feature-extraction-time-based

June 1, 2025

[ ]:
```python
import pandas as pd
import numpy as np
from scipy import __version__ as scipy_version
from scipy.stats import skew, kurtosis
from scipy.signal import welch
import antropy as ant
import matplotlib.pyplot as plt
import seaborn as sns
```

0.1 Data Exploration

[2]:
```python
#importing preprocessed dataset
df = pd.read_parquet('eeg_dataset_cleaned.parquet')
print(df.head())
```

```
   # FP1-F7    F7-T7         T7-P7     P7-O1    FP1-F3     F3-C3     C3-P3  \
0 -0.000007 -0.000009 -2.363858e-05 -0.000009 -0.000005 -0.000024 -0.000017
1 -0.000075  0.000126 -4.102564e-06 -0.000015  0.000043 -0.000034  0.000006
2 -0.000013  0.000024  1.953602e-07  0.000038 -0.000010 -0.000010  0.000013
3 -0.000021  0.000009 -5.958486e-05 -0.000074 -0.000021  0.000004 -0.000039
4  0.000119  0.000047 -9.826618e-05  0.000078  0.000141  0.000128 -0.000092

      P3-O1    FP2-F4         F4-C4  …           --4     PZ-OZ   FC1-Ref  \
0 -0.000002 -0.000013 -3.379731e-05  …  1.953602e-07 -0.000009  0.000003
1  0.000015  0.000032 -3.301587e-05  …  0.000000e+00 -0.000009  0.000003
2  0.000056 -0.000035  1.152625e-05  … -2.500000e-07 -0.000009  0.000003
3 -0.000088 -0.000017  1.934066e-05  … -5.000000e-07 -0.000009  0.000003
4 -0.000028  0.000083  1.953602e-07  … -7.500000e-07 -0.000009  0.000003

    FC2-Ref  FC5-Ref   FC6-Ref       CP1-Ref   CP2-Ref   CP5-Ref   CP6-Ref
0  0.000007  0.00001 -0.000019  4.884005e-07  0.000001  0.000012 -0.000008
1  0.000007  0.00001 -0.000019  4.884005e-07  0.000001  0.000012 -0.000008
2  0.000007  0.00001 -0.000019  4.884005e-07  0.000001  0.000012 -0.000008
3  0.000007  0.00001 -0.000019  4.884005e-07  0.000001  0.000012 -0.000008
4  0.000007  0.00001 -0.000019  4.884005e-07  0.000001  0.000012 -0.000008

[5 rows x 38 columns]
```

