Predicting plasticity outcomes based on LFP magnitude

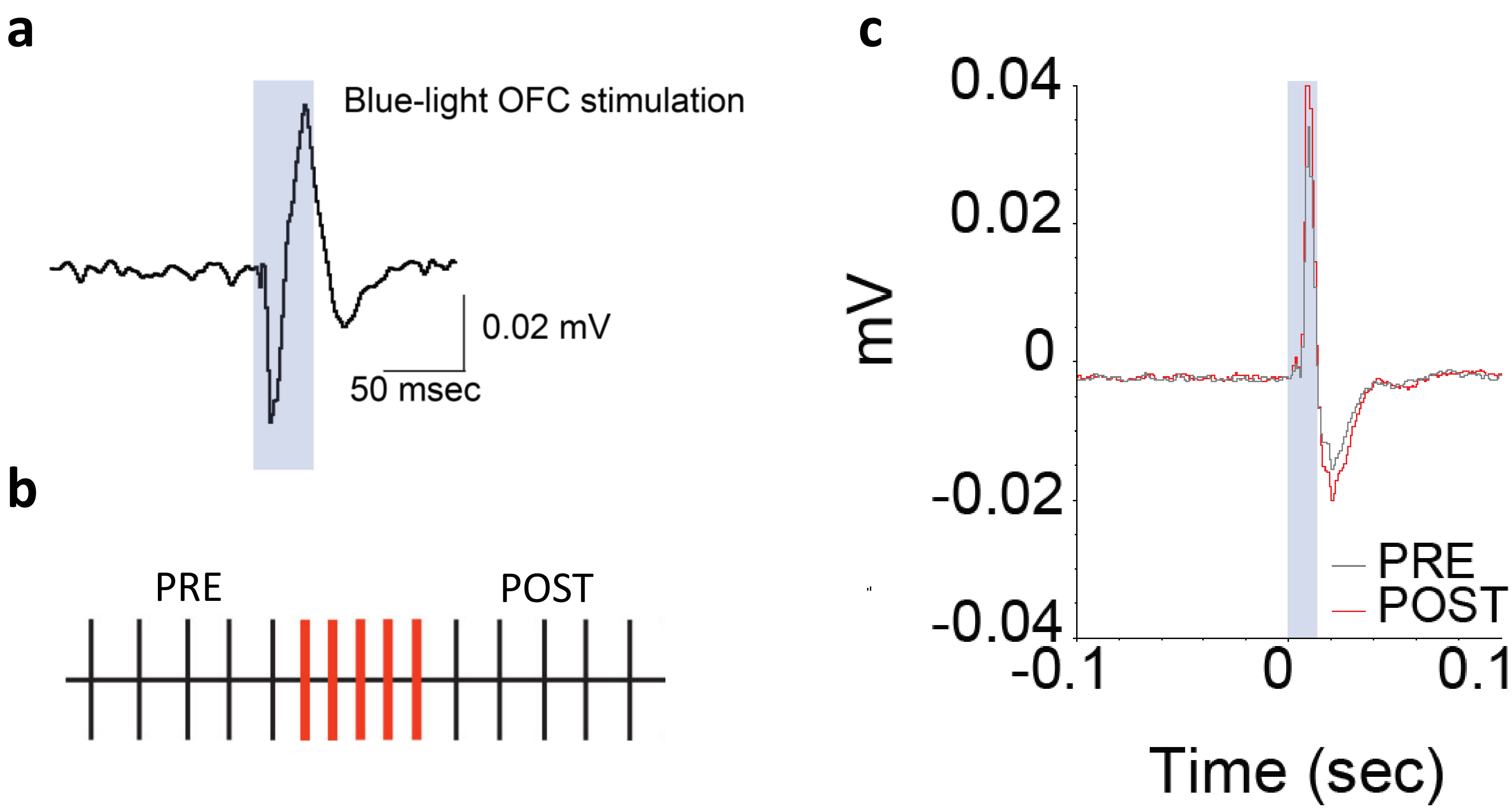
Sebastiano Bariselli &
Bridget Matikainen-Ankney

BIOF309, Spring 2018

Background

Neural plasticity:

The ability of connections between neurons to change



Question

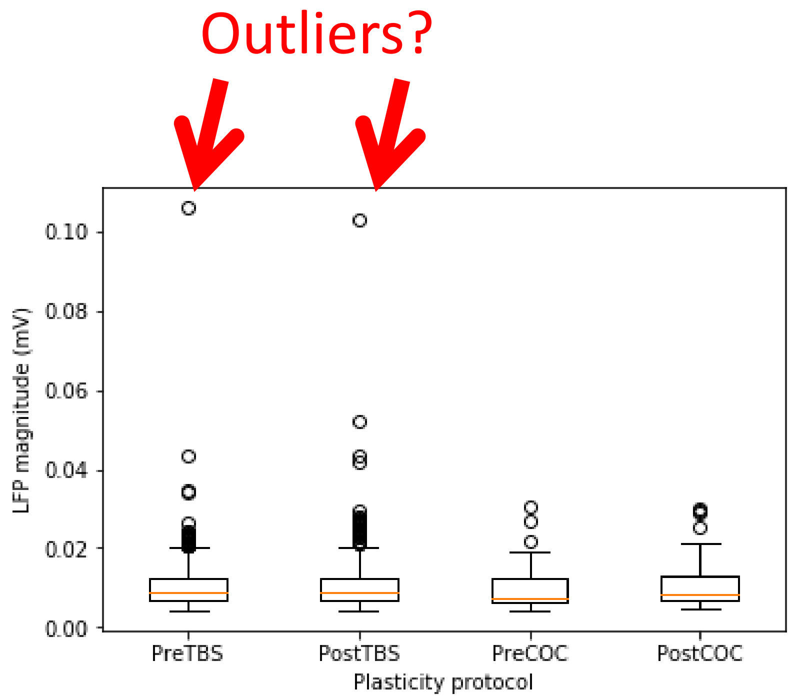
Can we predict plasticity outcomes based on local field potential magnitude during PRE period?

Project goals

- Import data
- Clean data (outliers)
- Data visualization
- Use prediction model

IMPORT DATA

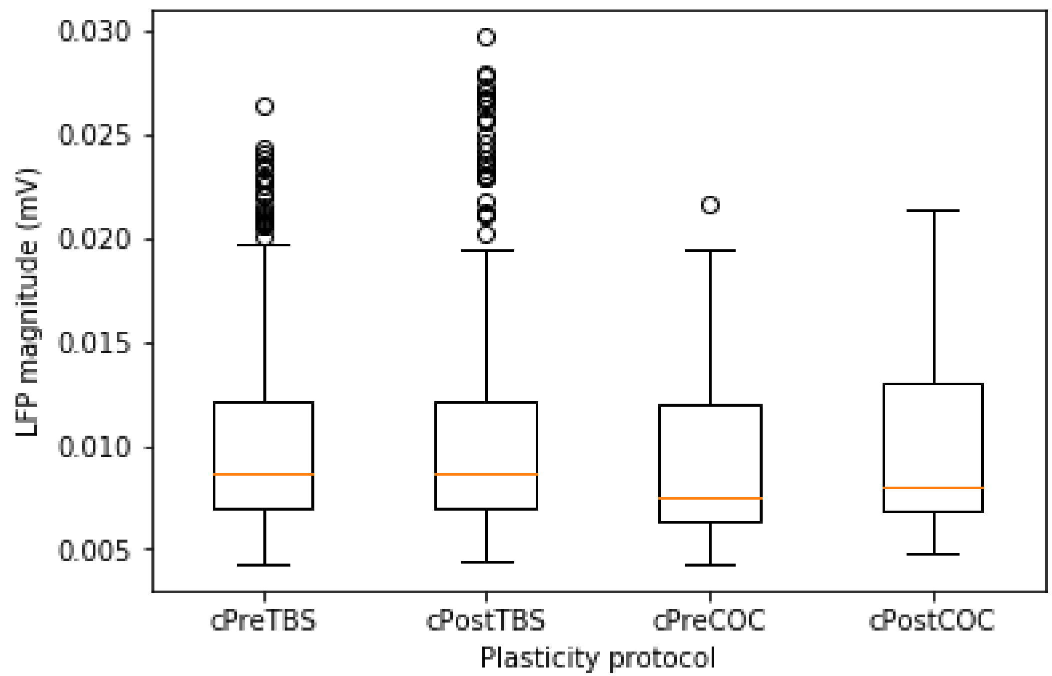
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

# Retrieve '.csv' files containing electrophysiology data in TBS and Cocaine conditions.
LFP_data_TBS = pd.read_csv('/Users/barisellisebastiano/Desktop/Excel_Practice4.csv')
LFP_data_COC = pd.read_csv('/Users/barisellisebastiano/Desktop/Excel_practice_Cocaine.csv')

