table['Mean (Y)']*table['X']
table['X^2']=table['X']**2
table=pd.DataFrame(table)
```

In [7]:

```
sum=pd.DataFrame(table.sum())
sum.columns=['Sigma']
sum

mean=pd.DataFrame(table.mean())
mean.columns=['Mean']
mean
```

Out[7]:

	Mean
Mean (Y)	27.838715
X	9.500000
XY	261.968731

X^2 117.166667

In [8]:

```
table=pd.concat([table,sum.T,mean.T])
```

table

Out[8]:

	Mean (Y)	X	XY	X^2
2000	29.059145	1.0	29.059145	1.000000
2001	28.542639	2.0	57.085279	4.000000
2002	28.769926	3.0	86.309777	9.000000
2003	28.396171	4.0	113.584684	16.000000
2004	28.180929	5.0	140.904647	25.000000
2005	28.116815	6.0	168.700889	36.000000
2006	27.883148	7.0	195.182037	49.000000
2007	27.463370	8.0	219.706963	64.000000
2008	27.444037	9.0	246.996333	81.000000
2009	27.533222	10.0	275.332222	100.000000
2010	27.503704	11.0	302.540741	121.000000
2011	27.660755	12.0	331.929057	144.000000
2012	27.624060	13.0	359.112782	169.000000
2013	27.750940	14.0	388.513158	196.000000
2014	27.549436	15.0	413.241541	225.000000
2015	27.354774	16.0	437.676391	256.000000
2016	27.186842	17.0	462.176316	289.000000
2017	27.076955	18.0	487.385188	324.000000
Sigma	501.096869	171.0	4715.437149	2109.000000
Mean	27.838715	9.5	261.968731	117.166667

In [9]:

```
numerator = table ['XY'  ] . loc [ 'Sigma' ] - table [ 'X' ].  loc [ 'Sigma' ] * table[
'Mean (Y)'  ] . loc [ 'Sigm a' ] / 20
numerator
```

Out[9]:

431.05891988180247

In [10]:

```
denominator = table [ 'X^2']  . loc [ 'Sigma'] -  table[  'X' ] . loc [ 'Sigma'
] **2  /20   denominator
```

Out[10]:

646.95

In [11]:

```
b=numerator/denominator
b
```

Out[11]:

0.6662940256307326

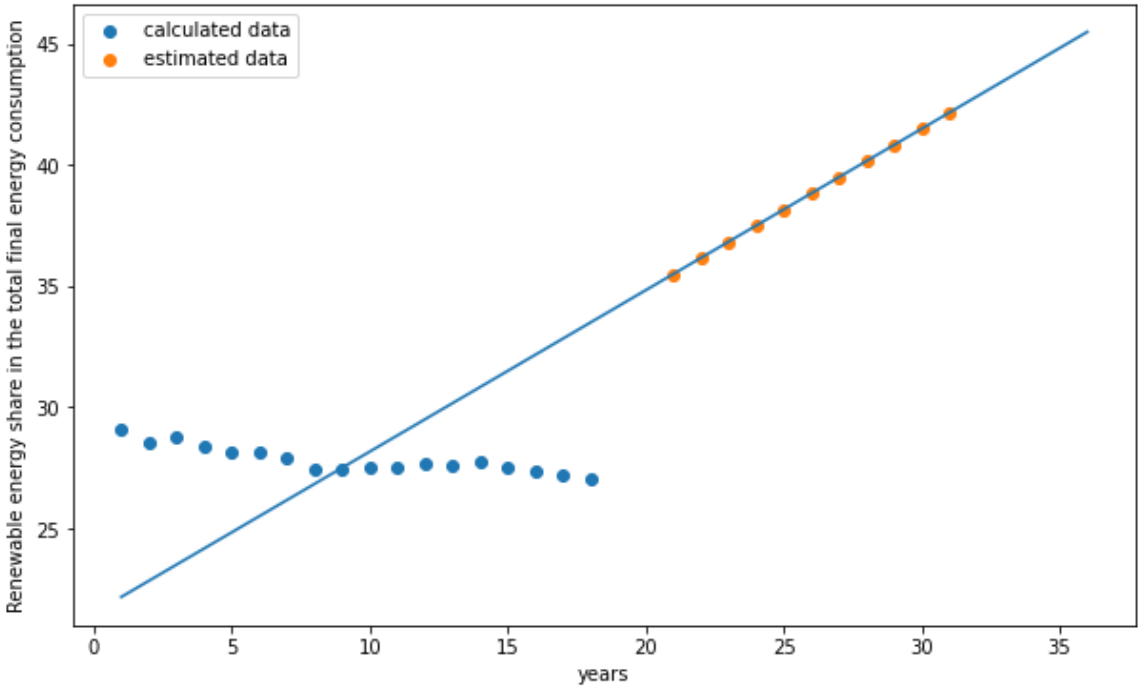
In [12]:

```
def  eq_regression ( x ):
     y = table [ 'Mean (Y)' ] . loc [ 'Mean'] +  b*  ( x-  table[  'X' ] .
 loc[  'Mean' ])      return ( y )
```

In [17]:

```
plt.figure(figsize=(10,6))
x_values=[]
y_values=[]
for x in range(21,32):
    x_values.append(x)
    y_values.append(eq_regression(x))
print("estimated Renewable energy share in the total final energy consumption for next
 11 years",y_values)

aabbccdd=[]
for i in mean_year.index:
    aabbccdd.append(int(i)-1999)
plt.scatter(aabbccdd,mean_year,label='calculated data')
plt.scatter([range(21,32)],y_values,label='estimated data')
plt.legend()

y_line=[]
x_line=[]
for x in range(1,37):
    x_line.append(x)
    y_line.append(eq_regression(x))
plt.plot(x_line,y_line)
plt.xlabel("years")
plt.ylabel("Renewable energy share in the total final energy consumption")
plt.show()
```

```
estimated Renewable energy share in the total final energy consumption for next
11 years [35.50109623591989, 36.16739026155062, 36.833684287181356, 3
7.49997831281209, 38.16627233844282, 38.83256636407356, 39.49886038970428
4, 40.16515441533502, 40.83144844096575, 41.497742466596485, 42.1640364922
2722]
```



In [ ]:

```
0.66629402563073,
```

Progress Report of Different Regions

In [3]:
```
import pandas as pd
df=pd.read_csv('usage_of_renewable_energy.csv')
df.index=range(1,7)
df
```

Out[3]:

	Cotinents	2017
1	Africa	54.38
2	Asia	16.01
3	Australia	9.54
4	Europe	13.46
5	North America	11.56
6	South America	34.21

In [5]:
```
#converting data into required matrix
import pandas as pd
import numpy as np
en_rank = pd.read_csv("Usage_ranking.csv")
df = pd.DataFrame(en_rank)
df.index=range(1,7)
df
```

Out[5]:

	Unnamed: 0	Africa	Asia	Australia	Europe	North America	South America
1	Africa	0	2	2	2	2	2
2	Asia	1	0	2	2	2	1
3	Australia	1	1	0	1	1	1
4	Europe	1	1	2	0	2	1
5	North America	1	1	2	1	0	1
6	South America	1	2	2	2	2	0

In [6]:
```
#converting data frame into matrix
a=df.iloc[:,1:].to_numpy()
print("Matrix:")
print(a)
```

```
Matrix:
[[0 2 2 2 2 2]
 [1 0 2 2 2 1]
 [1 1 0 1 1 1]
 [1 1 2 0 2 1]
 [1 1 2 1 0 1]
 [1 2 2 2 2 0]]
```

In [7]:
```
#calculating eigen value and eigen vector
import numpy as np
x0=np.array([[1],[1],[1],[1],[1],[1]])
j=1
#first iteration
x1=np.matmul(a,x0)
previous_eigen_value=np.round(max([abs(i) for i in x1]),4)
x1=x1/previous_eigen_value
print("Step",j,":",'\n',x1,'\n',"Eigen Value : ",previous_eigen_value)
j+=1
#second iteration
x3=np.matmul(a,x1)
present_eigen_value=np.round(max([abs(i) for i in x3]),4)
x3=np.round(x3/present_eigen_value,4)
print("Step",j,":",'\n',x3,'\n',"Eigen Value : ",present_eigen_value)
#while loop
while(previous_eigen_value!=present_eigen_value):#will calculate until previous and present eigen value are same
    j+=1
    previous_eigen_value=present_eigen_value
    x1=np.matmul(a,x3)
    present_eigen_value=np.round(max([abs(i) for i in x1]),4)
    x1=x1/present_eigen_value
    print("Step",j,"\n",x1,'\n',"Eigen Value : ",present_eigen_value)
    x3=x1
```

```
Step 1 :
 [[1. ]
 [0.8]
 [0.5]
 [0.7]
 [0.6]
 [0.9]]
 Eigen Value :  [10]
Step 2 :
 [[1.    ]
 [0.7857]
 [0.5714]
 [0.7   ]
 [0.6286]
 [0.8857]]
 Eigen Value :  [7.]
Step 3
 [[1.        ]
 [0.79600437]
 [0.56000448]
 [0.71000168]
 [0.63199306]
 [0.89200314]]
 Eigen Value :  [7.1428]
Step 4
 [[1.00000187]
 [0.79331498]
 [0.56128165]
 [0.70640704]
 [0.62925044]
 [0.8908088 ]]
 Eigen Value :  [7.18]
Step 5
 [[1.00000361]
 [0.79371817]
 [0.56125761]
 [0.70722132]
 [0.63013585]
 [0.89087141]]
 Eigen Value :  [7.1621]
Step 6
 [[1.00000122]
 [0.79371855]
 [0.56122326]
 [0.70710261]
 [0.62992991]
 [0.89091727]]
 Eigen Value :  [7.1664]
Step 7
 [[0.99999765]
 [0.79369087]
 [0.56123106]
 [0.70710086]
 [0.62996262]
 [0.89089144]]
 Eigen Value :  [7.1658]
Step 8
 [[0.99999354]
 [0.79369759]
 [0.56122742]
 [0.70710421]
 [0.62995659]
 [0.89089404]]
 Eigen Value :  [7.1658]
```

Ranking is determined on the basis of the elements present in the most dominant Eigen Vector

1. Africa (0.999)
2. South America (0.890)
3. Asia (0.794)
4. Europe (0.707)
5. North America (0.630)
6. Australia (0.561)

Objective 3: (Click to view all Objectives)

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import scipy
from collections import OrderedDict
import math
import numpy as np
import io
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [31]:
```
data=pd.read_csv('1 EG_EGY_CLEAN.csv',index_col='S.No.')
data
```

Out[31]:

S.No.	GeoAreaName	2001	2017
1	Afghanistan	11	34
2	Albania	41	78
3	Algeria	95	95
4	Andorra	95	95
5	Angola	44	48
...	...	...	...
204	Viet Nam	16	62
205	Western Asia	88	93
206	Yemen	57	61
207	Zambia	14	14
208	Zimbabwe	34	29

208 rows × 3 columns

Since This is a Percentage Data we will use Test of Proportion

In [32]:
```
#for calculating mean
def cal_mean(data,column):
  column=str(column)
  count=len(data[column])
  sum=0
  for index,row in data.iterrows():
      sum=sum+row[column]
  u=round(sum/count,2)
  return u
```

In

[33]:
```
#for calculating standard deviation
def cal_dev(data,column):
  sum=0
  column=str(column)
  count=len(data[column])
  for index,row in data.iterrows():
      diff=round((row[column]-cal_mean(data,column))**2,2)
      sum=sum+diff
  sd=round(math.sqrt(sum/count),2)
  return sd
```

For 2001

UN Claim is 50%

In [34]:
```
#calculating mean and standard deviation
mean_1=cal_mean(data,2001)
dev_1=cal_dev(data,2001)
print('Mean: ',mean_1)
print('Standard Deviation: ',dev_1)
```

```
Mean:  58.17
Standard Deviation:  36.9
```

In [35]:
```
#creating hypothesis from claim print('Null Hypothesis : h0=50%
\nAlternate Hypothesis : h1≠50%')
```

```
Null Hypothesis : h0=50%
Alternate Hypothesis : h1≠50%
```

In [36]:
```
p0_1=50/100
p_1=round(mean_1/100,2)
n_1=len(data['2001'])
```

In [37]:
```
def z_test_single_prop(p0,p,n):
  p0=float(p0)
  p=float(p)
  n=float(n)
  a=(p-p0)
  q=float(1-p0)
  b=float(((p0)*q)/n)
  z=round(a/(b**(1/2)),2)
  if -1.69<z<1.69:
    print("Hypothesis Accepted")
  else:
    print("Hypothesis Rejected")
  return z
```

In [38]:
```
z_test_single_prop(p0_1,p_1,n_1)
```

```
Hypothesis Rejected
```

Out[38]:
```
2.31
```

In

For 2017

UN Claim is 62%

[39]:
```
#calculating mean and standard deviation
mean_2=cal_mean(data,2017)
dev_2=cal_dev(data,2017)
print('Mean: ',mean_2)
print('Standard Deviation: ',dev_2)
```

```
Mean:  64.5
Standard Deviation:  35.27
```

In [40]:
```
#creating hypothesis from claim print('Null Hypothesis : h0=62%
\nAlternate Hypothesis : h1≠62%')
```

```
Null Hypothesis : h0=62%
Alternate Hypothesis : h1≠62%
```

In [41]:
```
p0_2=62/100
p_2=round(mean_2/100,2)
n_2=len(data['2017'])
```

In [42]:
```
z_test_single_prop(p0_2,p_2,n_2)
```

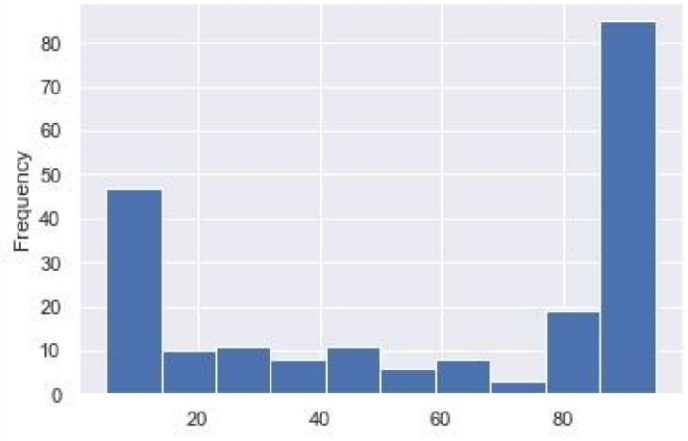
```
Hypothesis Accepted
```

Out[42]:
```
0.89
```

Graphs For Year 2001

In [80]:
```
data['2001'].plot.hist(bins=10)
```

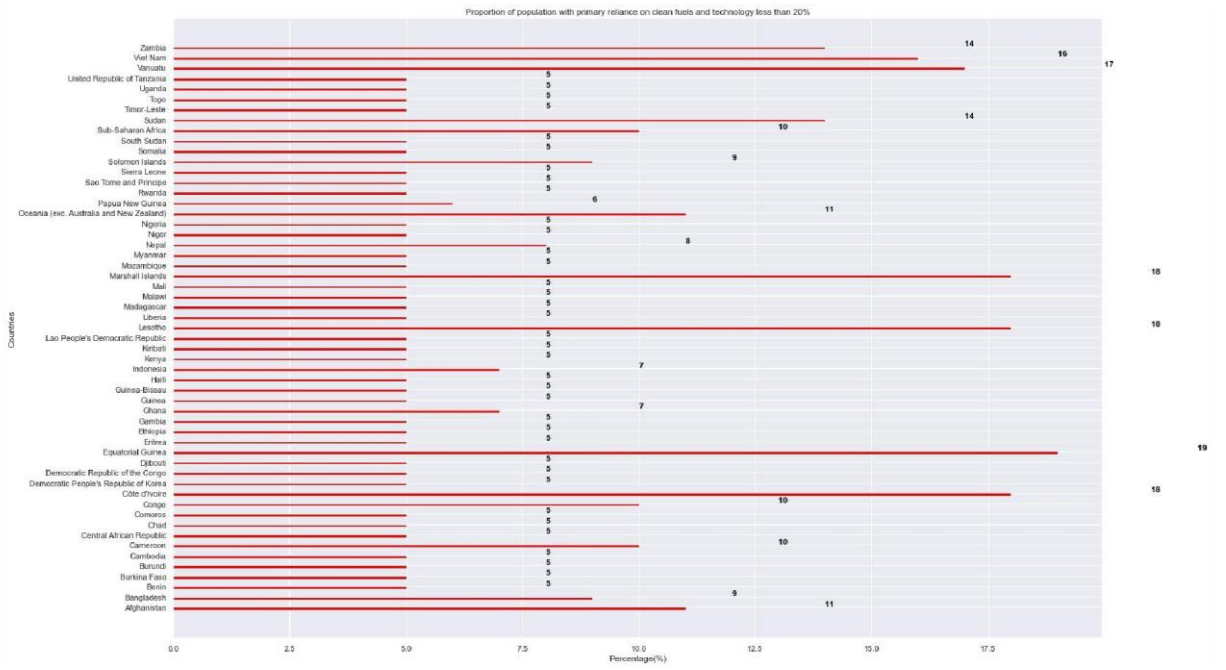
Out[80]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2508458a790>
```



In [44]:
```
el1=data.loc[data['2001']<20].drop(columns=['2017'])
el2=data.loc[(data['2001']>=20)&(data['2001']<40)].drop(columns=['2017'])
el3=data.loc[(data['2001']>=40)&(data['2001']<60)].drop(columns=['2017'])
el4=data.loc[(data['2001']>=60)&(data['2001']<80)].drop(columns=['2017'])
el5=data.loc[(data['2001']>=80)&(data['2001']<90)].drop(columns=['2017'])
el6=data.loc[(data['2001']>=90)&(data['2001']<100)].drop(columns=['2017'])
```

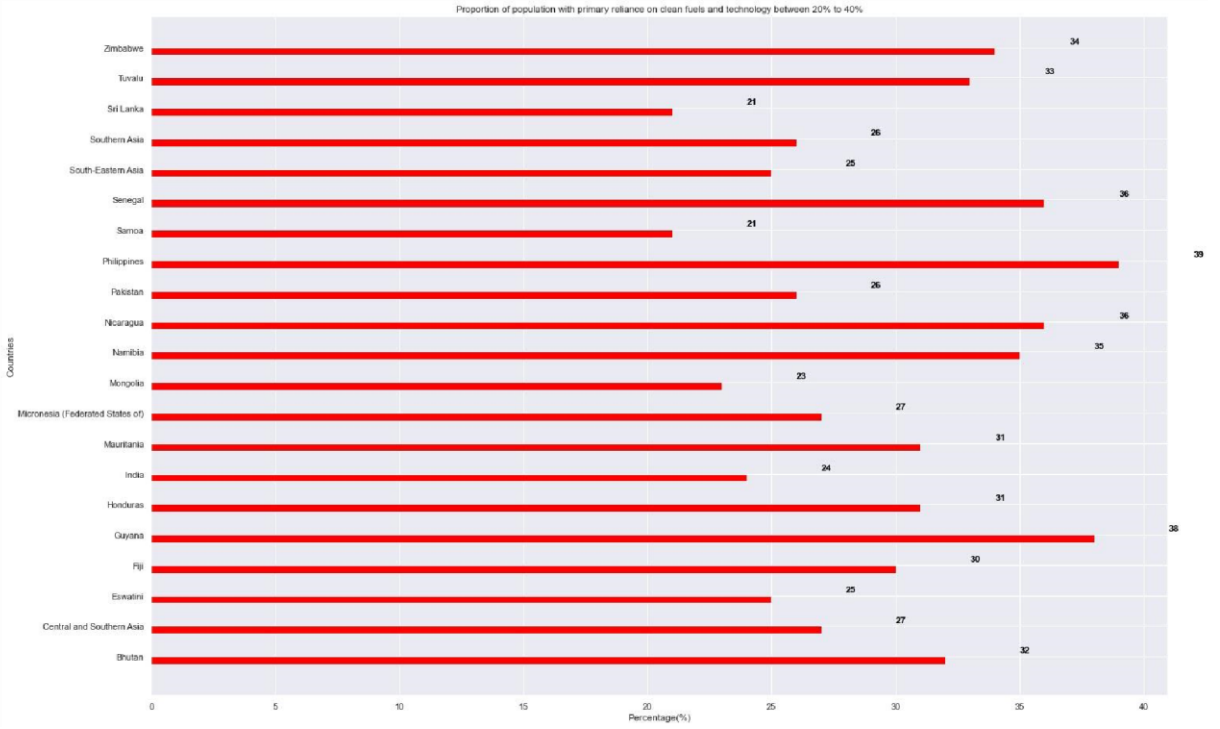
In [62]:

```
x = el1['GeoAreaName']
y = el1['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



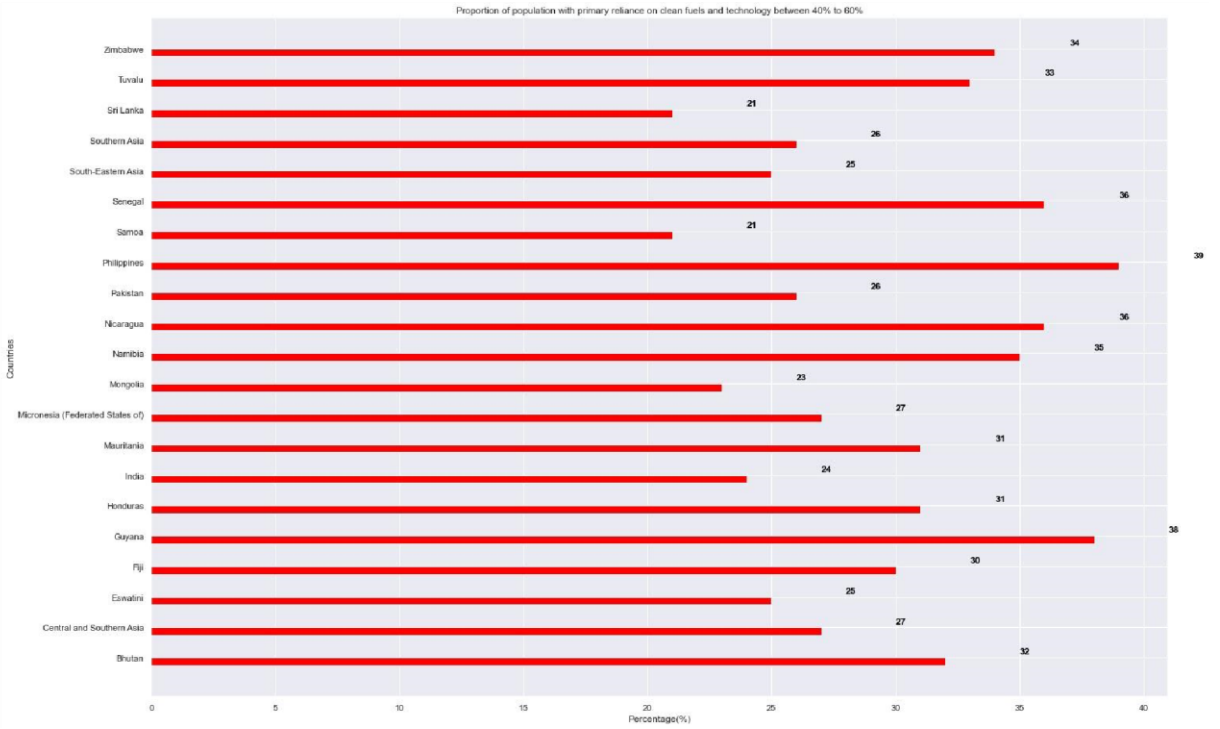
In

[63]:
```
x = el2['GeoAreaName']
y = el2['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

[64]:
```
x = el2['GeoAreaName']
y = el2['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

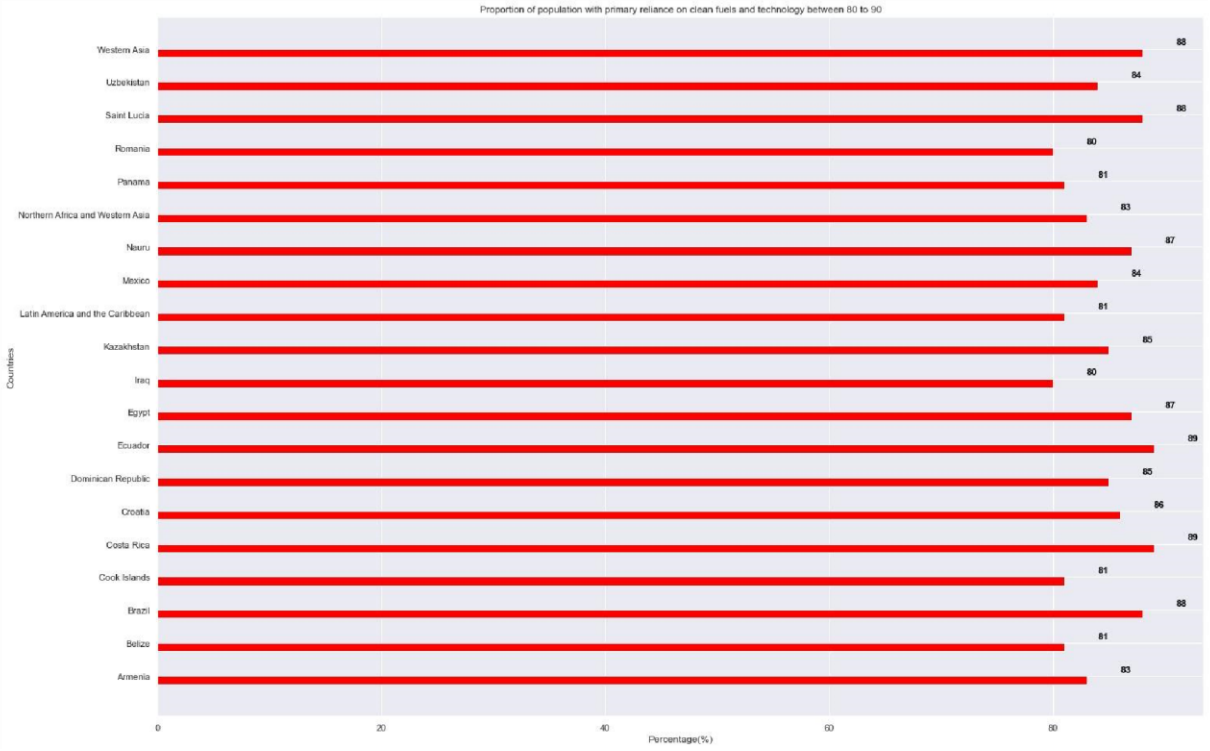
[65]:
```
x = el3['GeoAreaName']
y = el3['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

In

[58]:
```
x = el4['GeoAreaName']
y = el4['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

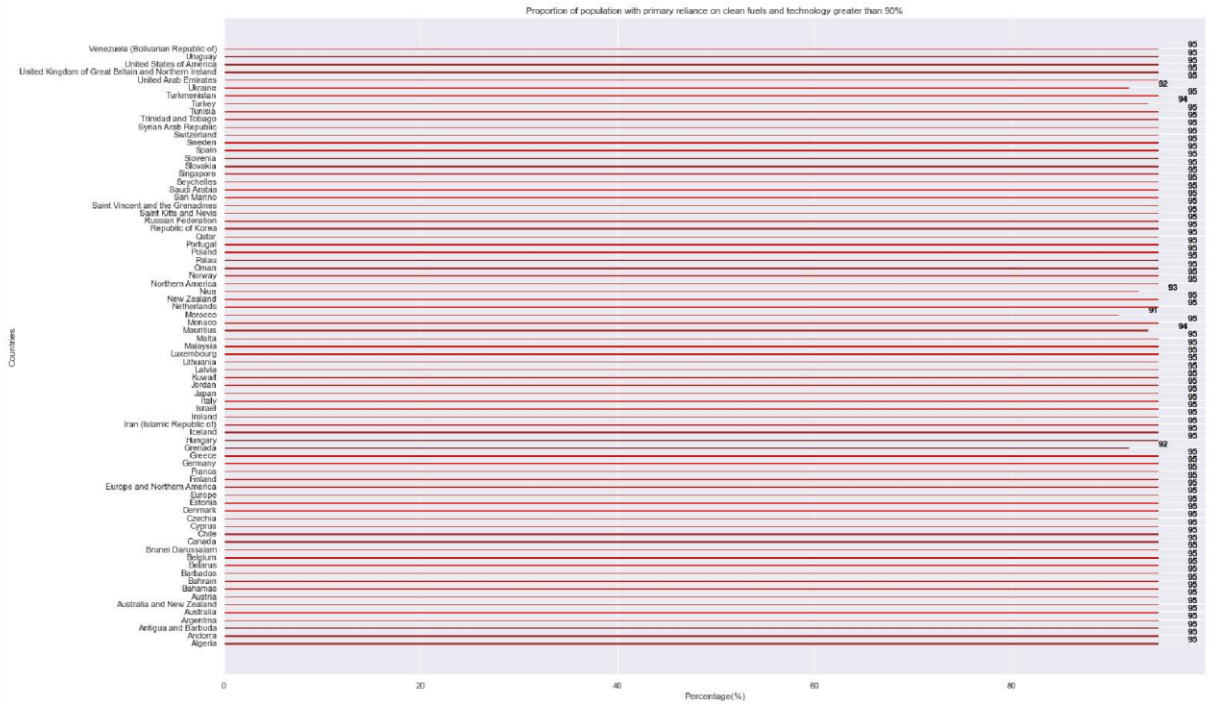
In

[66]:
```
x = el5['GeoAreaName']
y = el5['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



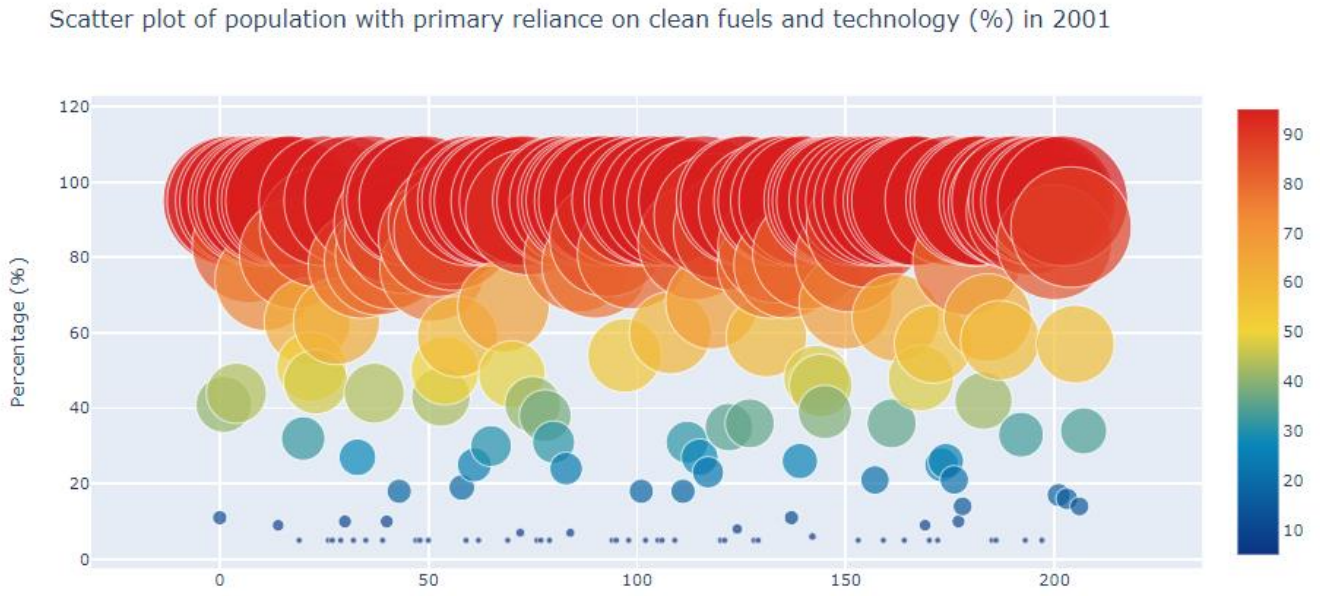
In

[67]:
```
x = el6['GeoAreaName']
y = el6['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



In

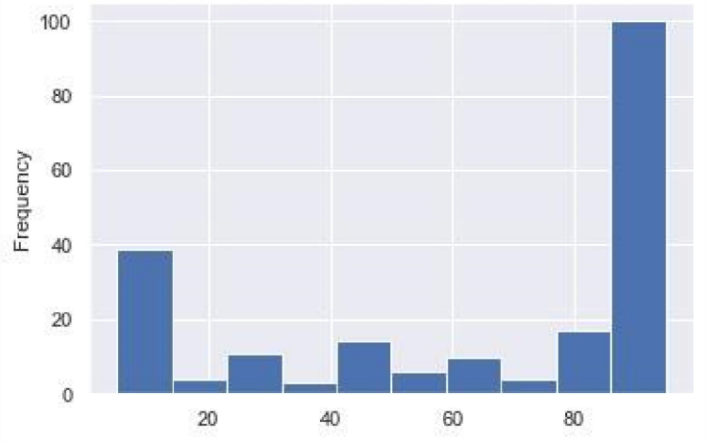
[77]:
```
l=[]
trace0= go.Scatter(
        y= data['2001'],
        mode= 'markers',
        name='Percentage (%)',
        marker= dict(size= data['2001'].values,
                     line= dict(width=1),
                     color= data['2001'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of population with primary reliance on clean fuels and technolo
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Percentage (%)',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```



Graphs For Year 2017

In [81]:
```
data['2017'].plot.hist(bins=10)
```

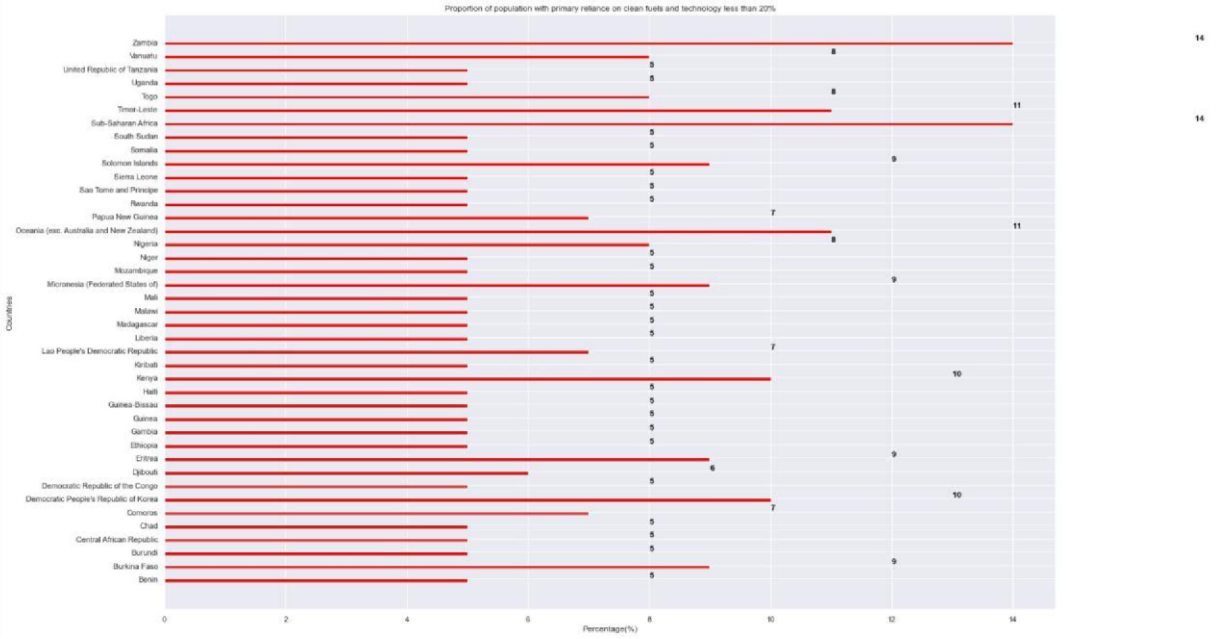
Out[81]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25082e8e040>
```



In [68]:
```
em1=data.loc[data['2017']<20].drop(columns=['2001'])
em2=data.loc[(data['2017']>=20)&(data['2017']<40)].drop(columns=['2001'])
em3=data.loc[(data['2017']>=40)&(data['2017']<60)].drop(columns=['2001'])
em4=data.loc[(data['2017']>=60)&(data['2017']<80)].drop(columns=['2001'])
em5=data.loc[(data['2017']>=80)&(data['2017']<90)].drop(columns=['2001'])
em6=data.loc[(data['2017']>=90)&(data['2017']<100)].drop(columns=['2001'])
```

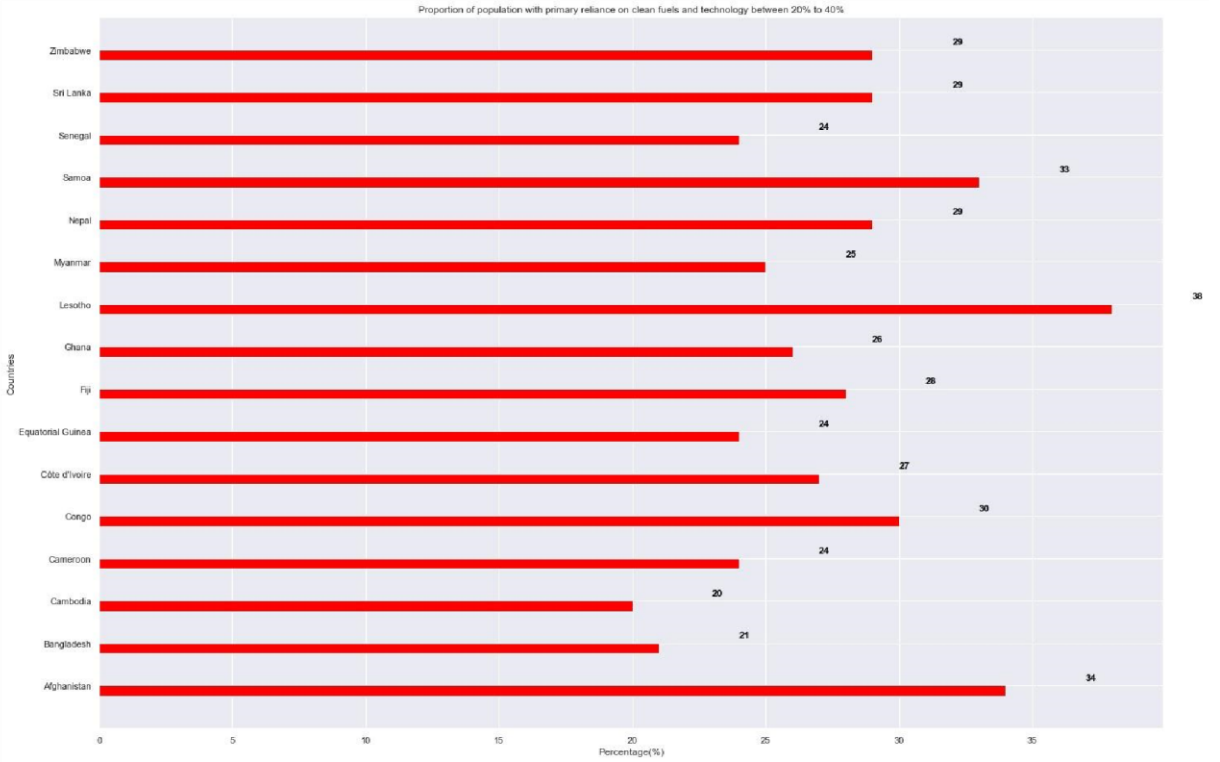
In

[70]:
```
x = em1['GeoAreaName']
y = em1['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



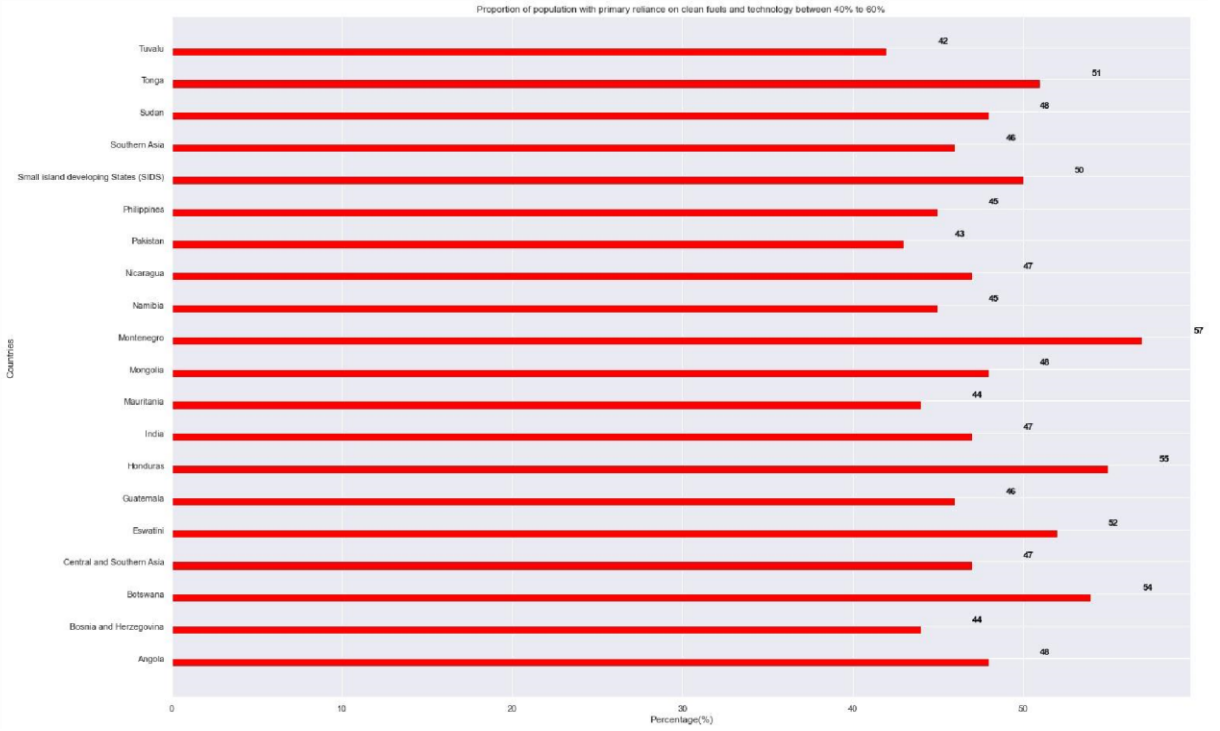
In

[71]:
```
x = em2['GeoAreaName']
y = em2['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



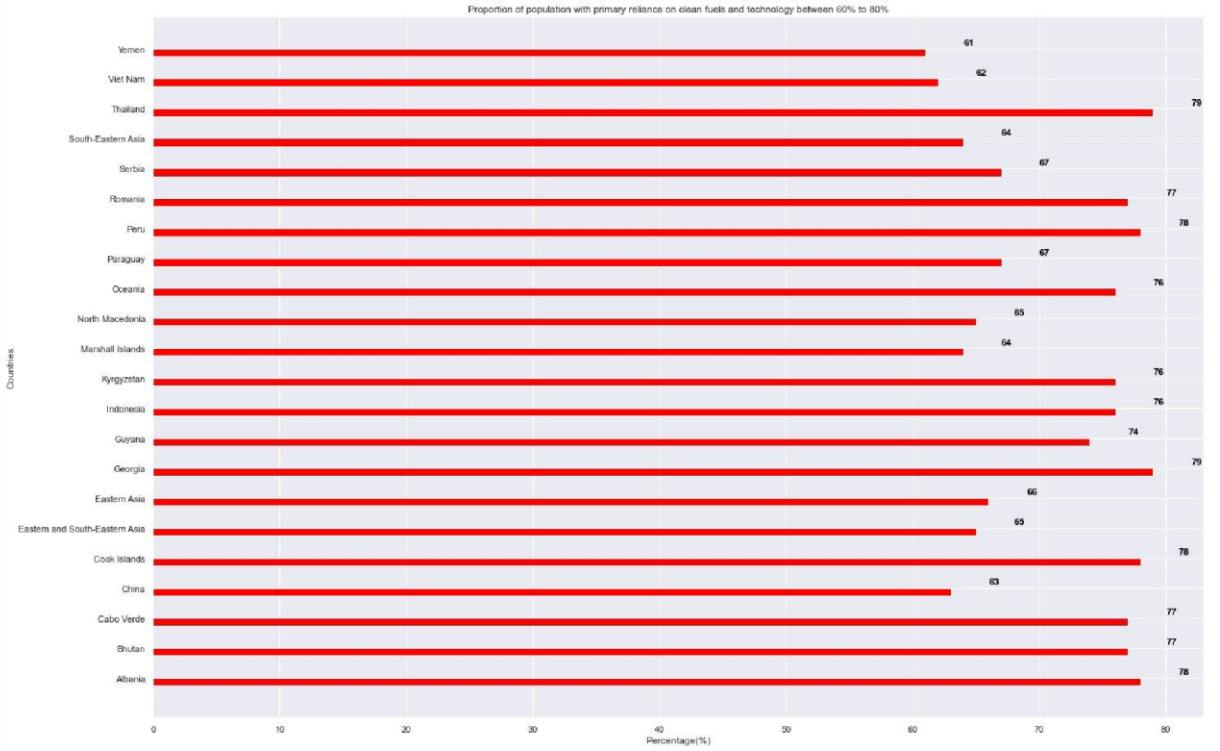
In

[72]:
```
x = em3['GeoAreaName']
y = em3['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



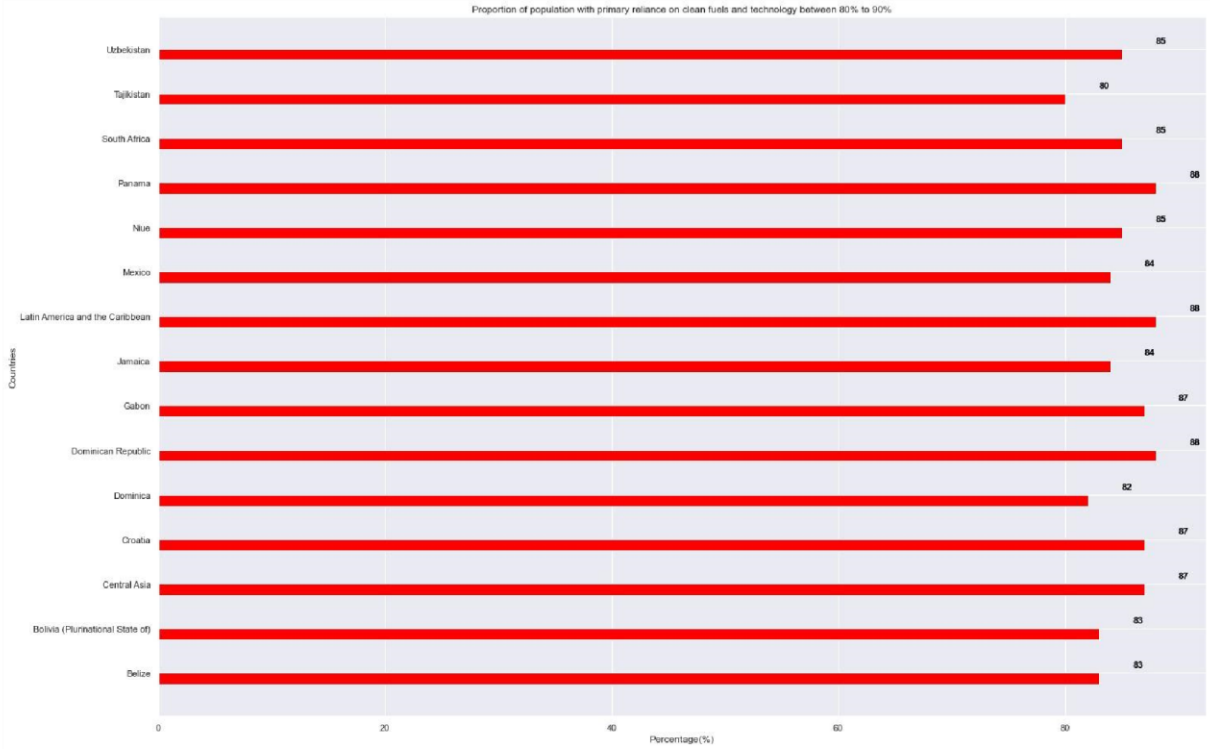
In

[73]:
```
x = em4['GeoAreaName']
y = em4['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



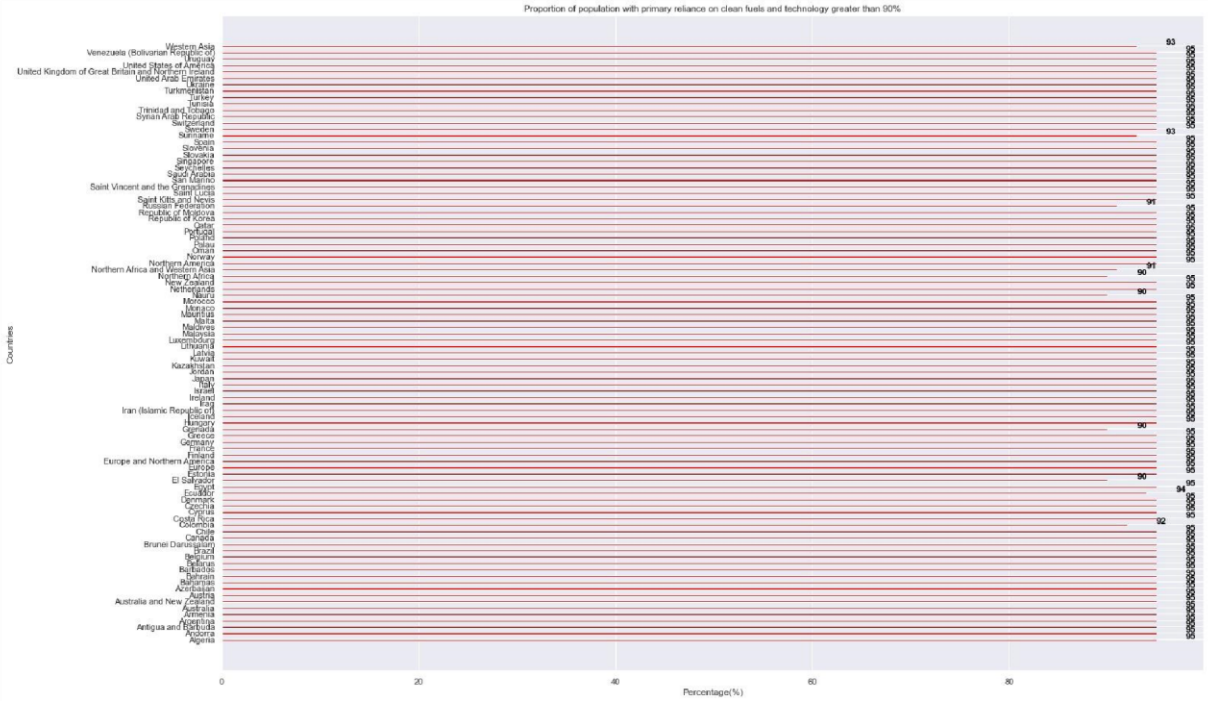
In

[74]:
```
x = em5['GeoAreaName']
y = em5['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



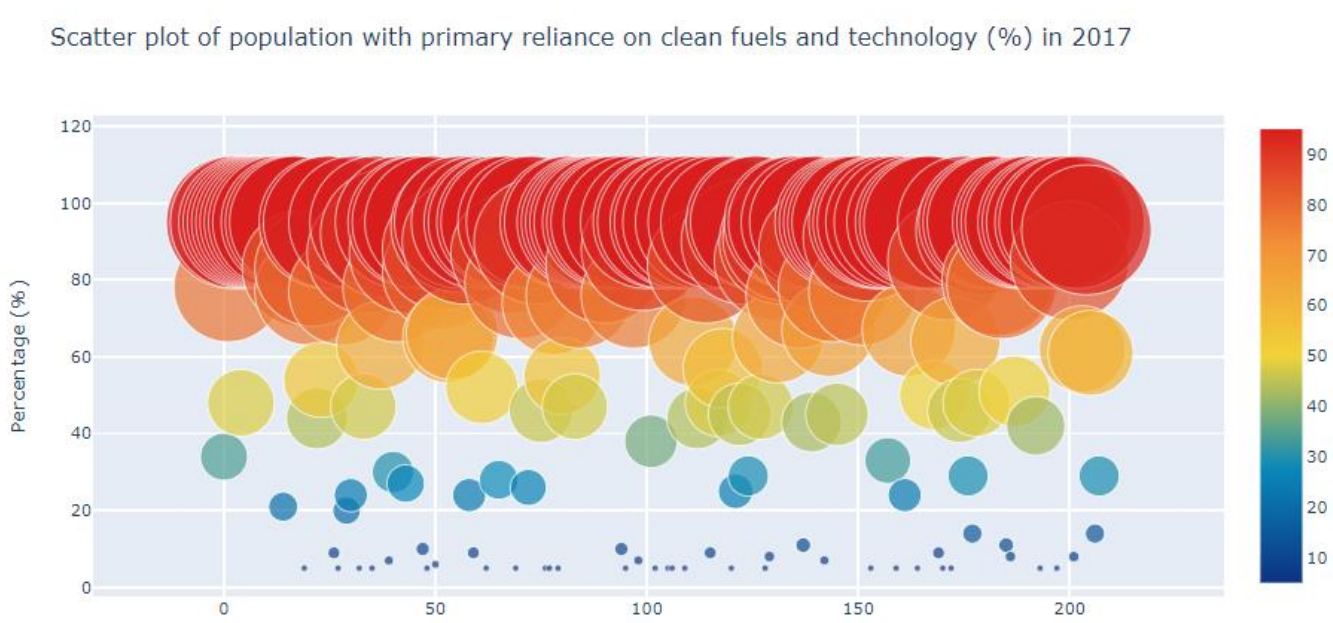
In

[76]:
```
x = em6['GeoAreaName']
y = em6['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Proportion of population with primary reliance on clean fuels and technology
plt.xlabel('Percentage(%)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



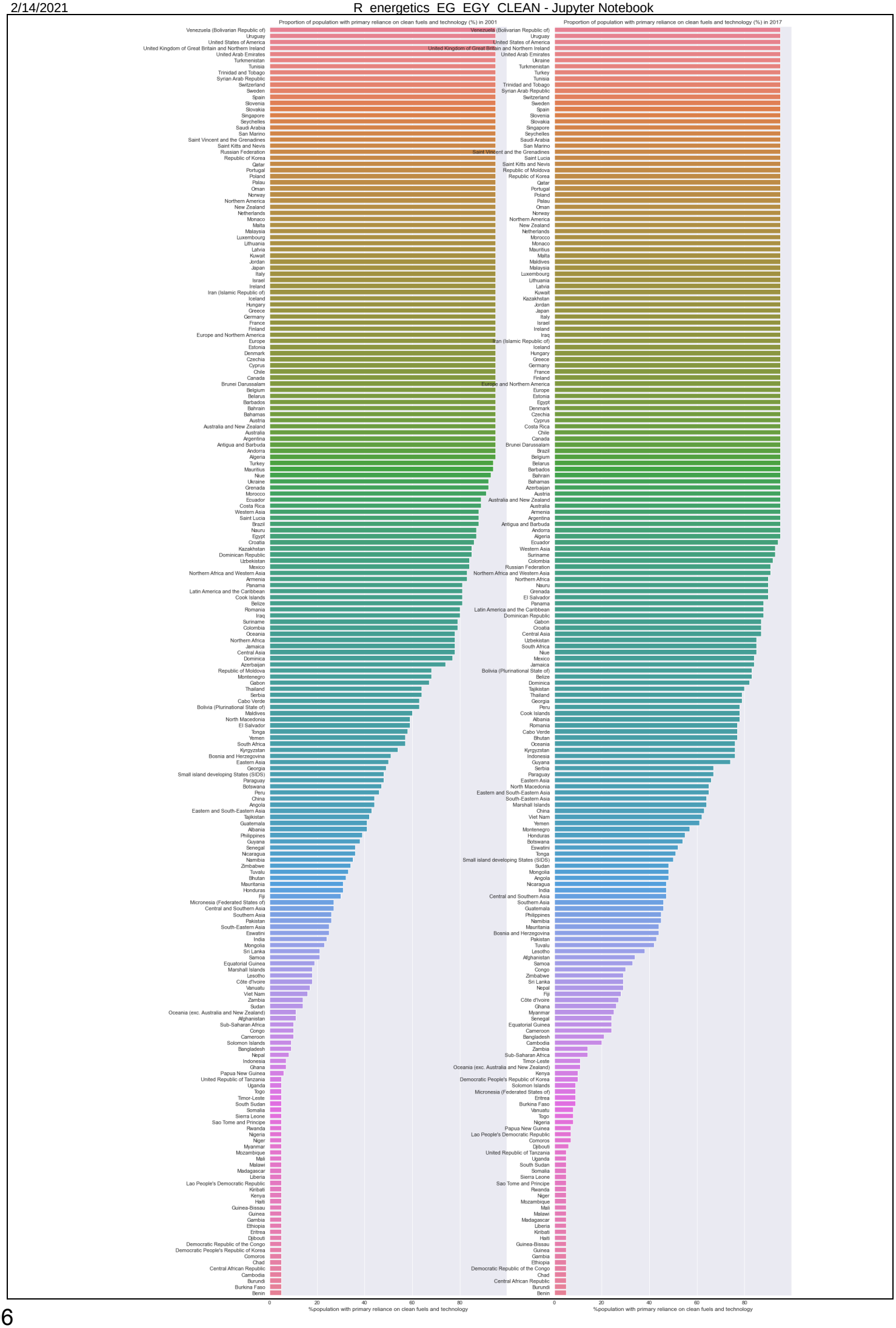
In

[79]:
```
l=[]
trace0= go.Scatter(
        y= data['2017'],
        mode= 'markers',
        name='Percentage (%)',
        marker= dict(size= data['2017'].values,
                     line= dict(width=1),
                     color= data['2017'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of population with primary reliance on clean fuels and technolo
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Percentage (%)',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```



In

47]:
```
# Tried with Plotly now going with seaborn
twoyearchange1_bar, countries_bar1 = (list(x) for x in zip(*sorted(zip(data['2001'], dat
                                                                   reverse = True)))

twoyearchange2_bar, countries_bar2 = (list(x) for x in zip(*sorted(zip(data['2017'], dat
                                                                   reverse = True)))

# Another direct way of sorting according to values is creating distinct sorted datafram
# passing their values directly as in below mentioned code to achieve the same effect as

# df_country_sorted=df_country.sort(columns='2014-2012 change',ascending=False)
# df_country_sorted.head()


sns.set(font_scale=1)
fig, axes = plt.subplots(1,2,figsize=(20, 50))
colorspal = sns.color_palette('husl', len(data['2001']))
sns.barplot(twoyearchange1_bar, countries_bar1, palette = colorspal,ax=axes[0])
sns.barplot(twoyearchange2_bar, countries_bar2, palette = colorspal,ax=axes[1])
axes[0].set(xlabel='%population with primary reliance on clean fuels and technology', ti
axes[1].set(xlabel='%population with primary reliance on clean fuels and technology', ti
fig.savefig('output.png')
```


Proportion of population with primary reliance on clean fuels and technology (%) in 2001
Venezuela (Bolivarian Republic of)
Uruguay
United States of America
United Kingdom of Great Britain and Northern Ireland
United Arab Emirates
Turkmenistan
Tunisia
Trinidad and Tobago
Syrian Arab Republic
Switzerland
Sweden
Spain
Slovenia
Slovakia
Singapore
Seychelles
Saudi Arabia
San Marino
Saint Vincent and the Grenadines
Saint Kitts and Nevis
Russian Federation
Republic of Korea
Qatar
Portugal
Poland
Palau
Oman
Norway
Northern America
New Zealand
Netherlands
Monaco
Malta
Malaysia
Luxembourg
Lithuania
Latvia
Kuwait
Jordan
Japan
Italy
Israel
Ireland
Iran (Islamic Republic of)
Iceland
Hungary
Greece
Germany
France
Finland
Europe and Northern America
Europe
Estonia
Denmark
Czechia
Cyprus
Chile
Canada
Brunei Darussalam
Belgium
Belarus
Barbados
Bahrain
Bahamas
Austria
Australia and New Zealand
Australia
Argentina
Antigua and Barbuda
Andorra
Algeria
Turkey
Mauritius
Niue
Ukraine
Grenada
Morocco
Ecuador
Costa Rica
Western Asia
Saint Lucia
Brazil
Nauru
Egypt
Croatia
Kazakhstan
Dominican Republic
Uzbekistan
Mexico
Northern Africa and Western Asia
Armenia
Panama
Latin America and the Caribbean
Cook Islands
Belize
Romania
Iraq
Suriname
Colombia
Oceania
Northern Africa
Jamaica
Central Asia
Dominica
Azerbaijan
Republic of Moldova
Montenegro
Gabon
Thailand
Serbia
Cabo Verde
Bolivia (Plurinational State of)
Maldives
North Macedonia
El Salvador
Tonga
Yemen
South Africa
Kyrgyzstan
Bosnia and Herzegovina
Eastern Asia
Georgia
Small island developing States (SIDS)
Paraguay
Botswana
Peru
China
Angola
Eastern and South-Eastern Asia
Tajikistan
Guatemala
Albania
Philippines
Guyana
Senegal
Nicaragua
Namibia
Zimbabwe
Tuvalu
Bhutan
Mauritania
Honduras
Fiji
Micronesia (Federated States of)
Central and Southern Asia
Southern Asia
Pakistan
South-Eastern Asia
Eswatini
India
Mongolia
Sri Lanka
Samoa
Equatorial Guinea
Marshall Islands
Lesotho
Côte d'Ivoire
Vanuatu
Viet Nam
Zambia
Sudan
Oceania (exc. Australia and New Zealand)
Afghanistan
Sub-Saharan Africa
Congo
Cameroon
Solomon Islands
Bangladesh
Nepal
Indonesia
Ghana
Papua New Guinea
United Republic of Tanzania
Uganda
Togo
Timor-Leste
South Sudan
Somalia
Sierra Leone
Sao Tome and Principe
Rwanda
Nigeria
Niger
Myanmar
Mozambique
Mali
Malawi
Madagascar
Liberia
Lao People's Democratic Republic
Kiribati
Kenya
Haiti
Guinea-Bissau
Guinea
Gambia
Ethiopia
Eritrea
Djibouti
Democratic Republic of the Congo
Democratic People's Republic of Korea
Comoros
Chad
Central African Republic
Cambodia
Burundi
Burkina Faso
Benin
0
20
40
60
80
%population with primary reliance on clean fuels and technology
Proportion of population with primary reliance on clean fuels and technology (%) in 2017
Venezuela (Bolivarian Republic of)
Uruguay
United States of America
United Kingdom of Great Britain and Northern Ireland
United Arab Emirates
Ukraine
Turkmenistan
Turkey
Tunisia
Trinidad and Tobago
Syrian Arab Republic
Switzerland
Sweden
Spain
Slovenia
Slovakia
Singapore
Seychelles
Saudi Arabia
San Marino
Saint Vincent and the Grenadines
Saint Lucia
Saint Kitts and Nevis
Republic of Moldova
Republic of Korea
Qatar
Portugal
Poland
Palau
Oman
Norway
Northern America
New Zealand
Netherlands
Morocco
Monaco
Mauritius
Malta
Maldives
Malaysia
Luxembourg
Lithuania
Latvia
Kuwait
Kazakhstan
Jordan
Japan
Italy
Israel
Ireland
Iraq
Iran (Islamic Republic of)
Iceland
Hungary
Greece
Germany
France
Finland
Europe and Northern America
Europe
Estonia
Egypt
Denmark
Czechia
Cyprus
Costa Rica
Chile
Canada
Brunei Darussalam
Brazil
Belgium
Belarus
Barbados
Bahrain
Bahamas
Azerbaijan
Austria
Australia and New Zealand
Australia
Armenia
Argentina
Antigua and Barbuda
Andorra
Algeria
Ecuador
Western Asia
Suriname
Colombia
Russian Federation
Northern Africa and Western Asia
Northern Africa
Nauru
Grenada
El Salvador
Panama
Latin America and the Caribbean
Dominican Republic
Gabon
Croatia
Central Asia
Uzbekistan
South Africa
Niue
Mexico
Jamaica
Bolivia (Plurinational State of)
Belize
Dominica
Tajikistan
Thailand
Georgia
Peru
Cook Islands
Albania
Romania
Cabo Verde
Bhutan
Oceania
Kyrgyzstan
Indonesia
Guyana
Serbia
Paraguay
Eastern Asia
North Macedonia
Eastern and South-Eastern Asia
South-Eastern Asia
Marshall Islands
China
Viet Nam
Yemen
Montenegro
Honduras
Botswana
Eswatini
Tonga
Small island developing States (SIDS)
Sudan
Mongolia
Angola
Nicaragua
India
Central and Southern Asia
Southern Asia
Guatemala
Philippines
Namibia
Mauritania
Bosnia and Herzegovina
Pakistan
Tuvalu
Lesotho
Afghanistan
Samoa
Congo
Zimbabwe
Sri Lanka
Nepal
Fiji
Côte d'Ivoire
Ghana
Myanmar
Senegal
Equatorial Guinea
Cameroon
Bangladesh
Cambodia
Zambia
Sub-Saharan Africa
Timor-Leste
Oceania (exc. Australia and New Zealand)
Kenya
Democratic People's Republic of Korea
Solomon Islands
Micronesia (Federated States of)
Eritrea
Burkina Faso
Vanuatu
Togo
Nigeria
Papua New Guinea
Lao People's Democratic Republic
Comoros
Djibouti
United Republic of Tanzania
Uganda
South Sudan
Somalia
Sierra Leone
Sao Tome and Principe
Rwanda
Niger
Mozambique
Mali
Malawi
Madagascar
Liberia
Kiribati
Haiti
Guinea-Bissau
Guinea
Gambia
Ethiopia
Democratic Republic of the Congo
Chad
Central African Republic
Burundi
Benin
0
20
40
60
80
%population with primary reliance on clean fuels and technology

In [49]:
```
x_data = ['2001','2017']

y0 = data['2001']
y1 = data['2017']


y_data = [y0,y1]

colors = ['rgba(93, 164, 214, 0.5)', 'rgba(255, 144, 14, 0.5)']

traces = []

for xd, yd, color in zip(x_data, y_data, colors):
        traces.append(go.Box(
            y=yd,
            name=xd,
            boxpoints='all',
            whiskerwidth=0.2,
            fillcolor=color,
            marker=dict(
                size=2,
            ),
            boxmean=True,
            line=dict(width=1),
        ))

layout = go.Layout(
    title='Distribution of Data',
    xaxis=dict(
        title='Year'
    ),
    yaxis=dict(
        title='Percentage (%)',
        autorange=True,
        showgrid=True,
        zeroline=False,
        dtick=5,
        gridcolor='rgb(255, 255, 255)',
        gridwidth=1,
#         zerolinecolor='rgb(255, 255, 255)',
#         zerolinewidth=2,
    ),
    margin=dict(
        l=40,
        r=30,
        b=80,
        t=100,
    ),
    paper_bgcolor='rgb(243, 243, 243)',
    plot_bgcolor='rgb(243, 243, 243)',
    showlegend=False
)

fig = go.Figure(data=traces, layout=layout)
py.iplot(fig)
```


Distribution of Data
Percentage (%)
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
2001
2017
Year

Regression Equation

In [3]:

```
import pandas as pd
df = pd.read_csv('data1.csv')
df.index=range(1,212)
df
```

Out[3]:

GeoAreaName 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 201

1	Afghanistan 19	20	11 2	11	12	13	14	14	15	16	18
2	Albania 67	39 7	41	43	46	49	52	55	58	62	64
3	Algeria 95	95 9	95	95	95	95	95	95	95	95	95
4	Andorra 95	95 9	95	95	95	95	95	95	95	95	95
5	Angola 47	44 4	44	45	45	46	46	46	46	46	46
...	...	...	...	...	...	...	...	...	...	...	...	...	.
207	Viet Nam 46	49	13 5	16	20	24	28	32	35	39	43
208	Western Asia 92	92	87 9	88	88	89	90	90	91	91	92
209	Yemen 60	56 6	57	57	58	58	58	59	59	59	59
210	Zambia 16	14 1	14	15	15	15	16	16	16	16	16
211	Zimbabwe 31	30	34 3	34	34	33	33	32	32	32	31

211 rows × 20 columns

In [4]:

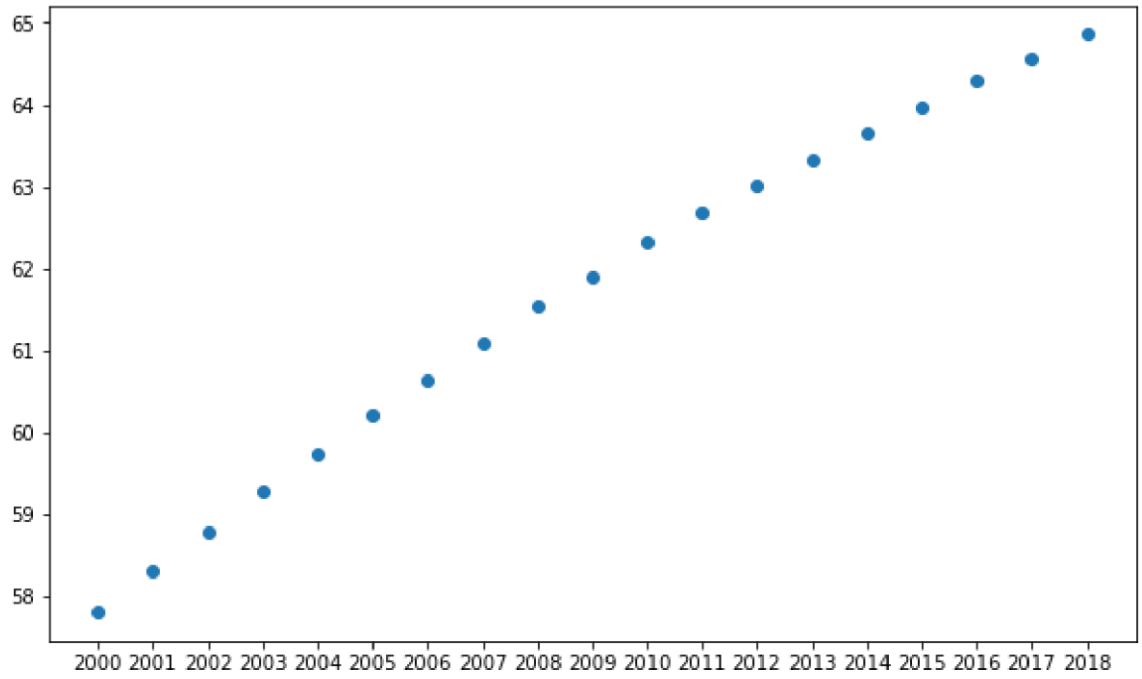
```
mean_year=pd.Series(df[1:].mean())
mean_year
```