[3]:
```python
print(df.shape)
print(df.columns)
print(df.describe())
```

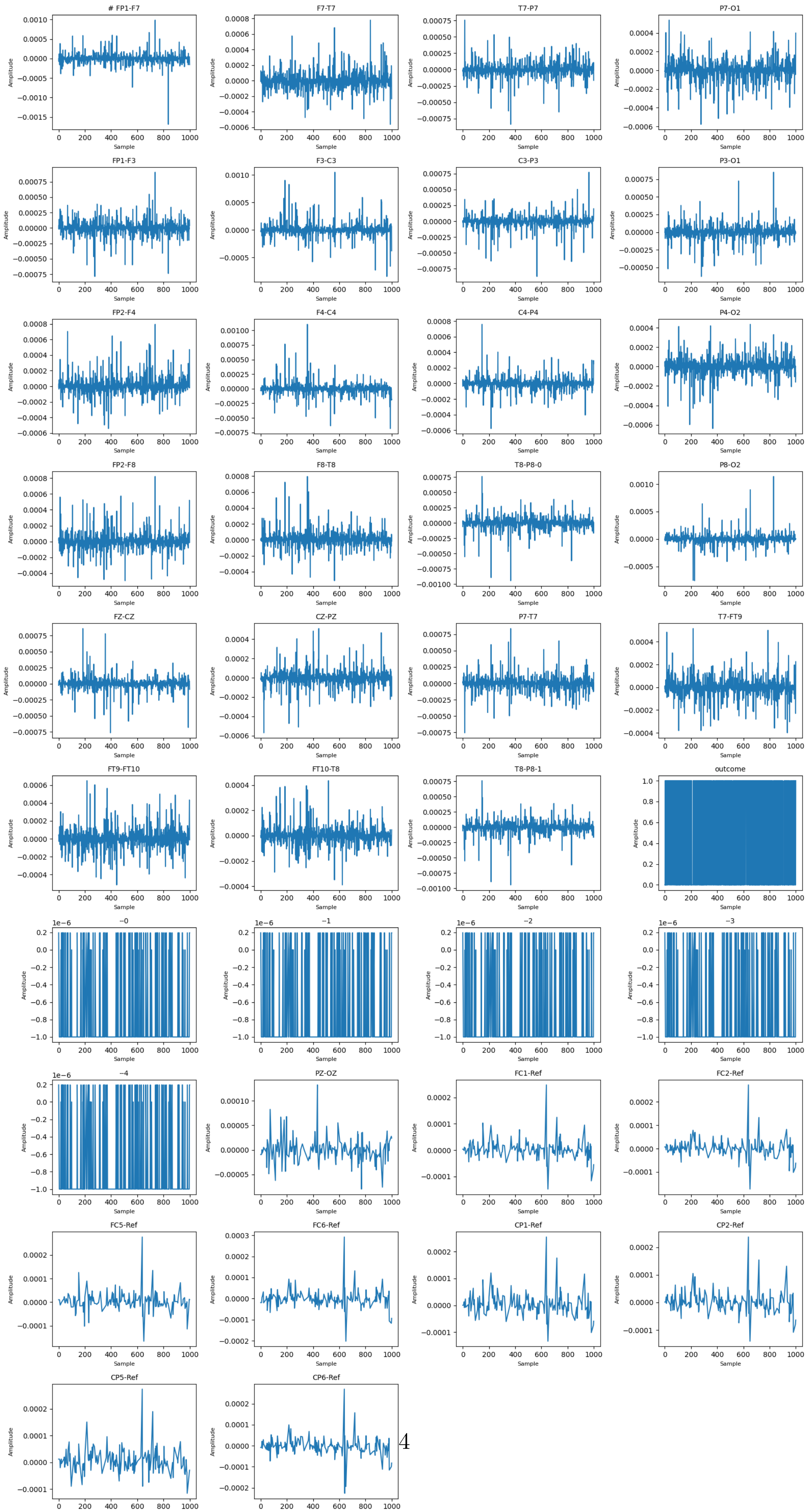
```
(5845760, 38)
Index(['# FP1-F7', 'F7-T7', 'T7-P7', 'P7-O1', 'FP1-F3', 'F3-C3', 'C3-P3',
       'P3-O1', 'FP2-F4', 'F4-C4', 'C4-P4', 'P4-O2', 'FP2-F8', 'F8-T8',
       'T8-P8-0', 'P8-O2', 'FZ-CZ', 'CZ-PZ', 'P7-T7', 'T7-FT9', 'FT9-FT10',
       'FT10-T8', 'T8-P8-1', 'outcome', '--0', '--1', '--2', '--3', '--4',
       'PZ-OZ', 'FC1-Ref', 'FC2-Ref', 'FC5-Ref', 'FC6-Ref', 'CP1-Ref',
       'CP2-Ref', 'CP5-Ref', 'CP6-Ref'],
      dtype='object')
           # FP1-F7         F7-T7         T7-P7         P7-O1        FP1-F3  \
count  5.845760e+06  5.845760e+06  5.845760e+06  5.845760e+06  5.845760e+06
mean   2.967544e-07  3.122452e-07  2.507915e-07  2.667591e-07  3.020567e-07
std    9.744611e-05  1.025508e-04  9.661989e-05  8.905929e-05  1.004107e-04
min   -2.391404e-03 -1.709988e-03 -1.611136e-03 -1.511795e-03 -1.662320e-03
25%   -2.598291e-05 -2.832723e-05 -2.686203e-05 -2.442002e-05 -2.881563e-05
50%   -1.953602e-07  1.953602e-07  2.930403e-07  4.884005e-07 -4.884005e-07
75%    2.363858e-05  2.783883e-05  2.759463e-05  2.481074e-05  2.598291e-05
max    2.067497e-03  1.663492e-03  1.686154e-03  1.296215e-03  1.631453e-03

              F3-C3         C3-P3         P3-O1        FP2-F4         F4-C4  \
count  5.845760e+06  5.845760e+06  5.845760e+06  5.845760e+06  5.845760e+06
mean   2.725279e-07  2.620222e-07  2.862242e-07  2.723094e-07  2.700204e-07
std    1.064859e-04  8.168193e-05  8.610181e-05  1.018910e-04  9.781508e-05
min   -1.864322e-03 -2.092894e-03 -1.647082e-03 -1.344664e-03 -2.055678e-03
25%   -2.363858e-05 -1.836386e-05 -2.285714e-05 -2.832723e-05 -2.188034e-05
50%    2.930403e-07  4.884005e-07  2.930403e-07 -1.953602e-07  3.907204e-07
75%    2.402930e-05  1.894994e-05  2.300366e-05  2.598291e-05  2.207570e-05
max    2.177289e-03  2.003810e-03  1.763516e-03  1.800147e-03  1.729524e-03

       …           --4         PZ-OZ       FC1-Ref       FC2-Ref  \
count  …  5.845760e+06  5.845760e+06  5.845760e+06  5.845760e+06
mean   … -7.924557e-07  5.057656e-07  4.500398e-07  4.526680e-07
std    …  3.928280e-07  2.063459e-05  3.003764e-05  3.132907e-05
min    … -1.000000e-06 -2.358974e-04 -9.020757e-04 -8.962149e-04
25%    … -1.000000e-06 -1.025641e-05 -1.064713e-05 -1.195444e-05
50%    … -1.000000e-06  4.884005e-07  1.074481e-06  1.139601e-06
75%    … -8.007733e-07  1.117216e-05  1.318681e-05  1.440781e-05
max    …  1.953602e-07  3.003663e-04  7.995116e-04  5.142857e-04

            FC5-Ref       FC6-Ref       CP1-Ref       CP2-Ref       CP5-Ref  \
count  5.845760e+06  5.845760e+06  5.845760e+06  5.845760e+06  5.845760e+06
mean   4.191789e-07  4.510952e-07  4.601107e-07  4.729205e-07  4.240870e-07
std    3.214689e-05  3.276566e-05  3.570799e-05  3.566193e-05  3.876480e-05
min   -8.864469e-04 -8.991453e-04 -9.089133e-04 -9.098901e-04 -9.020757e-04
25%   -1.179138e-05 -1.221001e-05 -1.465201e-05 -1.465201e-05 -1.622575e-05
```

```
50%    1.074481e-06  1.058201e-06  1.221001e-06  1.198801e-06  1.302401e-06
75%    1.416361e-05  1.455433e-05  1.728938e-05  1.716916e-05  1.904762e-05
max    5.054945e-04  5.250305e-04  5.211233e-04  7.272283e-04  5.172161e-04

            CP6-Ref
count  5.845760e+06
mean   4.357542e-07
std    3.860677e-05
min   -1.549695e-03
25%   -1.427215e-05
50%    1.074481e-06
75%    1.692164e-05
max    1.597558e-03

[8 rows x 38 columns]
```

0.2 Interpretation of Dataset Exploration Output

- **Dataset Shape**: `(5845760, 38)` indicates 5.8 million time points across 38 EEG channels, suggesting high-resolution data.
- **Column Names**: Lists 38 channels (e.g., FP1-F7, PZ-OZ, FC1-Ref) and an `outcome` column, likely for classification tasks.
- **Zero-Centered Means**: Near-zero means (e.g., 2.967544e-07 for FP1-F7) confirm baseline-corrected EEG signals.
- **Signal Variability**: Standard deviations vary (e.g., 1.06e-04 for F3-C3 vs. 2.06e-05 for PZ-OZ), indicating diverse channel dynamics.
- **Microvolt Range**: Min/max values (e.g., -2.39e-03 to 2.06e-03 for FP1-F7) confirm typical EEG amplitudes with potential outliers.

