

30538 Final Project

Mohamed Mohamed

1. Data wrangling

```
import pandas as pd

# Loading the datasets
primary_energy = pd.read_csv("C:/Users/mmmoh/DPPS
    ↪ Python/PSet/DPPP_Python_programming_2_Final_Project/primary-energy-cons.csv")
electricity_generation = pd.read_csv("C:/Users/mmmoh/DPPS
    ↪ Python/PSet/DPPP_Python_programming_2_Final_Project/all-data.csv",
    ↪ encoding='latin1') # Specify encoding

# Filtering the data for Somalia
primary_energy_somalia = primary_energy.query("Country == 'Somalia' and Year
    ↪ <= 2021")
electricity_generation_somalia = electricity_generation.query("Country ==
    ↪ 'Somalia' and Year <= 2021")

#primary_energy_somalia.to_csv("primary_energy_somalia.csv", index=False)
#electricity_generation_somalia.to_csv("electricity_generation_somalia.csv",
    ↪ index=False)

primary_energy_somalia = primary_energy_somalia[['Year', 'Primary energy
    ↪ consumption (TWh)']]
electricity_generation_somalia = electricity_generation_somalia[['Year',
    ↪ 'Electricity Generation (GWh)', 'Technology']]

# Putting together the datasets with a full outer join
merged_data = pd.merge(
    primary_energy_somalia,
    electricity_generation_somalia,
    on="Year",
```

```

        how="outer"
    )

# Filtering the merged data
merged_data = merged_data.query("Year <= 2021")

# Handle missing data
merged_data['Electricity Generation (GWh)'] = merged_data['Electricity
↳ Generation (GWh)'].fillna(0)
merged_data.fillna('NA', inplace=True) # Use 'NA' for all other missing
↳ values

# Saving the merged dataset
merged_data.to_csv("somalia_energy_with_technology.csv", index=False)

```

2. Plotting

```

import altair as alt
import pandas as pd

# Loading merged dataset
data = pd.read_csv("somalia_energy_with_technology.csv")

# Ensuring numeric conversion for 'Electricity Generation (GWh)' and 'Primary
↳ energy consumption (TWh)'
data['Electricity Generation (TWh)'] = data['Electricity Generation
↳ (GWh)'].apply(
    lambda x: float(x) / 1000 if x != 'NA' else None
)
data['Primary energy consumption (TWh)'] = pd.to_numeric(data['Primary energy
↳ consumption (TWh)'], errors='coerce')

# Creating variable 'Actual Electricity Generation (TWh)'
data['Actual Electricity Generation (TWh)'] = data['Electricity Generation
↳ (TWh)'].apply(
    lambda x: x if x != 0 else None
)

# Aggregating data by year
aggregated_data = data.groupby('Year', as_index=False).agg({
    'Primary energy consumption (TWh)': 'sum',
    'Actual Electricity Generation (TWh)': 'sum'
})

```

```

})

# Saving the aggregated dataset
aggregated_data.to_csv("somalia_aggregated_energy.csv", index=False)

# Reshaping data
melted_data = pd.melt(aggregated_data, id_vars=['Year'],
    value_vars=['Primary energy consumption (TWh)', 'Actual Electricity
↪ Generation (TWh)'],
    var_name='Metric', value_name='Value')

# Creating line plot
line_plot = alt.Chart(melted_data).mark_line().encode(
    x=alt.X('Year:O', title="Year"),
    y=alt.Y('Value:Q', title="Value (TWh)"),
    color=alt.Color('Metric:N', title="Metric"),
    tooltip=['Year', 'Metric', 'Value']
).properties(
    title='Somalia Aggregated Energy Metrics Over Time',
    width=600,
    height=400
)

line_plot.save('chart.html')

# Loadint merged dataset
bar_data = pd.read_csv("somalia_energy_with_technology.csv")

# Cleaning NA values in Technology column
bar_data['Technology'] = bar_data['Technology'].fillna('Unknown')

# Ensuring numeric conversion for 'Electricity Generation (GWh)'
bar_data['Electricity Generation (TWh)'] = bar_data['Electricity Generation
↪ (GWh)'].apply(
    lambda x: float(x) / 1000 if x != 'NA' else None
)

# Filtering data for years after 2000
bar_data = bar_data[bar_data['Year'] >= 2000]

# Aggregating data by year and technology

```

```

aggregated_data = bar_data.groupby(['Year', 'Technology'],
    ↪ as_index=False).agg({
    'Electricity Generation (TWh)': 'sum'
})

```

```

# Reshaping data
melted_data = pd.melt(aggregated_data, id_vars=['Year', 'Technology'],
    value_vars=['Electricity Generation (TWh)'],
    var_name='Metric', value_name='Value')

# Creating bar chart
bar_chart = alt.Chart(melted_data).mark_bar().encode(
    x=alt.X('Year:O', title="Year", axis=alt.Axis(labelAngle=-45)),
    y=alt.Y('Value:Q',
        title="Electricity Generation (TWh)",
        scale=alt.Scale(domain=[0, 0.40])),
    color=alt.Color('Technology:N',
        title="Technology",
        scale=alt.Scale(scheme='category20')),
    tooltip=['Year', 'Technology', 'Value']
).properties(
    title='Electricity Generation by Technology in Somalia (2000-2021)',
    width=600,
    height=400
)

bar_chart.save('bar.html')

```