Concept 1.5: Data Visualization and Representation in Python

- The Anscombe Quartet and the importance of visualizing data.

```python
# Import plotting library
import matplotlib.pyplot as plt

plt.figure(figsize=(7,4))
plt.xticks(rotation=90)
fuel_unit = pd.DataFrame({'unit':['BBL', 'GAL', 'GRAMSU', 'KGU', 'MCF', 'MMBTU',
'MWDTH', 'MWHTH', 'TON'],
            'count':[7998, 84, 464, 110, 11354, 180, 95, 100, 8958]})
sns.barplot(data=fuel_unit, x='unit', y='count')
plt.xlabel('Fuel Unit')
```

Because of the extreme range of the values for the fuel unit, we can plot the
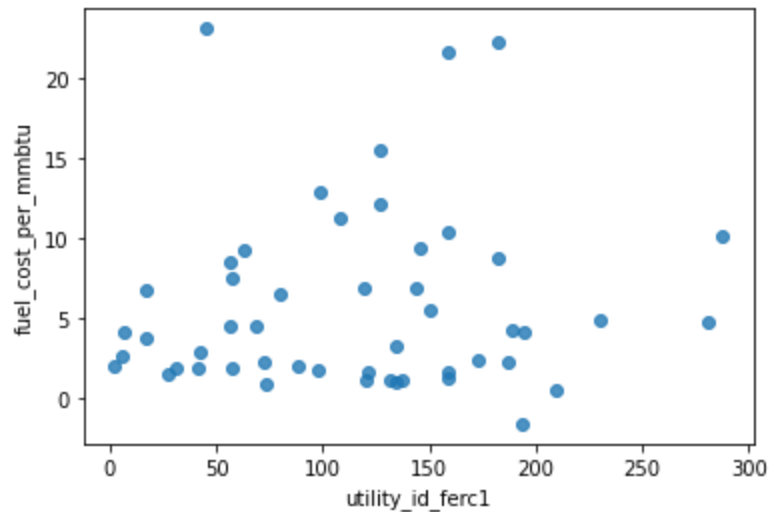barchart by taking the logarithm of the y-axis as follows:

```python
g = sns.barplot(data=fuel_unit, x='unit', y='count')
g.set_yscale("log")
g.set_ylim(1, 12000)
plt.xlabel('Fuel Unit')
```
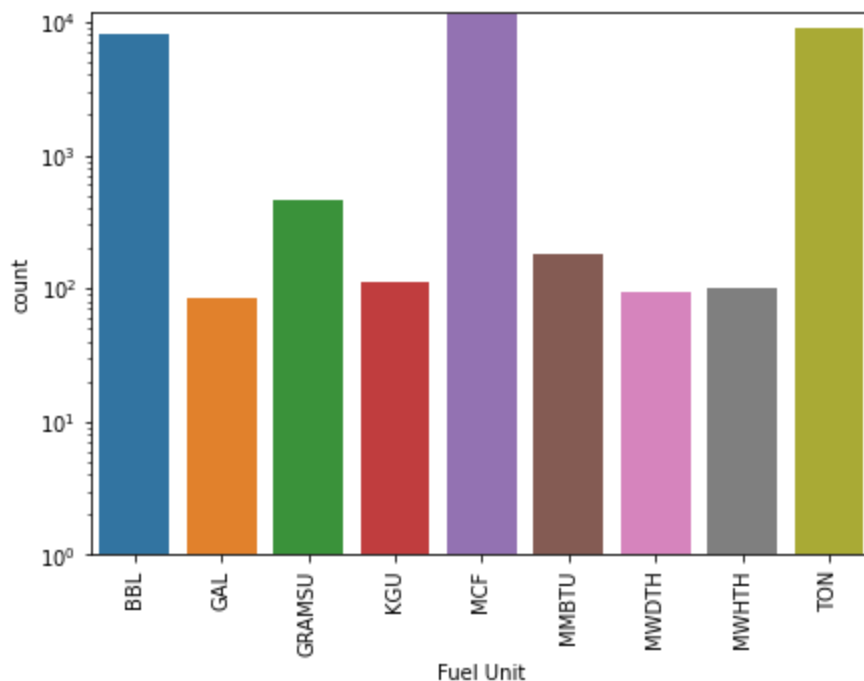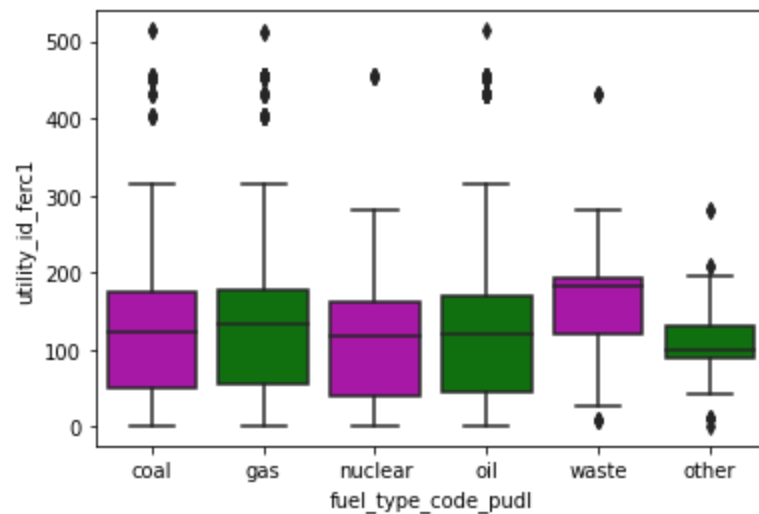
```
# Select a sample of the dataset
sample_df = fuel_data.sample(n=50, random_state=4)
sns.regplot(x=sample_df["utility_id_ferc1"], y=sample_df["fuel_cost_per_mmbtu"],
fit_reg=False)
```


- Advanced plotting: Kernel Density Estimate plots, box plots and violin plots


```
# Import plotting library
import seaborn as sns
```

```
# Box plot
```

```
sns.boxplot(x="fuel_type_code_pudl", y="utility_id_ferc1",
            palette=["m", "g"], data=fuel_data)
# KDE plot
sns.kdeplot(sample_df['fuel_cost_per_unit_burned'], shade=True, color="b")
```