Out[4]:

```
2000  57.804762
2001  58.309524
2002  58.780952
2003  59.271429
2004  59.728571
2005  60.214286
2006  60.628571
2007  61.080952
2008  61.542857
2009  61.895238
2010  62.314286
```

```
2011  62.680952
2012  63.023810
2013  63.333333
2014  63.647619
2015  63.957143
2016  64.285714
2017  64.566667 2018    64.857143 dtype: float64 In [16]:
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
plt.scatter(mean_year.index,mean_year)
plt.show()
```



In [17]:

```
table=pd.DataFrame(mean_year)
table.columns=['Y']
```

In [19]:

```
table['X']=range(1,20)
table['XY']=table['Y']*table['X']
table['X^2']=table['X']**2
table=pd.DataFrame(table)
```

In [20]:

```
sum=pd.DataFrame(table.sum())
sum.columns=['Sigma']
sum

mean=pd.DataFrame(table.mean())
mean.columns=['Mean']
mean
```

Out[20]:

	Mean
Y	61.680201
X	10.000000
XY	628.560150
X^2	130.000000

In [21]:

```
table=pd.concat([table,sum.T,mean.T])
```

```
table
```

Out[21]:

	Y	X	XY	X^2
2000	57.804762	1.0	57.804762	1.0
2001	58.309524	2.0	116.619048	4.0
2002	58.780952	3.0	176.342857	9.0
2003	59.271429	4.0	237.085714	16.0
2004	59.728571	5.0	298.642857	25.0
2005	60.214286	6.0	361.285714	36.0
2006	60.628571	7.0	424.400000	49.0
2007	61.080952	8.0	488.647619	64.0
2008	61.542857	9.0	553.885714	81.0
2009	61.895238	10.0	618.952381	100.0
2010	62.314286	11.0	685.457143	121.0
2011	62.680952	12.0	752.171429	144.0
2012	63.023810	13.0	819.309524	169.0
2013	63.333333	14.0	886.666667	196.0
2014	63.647619	15.0	954.714286	225.0
2015	63.957143	16.0	1023.314286	256.0
2016	64.285714	17.0	1092.857143	289.0
2017	64.566667	18.0	1162.200000	324.0
2018	64.857143	19.0	1232.285714	361.0
Sigma	1171.923810	190.0	11942.642857	2470.0
Mean	61.680201	10.0	628.560150	130.0

In [24]:

```
numerator=(table['XY'].loc['Sigma']-table['X'].loc['Sigma']*table['Y'].loc['Sigma']/20)
numerator
```

Out[24]:

```
1.2170927318295715
```

In [11]:

```
denominator = table [ 'X^2'] . loc [ 'Sigma' ] - table[ 'X' ] . loc [ 'Sigma' ] ** 2 /20
denominator
```

Out[11]:

```
665.0
```

In [12]:

```
b=numerator/denominator
b
```

Out[12]:

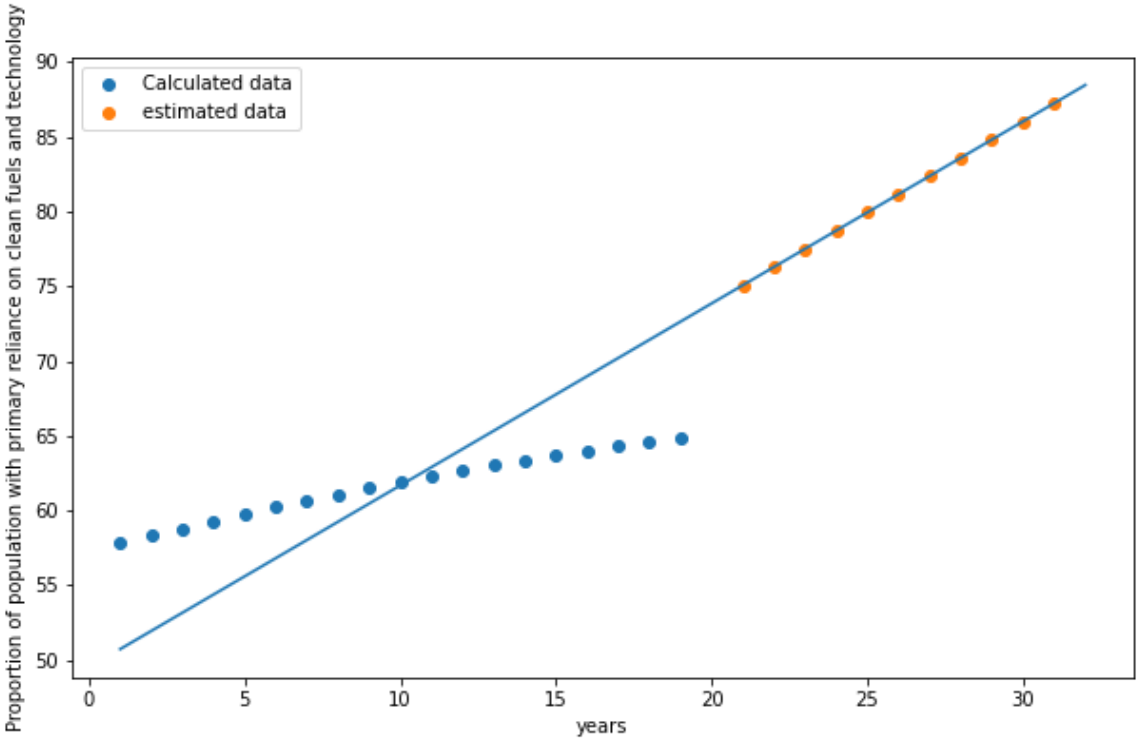
1.2170927318295715 In

```
def eq_regression(x):
    y=table['Y'].loc['Mean']+b*(x-table['X'].loc['Mean'])
    return(y)
```

In [32]:

```
plt.figure(figsize=(10,6))
x=[]
y=[]
for i in range(21,32):
    x.append(i)
    y.append(eq_regression(i))

print("estimated Proportion of population with primary reliance on clean fuels and tech
nology for next 11 years",y)

a=[]
for i in mean_year.index:
    a.append(int(i)-1999)
plt.scatter(a,mean_year,label='Calculated data')
plt.scatter([range(21,32)],y,label='estimated data')
plt.legend()

y_line=[]
x_line=[]
for i in range(1,33):
    x_line.append(i)
    y_line.append(eq_regression(i))
plt.plot(x_line,y_line)
plt.legend()
plt.xlabel("years")
plt.ylabel("Proportion of population with primary reliance on clean fuels and technolog
y")
plt.show()
```

```
estimated Proportion of population with primary reliance on clean fuels an d technology
for next 11 years [75.06822055137842, 76.28531328320798, 77.5 0240601503756,
78.71949874686713, 79.9365914786967, 81.15368421052628, 82. 37077694235585,
83.58786967418541, 84.804962406015, 86.02205513784456, 87. 23914786967413]
```


Calculated data
estimated data
Proportion of population with primary reliance on clean fuels and technology
years
90
85
80
75
70
65
60
55
50
0
5
10
15
20
25
30

In [ ]:

Progress Report of Different Regions

In [7]:
```
import pandas as pd
df=pd.read_csv('data 3(africa).csv')
df
```

Out[7]:

	GeoAreaName	2014	2015	2016	2017	2018
0	Central African Republic	5	5	5	5	5
1	Northern Africa	88	89	89	90	90
2	Northern Africa and Western Asia	91	91	91	91	91
3	South Africa	82	83	84	85	85
4	Sub-Saharan Africa	13	13	14	14	15

In [8]:
```
a=df.iloc[:,1:].to_numpy()
print("Matrix:")
print(a)
```

```
Matrix:
[[ 5  5  5  5  5]
 [88 89 89 90 90]
 [91 91 91 91 91]
 [82 83 84 85 85]
 [13 13 14 14 15]]
```

In [9]:
```
import numpy as np
x0=np.array([[1],[1],[1],[1],[1]])
j=1
#first iteration
x1=np.matmul(a,x0)
previous_eigen_value=np.round(max([abs(i) for i in x1]),4)
x1=x1/previous_eigen_value
print("Step",j,":",'\n',x1,'\n',"Eigen Value : ",previous_eigen_value)
j+=1
#second iteration
x3=np.matmul(a,x1)
present_eigen_value=np.round(max([abs(i) for i in x3]),4)
x3=np.round(x3/present_eigen_value,4)
print("Step",j,":",'\n',x3,'\n',"Eigen Value : ",present_eigen_value)
#while loop
while(previous_eigen_value!=present_eigen_value):#will calculate until previous and present eigen value are same
    j+=1
    previous_eigen_value=present_eigen_value
    x1=np.matmul(a,x3)
    present_eigen_value=np.round(max([abs(i) for i in x1]),4)
    x1=x1/present_eigen_value
    print("Step",j,"\n",x1,'\n',"Eigen Value : ",present_eigen_value)
    x3=x1
```

```
Step 1 :
 [[0.05494505]
 [0.98021978]
 [1.        ]
 [0.92087912]
 [0.15164835]]
 Eigen Value :  [455]
Step 2 :
 [[0.0549]
 [0.9816]
 [1.    ]
 [0.923 ]
 [0.1507]]
 Eigen Value :  [282.8]
```

```
Step 3
 [[0.05494505]
 [0.98162162]
 [1.        ]
 [0.92301439]
 [0.15071643]]
 Eigen Value :  [283.0282]
Step 4
 [[0.05494505]
 [0.98162136]
 [0.9999999 ]
 [0.92301401]
 [0.15071634]]
 Eigen Value :  [283.0371]
Step 5
 [[0.05494505]
 [0.98162144]
 [0.99999998]
 [0.92301408]
 [0.15071635]]
 Eigen Value :  [283.037]
Step 6
 [[0.05494506]
 [0.98162152]
 [1.00000007]
 [0.92301416]
 [0.15071636]]
 Eigen Value :  [283.037]
```

Ranking is determined on the basis of the elements present in the most dominant Eigen Vector

1. Northern Africa and Western Asia (1.00)
2. Northern Africa (0.982)
3. South Africa (0.923)
4. Sub-Saharan Africa (0.150)
5. Central African Republic (0.055)

Objective 4: (Click to view all Objectives)

In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt
import scipy
from collections import OrderedDict
import math
import numpy as np
import io
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [11]:
```
data=pd.read_csv('4 EG_EGY_RNEW.csv',index_col='S.No.')
data
```

Out[11]:

S.No.	GeoAreaName	2001	2017
1	Afghanistan	8.863	9.792
2	Algeria	8.794	16.024
3	American Samoa	0.000	75.085
4	Angola	13.904	81.472
5	Anguilla	0.000	146.764
...	...	...	...
166	Venezuela (Bolivarian Republic of)	535.898	564.462
167	Viet Nam	48.733	192.539
168	Yemen	0.000	3.593
169	Zambia	168.558	144.850
170	Zimbabwe	61.204	62.326

170 rows × 3 columns

[12]:
```
#for calculating mean
def cal_mean(data,column):
  column=str(column)
  count=len(data[column])
  sum=0
  for index,row in data.iterrows():
      sum=sum+row[column]
  u=round(sum/count,2)
  return u
#for calculating standard deviation
def cal_dev(data,column):
  sum=0
  column=str(column)
  count=len(data[column])
  for index,row in data.iterrows():
      diff=round((row[column]-cal_mean(data,column))**2,2)
      sum=sum+diff
  sd=round(math.sqrt(sum/count),2)
  return sd
```

For 2001

UN Claim is 65.202

In [4]:
```
mean_1=cal_mean(data,2001)
dev_1=cal_dev(data,2001)
print('Mean: ',mean_1)
print('Standard Deviation: ',dev_1)
```

```
Mean:  79.26
Standard Deviation:  165.43
```

Forming Hypothesis
Null Hypothesis H0: μ1=65.202
Alternate Hypothesis H1: μ1≠65.202
Two Tail Z - Test

In [5]:
```
def z_test_twotail(claim,samplemean,data,deviation,column):
  column=str(column)
  σ=deviation
  μ=claim
  x=samplemean
  n=len(data[column])
  a=x-μ
  b=σ/((n)**(1/2))
  z=a/b
  if -1.96<z<1.96:
    print("Null Hypothesis Accepted")
  else:
    print("Null Hypothesis Rejected")
  return z
```

In [6]:
```
z_test_twotail(65.202,mean_1,data,dev_1,2001)
```

```
Null Hypothesis Accepted
```

Out[6]: 1.107984614789807

For 2017
UN Claim is 188.01

In [7]:
```
mean_2=cal_mean(data,2017)
dev_2=cal_dev(data,2017)
print('Mean: ',mean_2)
print('Standard Deviation: ',dev_2)
```

```
Mean:  158.19
Standard Deviation:  264.66
```

Forming Hypothesis Null Hypothesis H0: μ1=188.01
Alternate Hypothesis H1: μ1≠188.01
Two Tail Z - Test

In [8]:
```
z_test_twotail(188.01,mean_2,data,dev_2,2017)
```

```
Null Hypothesis Accepted
```

Out[8]: -1.4690744028046772

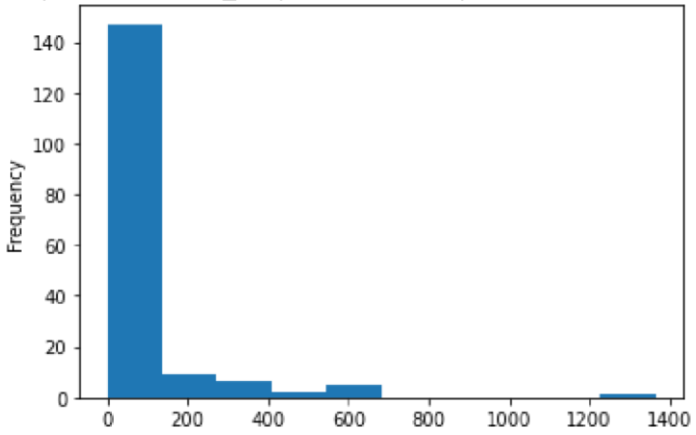
In [9]:
```
data['2001'].corr(data['2017'])
```

Out[9]: 0.7627728949821215

Graphs For Year 2001

In [13]:
```
data['2001'].plot.hist(bins=10)
```

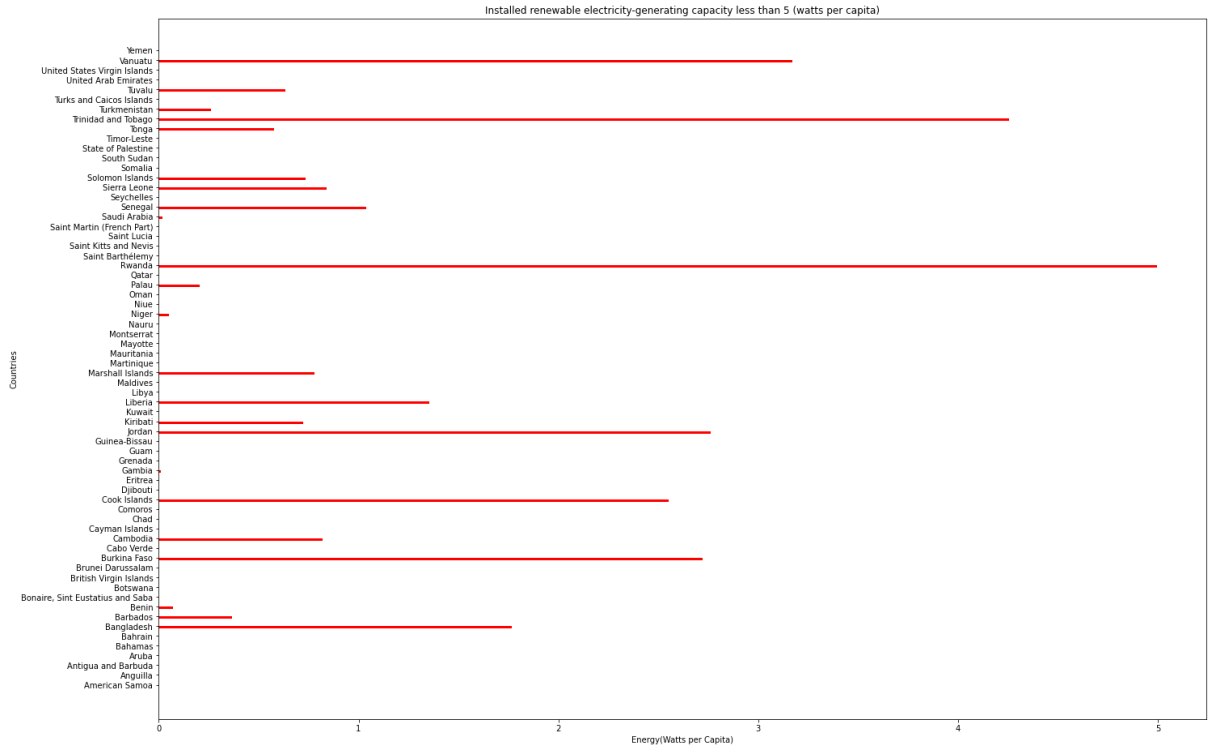
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x24196178b50>



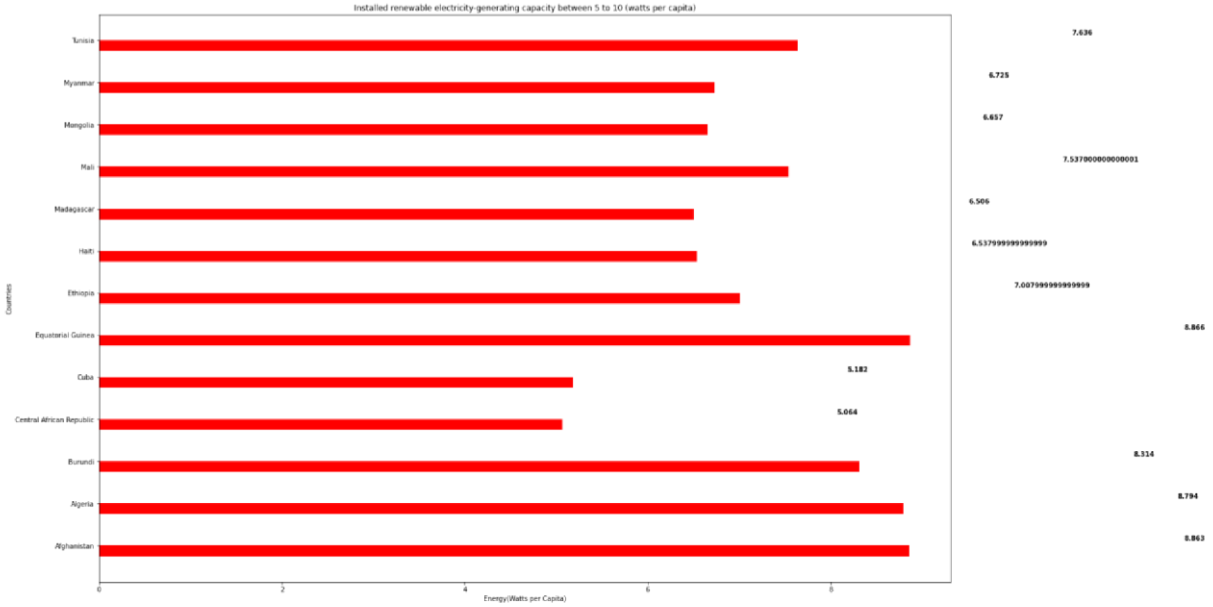
In [44]:
```
ell=data.loc[(data['2001']<5)].drop(columns=['2017'])
el=data.loc[(data['2001']<10)&(data['2001']>=5)].drop(columns=['2017'])
el1=data.loc[(data['2001']<20)&(data['2001']>=10)].drop(columns=['2017'])
el2=data.loc[(data['2001']>=20)&(data['2001']<40)].drop(columns=['2017'])
el3=data.loc[(data['2001']>=40)&(data['2001']<60)].drop(columns=['2017'])
el4=data.loc[(data['2001']>=60)&(data['2001']<80)].drop(columns=['2017'])
el5=data.loc[(data['2001']>=80)&(data['2001']<100)].drop(columns=['2017'])
el6=data.loc[(data['2001']>=100)].drop(columns=['2017'])
```

[32]:
```
x = ell['GeoAreaName']
y = ell['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity less than 5 (watts per ca
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
```

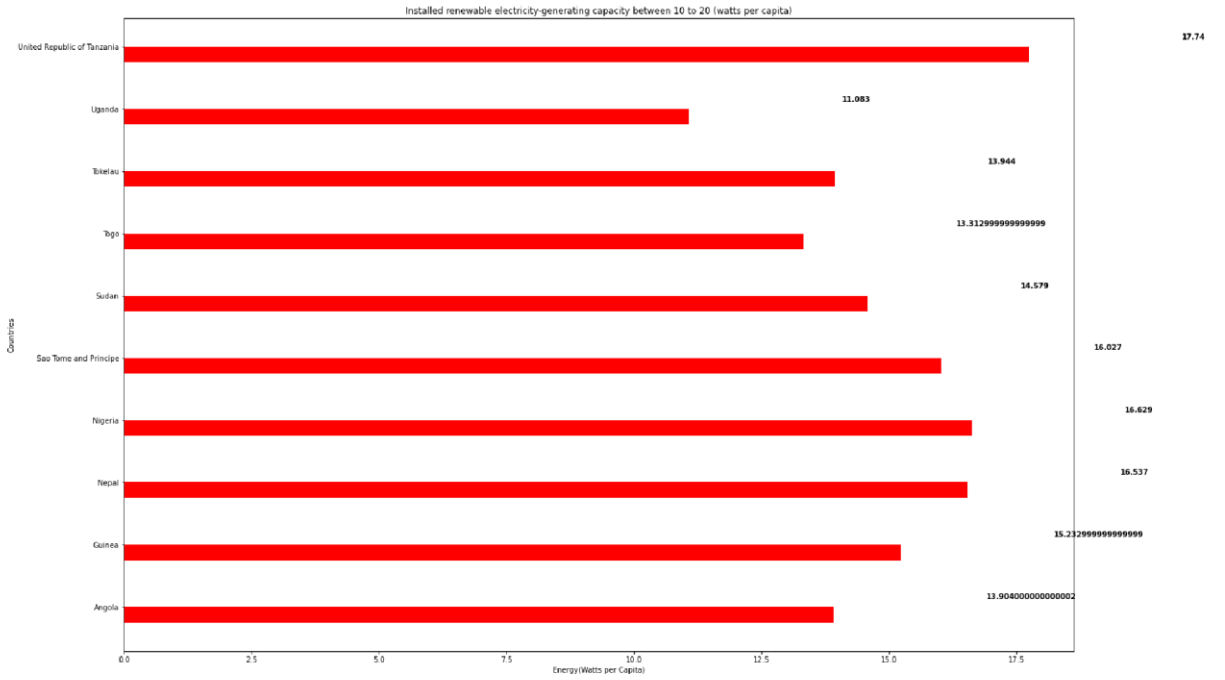
Out[32]: Text(0, 0.5, 'Countries')