[4]:
```
channels = df.columns
n_channels = len(channels)

n_cols = 4
n_rows = (n_channels + n_cols - 1) // n_cols

plt.figure(figsize=(4 * n_cols, 3 * n_rows))

for i, ch in enumerate(channels):
    plt.subplot(n_rows, n_cols, i + 1)
    plt.plot(df[ch].values[:1000])
    plt.title(ch, fontsize=10)
    plt.xlabel("Sample", fontsize=8)
    plt.ylabel("Amplitude", fontsize=8)
    plt.tight_layout()

plt.show()
```



FP1-F7
F7-T7
T7-P7
P7-O1
FP1-F3
F3-C3
C3-P3
P3-O1
FP2-F4
F4-C4
C4-P4
P4-O2
FP2-F8
F8-T8
T8-P8-0
P8-O2
FZ-CZ
CZ-PZ
P7-T7
T7-FT9
FT9-FT10
FT10-T8
T8-P8-1
outcome
−0
−1
−2
−3
−4
PZ-OZ
FC1-Ref
FC2-Ref
FC5-Ref
FC6-Ref
CP1-Ref
CP2-Ref
CP5-Ref
CP6-Ref
Amplitude
Sample

0.3 Interpretation of Feature Extraction Output

- **Feature DataFrame**: Displays `features_df` with 152 columns (mean, std, skew, kurtosis for each of 38 channels).
- **Mean Values**: Near-zero means (e.g., 2.967544e-07 for FP1-F7) confirm baseline-corrected EEG signals.
- **Std Variability**: Standard deviations (e.g., 0.000097 for FP1-F7) indicate varying signal fluctuations across channels.
- **Skew and Kurtosis**: High skew (e.g., 0.48946 for FP1-F7) and kurtosis (e.g., 64.627033 for CP6-Ref) suggest non-Gaussian distributions and potential outliers.
- **Single Row**: Represents features for one time window, suitable for further analysis or visualization.

1 Time-based feature extraction

[5]:
```python
features = {}

for col in df.columns:
    signal = df[col].values
    features[f'{col}_mean'] = np.mean(signal)
    features[f'{col}_std'] = np.std(signal)
    features[f'{col}_skew'] = skew(signal)
    features[f'{col}_kurtosis'] = kurtosis(signal)
```

[6]:
```python
features_df = pd.DataFrame([features])
```

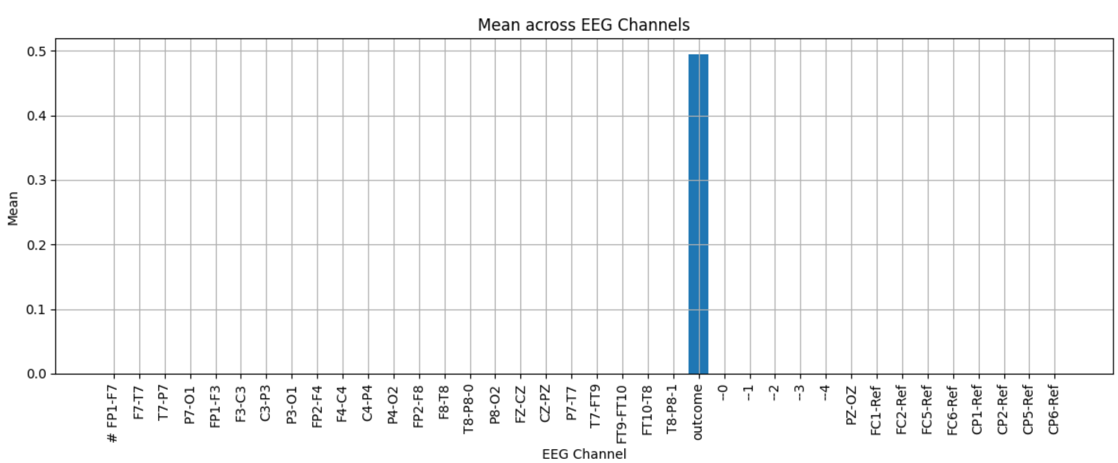
[7]:
```python
print("Extracted time-based features:")
display(features_df)
```

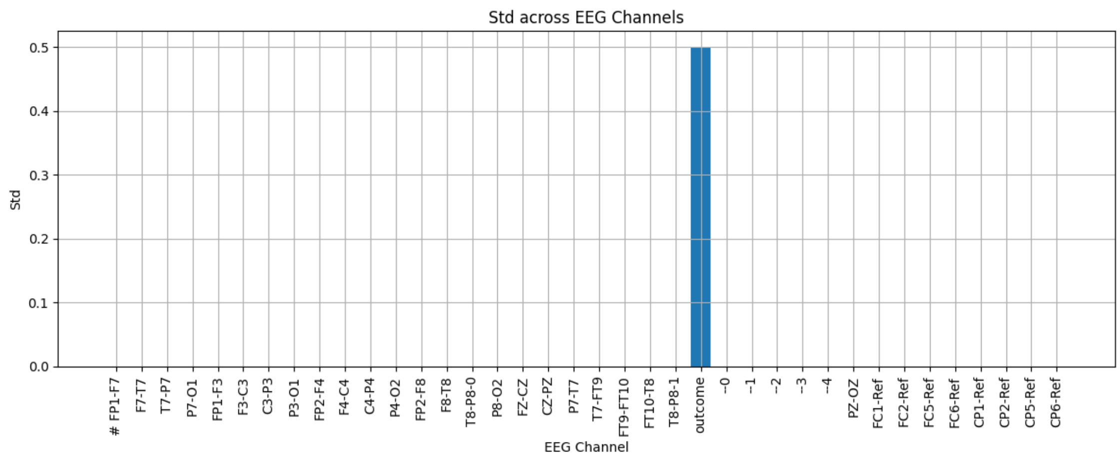
```
Extracted time-based features:

   # FP1-F7_mean  # FP1-F7_std  # FP1-F7_skew  # FP1-F7_kurtosis  \
0   2.967544e-07      0.000097        0.48946          16.847466

     F7-T7_mean  F7-T7_std  F7-T7_skew  F7-T7_kurtosis    T7-P7_mean  \
0  3.122452e-07   0.000103    0.325162       13.283395  2.507915e-07

   T7-P7_std  …  CP2-Ref_skew  CP2-Ref_kurtosis  CP5-Ref_mean  CP5-Ref_std  \
0   0.000097  …     -1.321515         19.952242  4.240870e-07     0.000039

   CP5-Ref_skew  CP5-Ref_kurtosis  CP6-Ref_mean  CP6-Ref_std  CP6-Ref_skew  \
0     -1.092546          15.07464  4.357542e-07     0.000039      -1.25888

   CP6-Ref_kurtosis
0         64.627033
```

