Class 2B — By Your Powers Combined

This is the team activity for **Class 2B** (Thursday, Sep. 2, 2021).

When you have completed the activity, upload the notebook and a PDF export to the *Weeek 2 Assignment* in Canvas, and make sure all team members have a copy of the final notebook.

This notebook is intended for you to fill out. The instructions are written inline, with empty cells for you to work. Feel free to add additional code and/or markdown cells as needed, to present the results and to provide appropriate interpretive commentary.

🗄️ Data

This assignment uses Version 1.3.0 of the Global Power Plant Database. Download and unpack those files.

One of the files, `A_Global_Database_of_Power_Plants.pdf` , contains documentation about the data.

🛠️ Setup

As usual, we need to start by setting up our Python environment.

In [51]:

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

Turn on Matplotlib rendering:

In [52]:

```python
%matplotlib inline
```

And read the data, using `pandas.read_csv` :

In [53]:

```python
df = pd.read_csv("global_power_plant_database.csv", low_memory=False)
```

🏘️ Structural Description

How many **observations** are in this data?

In [54]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34936 entries, 0 to 34935
Data columns (total 36 columns):
 #   Column                          Non-Null Count  Dtype
```

```
---  ------                          --------------  -----
 0   country                         34936 non-null  object
 1   country_long                    34936 non-null  object
 2   name                            34936 non-null  object
 3   gppd_idnr                       34936 non-null  object
 4   capacity_mw                     34936 non-null  float64
 5   latitude                        34936 non-null  float64
 6   longitude                       34936 non-null  float64
 7   primary_fuel                    34936 non-null  object
 8   other_fuel1                     1944 non-null   object
 9   other_fuel2                     276 non-null    object
 10  other_fuel3                     92 non-null     object
 11  commissioning_year              17447 non-null  float64
 12  owner                           20868 non-null  object
 13  source                          34921 non-null  object
 14  url                             34918 non-null  object
 15  geolocation_source              34517 non-null  object
 16  wepp_id                         16234 non-null  object
 17  year_of_capacity_data           14887 non-null  float64
 18  generation_gwh_2013             6417 non-null   float64
 19  generation_gwh_2014             7226 non-null   float64
 20  generation_gwh_2015             8203 non-null   float64
 21  generation_gwh_2016             9144 non-null   float64
 22  generation_gwh_2017             9500 non-null   float64
 23  generation_gwh_2018             9637 non-null   float64
 24  generation_gwh_2019             9659 non-null   float64
 25  generation_data_source          11400 non-null  object
 26  estimated_generation_gwh_2013   16120 non-null  float64
 27  estimated_generation_gwh_2014   16503 non-null  float64
 28  estimated_generation_gwh_2015   17050 non-null  float64
 29  estimated_generation_gwh_2016   17570 non-null  float64
 30  estimated_generation_gwh_2017   33138 non-null  float64
 31  estimated_generation_note_2013  34936 non-null  object
 32  estimated_generation_note_2014  34936 non-null  object
 33  estimated_generation_note_2015  34936 non-null  object
 34  estimated_generation_note_2016  34936 non-null  object
 35  estimated_generation_note_2017  34936 non-null  object
dtypes: float64(17), object(19)
memory usage: 9.6+ MB
```

There are 34936 observations.

How many **variables** are in this data?

```
In [55]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34936 entries, 0 to 34935
Data columns (total 36 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   country                         34936 non-null  object
 1   country_long                    34936 non-null  object
 2   name                            34936 non-null  object
 3   gppd_idnr                       34936 non-null  object
 4   capacity_mw                     34936 non-null  float64
 5   latitude                        34936 non-null  float64
 6   longitude                       34936 non-null  float64
 7   primary_fuel                    34936 non-null  object
 8   other_fuel1                     1944 non-null   object
 9   other_fuel2                     276 non-null    object
 10  other_fuel3                     92 non-null     object
 11  commissioning_year              17447 non-null  float64
 12  owner                           20868 non-null  object
 13  source                         34921 non-null  object 
 14  url                            34918 non-null  object 
 15  geolocation_source             34517 non-null  object 
 16  wepp_id                        16234 non-null  object 
 17  year_of_capacity_data          14887 non-null  float64
 18  generation_gwh_2013            6417 non-null   float64
 19  generation_gwh_2014            7226 non-null   float64
 20  generation_gwh_2015            8203 non-null   float64
 21  generation_gwh_2016            9144 non-null   float64
 22  generation_gwh_2017            9500 non-null   float64
 23  generation_gwh_2018            9637 non-null   float64
 24  generation_gwh_2019            9659 non-null   float64
 25  generation_data_source         11400 non-null  object 
 26  estimated_generation_gwh_2013  16120 non-null  float64
 27  estimated_generation_gwh_2014  16503 non-null  float64
 28  estimated_generation_gwh_2015  17050 non-null  float64
 29  estimated_generation_gwh_2016  17570 non-null  float64
 30  estimated_generation_gwh_2017  33138 non-null  float64
 31  estimated_generation_note_2013  34936 non-null  object 
 32  estimated_generation_note_2014  34936 non-null  object 
 33  estimated_generation_note_2015  34936 non-null  object 
 34  estimated_generation_note_2016  34936 non-null  object 
 35  estimated_generation_note_2017  34936 non-null  object 
dtypes: float64(17), object(19)
memory usage: 9.6+ MB
```

There are 36 variables.

What are some of the variables in this data? Look at both the column names, and the documentation (particularly Table 3), to identify some of the variables we have here.

In [56]:

