

# Meeting #10

11-11-21

Michael Lee

# Deliverables

- Use epoch\_bm dictionary instead of dummy data
- Create a brain figure for 2-3 bands and display the electrodes that have the highest activity for that band

# Methodology and Learnings

- Help from team members
- `list(range(start, end))`
- dictionary
- layout

# Results

```
21 def highest_electrodes(epochbm_dict, epoch, band):
22     # Get the correct dataframe for a certain epoch
23     epoch = str(epoch)
24     df = epochbm_dict[epoch][1][band]
25
26     # Sort descending
27     mylist = df.sort_values(ascending=False)
28
29     # get the first 3 items of the list
30     # get the index of the first 3 items of the list
31     first3 = mylist[0:3:]
32     electrode_list = first3.index.values
33
34     return electrode_list
35
```

```
139 # Updates 3d brain 1 (Theta band) with the electrode to plot when slider is changed
140 @app.callback(
141     dash.dependencies.Output('image-display-graph-3d-1', 'figure'),
142     dash.dependencies.Input(slider.id, 'value') # for the electrode value
143 )
144 def update_graph(selected_value):
145     high3 = he.highest_electrodes(epochbm_dict, selected_value, "Theta")
146     fig = b3.make_3d_fig(data_brain, df, high3, "Theta")
147     return fig
148
149 # Updates 3d brain 2 (Alpha band) with the electrode to plot when slider is changed
150 @app.callback(
151     dash.dependencies.Output('image-display-graph-3d-2', 'figure'),
152     dash.dependencies.Input(slider.id, 'value') # for the electrode value
153 )
154 def update_graph(selected_value):
155     high3 = he.highest_electrodes(epochbm_dict, selected_value, "Alpha")
156     fig = b3.make_3d_fig(data_brain, df, high3, "Alpha")
157     return fig
```

# Results

```
54 # function that returns a figure with the biometric graph for a selected epoch
55 # INPUTS - the dataframe and biometric to be graphed - string, i.e HR or SpO2, and epochbm dict
56 #         and epoch as an int
57 # OUTPUTS - the figure
58 def graph_biometric_selection(biometric, epochbm_dict, epoch):
59     epoch = str(epoch)
60
61     # Get the x values from the dictionary,
62     #   numerate from each endpoint,
63     #   and convert to seconds by dividing by 500
64     x_values = epochbm_dict[epoch][0]
65     x_values = list(range(x_values[0], x_values[1]))
66     x_values = [number / 500 for number in x_values]
67
68     y_df = epochbm_dict[epoch][2][biometric]
69     y_values = y_df.values.tolist()
70
71     fig = go.Figure(data=[go.Scatter(x=x_values, y=y_values)])
72     fig.update_layout(
73         title="%s for Epoch %s" % (biometric, epoch),
74         title_x=0.5,
75         xaxis_title="Time (sec)",
76         yaxis_title=biometric
77     )
78     return fig
```

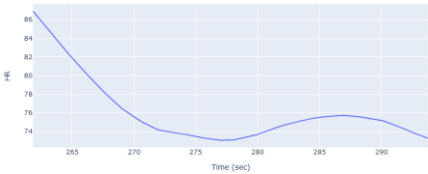
# Results

Theta Band Highest Electrodes



- Fp2
- Fp3
- Fp1

HR for Epoch 5

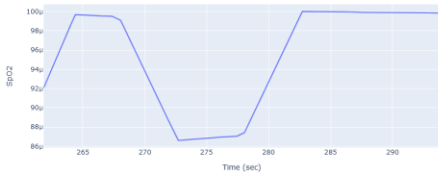


Alpha Band Highest Electrodes



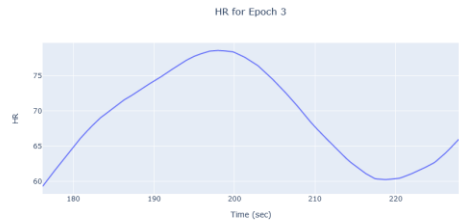
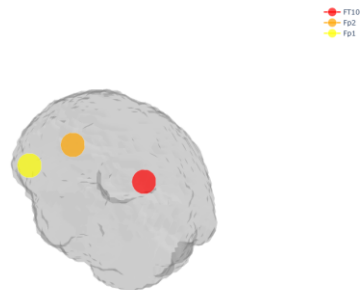
- P6
- P8
- Fp1

SpO2 for Epoch 5



# Results

Theta Band Highest Electrodes



Alpha Band Highest Electrodes

