import libraries

In [1]:

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

%matplotlib inline
```

In [2]:

```
data = 'https://raw.githubusercontent.com/WalePhenomenon/climate_change/master/fuel_ferc1.c
fuel_data = pd.read_csv(data)
```

In [3]:

```
fuel_data.to_csv('fuel_data_copy.csv', index=False) # creating a copy of the data
```

In [4]:

```
df = pd.read_csv('fuel_data_copy.csv')
```

In [5]:

```
#checking the basic information about the data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29523 entries, 0 to 29522
Data columns (total 11 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   record_id                     29523 non-null  object 
 1   utility_id_ferc1              29523 non-null  int64  
 2   report_year                   29523 non-null  int64  
 3   plant_name_ferc1              29523 non-null  object 
 4   fuel_type_code_pudl           29523 non-null  object 
 5   fuel_unit                     29343 non-null  object 
 6   fuel_qty_burned               29523 non-null  float64
 7   fuel_mmbtu_per_unit           29523 non-null  float64
 8   fuel_cost_per_unit_burned     29523 non-null  float64
 9   fuel_cost_per_unit_delivered  29523 non-null  float64
 10  fuel_cost_per_mmbtu           29523 non-null  float64
dtypes: float64(5), int64(2), object(4)
memory usage: 2.5+ MB
```

In [6]:

```
#checking the shape of the data
df.shape
```

Out[6]:

```
(29523, 11)
```

In [7]:

```
# To display the top 5 rows
df.head()
```

Out[7]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel_type_code_pudl
0	f1_fuel_1994_12_1_0_7	1	1994	rockport	coal
1	f1_fuel_1994_12_1_0_10	1	1994	rockport total plant	coal
2	f1_fuel_1994_12_2_0_1	2	1994	gorgas	coal
3	f1_fuel_1994_12_2_0_7	2	1994	barry	coal
4	f1_fuel_1994_12_2_0_10	2	1994	chickasaw	gas

In [8]:

```
# To display the bottom 5 rows
df.tail()
```

Out[8]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel_type_code_p
29518	f1_fuel_2018_12_12_0_13	12	2018	neil simpson ct #1	
29519	f1_fuel_2018_12_12_1_1	12	2018	cheyenne prairie 58%	
29520	f1_fuel_2018_12_12_1_10	12	2018	lange ct facility	
29521	f1_fuel_2018_12_12_1_13	12	2018	wygen 3 bhp 52%	
29522	f1_fuel_2018_12_12_1_14	12	2018	wygen 3 bhp 52%	

In [9]:

```
# Checking the data type
df.dtypes
```

Out[9]:

```
record_id                       object
utility_id_ferc1                 int64
report_year                      int64
plant_name_ferc1                object
fuel_type_code_pudl             object
fuel_unit                       object
fuel_qty_burned                float64
fuel_mmbtu_per_unit            float64
fuel_cost_per_unit_burned      float64
fuel_cost_per_unit_delivered   float64
fuel_cost_per_mmbtu            float64
dtype: object
```

In [10]:

```
# checking statistical data on numerial data
df.describe(include='all')
```

Out[10]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel_type_code_
count	29523	29523.000000	29523.000000	29523	2
unique	29523	NaN	NaN	2315	
top	f1_fuel_2014_12_6_0_13	NaN	NaN	big stone	
freq	1	NaN	NaN	156	
mean	NaN	118.601836	2005.806050	NaN	
std	NaN	74.178353	7.025483	NaN	
min	NaN	1.000000	1994.000000	NaN	
25%	NaN	55.000000	2000.000000	NaN	
50%	NaN	122.000000	2006.000000	NaN	
75%	NaN	176.000000	2012.000000	NaN	
max	NaN	514.000000	2018.000000	NaN	

In [11]:

```
# check all column names
df.columns
```

Out[11]:

```
Index(['record_id', 'utility_id_ferc1', 'report_year', 'plant_name_ferc1',
       'fuel_type_code_pudl', 'fuel_unit', 'fuel_qty_burned',
       'fuel_mmbtu_per_unit', 'fuel_cost_per_unit_burned',
       'fuel_cost_per_unit_delivered', 'fuel_cost_per_mmbtu'],
      dtype='object')
```

there is no irrelevant column, so no need to drop column(s)

In [12]:

```
# Rows containing duplicate data
duplicate_rows_df = df[df.duplicated()]
print("number of duplicate rows: ", duplicate_rows_df.shape)
```

```
number of duplicate rows:  (0, 11)
```

removing duplicate rows

In [13]:

```
df = df.drop_duplicates()
```

In [14]:

```
df.shape
```

Out[14]:

```
(29523, 11)
```

check unique values

In [15]:

```
df.nunique()
```

Out[15]:

```
record_id                       29523
utility_id_ferc1                  185
report_year                        25
plant_name_ferc1                 2315
fuel_type_code_pudl                 6
fuel_unit                           9
fuel_qty_burned                 26432
fuel_mmbtu_per_unit             11227
fuel_cost_per_unit_burned       19416
fuel_cost_per_unit_delivered    16675
fuel_cost_per_mmbtu             12605
dtype: int64
```

In [16]:

```
df['fuel_type_code_pudl'].unique()
```

Out[16]:

```
array(['coal', 'gas', 'nuclear', 'oil', 'waste', 'other'], dtype=object)
```

In [17]:

```
df['fuel_unit'].unique()
```

Out[17]:

```
array(['ton', 'mcf', 'kgU', 'bbl', 'gramsU', nan, 'mwdth', 'mmbtu',
       'mwhth', 'gal'], dtype=object)
```

Cleaning Data / Data Wrangling

In [18]:

```
# sorting dataset in descending order
df.sort_values(by = "record_id", ascending=False)
```

Out[18]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel_type_code_
28986	f1_fuel_2018_12_99_1_4	99	2018	sweatt ct	
28988	f1_fuel_2018_12_99_1_13	99	2018	ratcliffe	
28987	f1_fuel_2018_12_99_1_10	99	2018	daniel cc	
28983	f1_fuel_2018_12_99_0_7	99	2018	watson	
28982	f1_fuel_2018_12_99_0_2	99	2018	daniel	
...	...	...	...	...	
926	f1_fuel_1994_12_100_0_3	100	1994	independence	
930	f1_fuel_1994_12_100_0_15	100	1994	baxter wilson	
929	f1_fuel_1994_12_100_0_14	100	1994	baxter wilson	
928	f1_fuel_1994_12_100_0_11	100	1994	delta	
925	f1_fuel_1994_12_100_0_1	100	1994	independence	

29523 rows × 11 columns

grouping

In [19]:

```
fuel_data.groupby('report_year')['report_year'].count()
```

Out[19]:

```
report_year
1994    1235
1995    1201
1996    1088
1997    1094
1998    1107
1999    1050
2000    1373
2001    1356
2002    1205
2003    1211
2004    1192
2005    1269
2006    1243
2007    1264
2008    1228
2009    1222
2010    1261
2011    1240
2012    1243
2013    1199
2014    1171
2015    1093
2016    1034
2017     993
2018     951
Name: report_year, dtype: int64
```

In [20]:

```
#group by the fuel type code year and print the first entries in all the groups formed
fuel_data.groupby('fuel_type_code_pudl').first()
```

Out[20]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel
fuel_type_code_pudl					
coal	f1_fuel_1994_12_1_0_7	1	1994	rockport	
gas	f1_fuel_1994_12_2_0_10	2	1994	chickasaw	
nuclear	f1_fuel_1994_12_2_1_1	2	1994	joseph m. farley	
oil	f1_fuel_1994_12_6_0_2	6	1994	clinch river	
other	f1_fuel_1994_12_11_0_6	11	1994	w.f. wyman	
waste	f1_fuel_1994_12_9_0_3	9	1994	b.l. england	

Merging

In [21]:

```
fuel_df1 = fuel_data.iloc[0:19000].reset_index(drop=True)
fuel_df2 = fuel_data.iloc[19000:].reset_index(drop=True)

#check that the length of both dataframes sum to the expected length
assert len(fuel_data) == (len(fuel_df1) + len(fuel_df2))
```

In [22]:

```
#an inner merge will lose rows that do not match in both dataframes
pd.merge(fuel_df1, fuel_df2, how="inner")
```

Out[22]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel_type_code_pudl	fuel_unit	fuel_

In [23]:

```
#outer merge returns all rows in both dataframes
pd.merge(fuel_df1, fuel_df2, how="outer")
```

Out[23]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel_type_code_p
0	f1_fuel_1994_12_1_0_7	1	1994	rockport	
1	f1_fuel_1994_12_1_0_10	1	1994	rockport total plant	
2	f1_fuel_1994_12_2_0_1	2	1994	gorgas	
3	f1_fuel_1994_12_2_0_7	2	1994	barry	
4	f1_fuel_1994_12_2_0_10	2	1994	chickasaw	
...	...	...	...	...	
29518	f1_fuel_2018_12_12_0_13	12	2018	neil simpson ct #1	
29519	f1_fuel_2018_12_12_1_1	12	2018	cheyenne prairie 58%	
29520	f1_fuel_2018_12_12_1_10	12	2018	lange ct facility	
29521	f1_fuel_2018_12_12_1_13	12	2018	wygen 3 bhp 52%	
29522	f1_fuel_2018_12_12_1_14	12	2018	wygen 3 bhp 52%	

29523 rows × 11 columns

In [24]:

```
#removes rows from the right dataframe that do not have a match with the left
#and keeps all rows from the left

pd.merge(fuel_df1, fuel_df2, how="left")
```

Out[24]:

	record_id	utility_id_ferc1	report_year	plant_name_ferc1	fuel_type_code_
0	f1_fuel_1994_12_1_0_7	1	1994	rockport	
1	f1_fuel_1994_12_1_0_10	1	1994	rockport total plant	
2	f1_fuel_1994_12_2_0_1	2	1994	gorgas	
3	f1_fuel_1994_12_2_0_7	2	1994	barry	
4	f1_fuel_1994_12_2_0_10	2	1994	chickasaw	
...	...	...	...	...	
18995	f1_fuel_2009_12_182_1_9	182	2009	lake road	
18996	f1_fuel_2009_12_182_1_10	182	2009	lake road	
18997	f1_fuel_2009_12_182_1_13	182	2009	iatan (18%)	
18998	f1_fuel_2009_12_182_1_14	182	2009	iatan (18%)	
18999	f1_fuel_2009_12_79_0_1	79	2009	montrose	

19000 rows × 11 columns

Checking for duplicates

In [25]:

```
# number of NaN/Null values
df.isnull().sum()
```

Out[25]:

```
record_id                      0
utility_id_ferc1               0
report_year                    0
plant_name_ferc1               0
fuel_type_code_pudl            0
fuel_unit                    180
fuel_qty_burned                0
fuel_mmbtu_per_unit            0
fuel_cost_per_unit_burned      0
fuel_cost_per_unit_delivered   0
fuel_cost_per_mmbtu            0
dtype: int64
```

In [ ]:

In [26]:

```
# Replacing the missing values with "mcf".
df_replace_null = df.fillna('mcf')
```

In [27]:

```
#confirm null values been filled
df_replace_null.isnull().sum()
```

Out[27]:

```
record_id                       0
utility_id_ferc1                0
report_year                     0
plant_name_ferc1                0
fuel_type_code_pudl             0
fuel_unit                       0
fuel_qty_burned                 0
fuel_mmbtu_per_unit             0
fuel_cost_per_unit_burned       0
fuel_cost_per_unit_delivered    0
fuel_cost_per_mmbtu             0
dtype: int64
```

In [28]:

```
df_replace_null.duplicated().any() # checks for duplicate rows again
```

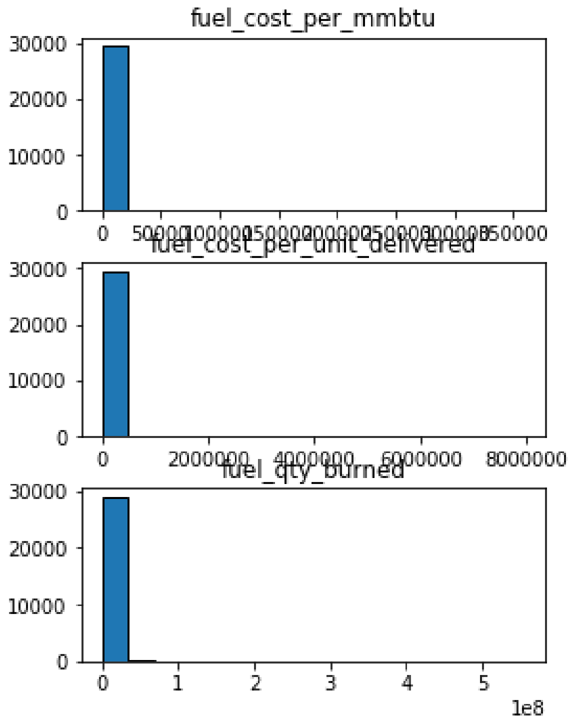
Out[28]:

```
False
```

Relationship Analysis

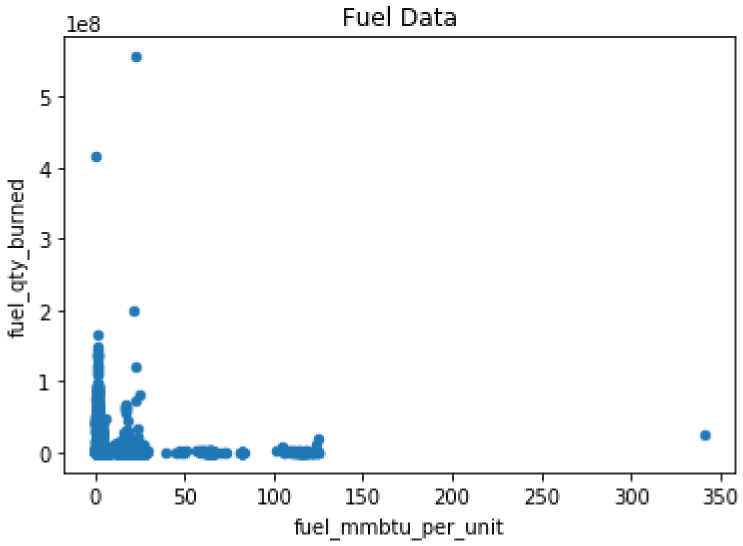
In [29]:

```
df_replace_null[['fuel_qty_burned', 'fuel_mmbtu_per_unit', 'fuel_cost_per_unit_burned',
       'fuel_cost_per_unit_delivered', 'fuel_cost_per_mmbtu']].hist(figsize=(10,6), bins=16
plt.show()
```



In [32]:

```
df_replace_null.plot(kind='scatter', x='fuel_mmbtu_per_unit', y='fuel_qty_burned', title='F
```



In [ ]: