Efficient green energy sources report

Alejandro Barajas, Gary Layman, Trevor Olmo, Add your names here

2025-10-14

R and Python libary set up

- **Overall goal** The goal is to identify which sources deliver the most energy per dollar while minimizing environmental impact

Data Preparation

```
#### Commented this out because we no longer need it
# energy <- eia_data(
# dir = "electricity/electric-power-operational-data",
#  data = c("ash-content", "consumption-for-eg", "consumption-for-eg-btu", "consumption-uto"
#  freq = "quarterly",
#  start = "2020",
#  end = "2025"
#)
# All columns are <char> and will need to be cleaned
```

```
# YOUR CODE HERE
```

Data Collection & Cleaning

- What sources are used?
- What cleaning steps are necessary?

We already scraped the data from the EIA website, so we can use that data directly. We just need to read it in from the csvs it has been saved in.

```
# YOUR CODE HERE
#Find all csv files in the directory, read them in, and store them in a single df
eia_csvs = glob("datasets/*.csv")
dfs = [pd.read_csv(csv) for csv in eia_csvs]
energy_df = pd.concat(dfs, ignore_index=True)
energy_df.head()
```

```
    period location  ... total-consumption-btu  total-consumption-btu-units
0  2016-Q3       IN  ...               8.93140                million MMBtu
1  2016-Q3       IN  ...               0.31604                million MMBtu
2  2016-Q3       MD  ...                   NaN                million MMBtu
3  2016-Q3       MD  ...               0.00000                million MMBtu
4  2016-Q3       MD  ...               1.65269                million MMBtu

[5 rows x 37 columns]
```

Data Organization & Conditioning

- How is the data structured?
- Are there transformations or feature engineering steps?

Here I just check number of rows and columns.

```
# YOUR CODE HERE
# Number of rows
print("Rows: ", energy_df.shape[0])
# Number of columns
print("Columns: ", energy_df.shape[1])
```

```
Rows:  530986
Columns:  37
```

Columns Types

```
#Data types
energy_df.dtypes
```

```
period                           object
location                         object
stateDescription                 object
```

```
sectorid                           int64
sectorDescription                 object
fueltypeid                        object
fuelTypeDescription               object
ash-content                      float64
ash-content-units                 object
consumption-for-eg               float64
consumption-for-eg-units          object
consumption-for-eg-btu           float64
consumption-for-eg-btu-units      object
consumption-uto                  float64
consumption-uto-units             object
consumption-uto-btu              float64
consumption-uto-btu-units         object
cost                             float64
cost-units                        object
cost-per-btu                     float64
cost-per-btu-units                object
generation                       float64
generation-units                  object
heat-content                     float64
heat-content-units                object
receipts                         float64
receipts-units                    object
receipts-btu                     float64
receipts-btu-units                object
stocks                           float64
stocks-units                      object
sulfur-content                   float64
sulfur-content-units              object
total-consumption                float64
total-consumption-units           object
total-consumption-btu            float64
total-consumption-btu-units       object
dtype: object
```

Data Storage

- Where and how is the data stored?
- Is it accessible and secure?

Find the number of missing values in each column.

```python
# YOUR CODE HERE
energy_df.isna().sum()
```

```
period                             0
location                           0
stateDescription                   0
sectorid                           0
sectorDescription                  0
fueltypeid                         0
fuelTypeDescription                0
ash-content                   189444
ash-content-units                  0
consumption-for-eg             49532
consumption-for-eg-units           0
consumption-for-eg-btu         49532
consumption-for-eg-btu-units       0
consumption-uto                49532
consumption-uto-units              0
consumption-uto-btu            49532
consumption-uto-btu-units          0
cost                          433700
cost-units                         0
cost-per-btu                  485548
cost-per-btu-units                 0
generation                     15880
generation-units                   0
heat-content                  170783
heat-content-units                 0
receipts                      170783
receipts-units                     0
receipts-btu                  166205
receipts-btu-units                 0
stocks                        507804
stocks-units                       0
sulfur-content                187923
sulfur-content-units               0
total-consumption              49532
total-consumption-units            0
total-consumption-btu          49532
total-consumption-btu-units        0
dtype: int64
```

