MP #2: Peer Feedback #1 (Pre-Feedback Work to Date)

Libo Zhang (lz200)

The structure of my work to date format will follow the recommended project milestones.

Note: Only the code written for testing my algorithm will be displayed to help peer review/feedback.

Week 1:

(1) Load/Read the data

[2]:
```
filename1 = "feaSubEImg_1.csv"
filename2 = "feaSubEImg_2.csv"
filename3 = "feaSubEOvert_1.csv"
filename4 = "feaSubEOvert_2.csv"
filename5 = "sensors102.csv"

df_img1       = pd.read_csv(filename1, header = None)
df_img2       = pd.read_csv(filename2, header = None)
df_overt1     = pd.read_csv(filename3, header = None)
df_overt2     = pd.read_csv(filename4, header = None)
df_sensors102 = pd.read_csv(filename5, header = None)
```

[3]:
```
img1       = df_img1.to_numpy()
img2       = df_img2.to_numpy()
overt1     = df_overt1.to_numpy()
overt2     = df_overt2.to_numpy()
sensors102 = df_sensors102.to_numpy()

print(img1.shape)
print(img2.shape)
print(overt1.shape)
print(overt2.shape)
print(sensors102.shape)
```

```
(204, 120)
(204, 120)
(204, 120)
(204, 120)
(102, 6)
```

Block [2] proves that I can successfully Load the data, and Block [3] proves that I can successfully Read the data.

[4]:
```
# Each trial (observation) is represented by a 204 * 1 vector
print(img1[:, 0].shape)
# Print out the first 5 feature values of the first trial 204 * 1 vector
print(img1[0:5, 0])

# The i-th feature vector is a 1 * 120 vector
print(img1[0, :].shape)
# Print out the first 5 trials of the first feature vector
print(img1[0, 0:5])
```

```
(204,)
[ 102.84934331   48.46989141  -83.16721505   83.92515322 -174.96255643]
(120,)
[ 102.84934331 -406.15754407  294.53639568  156.37028017 -157.56413992]
```

Block [4] further proves that I understand what the data represents.

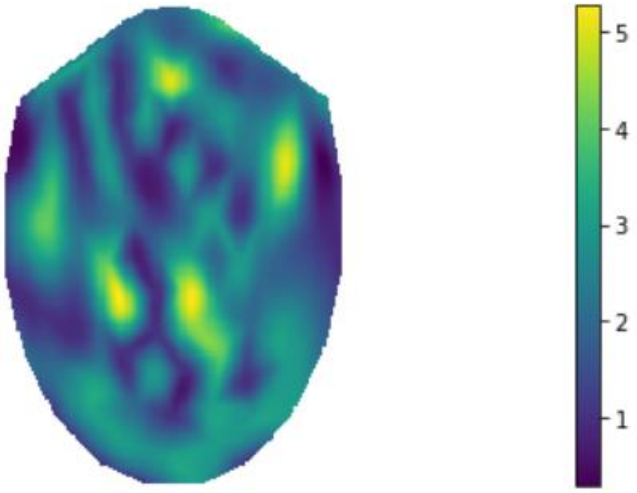
(2) Plot (image) channel weights on the brain surface

(You can create your own vectors of 204 positive numbers to test)

[5]:
```
test_weights = np.linspace(1, 5, num = 5)
chanVal = np.random.choice(a = test_weights, size = 204,
                           p = [0.30, 0.30, 0.30, 0.05, 0.05])
print(chanVal.shape)
print(chanVal[0:10])
```

```
(204,)
[3. 2. 1. 3. 1. 3. 3. 4. 1. 1.]
```

[6]:
```
show_chanWeights(chanVal)
```



Blocks [5] and [6] demonstrate that I have successfully tested the auxiliary plotting function for plotting channel weights on the brain surface.