[33]:
```
x = el['GeoAreaName']
y = el['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 5 to 10 (watts pe
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

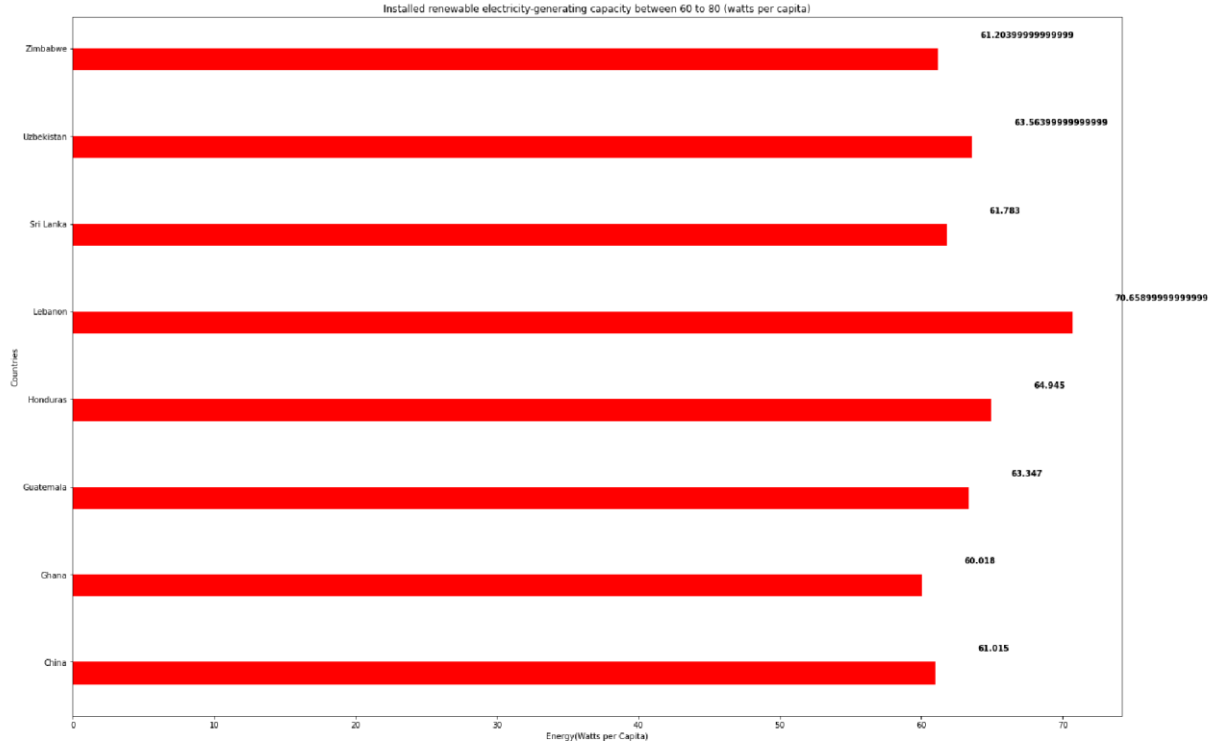


[50]:
```
x = el1['GeoAreaName']
y = el1['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 10 to 20 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

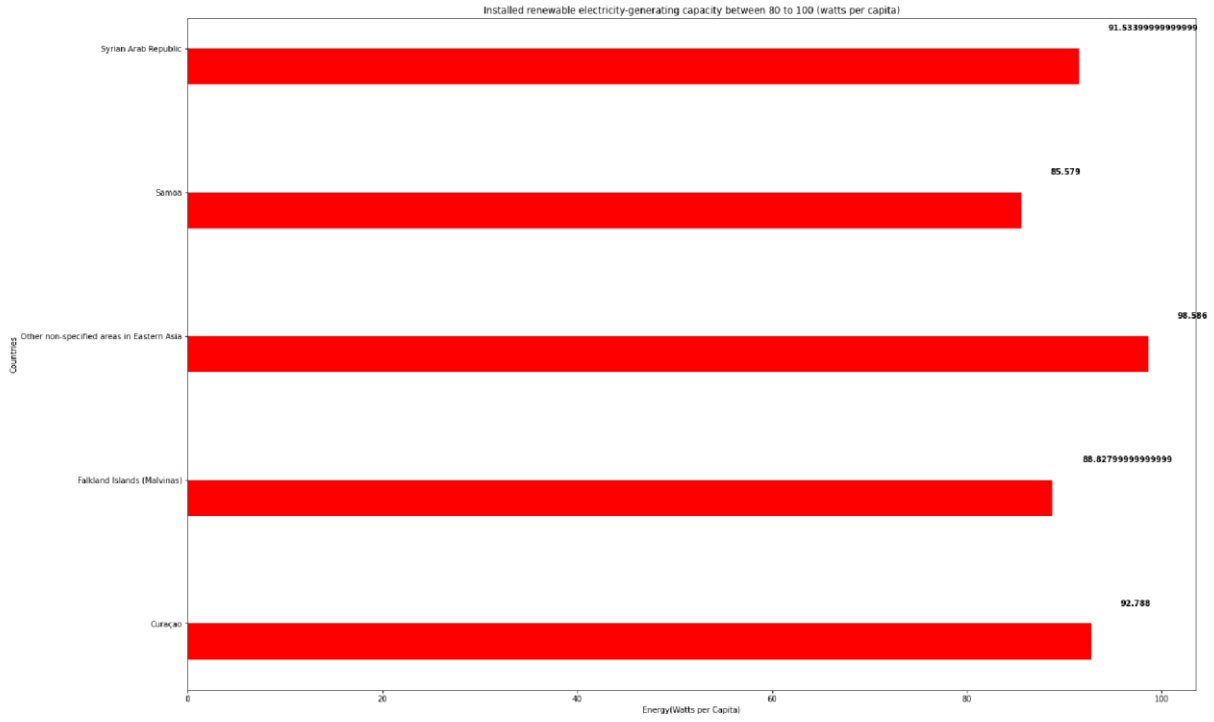


[51]:
```
x = el2['GeoAreaName']
y = el2['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 20 to 40 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

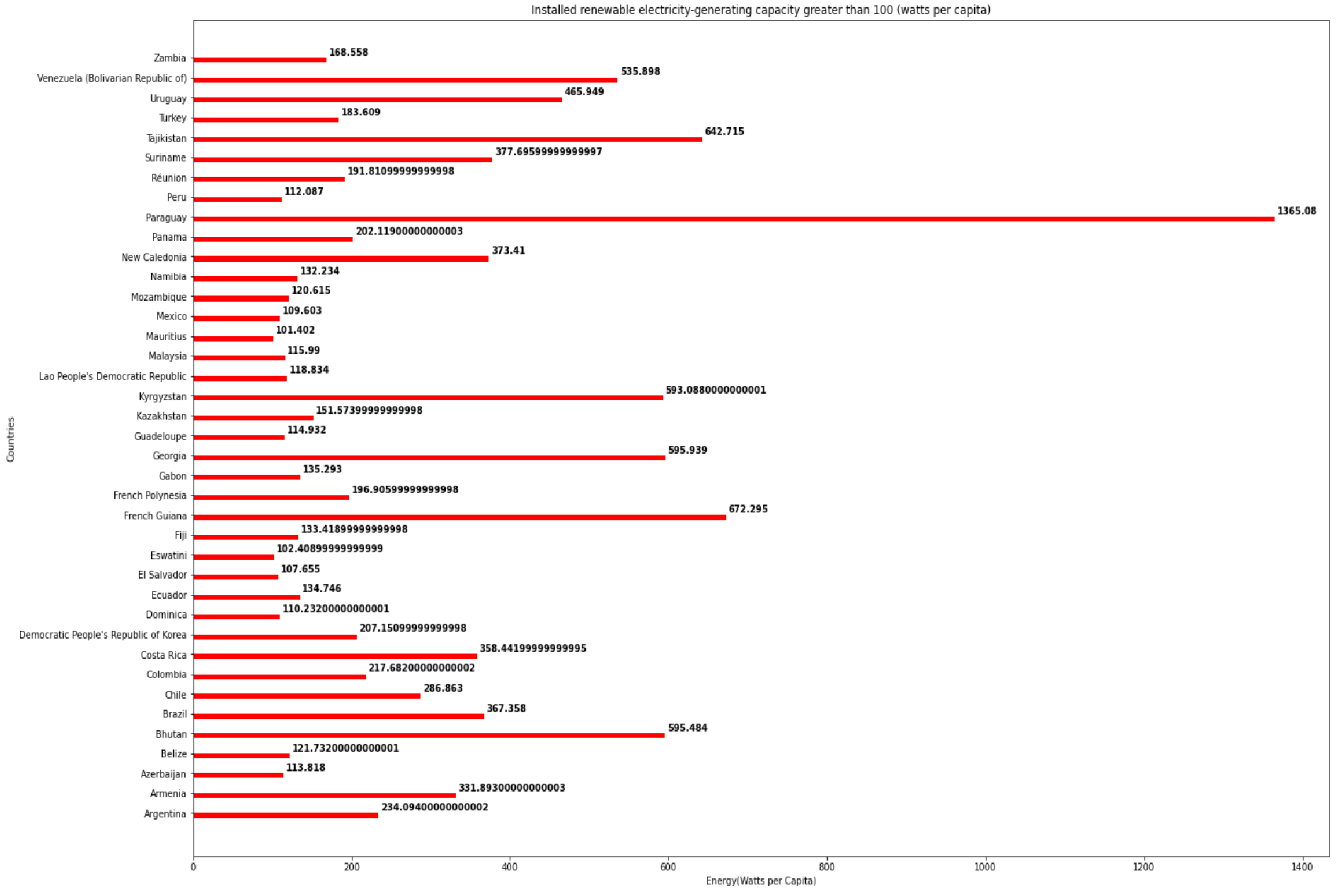
[53]:
```
x = el4['GeoAreaName']
y = el4['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 60 to 80 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



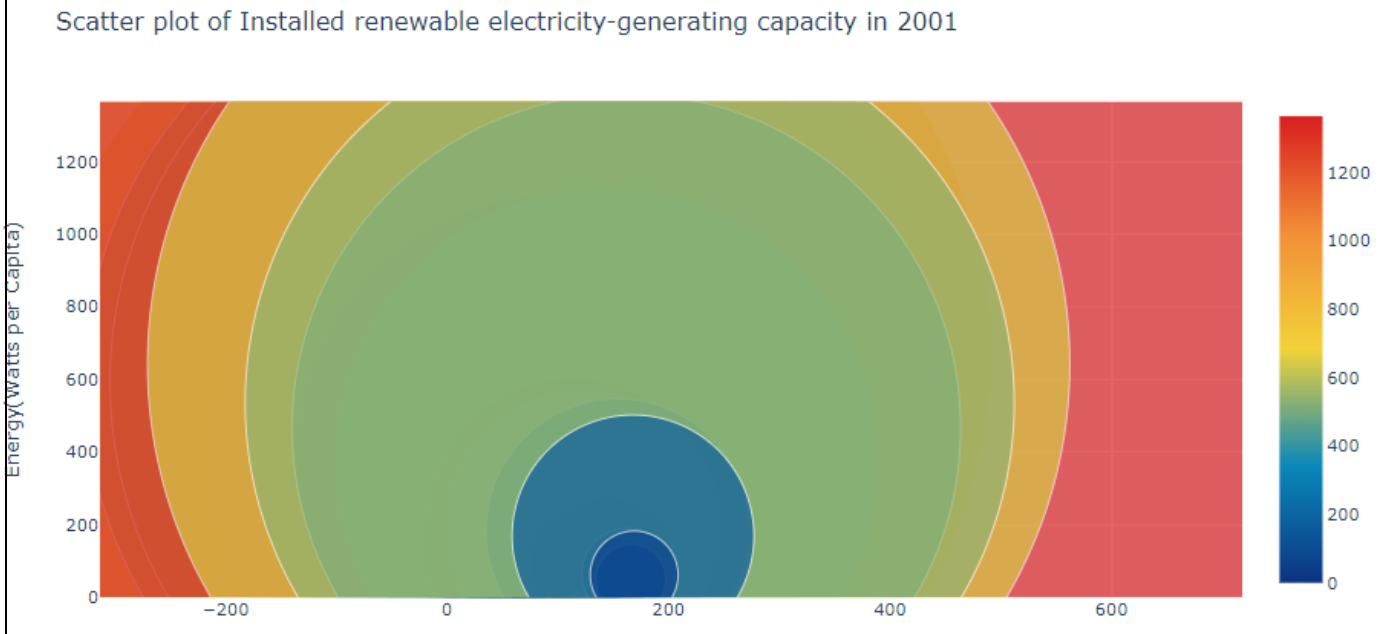
[54]:
```
x = el5['GeoAreaName']
y = el5['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 80 to 100 (watts
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



[55]:
```
x = el6['GeoAreaName']
y = el6['2001']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity greater than 100 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



[56]:
```
l=[]
trace0= go.Scatter(
        y= data['2001'],
        mode= 'markers',
        name='Energy(Watts per Capita)',
        marker= dict(size= data['2001'].values,
                     line= dict(width=1),
                     color= data['2001'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of Installed renewable electricity-generating capacity in 2001',
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Energy(Watts per Capita)',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```



Graphs For Year 2017

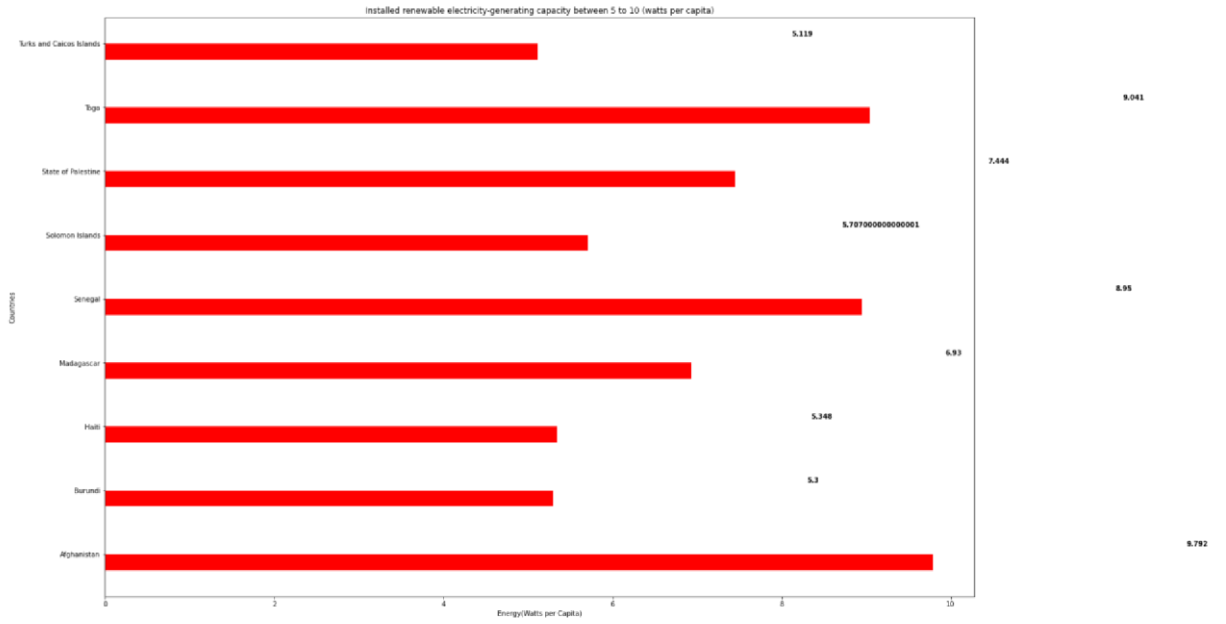
In [57]:
```
eml=data.loc[(data['2017']<5)].drop(columns=['2001'])
em=data.loc[(data['2017']<10)&(data['2017']>=5)].drop(columns=['2001'])
em1=data.loc[(data['2017']<20)&(data['2017']>=10)].drop(columns=['2001'])
em2=data.loc[(data['2017']>=20)&(data['2017']<40)].drop(columns=['2001'])
em3=data.loc[(data['2017']>=40)&(data['2017']<60)].drop(columns=['2001'])
em4=data.loc[(data['2017']>=60)&(data['2017']<80)].drop(columns=['2001'])
em5=data.loc[(data['2017']>=80)&(data['2017']<100)].drop(columns=['2001'])
em6=data.loc[(data['2017']>=100)].drop(columns=['2001'])
```

In [59]:
```
x = eml['GeoAreaName'] y =
eml['2017'] fig= plt.figure() ax =
fig.add_axes([3,3,3,3]) width =
0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the
groups ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2) ax.set_yticklabels(x,
minor=False)
plt.title('Installed renewable electricity-generating capacity less than 5 (watts per
ca plt.xlabel('Energy(Watts per Capita)') plt.ylabel('Countries')
```

Out[59]: Text(0, 0.5, 'Countries')

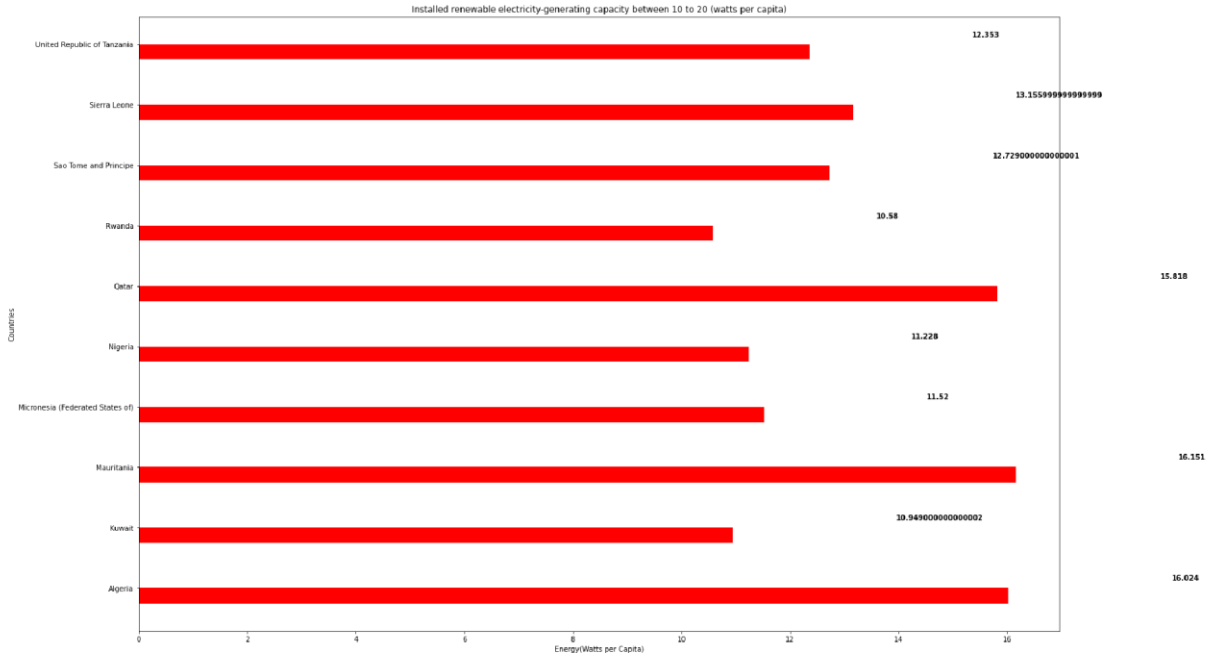
In

[60]:
```
x = em['GeoAreaName']
y = em['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 5 to 10 (watts pe
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



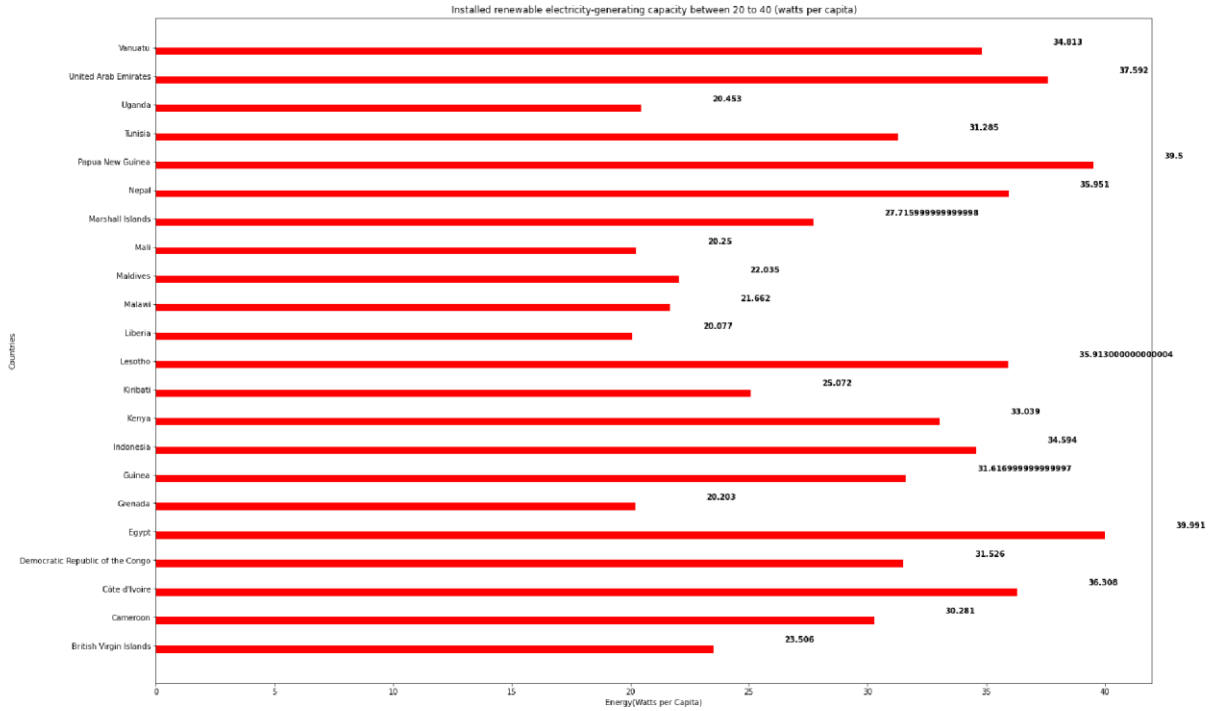
In

[61]:
```
x = em1['GeoAreaName']
y = em1['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 10 to 20 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



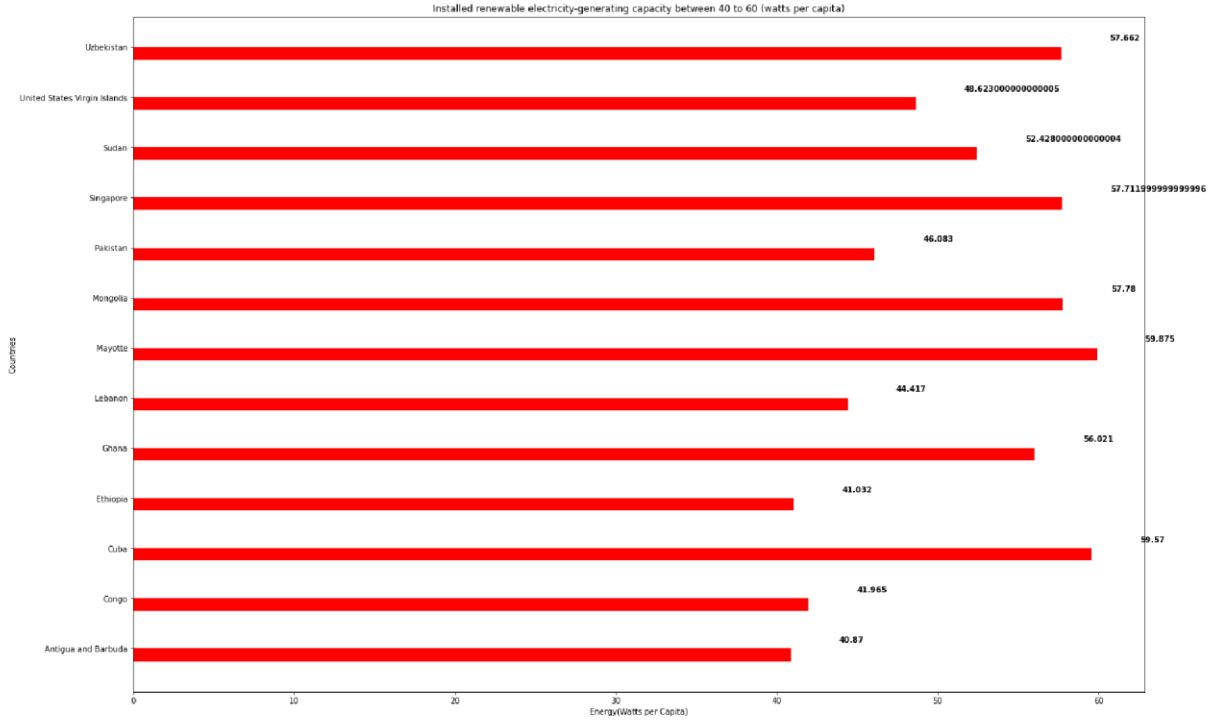
In

[62]:
```
x = em2['GeoAreaName']
y = em2['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 20 to 40 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



