In [8]:

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

In [9]:

```
import datetime as dt
from datetime import datetime
```

In [10]:

```
import os
```

In [11]:

```
# The given gppd_120_pr.csv consists of all the power plants which belongs to the Puerto Rico
global_power_plants = pd.read_csv("F:\gppd_120_pr.csv")
```

In [8]:

```
global_power_plants.head()
```

Out[8]:

	system:index	capacity_mw	commissioning_year	country	country_long	estimated_generation_gwh	generation_gwh_2013
0	000000000000000315a	15.0	1942.0	USA	United States of America	685.397712	0.0
1	0000000000000026e5	1492.0	1975.0	USA	United States of America	8334.010812	0.0
2	0000000000000002fda	990.0	1962.0	USA	United States of America	5529.940150	0.0
3	0000000000000003f76	602.0	1960.0	USA	United States of America	3362.650475	0.0
4	0000000000000002def	10.0	1915.0	USA	United States of America	456.931808	0.0

5 rows × 24 columns

In [12]:

```
global_power_plants.shape
```

Out[12]:

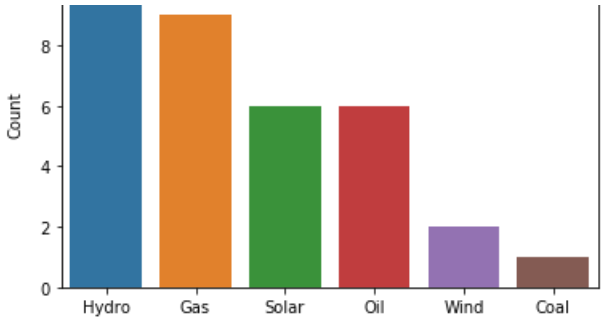
```
(35, 24)
```

In [13]:

```
# Let's check the different kinds of Power Plants based on primary Fuel used.
sns.barplot(x=global_power_plants['primary_fuel'].value_counts().index,y=global_power_plants['prima
ry_fuel'].value_counts())
plt.ylabel('Count')
```

Out[13]:

```
Text(0, 0.5, 'Count')
```



In [14]:

```
# How old are the plants
global_power_plants['commissioning_year'].value_counts()
```

Out[14]:

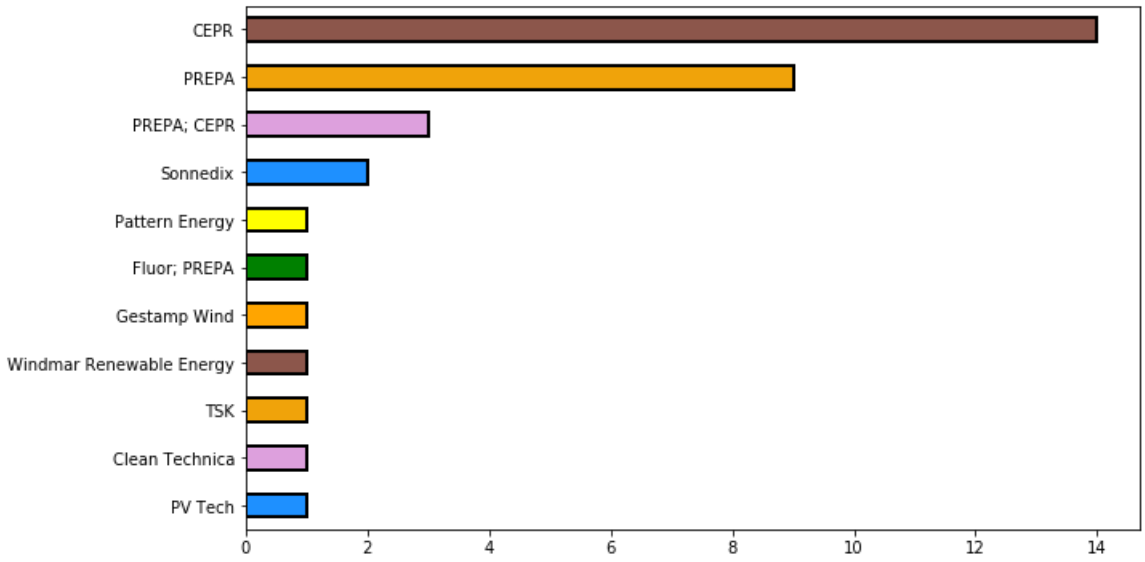
```
0.0       21
2012.0     2
1941.0     2
2015.0     1
2011.0     1
2009.0     1
1937.0     1
1929.0     1
1915.0     1
1960.0     1
1962.0     1
1975.0     1
1942.0     1
Name: commissioning_year, dtype: int64
```