# Plot a boxplot for the data to observe data points in 4 distributions: PreTBS, PostTBS, PreCOC, PostCOC
data1 = LFP_data_TBS['PRE'].values.reshape(-1, 1)
data2 = LFP_data_TBS['POST'].values.reshape(-1, 1)
data3 = LFP_data_COC['PRE'].values.reshape(-1, 1)
data4 = LFP_data_COC['POST'].values.reshape(-1, 1)
data = [data1, data2, data3, data4]
plt.boxplot(data)
plt.xticks([1, 2, 3, 4], ['PreTBS', 'PostTBS', 'PreCOC', 'PostCOC'])
plt.ylabel('LFP magnitude (mV)')
plt.xlabel('Plasticity protocol')
plt.show()
```



CLEANING THE DATAFRAME

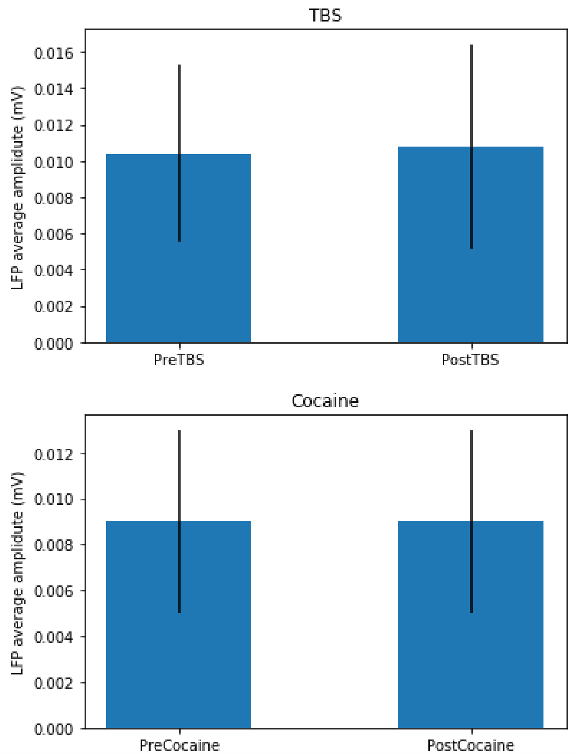
```
# Calculate mean and standard deviation for preTBS, postTBS, preCocaine and postCocaine and
LFP_data_TBS['MeanPre'] = LFP_data_TBS['PRE'].mean()
LFP_data_TBS['StDevPre'] = LFP_data_TBS['PRE'].std()
LFP_data_TBS['MeanPost'] = LFP_data_TBS['POST'].mean()
LFP_data_TBS['StDevPost'] = LFP_data_TBS['POST'].std()
LFP_data_COC['MeanPre'] = LFP_data_COC['PRE'].mean()
LFP_data_COC['StDevPre'] = LFP_data_COC['PRE'].std()
LFP_data_COC['MeanPost'] = LFP_data_COC['POST'].mean()
LFP_data_COC['StDevPost'] = LFP_data_COC['POST'].std()


# Write a function to detect 3 standard deviation (SD) outliers from 'PRE' column of a DataFrame
def remove_outliers_fromPRE(row):
    Pre = row['PRE']
    if Pre < (row['MeanPre'] - (3 * row['StDevPre'])) or Pre > (row['MeanPre'] + (3 * row['StDevPre'])):
        return NaN
    else:
        return Pre

# Write a function to detect 3 standard deviation (SD) outliers from 'POST' column of a DataFrame
def remove_outliers_fromPOST(row):
    Post = row['POST']
    if Post < (row['MeanPost'] - (3 * row['StDevPost'])) or Po:
        return NaN
    else:
        return Post

# Apply a function to TBS and Cocaine dataframe create columns
LFP_data_TBS['cPRE'] = LFP_data_TBS.apply(remove_outliers_froml
LFP_data_COC['cPRE'] = LFP_data_COC.apply(remove_outliers_froml
LFP_data_TBS['cPOST'] = LFP_data_TBS.apply(remove_outliers_fro
LFP_data_COC['cPOST'] = LFP_data_COC.apply(remove_outliers_fro
```

```
#Add a new column 'cCHANGE' to the TBS Dataframe tha
for row_index, row in LFP_data_TBS.iterrows():
    LFP_data_TBS['cCHANGE'] = LFP_data_TBS['POST']/L

# Drop the rows in TBS DataFrame that contain 'NaN'
LFP_data_TBS_cleaned = LFP_data_TBS.dropna(axis=0, h
```



Is there a plasticity?

