**Application and ETL Workflow Analysis**

**Application Overview**

The "US Baby Name Dashboard" is a Streamlit-based application designed to visualize baby names data from multiple perspectives. The dashboard leverages various data visualization libraries like Altair, Plotly, and Folium to create interactive and informative charts and maps. The primary features of the dashboard include:

1. **Yearly Trends Analysis**: Allows users to see the trends of baby names over selected years.
2. **Top Names Visualization**: Provides insights into the most popular names by state and overall trends for the past century.
3. **Biblical Names Percentage**: Displays the percentage of selected biblical names among all names chosen.
4. **Heatmap and Choropleth Maps**: Visualizes the distribution and count of names across different states and years.
5. **Unisex Names Identification**: Identifies and displays names used for both males and females.

**ETL Workflow**

The ETL (Extract, Transform, Load) workflow for this application involves several steps to ensure that the data is correctly processed and visualized. Below is a detailed breakdown:

**1. Extract**

The data extraction process involves reading multiple CSV files containing baby names data and additional datasets, such as biblical names and actor names:

@st.cache\_data

def load\_data():

try:

df\_top\_baby\_names\_yr = pd.read\_csv("Baby\_Names\_Start/top\_five\_names\_per\_state.csv")

biblical\_names\_df = pd.read\_csv("Baby\_Names\_Start/biblical\_names.csv")

df\_top\_five\_names\_per\_state = pd.read\_csv("Baby\_Names\_Start/top\_five\_names\_per\_state.csv")

top\_baby\_names\_100yrs\_df = pd.read\_csv("Baby\_Names\_Start/2top\_baby\_names.csv")

return df\_top\_baby\_names\_yr, biblical\_names\_df, df\_top\_five\_names\_per\_state, top\_baby\_names\_100yrs\_df

except Exception as e:

st.error(f"Error loading data: {e}")

return None, None, None, None

**2. Transform**

The transformation process includes several key operations:

* **Aggregation**: Aggregating data to get the top names over the last 100 years.

df\_aggregated = df\_top\_baby\_names\_yr.groupby('Name')['Count'].sum().reset\_index()

df\_aggregated = df\_aggregated.sort\_values(by='Count', ascending=False).head(5)

* **Filtering**: Filtering data based on user selections from the sidebar.

df\_filtered = df\_top\_baby\_names\_yr[df\_top\_baby\_names\_yr["Year"] == selected\_year]

if selected\_names:

df\_filtered = df\_filtered[df\_filtered["Name"].isin(selected\_names)]

* **Percentage Calculation**: Calculating the percentage of biblical names among the selected names.

selected\_biblical\_names = set(df\_filtered['Name']).intersection(set(biblical\_names\_df['Name']))

percent\_biblical = len(selected\_biblical\_names) / len(set(df\_filtered['Name'])) \* 100 if df\_filtered['Name'].any() else 0

* **Data Grouping**: Grouping and summarizing data for choropleth maps and heatmaps.

state\_data = df.groupby('State')['Count'].sum().reset\_index()

* **Unisex Names Identification**: Finding names that are both male and female.

def find\_unisex\_names(df):

male\_names = df[df["sex"] == "M"]["first\_name"].unique()

female\_names = df[df["sex"] == "F"]["first\_name"].unique()

unisex\_names = list(set(male\_names) & set(female\_names))

return unisex\_names

**3. Load**

The load phase involves presenting the transformed data through various interactive visualizations:

* **Donut Chart**: Showing the percentage of biblical names.

donut\_chart = make\_donut\_chart(percent\_biblical, selected\_color\_theme)

st.altair\_chart(donut\_chart, use\_container\_width=True)

* **Heatmap**: Visualizing the distribution of names across states and years.

heatmap\_chart = make\_heatmap(df\_filtered, input\_y, input\_x, input\_color, input\_color\_theme)

st.altair\_chart(heatmap\_chart, use\_container\_width=True)

* **Choropleth Map**: Displaying the count of names by state.

choropleth\_map = make\_choropleth(df\_filtered)

choropleth\_map.save('choropleth\_map.html')

with open('choropleth\_map.html', 'r', encoding='utf-8') as f:

html\_content = f.read()

html(html\_content, height=600)

* **Bar Chart and Trends**: Visualizing top names by state and yearly trends.

names\_chart = alt.Chart(top\_names).mark\_bar().encode(

x='Name:N',

y='Count:Q',

color=alt.Color('Count:Q', scale=alt.Scale(scheme=input\_color\_theme)),

tooltip=['Name', 'Count']

).properties(height=300, width=400)

st.altair\_chart(names\_chart, use\_container\_width=True)

In conclusion, this Streamlit application provides a comprehensive analysis and interactive visualization of baby names in the US, leveraging a well-structured ETL process to ensure data integrity and usability. Streamlit’s capabilities go far beyond what is done here and will be a proven tool in the future.