Shows the fuel types that have missing cost values

```python
# YOUR CODE HERE
missing_cost = energy_df[energy_df["cost"].isna()]
missing_cost["fuelTypeDescription"].unique()
```

```
array(['renewable', 'petroleum', 'residual fuel oil',
       'natural gas & other gases', 'natural gas', 'fossil fuels',
       'distillate fuel oil', 'all coal products',
       'coal, excluding waste coal', 'bituminous coal',
       'waste oil and other oils', 'onshore wind turbine', 'biomass',
       'other renewables', 'wood and wood wastes', 'wind',
       'renewable waste products', 'bituminous coal and synthetic coal',
       'all renewables', 'all fuels', 'solar', 'subbituminous coal',
       'solar photovoltaic', 'refined coal', 'petroleum liquids', 'other',
       'biogenic municipal solid waste', 'municiapl landfill gas',
       'landfill gas', 'conventional hydroelectric', 'petroleum coke',
       'nuclear', 'lignite coal', 'hydro-electric pumped storage',
       'solar thermal', 'waste coal', 'other gases', 'geothermal',
       'anthracite coal', 'offshore wind turbine'], dtype=object)
```

Finds the counts for each fuel type

```python
# YOUR CODE HERE
missing_cost["fuelTypeDescription"].value_counts()
```

```
fuelTypeDescription
biomass                                  38855
all fuels                                23127
fossil fuels                             21748
natural gas & other gases                20655
renewable                                20351
all renewables                           19817
petroleum                                19020
distillate fuel oil                      18418
renewable waste products                 15839
natural gas                              15181
petroleum liquids                        13586
coal, excluding waste coal               12871
solar photovoltaic                       12593
solar                                    12580
```

```
other                                  12218
municiapl landfill gas                 11917
landfill gas                           11126
conventional hydroelectric             11015
bituminous coal                        10502
bituminous coal and synthetic coal     10489
wind                                   10406
onshore wind turbine                   10318
other renewables                       10218
wood and wood wastes                    9830
all coal products                       9372
residual fuel oil                       8522
waste oil and other oils                8066
subbituminous coal                      7656
nuclear                                 4644
biogenic municipal solid waste          4461
other gases                             4277
refined coal                            3267
hydro-electric pumped storage           2614
waste coal                              2071
petroleum coke                          1986
geothermal                              1487
lignite coal                            1338
solar thermal                            796
offshore wind turbine                    236
anthracite coal                          227
Name: count, dtype: int64
```

Exploratory Data Analysis (EDA)

- What are the high-level patterns?
- What intuitive insights emerge?
- Is the data consistent across sources?
- What assumptions are we making?
- What new questions arise from the EDA?

Note: Be mindful of *confirmation bias*—avoid interpreting data only to support your initial hypothesis.

```
# YOUR CODE HERE
```

```
# YOUR CODE HERE
```

Model Planning

- **Model Type**: Classification, clustering, regression—what fits best?
- **Success/Failure Definitions**: Refine what constitutes a good or bad outcome.
- **EDA for Modeling**:
  - Identify relevant variables
  - Explore correlations
  - Apply domain knowledge
  - Verify assumptions
  - Prototype modeling ideas

```
# YOUR CODE HERE
```

```
# YOUR CODE HERE
```

Model Building

- What models are being trained?
- How are they tested?
- What metrics are used to evaluate performance?

```
# YOUR CODE HERE
```

```
# YOUR CODE HERE
```

Communicating Results

- Are the results **robust** across different scenarios?
- Are they **statistically significant**?
- Why do these results **matter** to stakeholders?
- How do they compare to your definitions of success and failure?

Operationalization

- **Presentation**:
  - Who is the audience?
  - Is the code and analysis well-documented and replicable?
- **Deployment**:
  - Are models deployed on live data?
  - Are they behaving as expected?