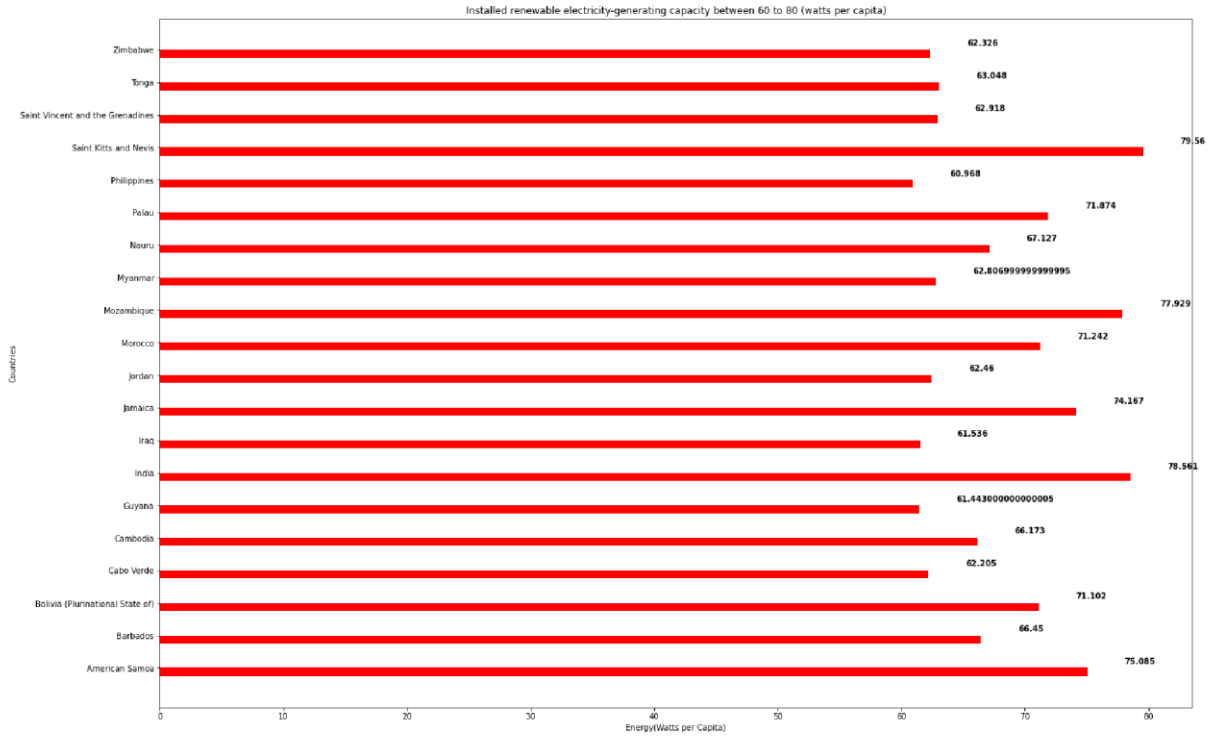
In

[63]:
```
x = em3['GeoAreaName']
y = em3['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 40 to 60 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



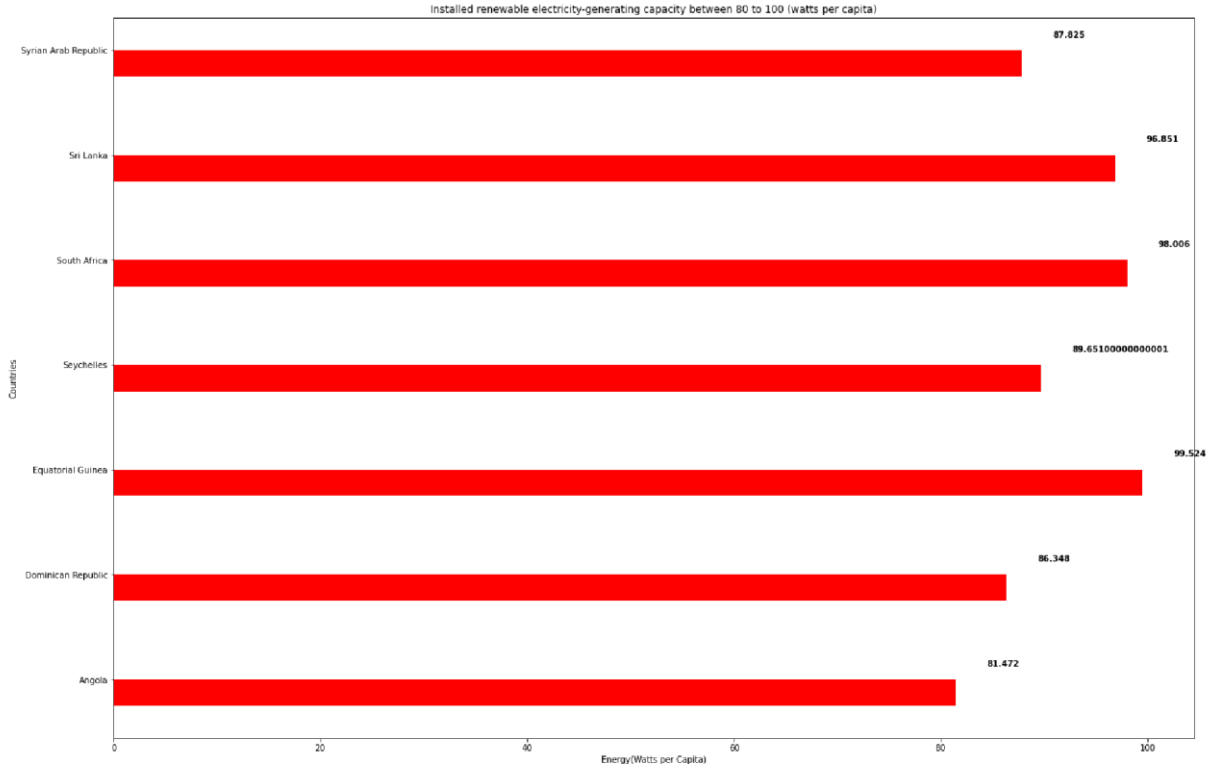
In

[64]:
```
x = em4['GeoAreaName']
y = em4['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 60 to 80 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



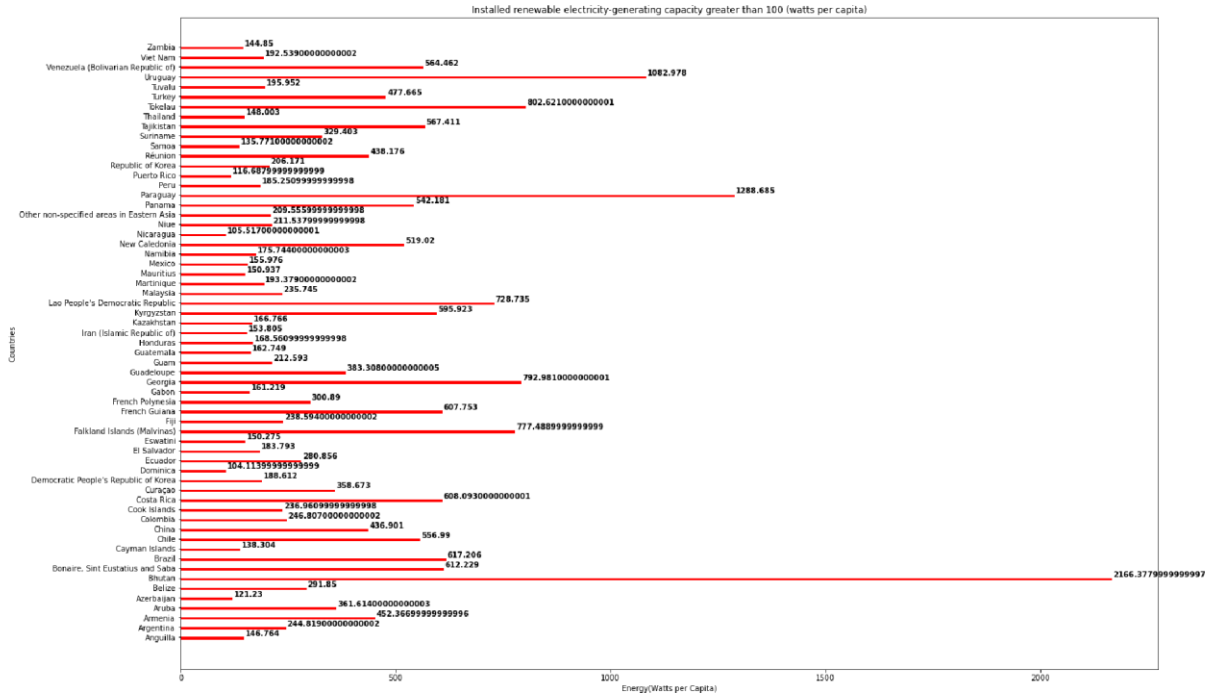
In

[66]:
```
x = em5['GeoAreaName']
y = em5['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity between 80 to 100 (watts
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```



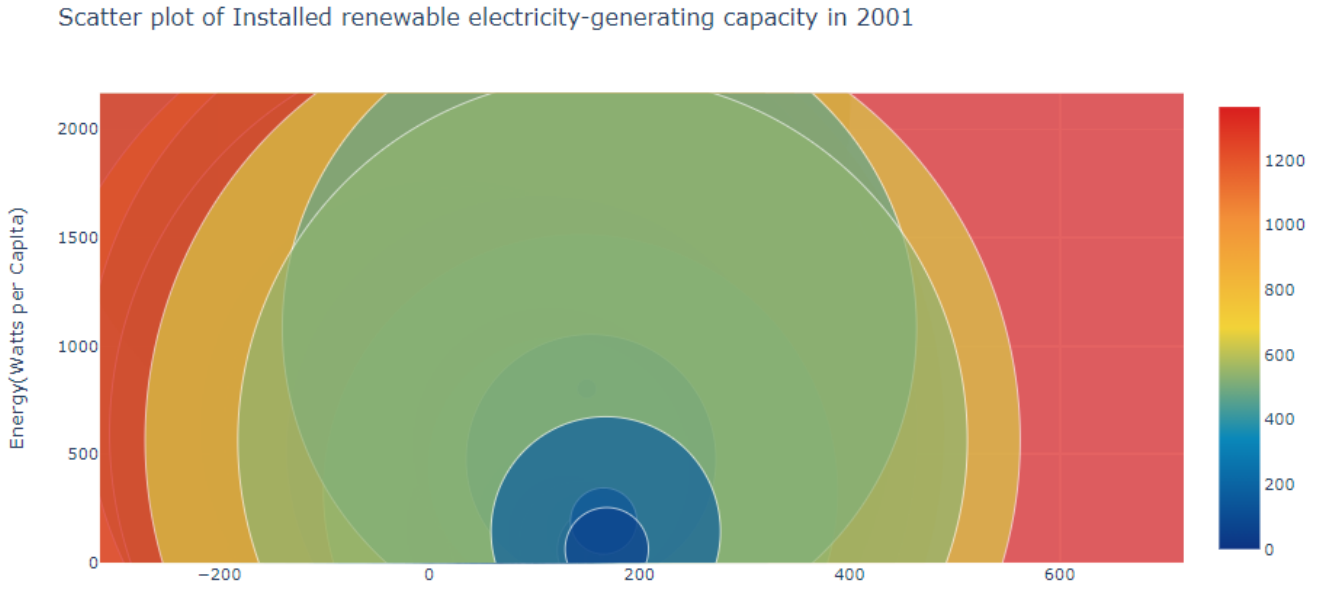
In

[67]:
```
x = em6['GeoAreaName']
y = em6['2017']
fig= plt.figure()
ax = fig.add_axes([3,3,3,3])
width = 0.25 # the width of the bars
ind = np.arange(len(y))  # the x locations for the groups
ax.barh(ind, y, width, color="red")
ax.set_yticks(ind+width/2)
ax.set_yticklabels(x, minor=False)
plt.title('Installed renewable electricity-generating capacity greater than 100 (watts p
plt.xlabel('Energy(Watts per Capita)')
plt.ylabel('Countries')
#plt.show()
for i, v in enumerate(y):
    ax.text(v + 3, i + .25, str(v), color='black', fontweight='bold')
```

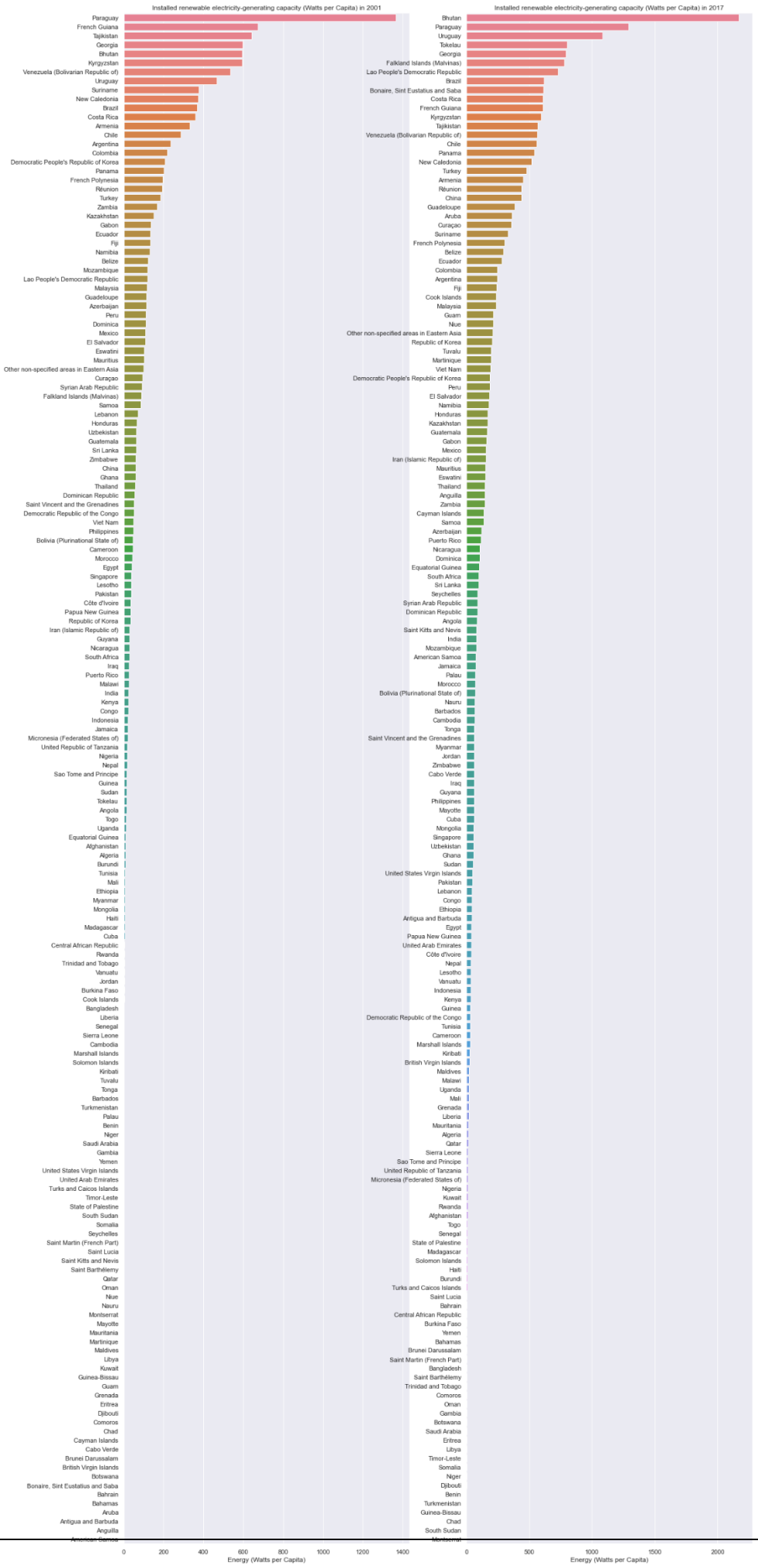


In

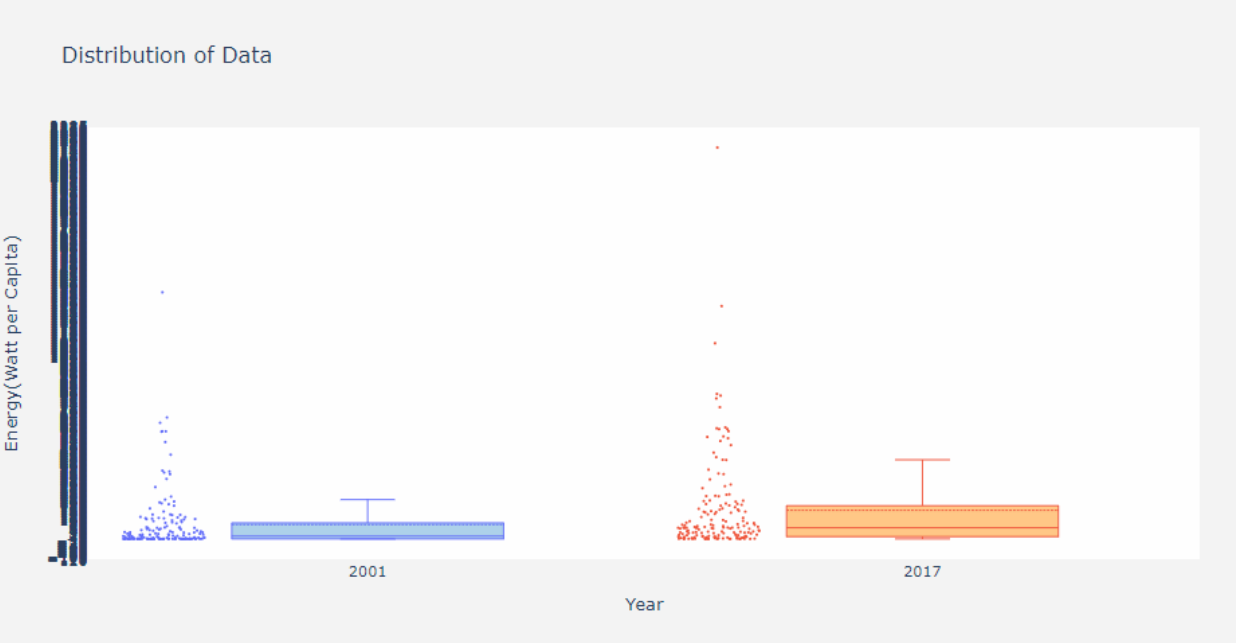
[68]:
```
l=[]
trace0= go.Scatter(
        y= data['2017'],
        mode= 'markers',
        name='Energy(Watts per Capita)',
        marker= dict(size= data['2001'].values,
                     line= dict(width=1),
                     color= data['2001'].values,
                     opacity= 0.7,
                     colorscale='Portland',
                     showscale=True),
        text= data['GeoAreaName'].values) # The hover text goes here...
l.append(trace0);

layout= go.Layout(
    title= 'Scatter plot of Installed renewable electricity-generating capacity in 2001',
    hovermode= 'closest',
    xaxis= dict(
#         title= 'Pop',
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Energy(Watts per Capita)',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False,
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```



[70]:
```
# Tried with Plotly now going with seaborn
twoyearchange1_bar, countries_bar1 = (list(x) for x in zip(*sorted(zip(data['2001'], dat
                                                                   reverse = True)))

twoyearchange2_bar, countries_bar2 = (list(x) for x in zip(*sorted(zip(data['2017'], dat
                                                                   reverse = True)))

# Another direct way of sorting according to values is creating distinct sorted datafram
# passing their values directly as in below mentioned code to achieve the same effect as

# df_country_sorted=df_country.sort(columns='2014-2012 change',ascending=False)
# df_country_sorted.head()


sns.set(font_scale=1)
fig, axes = plt.subplots(1,2,figsize=(20, 50))
colorspal = sns.color_palette('husl', len(data['2001']))
sns.barplot(twoyearchange1_bar, countries_bar1, palette = colorspal,ax=axes[0])
sns.barplot(twoyearchange2_bar, countries_bar2, palette = colorspal,ax=axes[1])
axes[0].set(xlabel='Energy (Watts per Capita)', title='Installed renewable electricity-g
axes[1].set(xlabel='Energy (Watts per Capita)', title='Installed renewable electricity-g
fig.savefig('output.png')
```