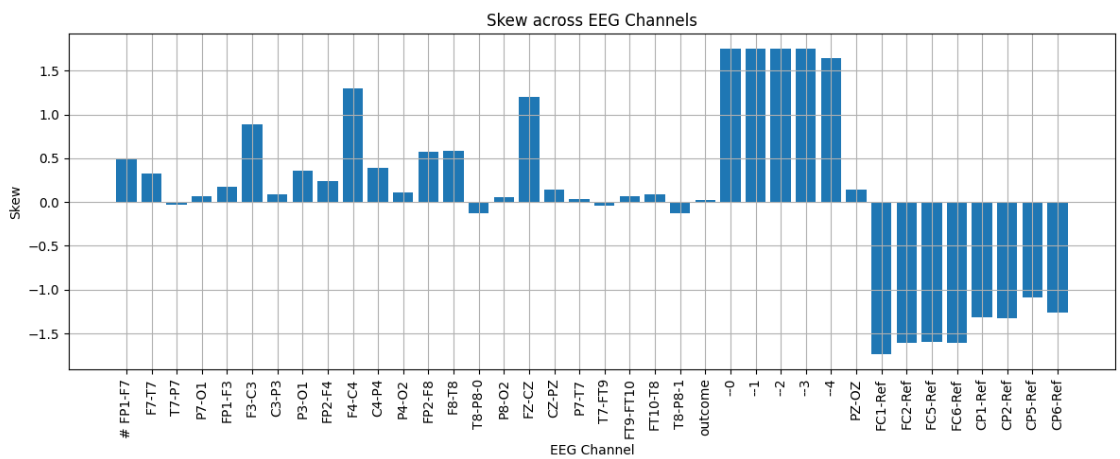
```
[1 rows x 152 columns]
```

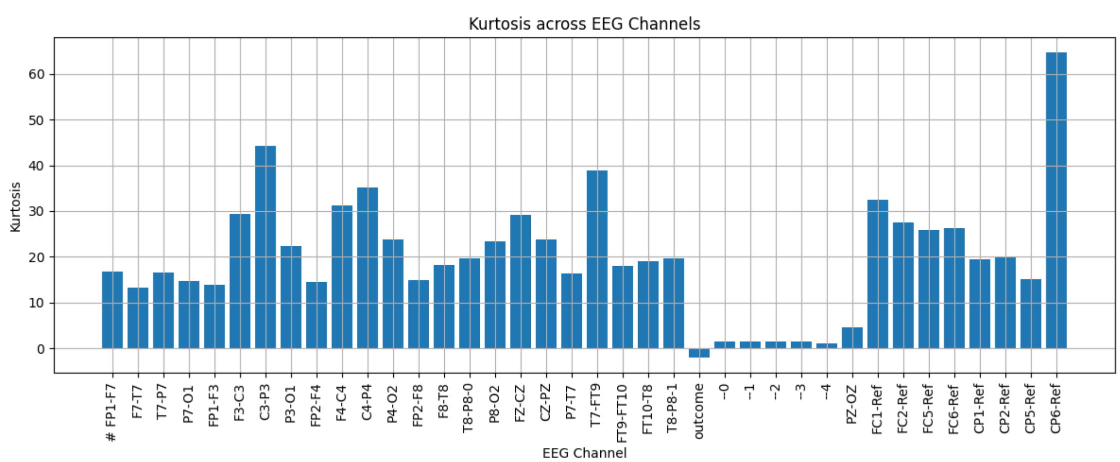
2 Visualization of data

[8]:
```python
# Extract feature names from column suffixes
features = ['mean', 'std', 'skew', 'kurtosis']

# Loop over each feature and plot values across channels
for feat in features:
    # Get all columns matching the current feature
    feat_cols = [col for col in features_df.columns if col.endswith(f"_{feat}")]

    # Map: 'FP1-F7_mean' → 'FP1-F7'
    channel_names = [col.replace(f"_{feat}", "") for col in feat_cols]

    # Extract values from first row
    values = features_df.loc[0, feat_cols].values

    # Plot
    plt.figure(figsize=(12, 5))
    plt.bar(channel_names, values)
    plt.title(f"{feat.capitalize()} across EEG Channels")
    plt.xlabel("EEG Channel")
    plt.ylabel(feat.capitalize())
    plt.xticks(rotation=90)
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```




Std across EEG Channels
Std
EEG Channel
0.0
0.1
0.2
0.3
0.4
0.5
FP1-F7
F7-T7
T7-P7
P7-O1
FP1-F3
F3-C3
C3-P3
P3-O1
FP2-F4
F4-C4
C4-P4
P4-O2
FP2-F8
F8-T8
T8-P8-0
P8-O2
FZ-CZ
CZ-PZ
P7-T7
T7-FT9
FT9-FT10
FT10-T8
T8-P8-1
outcome
−0
−1
−2
−3
−4
PZ-OZ
FC1-Ref
FC2-Ref
FC5-Ref
FC6-Ref
CP1-Ref
CP2-Ref
CP5-Ref
CP6-Ref


Skew across EEG Channels
Skew
EEG Channel
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
FP1-F7
F7-T7
T7-P7
P7-O1
FP1-F3
F3-C3
C3-P3
P3-O1
FP2-F4
F4-C4
C4-P4
P4-O2
FP2-F8
F8-T8
T8-P8-0
P8-O2
FZ-CZ
CZ-PZ
P7-T7
T7-FT9
FT9-FT10
FT10-T8
T8-P8-1
outcome
−0
−1
−2
−3
−4
PZ-OZ
FC1-Ref
FC2-Ref
FC5-Ref
FC6-Ref
CP1-Ref
CP2-Ref
CP5-Ref
CP6-Ref


Kurtosis across EEG Channels
Kurtosis
EEG Channel
0
10
20
30
40
50
60
FP1-F7
F7-T7
T7-P7
P7-O1
FP1-F3
F3-C3
C3-P3
P3-O1
FP2-F4
F4-C4
C4-P4
P4-O2
FP2-F8
F8-T8
T8-P8-0
P8-O2
FZ-CZ
CZ-PZ
P7-T7
T7-FT9
FT9-FT10
FT10-T8
T8-P8-1
outcome
−0
−1
−2
−3
−4
PZ-OZ
FC1-Ref
FC2-Ref
FC5-Ref
FC6-Ref
CP1-Ref
CP2-Ref
CP5-Ref
CP6-Ref

2.1 Interpretation of Bar Plot Visualization Output

- **Four Bar Plots**: Display mean, std, skew, and kurtosis across EEG channels for the first time window.
- **Mean Plot**: Near-zero values (e.g., 2.967544e-07 for FP1-F7) confirm baseline correction across channels.
- **Std Plot**: Higher values (e.g., 0.000097 for T7-P7) indicate greater signal variability in specific channels.
- **Skew Plot**: Non-zero values (e.g., -1.321515 for CP2-Ref) suggest asymmetric signal distributions.
- **Kurtosis Plot**: High values (e.g., 64.627033 for CP6-Ref) indicate heavy-tailed distributions, possibly due to spikes or artifacts.

If these plots don't help differentiate segments or clusters of data (e.g., seizure-like vs non-seizure), then add: - RMS (for power) - Entropy (for irregularity) - Zero-crossings (for frequency proxy) - Hjorth parameters (for signal shape)

[9]:
```
features = {}