```
df
```

Out[56]:

	country	country_long	name	gppd_idnr	capacity_mw	latitude	longitude	primary_
0	AFG	Afghanistan	Kajaki Hydroelectric Power Plant Afghanistan	GEODB0040538	33.0	32.3220	65.1190	Hy
1	AFG	Afghanistan	Kandahar DOG	WKS0070144	10.0	31.6700	65.7950	S
2	AFG	Afghanistan	Kandahar JOL	WKS0071196	10.0	31.6230	65.7920	S
3	AFG	Afghanistan	Mahipar Hydroelectric Power Plant Afghanistan	GEODB0040541	66.0	34.5560	69.4787	Hy
4	AFG	Afghanistan	Naghlu Dam Hydroelectric Power Plant Afghanistan	GEODB0040534	100.0	34.6410	69.7170	Hy
...	...	...	...	...	...	...	...	
34931	ZMB	Zambia	Ndola	WRI1022386	50.0	-12.9667	28.6333	
34932	ZMB	Zambia	Nkana	WRI1022384	20.0	-12.8167	28.2000	
34933	ZMB	Zambia	Victoria Falls	WRI1022380	108.0	-17.9167	25.8500	Hy

	country	country_long	name	gppd_idnr	capacity_mw	latitude	longitude	primary_
34934	ZWE	Zimbabwe	Hwange Coal Power Plant Zimbabwe	GEODB0040404	920.0	-18.3835	26.4700	
34935	ZWE	Zimbabwe	Kariba Dam South Hydroelectric Power Station Z...	GEODB0003803	750.0	-16.5222	28.7619	Hy

34936 rows × 36 columns

- capacity, mostly integer, float in pandas
- primary_fuel, string actual, object
- url, string actual, object in pandas
- owner, string actual, object in pandas

`

Do the Pandas types match what you would expect from the expected data type? Are there any surprises?

No, strings are stored as objects and some integer data types are stored as floats.

❔ Questions

Identify **2 questions** that you could try to answer with this data, and write them in the Markdown cell below.

- Which countries use different fuel types the most?
- How has the fuel distribution changed over the time period?

🙋‍♀️ Check-In Breakpoint 🙋‍♀️

This is where we're going to do an initial check-in and cross-team discussion.

Distributions

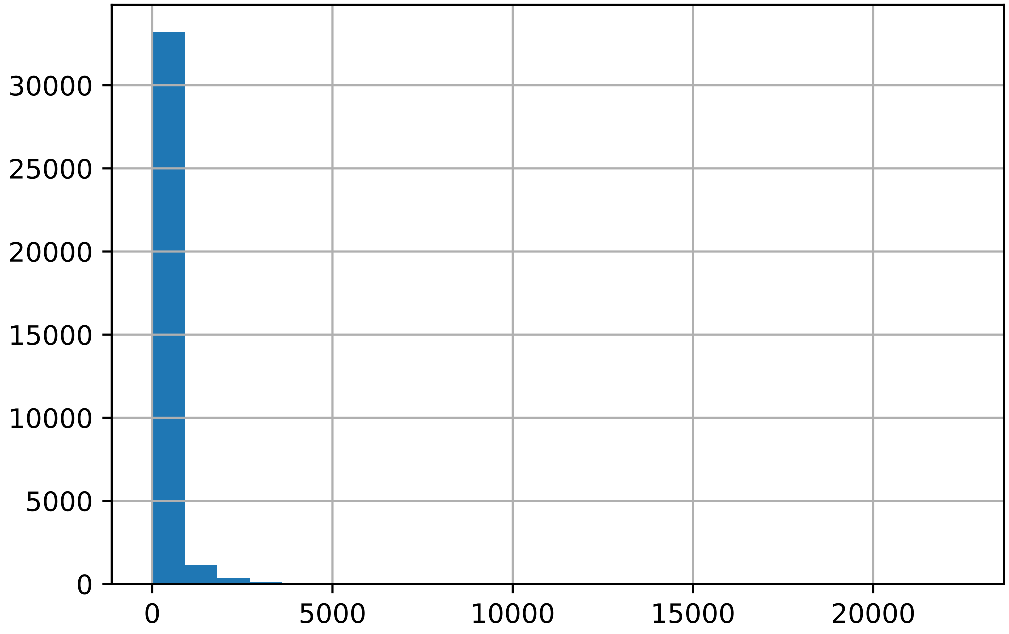
What is the distribution of power production capacity **per plant**? Describe graphically and numerically.

In [57]:

```
df["capacity_mw"].hist(bins=25)
```

Out[57]:

```
<AxesSubplot:>
```



In [58]:

```
df["capacity_mw"].describe()
```

Out[58]:

```
count    34936.000000
mean       163.355148
std        489.636072
min          1.000000
25%          4.900000
50%         16.745000
75%         75.344250
max      22500.000000
Name: capacity_mw, dtype: float64
```

What is the distribution of power production capacity **per country**? Describe graphically and numerically.

In [59]:

```
cap_per_country = df.groupby("country")["capacity_mw"].agg(["sum"])
cap_per_country.describe()
```

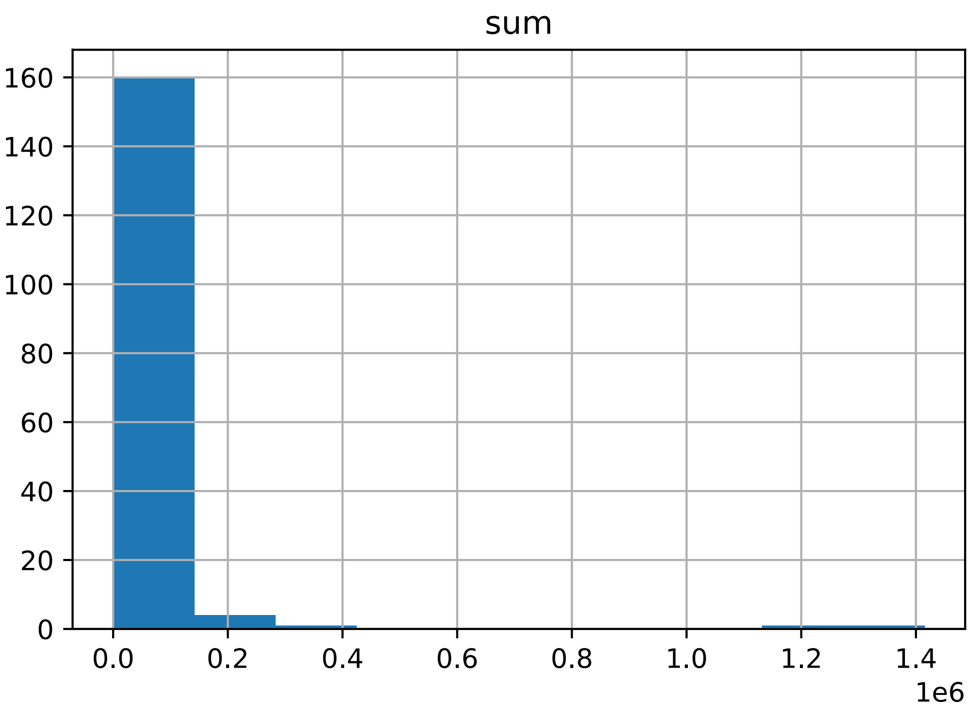
Out[59]:

	sum
count	1.670000e+02
mean	3.417351e+04
std	1.473412e+05
min	3.000000e+00
25%	8.724900e+02
50%	3.720100e+03
75%	1.470533e+04
max	1.415067e+06

In [60]:

```
cap_per_country.hist()
```

```
Out[60]: array([[<AxesSubplot:title={'center':'sum'}>]], dtype=object)
```



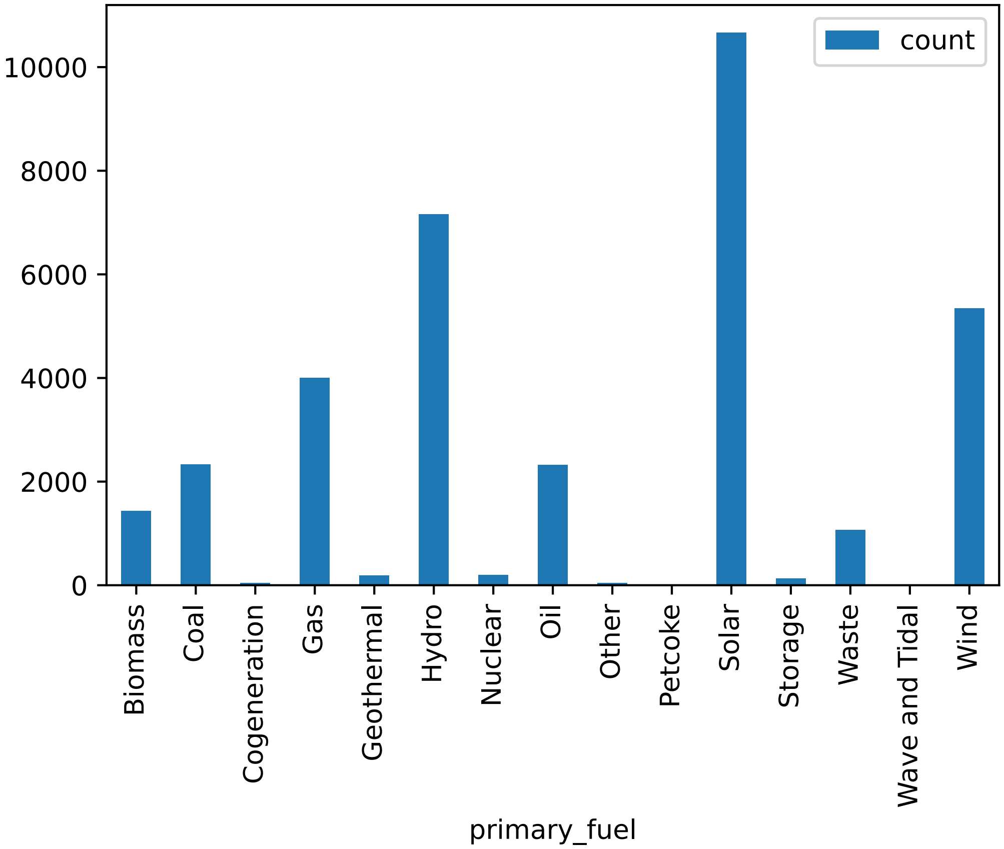
Exploration

How many power plants are there of each fuel type?

> **Hint:** This is easiest to view with a **horizontal bar plot**. You can do this by using Pandas group-by, and calling `.plot.barh()` on the resulting series; or by using Seaborn's `countplot`, with `y='primary_fuel'` instead of `x='primary_fuel'`.

```
In [61]: plant_per_fuel = df.groupby("primary_fuel")["gppd_idnr"].agg(["count"])
         plant_per_fuel.plot(kind="bar")
```

```
Out[61]: <AxesSubplot:xlabel='primary_fuel'>
```



How much **total production capacity** is there of each fuel type?

In [62]:

```
df.groupby("primary_fuel")["capacity_mw"].agg(["sum"])
```

Out[62]:

	sum
primary_fuel	
Biomass	3.428130e+04
Coal	1.965541e+06
Cogeneration	4.048000e+03
Gas	1.493051e+06
Geothermal	1.268775e+04
Hydro	1.053160e+06
Nuclear	4.079118e+05
Oil	2.618787e+05
Other	3.612860e+03
Petcoke	2.424577e+03
Solar	1.883123e+05
Storage	1.712300e+03
Waste	1.474871e+04
Wave and Tidal	5.522000e+02

	sum
primary_fuel	
Wind	2.630537e+05

❓ Question

Pick one of the questions (either one of yours, or one of the other teams'). I recommend picking a simple one! State the question:

What is the total capacity of fuels per fuel_type per country?

Describe, in English, a precise mechanism by which you will compute this measurement (including the variable(s), grouping, aggregates, etc. involved):

First we group the data by countries and then by fuel type and take the sum of capacity of the country.

Attempt to answer it with the tools we have seen so far:

In [63]:

```
df.groupby(by=["country", "primary_fuel"])["capacity_mw"].sum()
```

Out[63]:

```
country  primary_fuel
AFG      Gas               42.00
         Hydro            238.55
         Solar             20.00
AGO      Gas              163.68
         Hydro            770.60
                           ...
ZMB      Hydro           2160.00
         Oil              169.60
         Solar             47.50
ZWE      Coal             920.00
         Hydro            750.00
Name: capacity_mw, Length: 698, dtype: float64
```

🏁 We're Done!

Submit to Canvas