Installed renewable electricity-generating capacity (Watts per Capita) in 2001
Paraguay
French Guiana
Tajikistan
Georgia
Bhutan
Kyrgyzstan
Venezuela (Bolivarian Republic of)
Uruguay
Suriname
New Caledonia
Brazil
Costa Rica
Armenia
Chile
Argentina
Colombia
Democratic People's Republic of Korea
Panama
French Polynesia
Réunion
Turkey
Zambia
Kazakhstan
Gabon
Ecuador
Fiji
Namibia
Belize
Mozambique
Lao People's Democratic Republic
Malaysia
Guadeloupe
Azerbaijan
Peru
Dominica
Mexico
El Salvador
Eswatini
Mauritius
Other non-specified areas in Eastern Asia
Curaçao
Syrian Arab Republic
Falkland Islands (Malvinas)
Samoa
Lebanon
Honduras
Uzbekistan
Guatemala
Sri Lanka
Zimbabwe
China
Ghana
Thailand
Dominican Republic
Saint Vincent and the Grenadines
Democratic Republic of the Congo
Viet Nam
Philippines
Bolivia (Plurinational State of)
Cameroon
Morocco
Egypt
Singapore
Lesotho
Pakistan
Côte d'Ivoire
Papua New Guinea
Republic of Korea
Iran (Islamic Republic of)
Guyana
Nicaragua
South Africa
Iraq
Puerto Rico
Malawi
India
Kenya
Congo
Indonesia
Jamaica
Micronesia (Federated States of)
United Republic of Tanzania
Nigeria
Nepal
Sao Tome and Principe
Guinea
Sudan
Tokelau
Angola
Togo
Uganda
Equatorial Guinea
Afghanistan
Algeria
Burundi
Tunisia
Mali
Ethiopia
Myanmar
Mongolia
Haiti
Madagascar
Cuba
Central African Republic
Rwanda
Trinidad and Tobago
Vanuatu
Jordan
Burkina Faso
Cook Islands
Bangladesh
Liberia
Senegal
Sierra Leone
Cambodia
Marshall Islands
Solomon Islands
Kiribati
Tuvalu
Tonga
Barbados
Turkmenistan
Palau
Benin
Niger
Saudi Arabia
Gambia
Yemen
United States Virgin Islands
United Arab Emirates
Turks and Caicos Islands
Timor-Leste
State of Palestine
South Sudan
Somalia
Seychelles
Saint Martin (French Part)
Saint Lucia
Saint Kitts and Nevis
Saint Barthélemy
Qatar
Oman
Niue
Nauru
Montserrat
Mayotte
Mauritania
Martinique
Maldives
Libya
Kuwait
Guinea-Bissau
Guam
Grenada
Eritrea
Djibouti
Comoros
Chad
Cayman Islands
Cabo Verde
Brunei Darussalam
British Virgin Islands
Botswana
Bonaire, Sint Eustatius and Saba
Bahrain
Bahamas
Aruba
Antigua and Barbuda
Anguilla
American Samoa
0
200
400
600
800
1000
1200
1400
Energy (Watts per Capita)
Installed renewable electricity-generating capacity (Watts per Capita) in 2017
Bhutan
Paraguay
Uruguay
Tokelau
Georgia
Falkland Islands (Malvinas)
Lao People's Democratic Republic
Brazil
Bonaire, Sint Eustatius and Saba
Costa Rica
French Guiana
Kyrgyzstan
Tajikistan
Venezuela (Bolivarian Republic of)
Chile
Panama
New Caledonia
Turkey
Armenia
Réunion
China
Guadeloupe
Aruba
Curaçao
Suriname
French Polynesia
Belize
Ecuador
Colombia
Argentina
Fiji
Cook Islands
Malaysia
Guam
Niue
Other non-specified areas in Eastern Asia
Republic of Korea
Tuvalu
Martinique
Viet Nam
Democratic People's Republic of Korea
Peru
El Salvador
Namibia
Honduras
Kazakhstan
Guatemala
Gabon
Mexico
Iran (Islamic Republic of)
Mauritius
Eswatini
Thailand
Anguilla
Zambia
Cayman Islands
Samoa
Azerbaijan
Puerto Rico
Nicaragua
Dominica
Equatorial Guinea
South Africa
Sri Lanka
Seychelles
Syrian Arab Republic
Dominican Republic
Angola
Saint Kitts and Nevis
India
Mozambique
American Samoa
Jamaica
Palau
Morocco
Bolivia (Plurinational State of)
Nauru
Barbados
Cambodia
Tonga
Saint Vincent and the Grenadines
Myanmar
Jordan
Zimbabwe
Cabo Verde
Iraq
Guyana
Philippines
Mayotte
Cuba
Mongolia
Singapore
Uzbekistan
Ghana
Sudan
United States Virgin Islands
Pakistan
Lebanon
Congo
Ethiopia
Antigua and Barbuda
Egypt
Papua New Guinea
United Arab Emirates
Côte d'Ivoire
Nepal
Lesotho
Vanuatu
Indonesia
Kenya
Guinea
Democratic Republic of the Congo
Tunisia
Cameroon
Marshall Islands
Kiribati
British Virgin Islands
Maldives
Malawi
Uganda
Mali
Grenada
Liberia
Mauritania
Algeria
Qatar
Sierra Leone
Sao Tome and Principe
United Republic of Tanzania
Micronesia (Federated States of)
Nigeria
Kuwait
Rwanda
Afghanistan
Togo
Senegal
State of Palestine
Madagascar
Solomon Islands
Haiti
Burundi
Turks and Caicos Islands
Saint Lucia
Bahrain
Central African Republic
Burkina Faso
Yemen
Bahamas
Brunei Darussalam
Saint Martin (French Part)
Bangladesh
Saint Barthélemy
Trinidad and Tobago
Comoros
Oman
Gambia
Botswana
Saudi Arabia
Eritrea
Libya
Timor-Leste
Somalia
Niger
Djibouti
Benin
Turkmenistan
Guinea-Bissau
Chad
South Sudan
Montserrat
0
500
1000
1500
2000
Energy (Watts per Capita)

[72]:
```
x_data = ['2001','2017']

y0 = data['2001']
y1 = data['2017']


y_data = [y0,y1]

colors = ['rgba(93, 164, 214, 0.5)', 'rgba(255, 144, 14, 0.5)']

traces = []

for xd, yd, color in zip(x_data, y_data, colors):
        traces.append(go.Box(
            y=yd,
            name=xd,
            boxpoints='all',
            whiskerwidth=0.2,
            fillcolor=color,
            marker=dict(
                size=2,
            ),
            boxmean=True,
            line=dict(width=1),
        ))

layout = go.Layout(
    title='Distribution of Data',
    xaxis=dict(
        title='Year'
    ),
    yaxis=dict(
        title='Energy(Watt per Capita)',
        autorange=True,
        showgrid=True,
        zeroline=False,
        dtick=5,
        gridcolor='rgb(255, 255, 255)',
        gridwidth=1,
#         zerolinecolor='rgb(255, 255, 255)',
#         zerolinewidth=2,
    ),
    margin=dict(
        l=40,
        r=30,
        b=80,
        t=100,
    ),
    paper_bgcolor='rgb(243, 243, 243)',
    plot_bgcolor='rgb(243, 243, 243)',
    showlegend=False
)

fig = go.Figure(data=traces, layout=layout)
py.iplot(fig)
```


Distribution of Data
Energy(Watt per Capita)
2001
2017
Year

Regression Equation

In [1]:

```
import pandas as pd
data=pd.read_csv('implant.csv')
```

```
data
```

Out[1]:

	GeoAreaName	2000	2001	2002	2003	2004	2005	2006	2007
0	Afghanistan	9.216	8.863	8.473	8.092	7.753	7.508	7.402	7.249
1	Algeria	8.910	8.794	8.683	8.573	8.461	8.344	7.420	7.305
	American								
2	Samoa	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	Angola	14.369	13.904	13.457	13.014	26.442	25.533	24.634	36.270
4	Anguilla	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
...	...	...	...	...	...	...	...	...	...
336	Viet Nam	42.543	48.733	52.056	52.204	52.233	52.423	54.796	59.000
337	Yemen	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
338	Zambia	173.029	168.558	164.264	154.775	150.836	146.948	143.119	134.609
339	Zimbabwe	61.423	61.204	62.722	62.576	62.379	62.086	61.684	61.178
340	World	64.497	65.202	65.926	68.811	72.044	75.124	78.587	83.028
341	rows × 20 columns								

In [2]:

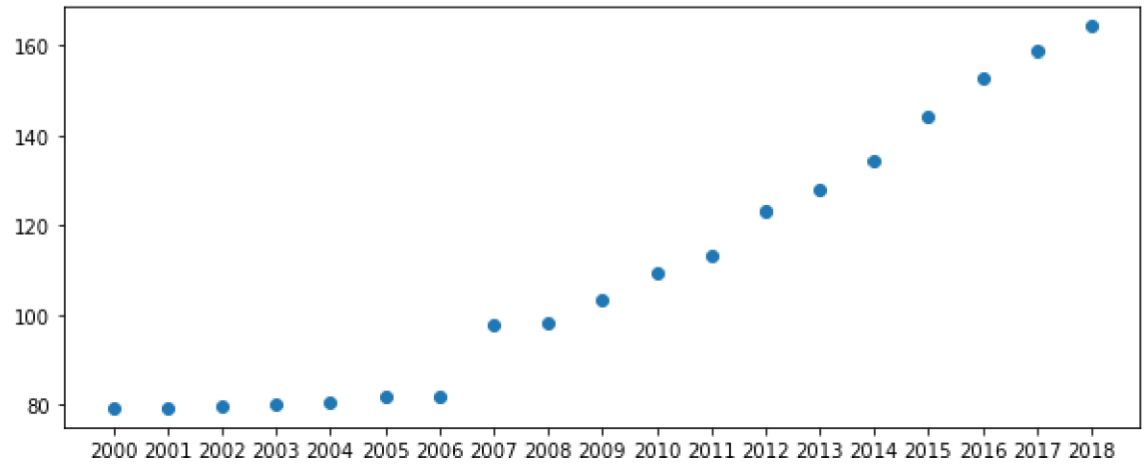
```
mean_year=pd.Series(data[1:].mean())
mean_year
```

Out[2]:

```
2000      79.382574
2001      79.425415
2002      79.896362
2003      80.067944
2004      80.802238
2005      82.091647
2006      81.954150
2007      97.985715
2008      98.195182
2009     103.446197
2010     109.402094
2011     113.099156
2012     123.144632
2013     127.734412
2014     134.270291
2015     144.015547
2016     152.657118
2017     158.716088 2018    164.306824 dtype: float64
```

In [3]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,4))
plt.scatter(mean_year.index,mean_year)
plt.show()
```



In [4]:

```
table=pd.DataFrame(mean_year)
table.columns=['Mean (Y)']
```

In [5]:

```
table['X']=range(1,20)
table['XY']=table['Mean (Y)']*table['X']
table['X^2']=table['X']**2
table=pd.DataFrame(table)
```

In [6]:

```
sum=pd.DataFrame(table.sum())
sum.columns=['Sigma']
sum

mean=pd.DataFrame(table.mean())
mean.columns=['Mean']
mean
```

Out[6]:

	Mean
Mean (Y)	110.031241
X	10.000000
XY	1253.289840
X^2	130.000000

In [7]:

```
table=pd.concat([table,sum.T,mean.T])
table
```

Out[7]:

	Mean (Y)	X	XY	X^2
2000	79.382574	1.0	79.382574	1.0
2001	79.425415	2.0	158.850829	4.0
2002	79.896362	3.0	239.689085	9.0
2003	80.067944	4.0	320.271776	16.0
2004	80.802238	5.0	404.011191	25.0
2005	82.091647	6.0	492.549882	36.0
2006	81.954150	7.0	573.679050	49.0
2007	97.985715	8.0	783.885718	64.0
2008	98.195182	9.0	883.756641	81.0
2009	103.446197	10.0	1034.461971	100.0
2010	109.402094	11.0	1203.423035	121.0
2011	113.099156	12.0	1357.189871	144.0
2012	123.144632	13.0	1600.880221	169.0
2013	127.734412	14.0	1788.281765	196.0
2014	134.270291	15.0	2014.054368	225.0
2015	144.015547	16.0	2304.248753	256.0
2016	152.657118	17.0	2595.171000	289.0
2017	158.716088	18.0	2856.889588	324.0
2018	164.306824	19.0	3121.829647	361.0
Sigma	2090.593585	190.0	23812.506965	2470.0
Mean	110.031241	10.0	1253.289840	130.0

In [8]:

```
numerator=table['XY'].loc['Sigma']-table['X'].loc['Sigma']*table['Mean (Y)'].loc['Sigm
a']/20
numerator
```

Out[8]:

3951.86790441176 In

[9]:

```
denominator = table [ 'X^2'] . loc [ 'Sigma' ] - table[ 'X' ] . loc [ 'Sigma' ] ** 2 /20
denominator
```

Out[9]:

665.0

In [10]:

```
b=numerator/denominator
b
```

Out[10]:

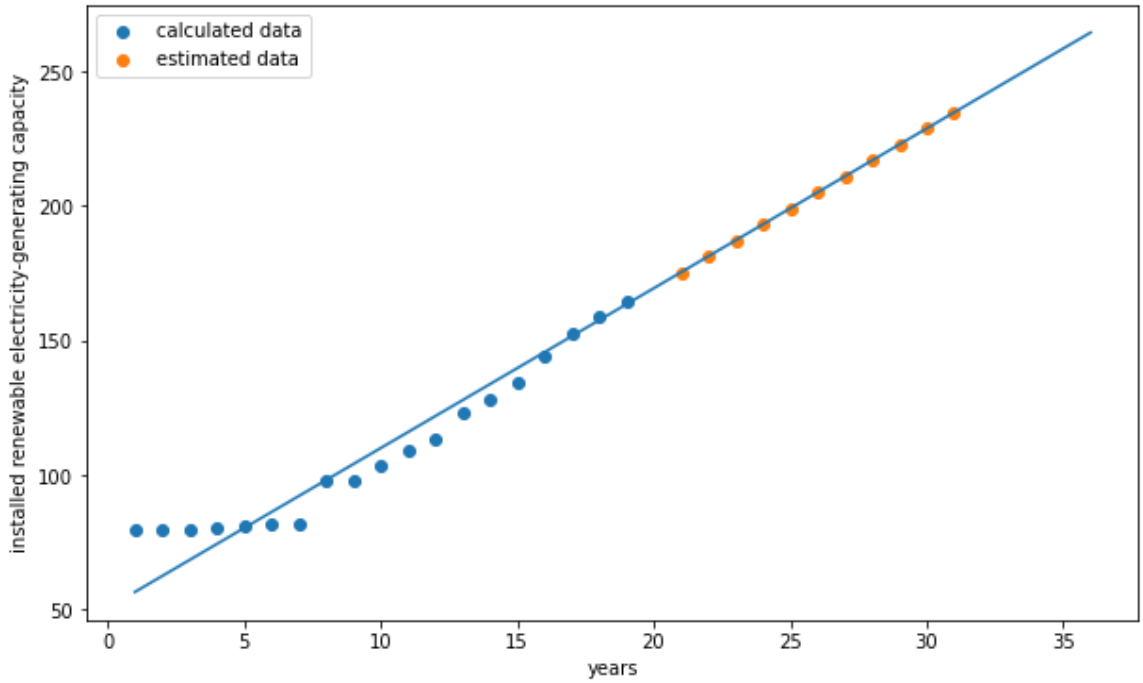
```
5.942658502874827 I
```

```
def eq_regression(x):
    y=table['Mean (Y)'].loc['Mean']+b*(x-table['X'].loc['Mean'])
    return(y)
```

In [16]:

```
plt.figure(figsize=(10,6))
x_values=[]
y_values=[]
for x in range(21,32):
    x_values.append(x)
    y_values.append(eq_regression(x))
print("estimated watts per capita for next 11 years",y_values)

aabbccdd=[]
for i in mean_year.index:
    aabbccdd.append(int(i)-1999)
plt.scatter(aabbccdd,mean_year,label='calculated data')
plt.scatter([range(21,32)],y_values,label='estimated data')
plt.legend()
y_line=[]
x_line=[]
for x in range(1,37):
    x_line.append(x)
    y_line.append(eq_regression(x))
plt.plot(x_line,y_line)
plt.xlabel("years")
plt.ylabel("installed renewable electricity-generating capacity")
plt.show()
```

```
estimated watts per capita for next 11 years [175.40048486289248, 181.3431 4336576728,
187.28580186864212, 193.22846037151695, 199.17111887439177, 20
5.11377737726662, 211.0564358801414, 216.99909438301626, 222.9417528858910
8, 228.8844113887659, 234.82706989164075]
```



Progress Report of Different Regions

In [6]:
```
import pandas as pd
df=pd.read_csv('data 4(asia).csv')
df
```

Out[6]:

	GeoAreaName	2011	2012	2013	2014	2015	2016	2017	2018
0	Central and Southern Asia	50.008	51.605	53.434	58.045	61.378	67.453	75.195	82.962
1	Central Asia	186.444	187.972	192.724	197.254	195.183	193.246	191.718	195.910
2	Eastern and South-Eastern Asia	151.303	169.470	198.222	224.897	256.608	286.602	323.310	357.960
3	Eastern Asia	188.427	210.990	248.897	285.375	328.201	368.596	420.719	469.257
4	Northern Africa and Western Asia	79.453	84.734	91.760	97.654	105.899	111.654	121.274	132.434
5	South-Eastern Asia	60.577	68.693	76.051	80.057	86.244	92.693	94.329	97.886
6	Southern Asia	44.990	46.570	48.267	52.855	56.365	62.717	70.787	78.669
7	Western Asia	119.708	129.251	142.033	152.396	166.990	177.467	193.598	209.238

In [2]:
```
a=df.iloc[:,1:].to_numpy()
print("Matrix:")
print(a)
```

```
Matrix:
[[ 50.008  51.605  53.434  58.045  61.378  67.453  75.195  82.962]
 [186.444 187.972 192.724 197.254 195.183 193.246 191.718 195.91 ]
 [151.303 169.47  198.222 224.897 256.608 286.602 323.31  357.96 ]
 [188.427 210.99  248.897 285.375 328.201 368.596 420.719 469.257]
 [ 79.453  84.734  91.76   97.654 105.899 111.654 121.274 132.434]
 [ 60.577  68.693  76.051  80.057  86.244  92.693  94.329  97.886]
 [ 44.99   46.57   48.267  52.855  56.365  62.717  70.787  78.669]
 [119.708 129.251 142.033 152.396 166.99  177.467 193.598 209.238]]
```

In [5]:
```
import numpy as np
x0=np.array([[1],[1],[1],[1],[1],[1],[1],[1]])
j=1
#first iteration
x1=np.matmul(a,x0)
previous_eigen_value=np.round(max([abs(i) for i in x1]),4)
x1=x1/previous_eigen_value
print("Step",j,":",'\n',x1,'\n',"Eigen Value : ",previous_eigen_value)
j+=1
#second iteration
x3=np.matmul(a,x1)
present_eigen_value=np.round(max([abs(i) for i in x3]),4)
x3=np.round(x3/present_eigen_value,4)
print("Step",j,":",'\n',x3,'\n',"Eigen Value : ",present_eigen_value)
#while loop
while(previous_eigen_value!=present_eigen_value):#will calculate until previous and present eigen value are same
    j+=1
    previous_eigen_value=present_eigen_value
    x1=np.matmul(a,x3)
    present_eigen_value=np.round(max([abs(i) for i in x1]),4)
    x1=x1/present_eigen_value
    print("Step",j,"\n",x1,'\n',"Eigen Value : ",present_eigen_value)
    x3=x1
```

```
Step 1 :
 [[0.19840807]
 [0.61117803]
 [0.78095682]
 [1.        ]
 [0.32726619]
 [0.26048002]
 [0.18299026]
 [0.51208112]]
 Eigen Value :  [2520.462]
Step 2 :
 [[0.2015]
 [0.6421]
 [0.7837]
 [1.    ]
 [0.3348]
 [0.2684]
 [0.1851]
 [0.5224]]
 Eigen Value :  [1166.7968]
Step 3
 [[0.20160535]
 [0.64231826]
 [0.78368963]
 [1.        ]
 [0.33485311]
 [0.26846527]
 [0.1851666 ]
 [0.52247413]]
 Eigen Value :  [1185.7081]
Step 4
 [[0.2016062 ]
 [0.64231935]
 [0.78368957]
 [1.00000002]
 [0.33485362]
 [0.26846507]
 [0.18516744]
 [0.52247467]]
 Eigen Value :  [1185.8757]
Step 5
 [[0.2016062 ]
 [0.64231933]
 [0.78368956]
 [1.        ]
 [0.33485361]
 [0.26846506]
 [0.18516744]
 [0.52247466]]
 Eigen Value :  [1185.8768]
Step 6
 [[0.20160619]
 [0.64231931]
 [0.78368954]
 [0.99999998]
 [0.33485361]
 [0.26846505]
 [0.18516744]
 [0.52247465]]
 Eigen Value :  [1185.8768]
```

Ranking is determined on the basis of the elements present in the most dominant Eigen Vector

1. Eastern Asia (0.999)
2. Eastern and Southern Asia (0.783)
3. Central Asia (0.642)
4. Western Asia (0.522)
5. Northern Africa and Western Africa (0.335)
6. South Eastern Asia (0.268)
7. Central and Southern Asia (0.202)
8. Southern Asia (0.185)

Correlation (Click to view all Objectives)

(Between Renewable energy share in the total final energy consumption & Installed renewable electricity generating capacity (watts per capita))

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import scipy
from collections import OrderedDict
import math
import numpy as np
import io
```

In [2]:
```
data1=pd.read_csv('2 EG_FEC_RNEW.csv',index_col='S.No.')
data1=data1.drop(columns=['2001'])
data1=data1.rename(columns={'2017':'Renewable energy share in the total final energy consumption (%)'})
data1
```

Out[2]:

S.No.	GeoAreaName	Renewable energy share in the total final energy consumption (%)
1	Afghanistan	24.65
2	Albania	37.19
3	Algeria	0.14
4	American Samoa	1.75
5	Americas	16.47
...	...	...
230	Viet Nam	31.98
231	Wallis and Futuna Islands	0.67
232	Yemen	4.85
233	Zambia	84.53
234	Zimbabwe	83.29

234 rows × 2 columns

In [4]:
```
data2=pd.read_csv('4 EG_EGY_RNEW.csv',index_col='S.No.')
data2=data2.drop(columns=['2001'])
data2=data2.rename(columns={'2017':'Installed renewable electricity-generating capacity (watts per capita)'})
data2
```

Out[4]:

S.No.	GeoAreaName	Installed renewable electricity-generating capacity (watts per capita)
1	Afghanistan	9.792
2	Algeria	16.024
3	American Samoa	75.085
4	Angola	81.472
5	Anguilla	146.764
...	...	...
166	Venezuela (Bolivarian Republic of)	564.462
167	Viet Nam	192.539
168	Yemen	3.593
169	Zambia	144.850
170	Zimbabwe	62.326

170 rows × 2 columns

In [5]:
```
common=pd.merge(data1, data2, how ='inner', on =['GeoAreaName'])
common
```

Out[5]:

	GeoAreaName	Renewable energy share in the total final energy consumption (%)	Installed renewable electricity-generating capacity (watts per capita)
0	Afghanistan	24.65	9.792
1	Algeria	0.14	16.024
2	American Samoa	1.75	75.085
3	Angola	56.16	81.472
4	Anguilla	0.18	146.764
...	...	...	...
157	Venezuela (Bolivarian Republic of)	14.78	564.462
158	Viet Nam	31.98	192.539
159	Yemen	4.85	3.593
160	Zambia	84.53	144.850
161	Zimbabwe	83.29	62.326

162 rows × 3 columns

In [6]:
```
correlation=(common['Renewable energy share in the total final energy consumption (%)'].corr(common['Installed renewable\xa0electi
if 20<=correlation<=80:
  print("Moderate Correlation")
elif correlation>80:
  print("Strong Correlation")
else:
  print("Weak Correlation")
print("Correlation between these data is:",round(correlation,2),"%")
```

```
Weak Correlation
Correlation between these data is: 0.28 %
```

(Between Proportion of population with primary reliance & Energy intensity level of primary energy (megajoules per constant 2011 purchasing power parity GDP))

In [8]:
```
data3=pd.read_csv('1 EG_EGY_CLEAN.csv',index_col='S.No.')
data3=data3.drop(columns=['2001'])
data3=data3.rename(columns={'2017':'Proportion of population with primary reliance on clean fuels and technology (%)'})
data3
```

Out[8]:

S.No.	GeoAreaName	Proportion of population with primary reliance on clean fuels and technology (%)
1	Afghanistan	34
2	Albania	78
3	Algeria	95
4	Andorra	95
5	Angola	48
...	...	...
204	Viet Nam	62
205	Western Asia	93
206	Yemen	61
207	Zambia	14
208	Zimbabwe	29

208 rows × 2 columns

In [9]:
```
data4=pd.read_csv('3 EG_EGY_PRIM.csv',index_col='S.No.')
data4=data4.drop(columns=['2001'])
data4=data4.rename(columns={'2017':'Energy intensity level of primary energy (megajoules per constant 2011 purchasing power pari
data4
```

Out[9]:

S.No.	GeoAreaName	Energy intensity level of primary energy (megajoules per constant 2011 purchasing power parity GDP)
1	Afghanistan	1.93
2	Albania	2.91
3	Algeria	4.05
4	Americas	4.79
5	Angola	3.41
...	...	...
201	Vanuatu	3.65
202	Venezuela (Bolivarian Republic of)	6.02
203	Yemen	2.04
204	Zambia	8.05
205	Zimbabwe	12.95

205 rows × 2 columns

In [10]:
```
common2=pd.merge(data3, data4, how ='inner', on =['GeoAreaName'])
common2
```

Out[10]:

	GeoAreaName	Proportion of population with primary reliance on clean fuels and technology (%)	Energy intensity level of primary energy (megajoules per constant 2011 purchasing power parity GDP)
0	Afghanistan	34	1.93
1	Albania	78	2.91
2	Algeria	95	4.05
3	Angola	48	3.41
4	Antigua and Barbuda	95	3.24
...	...	...	...
181	Vanuatu	8	3.65
182	Venezuela (Bolivarian Republic of)	95	6.02
183	Yemen	61	2.04
184	Zambia	14	8.05
185	Zimbabwe	29	12.95

186 rows × 3 columns

In [15]:
```
correlation2=(common2['Proportion of population with primary reliance on clean fuels and technology (%)'].corr(common2['Energy ir
if 20<=correlation2<=80:
  print("Moderate Correlation")
elif correlation2>80:
  print("Strong Correlation")
else:
  print("Weak Correlation")
```

```
Weak Correlation
```

ANALYSIS (Click to view all Objectives)

Objective 1:

Objective 1.1:

In this objective, the claim of UN that energy intensity between year 2001 and 2017 has decreased by 1.48 (megajoules per constant 2011 purchasing power parity GDP) has been investigated has been verified to be True.