for ch in df.columns:
    sig = df[ch].values

    # Time-domain stats
    features[f'{ch}_mean'] = np.mean(sig)
    features[f'{ch}_std'] = np.std(sig)
    features[f'{ch}_skew'] = skew(sig)
    features[f'{ch}_kurtosis'] = kurtosis(sig)

    # RMS
    features[f'{ch}_rms'] = np.sqrt(np.mean(sig**2))

    # Zero-crossing rate
    features[f'{ch}_zcr'] = ((sig[:-1] * sig[1:]) < 0).sum()

    # Hjorth parameters
    activity = np.var(sig)
    mobility, complexity = ant.hjorth_params(sig)

    features[f'{ch}_hjorth_activity'] = activity
    features[f'{ch}_hjorth_mobility'] = mobility
    features[f'{ch}_hjorth_complexity'] = complexity

    # Permutation entropy
    features[f'{ch}_perm_entropy'] = ant.perm_entropy(sig)
```

[10]:
```
# Convert to a single-row DataFrame for ML use
features_df = pd.DataFrame([features])
print("Extracted time-based features:")
```

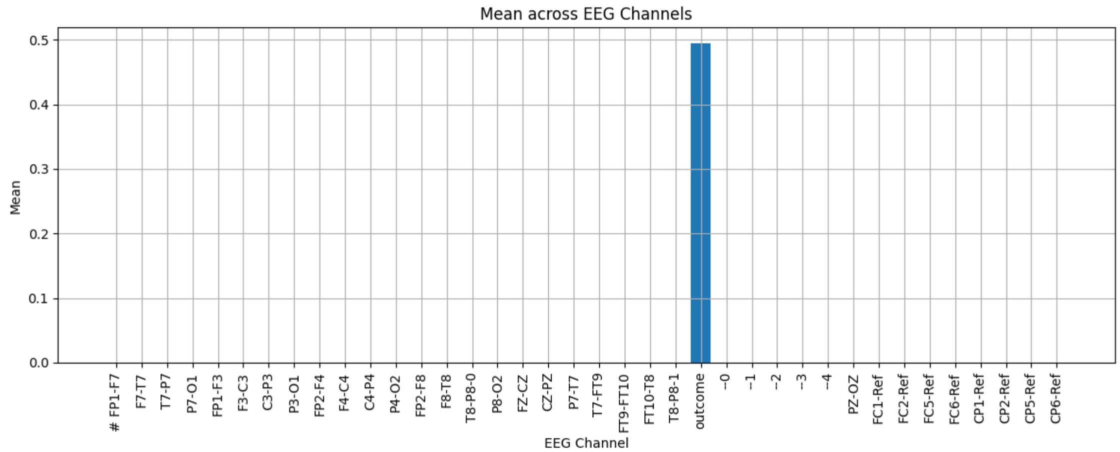
```
display(features_df)
```

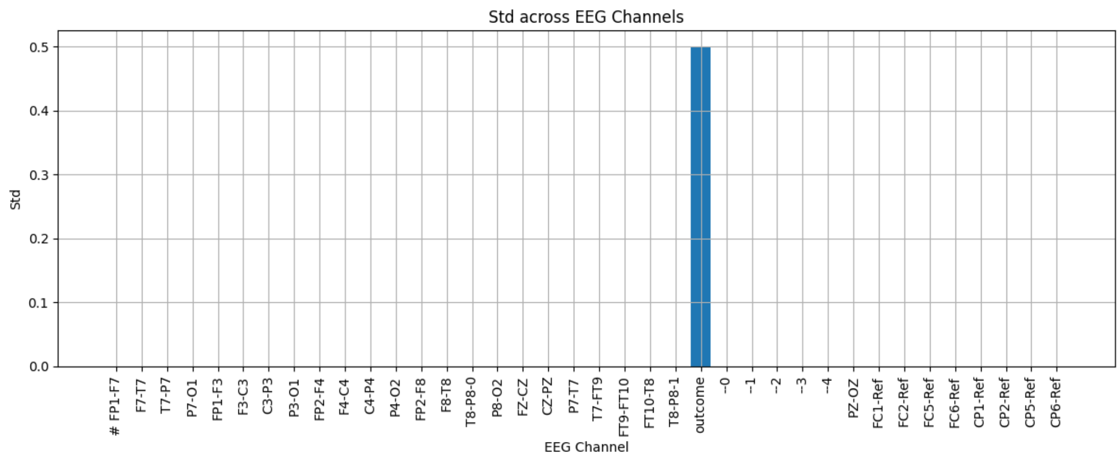
```
Extracted time-based features:

   # FP1-F7_mean  # FP1-F7_std  # FP1-F7_skew  # FP1-F7_kurtosis  \
0   2.967544e-07      0.000097       0.48946          16.847466

   # FP1-F7_rms  # FP1-F7_zcr  # FP1-F7_hjorth_activity  \
0      0.000097       2800818              9.495742e-09

   # FP1-F7_hjorth_mobility  # FP1-F7_hjorth_complexity  \
0                 1.383732                     1.24274

   # FP1-F7_perm_entropy  …  CP6-Ref_mean  CP6-Ref_std  CP6-Ref_skew  \
0              2.582426  …  4.357542e-07     0.000039      -1.25888