In [15]:

```
# Different source of data
fig = plt.gcf()
fig.set_size_inches(10, 6)
colors = ['dodgerblue', 'plum', '#F0A30A','#8c564b','orange','green','yellow']
global_power_plants['source'].value_counts(ascending=True).plot(kind='barh',color=colors,linewidth
=2,edgecolor='black')
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x28818b1a7c8>
```



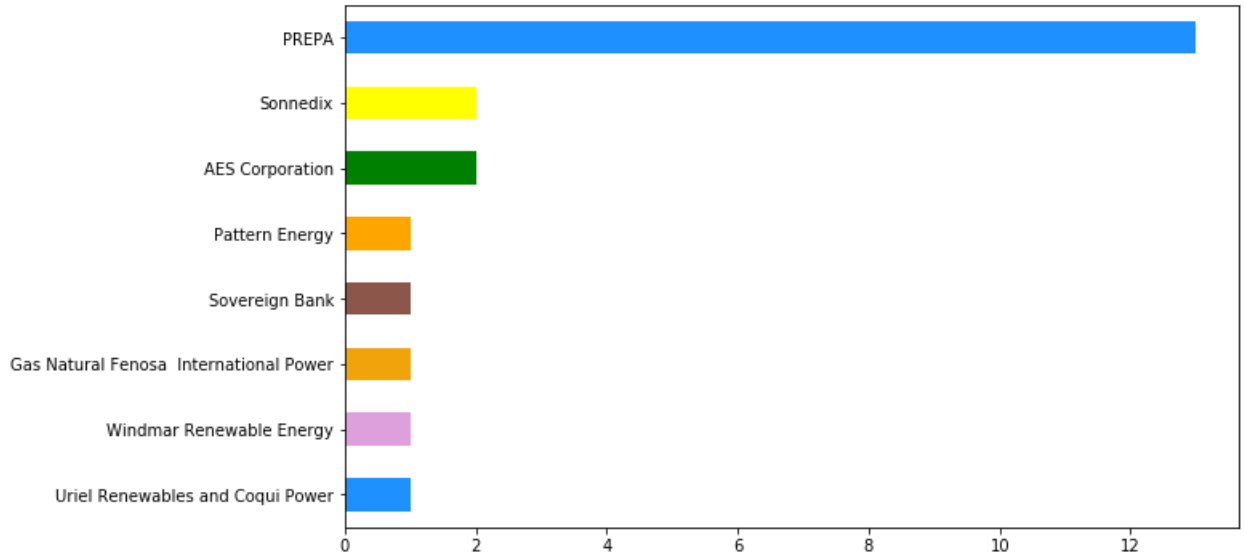
In [13]:

```
# Who owns the power plants
```

```
fig = plt.gcf()
fig.set_size_inches(10, 6)
colors = ['dodgerblue', 'plum', '#F0A30A','#8c564b','orange','green','yellow']
global_power_plants['owner'].value_counts(ascending=True).plot(kind='barh',color=colors)
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1953f6be448>
```



In [16]:

```
# Total capacity of all the plants
total_capacity_mw = global_power_plants['capacity_mw'].sum()
print('Total Installed Capacity: '+'{:.2f}'.format(total_capacity_mw) + ' MW')
```

```
Total Installed Capacity: 6148.45 MW
```

In [17]:

```
capacity = (global_power_plants.groupby(['primary_fuel'])['capacity_mw'].sum()).to_frame()
capacity = capacity.sort_values('capacity_mw',ascending=False)
capacity['percentage_of_total'] = (capacity['capacity_mw']/total_capacity_mw)*100
capacity
```

Out[17]:

primary_fuel	capacity_mw	percentage_of_total
Oil	4201.500000	68.334296
Gas	1105.000000	17.972009
Coal	454.299988	7.388854
Solar	154.650002	2.515268
Wind	124.599997	2.026527
Hydro	108.400000	1.763046