We have used difference of two means, two tailed Z-test to verify the claim.

As per UN data in 2001 energy intensity μ1= 6.49

As per UN data in 2017 energy intensity μ2 =5.01

From the data:

In year 2001

$\overline{x_1}$=6.86

$\boldsymbol{\sigma}_1$=4.88

$\boldsymbol{\sigma}_1^2$=23.8144

n1=205

In year 2017,

$\bar{x}_2$=5.14

$\boldsymbol{\sigma}_2$=3.02

$\boldsymbol{\sigma}_2^2$=9.1204

n2=205

Null Hypothesis will be (H_0): μ1- μ2=1.48

Alternate Hypothesis will be(H_a): μ1- μ2≠1.48

We have taken α (level of significance) = 0.05

Now the value of Z is-

$$z = \frac{\bar{x}_1 - \bar{x}_2 - \Delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Z=0.598

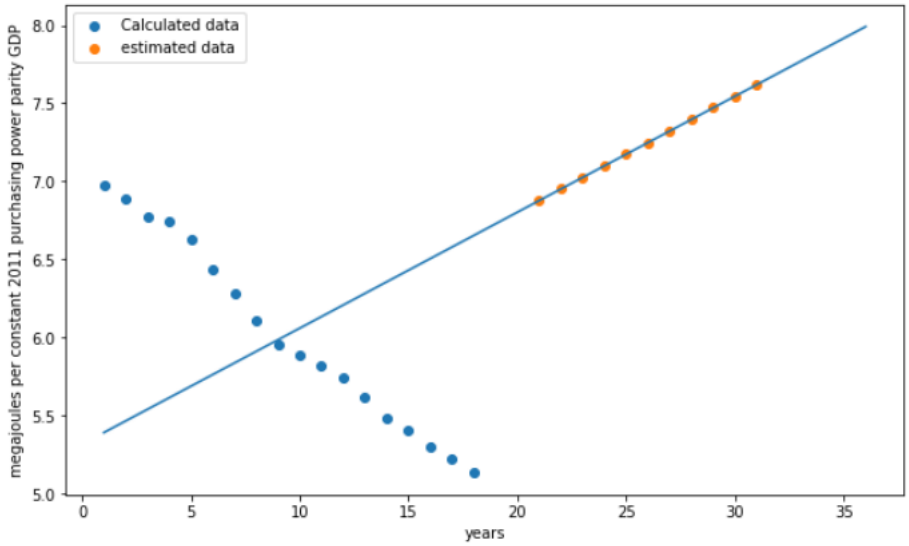
Rejection criteria

$|Z| > Z_{\alpha/2}$

Clearly 0.598<1.960

Hence Null Hypothesis is accepted, and UN claim is observed to be True.

Objective 1.2:



From the above graph we can analyze that to have sufficient Energy intensity level of primary energy by the year 2030 we should have the proportion from year 2018 to

2030 greater than [6.87, 6.95, 7.02, 7.09, 7.17, 7.24, 7.32, 7.39, 7.47, 7.54, 7.61,7.69] for the upcoming 12 years to reach the target by the UN till 2030. Now , for the upcoming year we can predict with the line(made from the data from 2000 to 2017) if in any of the year the data (scatter plot in the graph) is below the line (trend set by the observed data) then it means the observed data of that particular year is below the trend (decreased) same way if the data(scatter plot) is above the line (trend set by the observed data) then it means the observed data of that particular year is above the trend (increased).

Objective 1.3:

After observing the energy intensity measured in terms of primary energy and GDP of all the continents in the latest year (i.e. 2017). Ranks has been deduced for all continents through linear algebra concept (i.e. power iteration method)

Rank of all continents:

Rank	Continents	Energy intensity measured in terms of primary energy and GDP (megajoules)
1	South America	3.93
2	Europe	4.39
3	Australia	4.82
4	Asia	5.03
5	North America	5.3
6	Africa	5.65

After deducing rank of all the continents , it is observed that South America is the having best energy intensity level of primary energy (3.93) among all the continents in the world. Second is Europe and it is lagging behind South America (by 0.46 megajoules per constant 2011 purchasing power parity GDP difference).Third is Australia, it is lagging behind South America (by 0.89 megajoules per constant 2011 purchasing power parity GDP difference). Fourth is Asia is the , it is lagging behind South America (by 1.1 megajoules per constant 2011 purchasing power parity GDP difference). Fifth is North America, it is lagging behind South America (by 1.37 megajoules per constant 2011 purchasing power parity

GDP difference). Last is Africa , it is lagging behind South America (by 1.72 megajoules per constant 2011 purchasing power parity GDP difference).

Here Africa , North America and Asia really need to buck up to since a large difference between respective continent and South America.

Reason behind ranking all the continents is to check which continent is the best continent in energy efficiency of economy and where each continent stand. After deducing rank comparison of all the continents with the best continent in energy efficiency of economy is done to know by what percentage each continent is lagging behind, if the difference is large then that continent needs to buck up to make target by the time limit (by 2030) set by UN possible.

Objective 2: (Click to view all Objectives)

Objective 2.1:

In this objective, the claim of UN that the usage of renewable energy in Total energy has increased.

We have used difference of two proportions, right tailed Z-test to verify the claim.

From the data:

In year 2001,

$p_1=28.8$

$\boldsymbol{\sigma}_1=30.65$

$\boldsymbol{\sigma}_1^2=939.4225$

$n_1=234$

In year 2017,

p_2=5.14

σ_2=3.02

σ_2^2=9.1204

n_2=234

Null Hypothesis will be (H0): $p_1 - p_2 \leq 0$

Alternate Hypothesis will be(H_a): $p_1 - p_2 > 0$

We have taken α (level of significance) = 0.05

Now the value of Z is-

$$z = \frac{p_1 - p_2}{\sqrt{pq\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}}$$

After putting the values:

Z=0.452

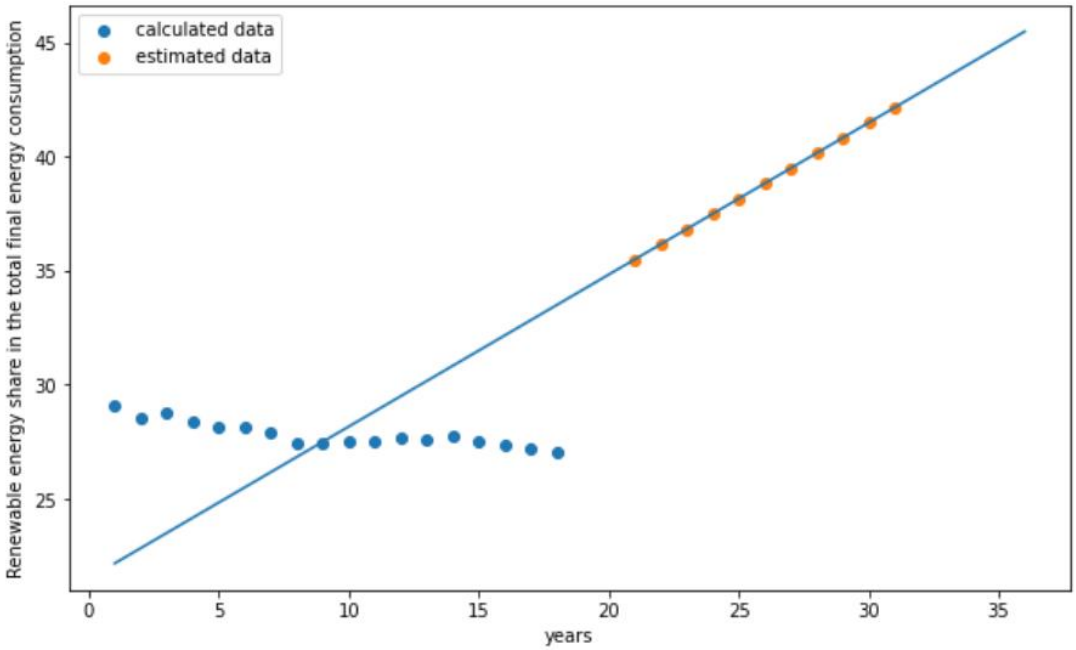
Rejection criteria

$Z > Z_\alpha$

Clearly 0.452<1.645

Hence Null Hypothesis is accepted and UN claim is observed to be true.

Objective 2.2:



From the above graph we can analyse that to have sufficient renewable energy share in the total final energy consumption by the year 2030 we should have the proportion from year 2018 to 2030 greater than [35.50, 36.16, 36.83, 37.49, 38.16, 38.83, 39.49, 40.16, 40.83, 41.49, 42.16,42.83]

for the upcoming 12 years to reach the target by the UN till 2030. Now , for the upcoming year we can predict with the line(made from the data from 2000 to 2017) if in any of the year the data (scatter plot in the graph) is below the line (trend set by the observed data) then it means the observed data of that particular year is below the trend (decreased) same way if the data(scatter plot) is above the line (trend set by the observed data) then it means the observed data of that particular year is above the trend (increased).

Objective 2.3:

After observing the consumption of renewable energy (in %) of all the continents in the latest year (i.e. 2017). Ranks has been deduced for all continents through linear algebra concept (i.e. power iteration method)

Rank of all continents:

Rank	Continents	Consumption of renewable energy (in %)
1	Africa	54.38
2	South America	34.21
3	Asia	16.01
4	Europe	13.46
5	North America	11.56
6	Australia	9.54

After deducing rank of all the continents , it is observed that Africa is the highest consumer (54.38 %) of renewable energy among all the continents in the world. South America is the second highest consumer and it is lagging behind Africa (highest consumer) by
20.17 %. Asia is the third highest consumer, it is lagging behind Africa (highest consumer) by 38.37%. Europe is the fourth highest consumer, it is lagging behind Africa (highest consumer) by 40.92 %. North America is the fifth highest consumer and it is lagging behind Africa (highest consumer) by 42.82%. Australia is the least consumer and it is lagging behind Africa (highest consumer) by 44.84 %.
Here Australia , North America and Europe really need to buck up to since a large difference between respective continent and Africa(highest consumer) in consumption of renewable energy is observed.

Reason behind ranking all the continents is to check which continent is the highest consumer and where each continent stand in terms of consumption of renewable energy . After deducing rank comparison of all the continents with the highest

consumer is done to know by what percentage each continent is lagging behind highest consumer , if the difference is large then that continent needs to buck up to make target by the time limit (by 2030) set by UN possible.

Objective 3: (Click to view all Objectives)

We have used difference of single proportions, right tailed Z-test to verify the claim.

From the data:

In year 2001,

$p=58.17$

$\boldsymbol{\sigma}=36.9$

$\boldsymbol{\sigma}^2=1361.61$

$n=208$

Null Hypothesis will be (H_0): $p_0=50\%$ (0.5)

Alternate Hypothesis will be (H_a): $p_0\neq50\%$ (0.5)

We have taken α (level of significance)=0.05

Now the value of Z is-

$$z = \frac{x-np_0}{\sqrt{np_0(1-p_0)}}$$

After putting the values:

Z=2.31

Rejection criteria

$|Z|>Z_{\alpha/2}$

Clearly 2.31>1.960

Hence Null Hypothesis is rejected, and UN claim is observed to be False.

In year 2017,

p=64.5

σ=35.27

σ^2=1243.9729

n=208

Null Hypothesis: H_0: p_0=62% (0.5)

Alternate Hypothesis will be(H_a): p_0≠62% (0.5)

We have taken α=0.05

Now the value of Z is-

$$z = \frac{x - np_0}{\sqrt{np_0(1-p_0)}}$$

After putting the values:

Z=0.89

Rejection criteria

$|Z| > Z_{\alpha/2}$

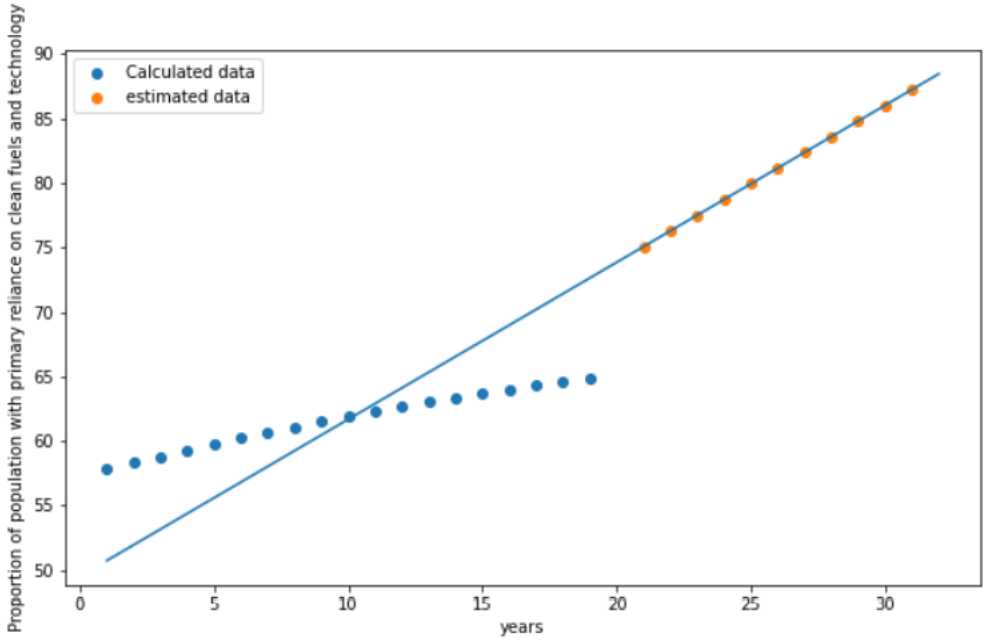
Clearly 0.89<1.960

Hence Null Hypothesis is accepted and UN claim is observed to be true.

Objective 3.2:

From the graph given below we can analyze that for the sufficient proportion of population to have primary reliance on clean energy by the year 2030 we should have

the proportion from year 2019 to 2030 greater than [75.06, 76.28,77.5, 78.72, 79.94, 81.15, 82.37, 83.58, 84.80, 86.02, 87.23] for the upcoming 11 years to reach the target by the UN till 2030. Now , for the upcoming year we can predict with the line(made from the data from 2000 to 2018) if in any of the year the data (scatter plot in the graph) is below the line (trend set by the observed data) then it means the observed data of that particular year is below the trend (decreased) same way if the data(scatter plot) is above the line (trend set by the observed data) then it means the observed data of that particular year is above the trend (increased).



Objective 3.3:

From the ranking given below, we can analyze the increment of population with primary reliance on clean energy in different regions of Africa over 5 years, as by given data we can conclude that progress of Northern Africa and Western Asia is impressive over the years while the progress of Central African Republic is mediocre over the years as observed from the calculated data.

Regions of Africa	Increment of Primary reliance on clean energy (%)	Rank
Northern Africa and Western Asia	91	1
Northern Africa	89.2	2
South Africa	83.8	3
Sub-Saharan Africa	13.8	4
Central African Republic	5	5

Objective 4: (Click to view all Objectives)

Objective 4.1

In this objective, the claim of UN that installed renewable electricity generating capacity (watts per capita) is 65.202 in 2001 and 188.01 in 2017.

We have used single, two tailed Z-test to verify the claim.

As per UN data in 2001 energy intensity μ= 65.202

As per UN data in 2017 energy intensity μ=188.01

From the data:

In year 2001

$\bar{x}$= 79.26

σ=165.43

σ=27367.0849

n=170

Null Hypothesis will be(H_0): μ=65.202

Alternate Hypothesis will be(H_a): μ≠65.202

We have taken α (level of significance) = 0.05

Now the value of Z is-

$$z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n}}$$

Z=1.108

Rejection criteria:

$|Z|>Z_{\alpha/2}$

Clearly 1.108<1.960

Hence Null Hypothesis is accepted and UN claim is observed to be True.

In year 2017,

$\bar{x}$= 158.19

σ=264.66

σ=70044.9156

n=170

Null Hypothesis will be (H_0): μ=188.01

Alternate Hypothesis will be (H_a): μ_1≠188.01

We have taken α (level of significance) = 0.05

Now the value of Z is-

$$z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n}}$$

Z=-1.469

Rejection criteria

$|Z|>Z_{\alpha/2}$

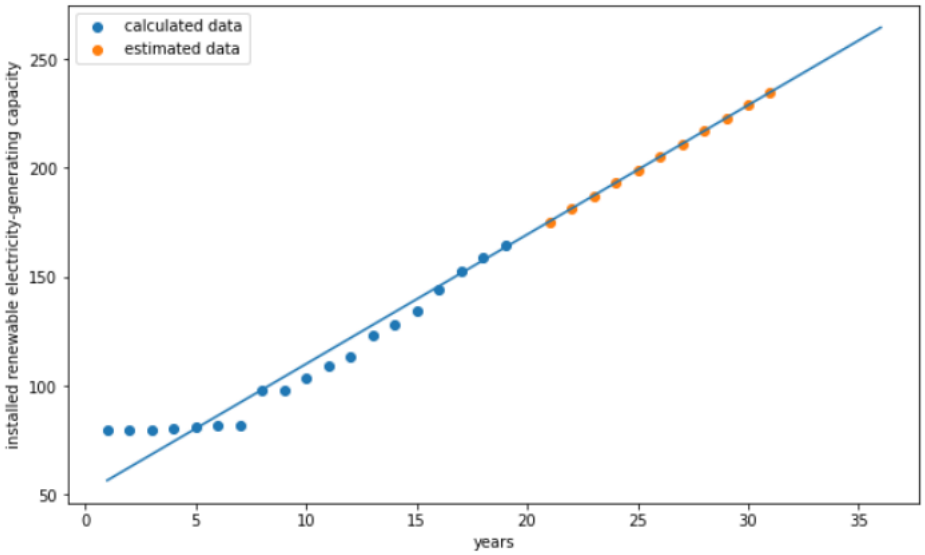
Clearly 1.469<1.960

Hence Null Hypothesis is accepted and UN claim is observed to be True.

Objective 4.2:

From the graph given below we can analyze that to have sufficient Installed renewable electricity-generating
capacity by the year 2030 we should have the proportion from 2019 to 2030 greater than

[175.40,181.34,187.28,193.22,199.17,205.11,211.05,216.99,222.94,228.88,234.82] for the upcoming 11 years. Now , for the upcoming year we can predict with the line(made from the data from 2000 to 2018) if in any of the year the data (scatter plot in the graph) is below the line (trend set by the observed data) then it means the observed data of that particular year is below the trend (decreased) same way if the data(scatter plot) is above the line (trend set by the observed data) then it means the observed data of that particular year is above the trend (increased).



Objective 4.3:

From the above ranking we can analyze the increment of watts per capita of different regions of Asia over 8 years, as by given data we can conclude that progress of eastern Asia is impressive over the years while the progress of central and southern Asia is mediocre over the years as observed from the calculated data.

Regions	Increment of Watts per capita over 8 years	Rank
Eastern Asia	315.0578	1
Eastern and southern Asia	246.0465	2
Central Asia	192.5564	3
Western Asia	161.3351	4
Northern Africa and western Asia	103.1078	5
South eastern Asia	82.06625	6
Central and southern Asia	62.51	7
Southern Asia	57.6525	8

Conclusion

As from the above report we can conclude that Energy advances in low-and centre pay nations will probably include decreases in customary wood fuel dependence for warming, cooking and limited scope modern energy arrangement, though nations as of now looking to enhance environmentally friendly power portfolios

In many settings, our assumption is a change that includes the enhancement of fuel sources that family units and organizations depend on as opposed to a total progress away from current energizes and advances.

While the natural impression is a significant test for all districts, helpless energy access is predominantly an issue for non-industrial countries. We consider that the proceeded and gigantic hunger for energy in USA is an indication that it neglects to satisfy the objective's interest for "present day" energy.

References

For collecting data-

https://unstats.un.org/sdgs/indicators/database

Literature of survey

1) Cary, M. (2019). Increasing Access to Clean Fuels and Clean Technologies: A Club Converge Approach *1*(1), 247-264.
2) Eder, L., & Provornaya, I., (2018). Analysis of energy intensity trend as a trend as a tool for long-term forecasting of energy consumption. 11, 1971-1997
3) Mehedintu, A., & Strepu, M., and Sova, G., (2018). Estimation and Forecasts for the Share of Renewable Energy Consumption in Final Energy Consumption by 2020. 10(5), 1515.
4) Vandaele, N., & Porter, W. (2015).Renewable Energy inn Developing and Developed Nations.

Individual Contribution

Student name	Objective	Problem statement	Contribution
Aditya Shukla	Objective-1	Verifying claim done by UN	Data analysis
		Estimating the condition by the year 2030	Python
		Region/Continent wise Ranking	Analysed the research paper by [Anca Mehedintu et al.(2018)]
Arushi Singh	Objective-2	Verifying claim done by UN	Data analysis
		Estimating the condition by the year 2030	Python
		Region/Continent wise Ranking	Analysed the research paper by [Leontiy Eder et al.(2018)]
Garv Baheti	Objective-3	Verifying claim done by UN	Data analysis
		Estimating the condition by the year 2030	Python
		Region/Continent wise Ranking	Analysed the research paper by [Michael Cary et al.(2019)]
Sparsh Goud	Objective-4	Verifying claim done by UN	Data analysis
		Estimating the condition by the year 2030	Python
		Region/Continent wise Ranking	Analysed the research paper by [Nicole Vandaele et.al(2015))]

Appendix

Files links

Objective : 1 Complete Files (Jupyter and CSV)

Objective: 2 Complete Files (Jupyter and CSV)

Objective: 3 Complete Files (Jupyter and CSV)

Objective: 4 Complete Files (Jupyter and CSV)

Correlation