   CP6-Ref_kurtosis  CP6-Ref_rms  CP6-Ref_zcr  CP6-Ref_hjorth_activity  \
0         64.627033     0.000039       508298             1.490482e-09

   CP6-Ref_hjorth_mobility  CP6-Ref_hjorth_complexity  CP6-Ref_perm_entropy
0                   0.4292                   2.992936              1.634824

[1 rows x 380 columns]
```

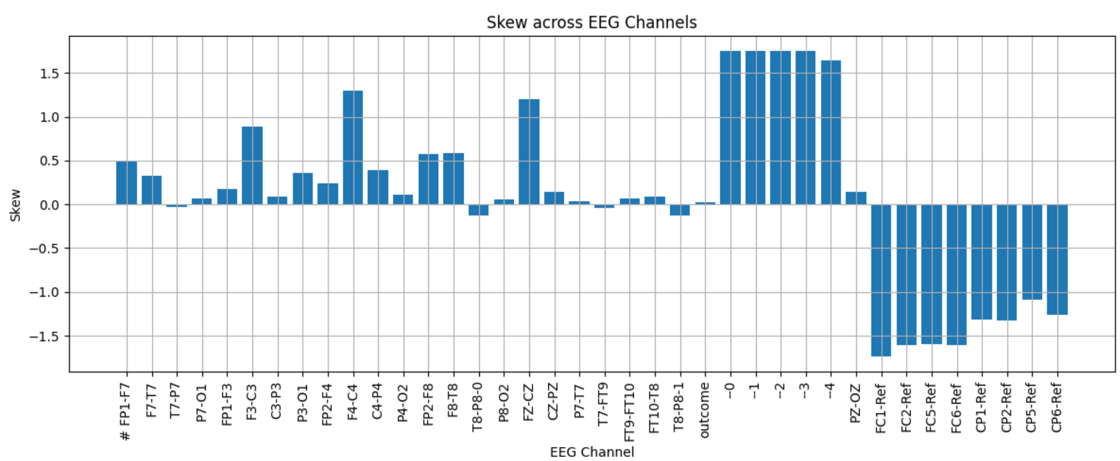
[11]:

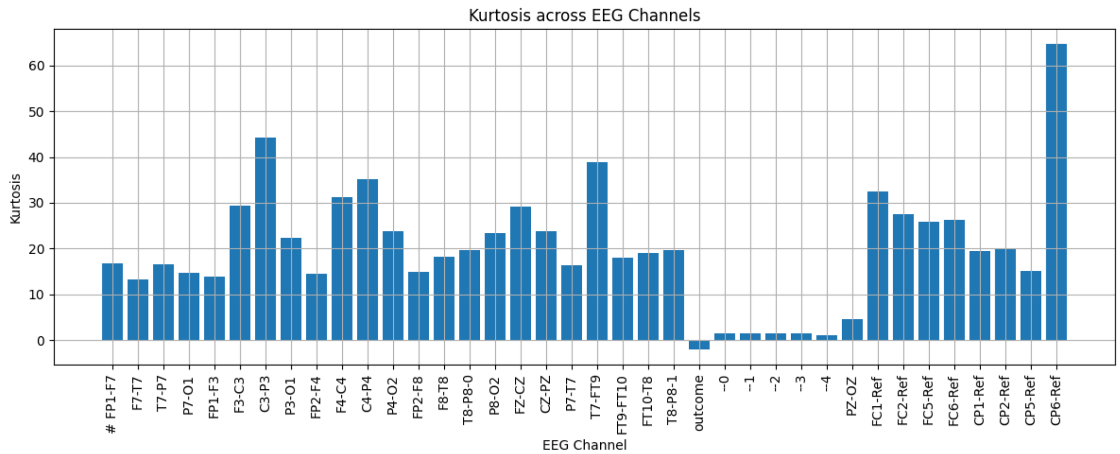
```python
# Extract feature names from column suffixes
features = ['mean', 'std', 'skew', 'kurtosis', 'rms', 'zcr', 'hjorth_activity',␣
↪'hjorth_mobility', 'hjorth_complexity', 'entropy']

# Loop over each feature and plot values across channels
for feat in features:
    # Get all columns matching the current feature
    feat_cols = [col for col in features_df.columns if col.endswith(f"_{feat}")]