```
# Execute a paired t-test to assess changes between pre- and post- for TBS and Cocaine conditions.
print('paired t-test for TBS = ' + str(stats.ttest_rel(LFP_data_TBS_cleaned['cPRE'], LFP_data_TBS_cleaned['cPOST'])))
print('paired t-test for COCAINE = ' + str(stats.ttest_rel(LFP_data_COC_cleaned['cPRE'], LFP_data_COC_cleaned['cPOST'])))
```



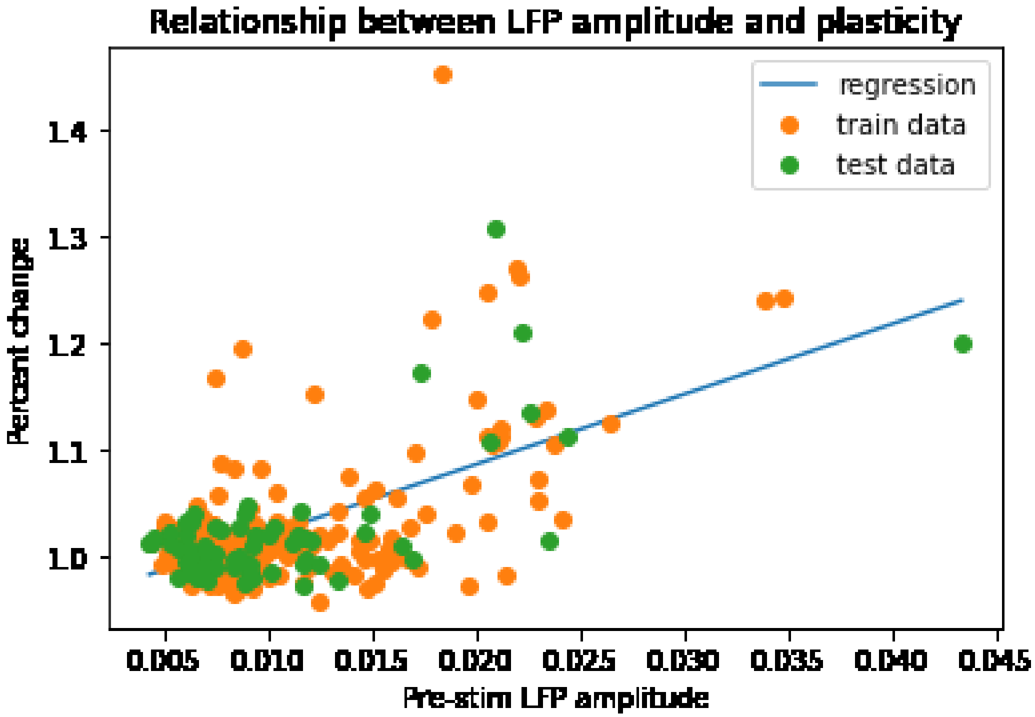

```
paired t-test for TBS = Ttest_relResult(statistic=-5.7939485436530358, pvalue=1.8844506960029739e-08)
paired t-test for COCAINE = Ttest_relResult(statistic=-9.7858542513832418, pvalue=8.5423497024256997e-18)
```

REGRESSION ANALYSIS

```python
#splitting our data set into 25% and 75%
from sklearn.model_selection import train_test_split
PreTBS_train, PreTBS_test, ChangeTBS_train, ChangeTBS_test = train_test_split(PreTBS, ChangeTBS, test_size=0.25, random_state=42)

#make a regression
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(PreTBS_train, ChangeTBS_train)

#print regression
print('coefficients: ', regressor.coef_)
print('y-axis intercept: ', regressor.intercept_)
print('Regression: % Change =', regressor.coef_,'*Pre-stim response amplitude +', regressor.intercept_)

#plotting regression to fit training data
min_pt = PreTBS.min() * regressor.coef_[0] + regressor.intercept_
max_pt = PreTBS.max() * regressor.coef_[0] + regressor.intercept_
plt.plot([PreTBS.min(), PreTBS.max()], [min_pt, max_pt], label="regression")
plt.plot(PreTBS_train, ChangeTBS_train, 'o', label="train data");
plt.show()

#predicting target data
Change_pred_train = regressor.predict(PreTBS_train)

#try test set
Change_pred_test = regressor.predict(PreTBS_test)
plt.plot(PreTBS_test, ChangeTBS_test, 'o', label="test data")

#evaluate preditction quantitatively
R2 = regressor.score(PreTBS_test, ChangeTBS_test)

plt.title("Relationship between LFP amplitude and plasticity")
plt.xlabel("Pre-stim LFP amplitude")
plt.ylabel("Percent change")
plt.legend(loc='best');
plt.show()
print("R squared =", R2)
```

Regression analysis

- Regression: % Change = [6.55694474] *Pre-stim response amplitude + 0.954975708918
- R squared = 0.471091816409



Conclusions

1) LFP magnitude at PRE does predict plasticity outcomes given a linear models ($R2 = 0.47$).

2) This code will be used to organize and analyze neural data.