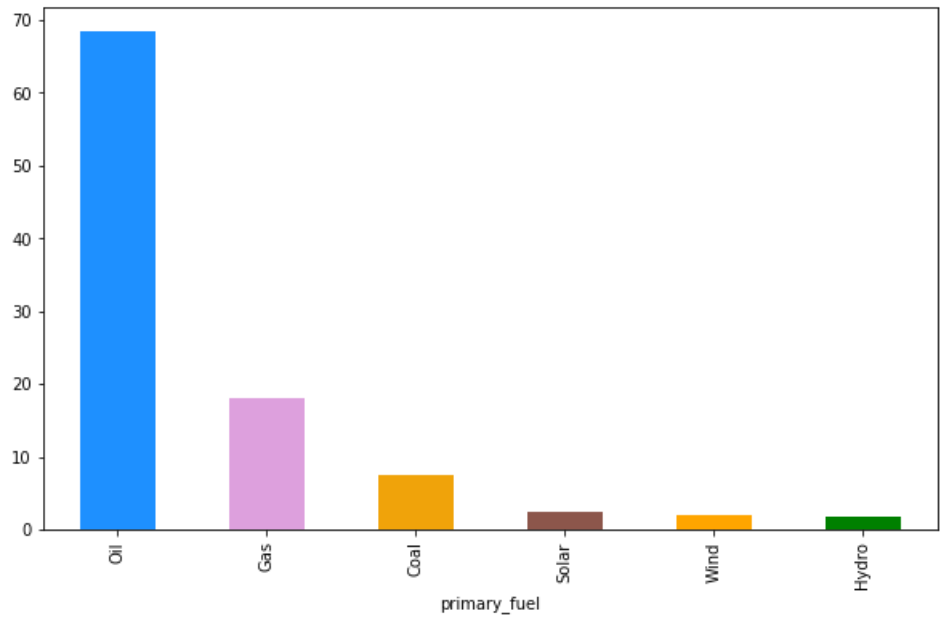
In [18]:

```
fig = plt.gcf()
fig.set_size_inches(10, 6)
colors = ['dodgerblue', 'plum', '#F0A30A','#8c564b','orange','green','yellow']
capacity['percentage_of_total'].plot(kind='bar',color=colors)
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1953f74f888>
```

<matplotlib.axes._subplots.AxesSubplot at 0x19531741668>



In [18]:

```
# Total generation of all the plants
total_gen_mw = global_power_plants['estimated_generation_gwh'].sum()
print('Total Generatation: '+'{:.2f}'.format(total_gen_mw) + ' GW')
```

```
Total Generatation: 486860.88 GW
```

In [19]:

```
generation = (global_power_plants.groupby(['primary_fuel'])
['estimated_generation_gwh'].sum()).to_frame()
generation = generation.sort_values('estimated_generation_gwh',ascending=False)
generation['percentage_of_total'] = (generation['estimated_generation_gwh']/total_gen_mw)*100
generation
```

Out[19]:

primary_fuel	estimated_generation_gwh	percentage_of_total
Coal	450562.692350	92.544444
Oil	23468.730848	4.820418
Gas	7785.243454	1.599069
Hydro	4953.140798	1.017363
Solar	68.962714	0.014165
Wind	22.104981	0.004540

In [20]:

```
import folium
import rasterio as rio
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-20-a8a0928602ce> in <module>
      1 import folium
----> 2 import rasterio as rio

ModuleNotFoundError: No module named 'rasterio'
```

In [4]:

```
# Code source: https://www.kaggle.com/paultimothymooney/overview-of-the-eie-analytics-challenge +ä
gvc b
def
plot_points_on_map(dataframe,begin_index,end_index,latitude_column,latitude_value,longitude_column
,longitude_value,zoom):
    df = dataframe[begin_index:end_index]
    location = [latitude_value,longitude_value]
    plot = folium.Map(location=location,zoom_start=zoom,tiles = 'Stamen Terrain')
    for i in range(0,len(df)):
        popup = folium.Popup(str(df.primary_fuel[i:i+1]))
        folium.Marker([df[latitude_column].iloc[i],
                       df[longitude_column].iloc[i]],
                       popup=popup,icon=folium.Icon(color='white',icon_color='red',icon
='bolt',prefix='fa',)).add_to(plot)
    return(plot)
def overlay_image_on_puerto_rico(file_name,band_layer,lat,lon,zoom):
    band = rio.open(file_name).read(band_layer)
    m = folium.Map([lat, lon], zoom_start=zoom)
    folium.raster_layers.ImageOverlay(
        image=band,
        bounds = [[18.6,-67.3,],[17.9,-65.2]],
        colormap=lambda x: (1, 0, 0, x),
    ).add_to(m)
    return m
def
split_column_into_new_columns(dataframe,column_to_split,new_column_one,begin_column_one,end_column_
one):
    for i in range(0, len(dataframe)):
        dataframe.loc[i, new_column_one] = dataframe.loc[i, column_to_split][begin_column_one:end_c
olumn_one]
    return dataframe
```

In [21]:

```
global_power_plants = split_column_into_new_columns(global_power_plants,'.geo','latitude',50,66)
global_power_plants = split_column_into_new_columns(global_power_plants,'.geo','longitude',31,48)
global_power_plants['latitude'] = global_power_plants['latitude'].astype(float)
a = np.array(global_power_plants['latitude'].values.tolist())
global_power_plants['latitude'] = np.where(a < 10, a+10, a).tolist()

lat=18.200178; lon=-66.664513 # Puerto Rico's co-ordinates
plot_points_on_map(global_power_plants,0,425,'latitude',lat,'longitude',lon,9)
```

Out[21]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [ ]:

```
```