    # Map: 'FP1-F7_mean' → 'FP1-F7'
    channel_names = [col.replace(f"_{feat}", "") for col in feat_cols]

    # Extract values from first row
    values = features_df.loc[0, feat_cols].values

    # Plot
    plt.figure(figsize=(12, 5))
    plt.bar(channel_names, values)
    plt.title(f"{feat.capitalize()} across EEG Channels")
    plt.xlabel("EEG Channel")
    plt.ylabel(feat.capitalize())
    plt.xticks(rotation=90)
    plt.grid(True)
    plt.tight_layout()
```

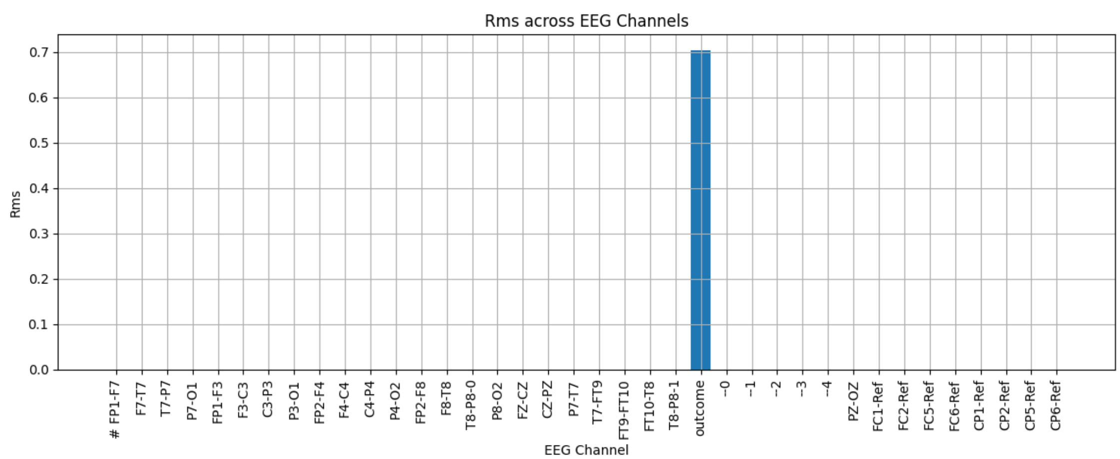
```
plt.show()
```

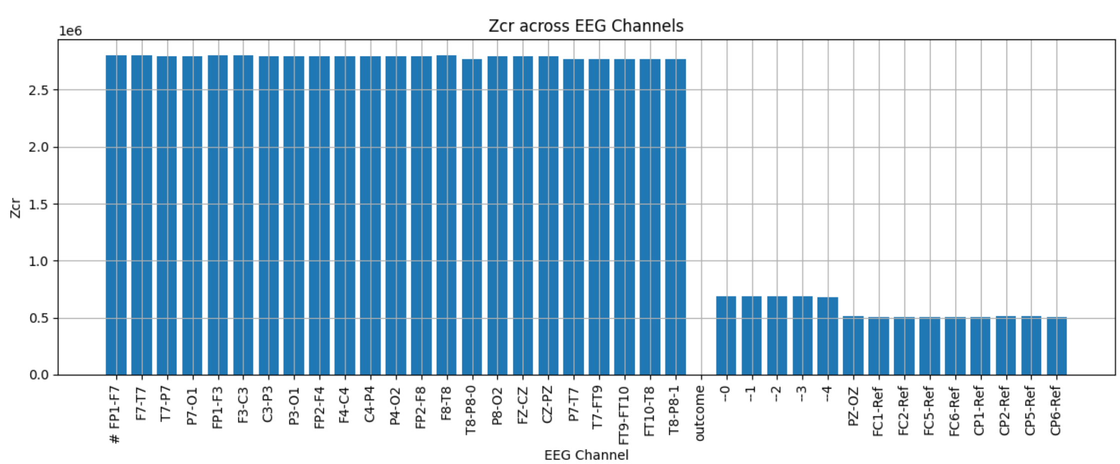

Mean across EEG Channels
Mean
EEG Channel

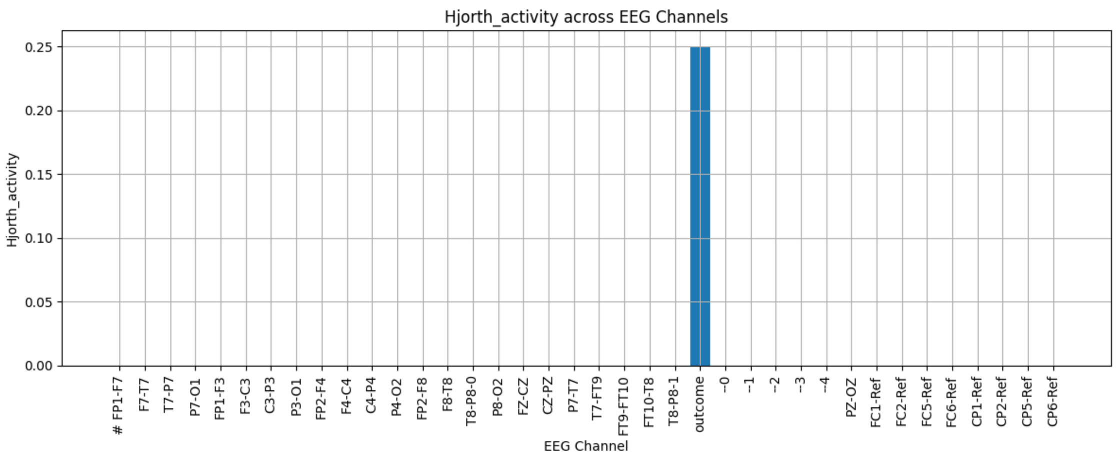

Std across EEG Channels
Std
EEG Channel

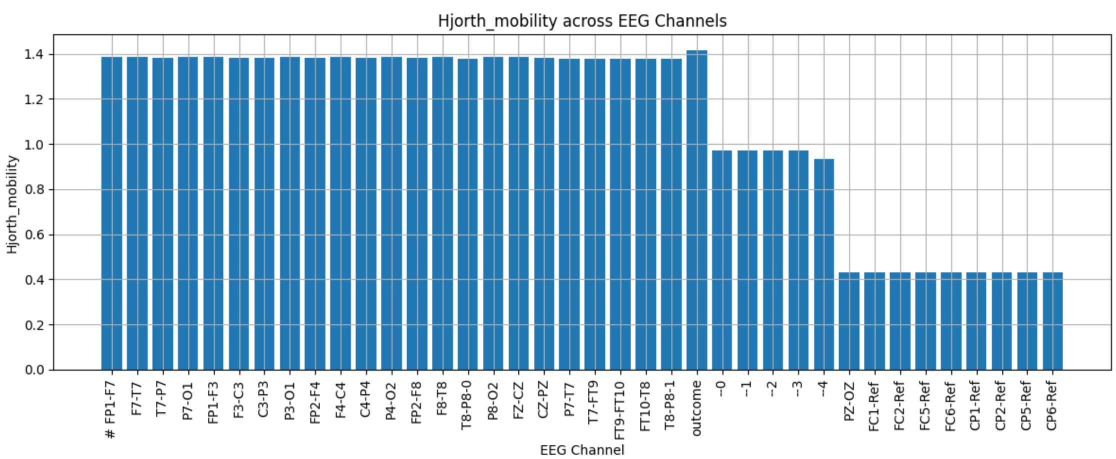

Skew across EEG Channels
Skew
EEG Channel

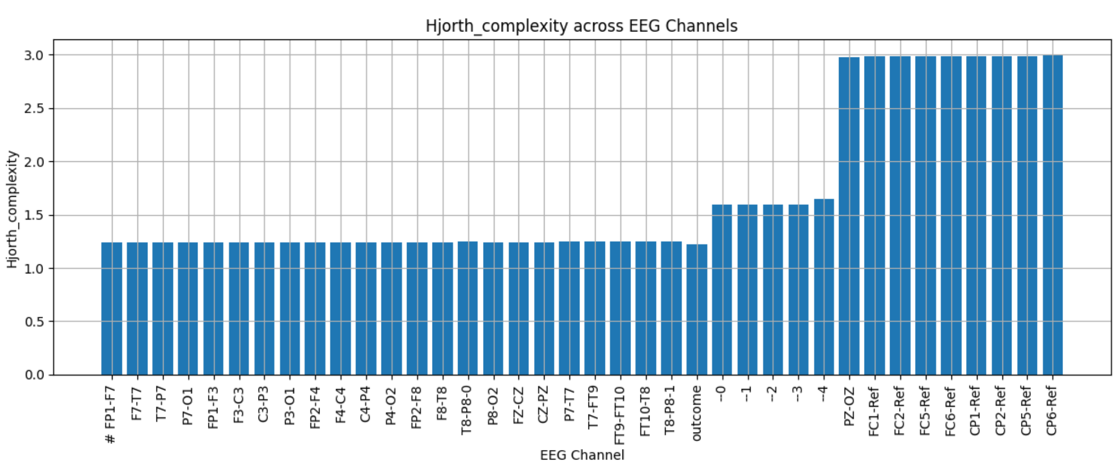

Kurtosis across EEG Channels
Kurtosis
EEG Channel

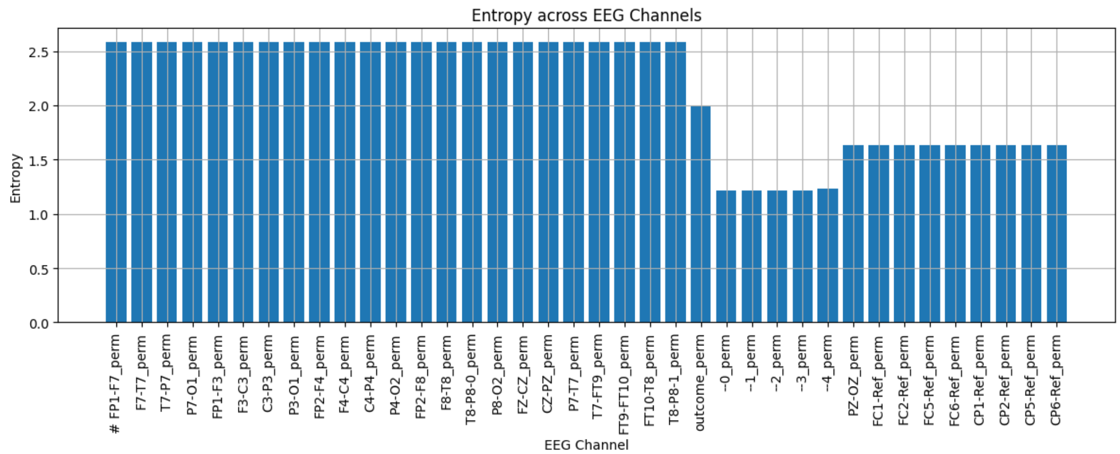

Rms across EEG Channels
Rms
EEG Channel


Zcr across EEG Channels
1e6
Zcr
EEG Channel


Hjorth_activity across EEG Channels
Hjorth_activity
EEG Channel
0.00
0.05
0.10
0.15
0.20
0.25
FP1-F7
F7-T7
T7-P7
P7-O1
FP1-F3
F3-C3
C3-P3
P3-O1
FP2-F4
F4-C4
C4-P4
P4-O2
FP2-F8
F8-T8
T8-P8-0
P8-O2
FZ-CZ
CZ-PZ
P7-T7
T7-FT9
FT9-FT10
FT10-T8
T8-P8-1
outcome
-0
-1
-2
-3
-4
PZ-OZ
FC1-Ref
FC2-Ref
FC5-Ref
FC6-Ref
CP1-Ref
CP2-Ref
CP5-Ref
CP6-Ref


Hjorth_mobility across EEG Channels
Hjorth_mobility
EEG Channel
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
FP1-F7
F7-T7
T7-P7
P7-O1
FP1-F3
F3-C3
C3-P3
P3-O1
FP2-F4
F4-C4
C4-P4
P4-O2
FP2-F8
F8-T8
T8-P8-0
P8-O2
FZ-CZ
CZ-PZ
P7-T7
T7-FT9
FT9-FT10
FT10-T8
T8-P8-1
outcome
-0
-1
-2
-3
-4
PZ-OZ
FC1-Ref
FC2-Ref
FC5-Ref
FC6-Ref
CP1-Ref
CP2-Ref
CP5-Ref
CP6-Ref


Hjorth_complexity across EEG Channels
Hjorth_complexity
EEG Channel
0.0
0.5
1.0
1.5
2.0
2.5
3.0
FP1-F7
F7-T7
T7-P7
P7-O1
FP1-F3
F3-C3
C3-P3
P3-O1
FP2-F4
F4-C4
C4-P4
P4-O2
FP2-F8
F8-T8
T8-P8-0
P8-O2
FZ-CZ
CZ-PZ
P7-T7
T7-FT9
FT9-FT10
FT10-T8
T8-P8-1
outcome
-0
-1
-2
-3
-4
PZ-OZ
FC1-Ref
FC2-Ref
FC5-Ref
FC6-Ref
CP1-Ref
CP2-Ref
CP5-Ref
CP6-Ref



[13]:
```python
features_df.to_csv('features-time-based.csv', index=False)
```

