

ML Volunteering Work

Visualization

AIE23134

This code is a Python script designed to visualize specific statistical and signal data from an Excel file named `group_stats.xlsx`. It first loads the Excel file and reads three sheets: **activeChannels**, **correlations**, and **L_condition**. From the `activeChannels` sheet, it creates a bar plot showing **T-values** across different brain channels, differentiated by group. A red dashed line is added at a T-value of 2 to indicate a common threshold for statistical significance. This plot helps to visually compare which channels show significant differences between groups.

Next, the script extracts a **Spearman correlation matrix** from the `correlations` sheet, focusing specifically on the HbO (oxygenated hemoglobin) metrics: Peak, Time to Peak, and AUC. This matrix is plotted as a heatmap to show the relationships among different variables related to brain activity. Lastly, from the `L_condition` sheet, it generates a **line plot of mean values** (HbO, HbR, HbT) across different channels. This helps visualize how mean hemoglobin values vary across brain regions under the `L_condition` experimental setup. Each plot is saved as a PNG image for documentation or reporting.

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Set your Excel file path
```

```
file_path = r"C:\Users\vanga\OneDrive - Amrita vishwa  
vidyapeetham\Documents\group_stats.xlsx"
```

```
# Load Excel file
```

```
excel_data = pd.ExcelFile(file_path)
```

```
# --- Visualization 1: T-values from activeChannels ---
```

```
active_channels = excel_data.parse("activeChannels")
```

```
plt.figure(figsize=(12, 6))
```

```
sns.barplot(data=active_channels, x="Channel", y="tval", hue="Group")
```

```
plt.axhline(y=2, color='red', linestyle='--', label="Approx. significance threshold")
```

```
plt.title("T-values by Channel and Group (activeChannels)")
```

```
plt.ylabel("T-value")
```

```
plt.xticks(rotation=90)
```

```
plt.legend()
```

```
plt.tight_layout()
```

```
plt.savefig("activeChannels_tval_plot.png")
```

```
plt.show()
```

```
# --- Visualization 2: Correlation heatmap for HbO ---
```

```
correlations = excel_data.parse("correlations")
```

```
# Extract and reshape correlation table
```

```
corr_hbo = correlations.iloc[2:5, 2:9].copy()
```

```
corr_hbo.columns = correlations.iloc[1, 2:9]
```

```
corr_hbo.index = ['Peak', 'Time to peak', 'AUC']
```

```
corr_hbo = corr_hbo.apply(pd.to_numeric, errors='coerce')
```

```
plt.figure(figsize=(10, 5))
```

```
sns.heatmap(corr_hbo, annot=True, cmap="coolwarm", center=0)

plt.title("Spearman Correlation (HbO)")

plt.tight_layout()

plt.savefig("correlation_hbo_heatmap.png")

plt.show()
```

--- Visualization 3: Line plot of Mean values across channels (L_condition) ---

```
l_condition = excel_data.parse("L_condition")
```

```
# Fix column names and extract relevant data
```

```
# Mean values seem to be in columns 11-13: HbO, HbR, HbT
```

```
l_condition_clean = l_condition.iloc[1:, [0, 10, 11, 12]]
```

```
l_condition_clean.columns = ['Channel', 'HbO', 'HbR', 'HbT']
```

```
l_condition_clean = l_condition_clean.dropna()
```

```
l_condition_clean[['HbO', 'HbR', 'HbT']] = l_condition_clean[['HbO', 'HbR',  
'HbT']].apply(pd.to_numeric, errors='coerce')
```

```
plt.figure(figsize=(12, 6))
```

```
for col in ['HbO', 'HbR', 'HbT']:
```

```
    plt.plot(l_condition_clean['Channel'], l_condition_clean[col], marker='o', label=col)
```

```
plt.title("Mean Values across Channels (L_condition)")
```

```
plt.xlabel("Channel")
```

```
plt.ylabel("Mean")
```

```
plt.xticks(rotation=90)
```

```
plt.legend()
```

```
plt.tight_layout()
```

```
plt.savefig("l_condition_mean_plot.png")
```

```
plt.show()
```