```
csv with time based features saved.
```

2.2 Interpretation of the features

- Mean: Captures the average signal level. Helps the model detect shifts in baseline brain activity, which can relate to seizures.
- Standard Deviation (Std): Measures signal variability or fluctuations. Seizure activity often shows higher variability, so std helps highlight that.
- Skewness: Shows signal asymmetry — whether the EEG wave tends to have more positive or negative spikes. Seizures can produce asymmetric waveforms, so skewness can signal those patterns.
- Kurtosis: Reflects how "spiky" or heavy-tailed the signal distribution is. Seizures often have sharp spikes, leading to high kurtosis — useful for detection.
- RMS (Root Mean Square): Represents signal power, which often increases during seizures, especially muscle artifacts or movement.
- Zero-Crossing Rate: Estimates frequency changes roughly. EEG changes frequency content during seizures; this helps catch that.
- Hjorth Parameters (Activity, Mobility, Complexity): Compact measures of signal shape and dynamics, sensitive to changes in EEG complexity during seizures.
- Entropy: Quantifies irregularity/unpredictability of the signal. Seizure EEGs often have altered complexity and regularity, so entropy helps the model learn those changes.

These features help a DL model: - Recognize patterns typical of seizures more reliably - Reduce noise and irrelevant details that confuse the model - Generalize better to new unseen EEG recordings