Phase 2: EMG Data

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

```
emg_eating.head()
```

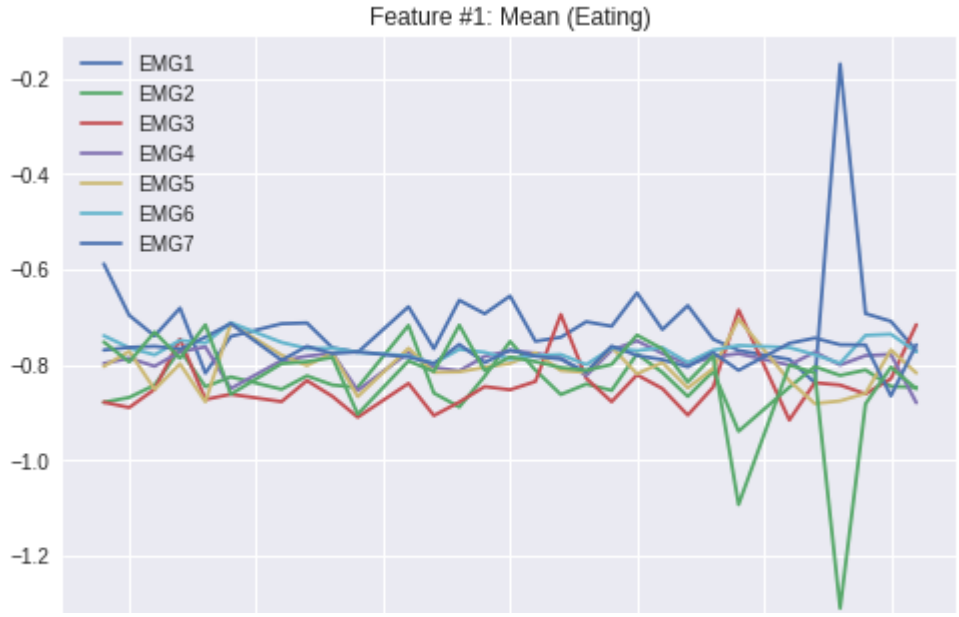
	Number	Timestamp	EMG1	EMG2	EMG3	EMG4	EMG5	EMG6	EMG7	EMG8	EMG9
0	9	1.500000e+12	-1.0	-2.0	-1.0	-3.0	-2.0	-1.0	2.0	0.0	1.0
1	9	1.500000e+12	0.0	0.0	0.0	-2.0	0.0	-1.0	0.0	0.0	1.0
2	9	1.500000e+12	-3.0	0.0	0.0	-1.0	-2.0	-1.0	-1.0	-1.0	1.0
3	9	1.500000e+12	0.0	1.0	-1.0	-1.0	-3.0	-2.0	-1.0	0.0	1.0
4	9	1.500000e+12	0.0	-3.0	-3.0	0.0	0.0	0.0	-4.0	-2.0	1.0

Feature 1: Mean

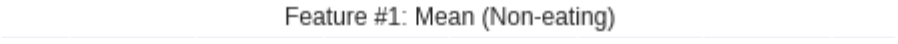
```
df = pd.read_csv("emgeatingfile.csv")
a = df['Number'].unique()
a.sort()

li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x1, y1 = {}, {}
# Mean of EMG1 for all Eating
means_eating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_eating[i] = df[col].mean() # means_eating[1] is mean of EMG1
    #print("Done for person ", i)
  lists = sorted(means_eating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x1[j], y1[j] = u, v
  print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x1[i], y1[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7'], loc='upper left')
plt.title("Feature #1: Mean (Eating)")
plt.show()
```

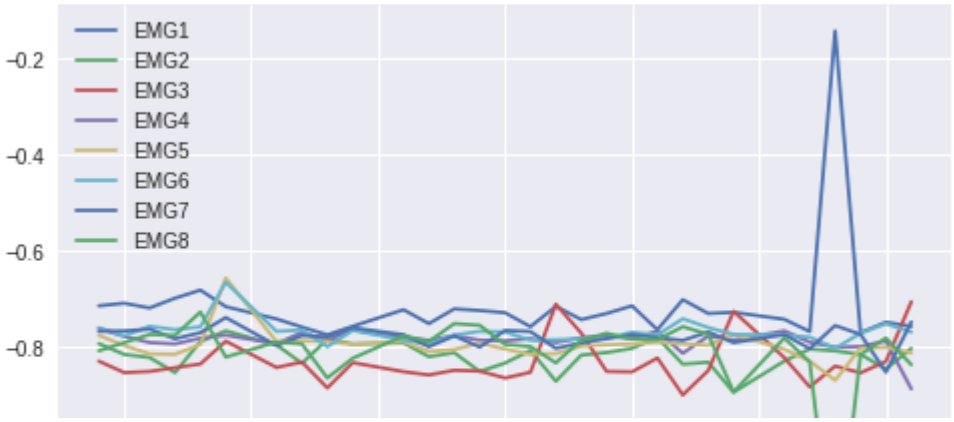
```
Done for col EMG  1
```

```
Done for col EMG  2
Done for col EMG  3
Done for col EMG  4
Done for col EMG  5
Done for col EMG  6
Done for col EMG  7
Done for col EMG  8
```

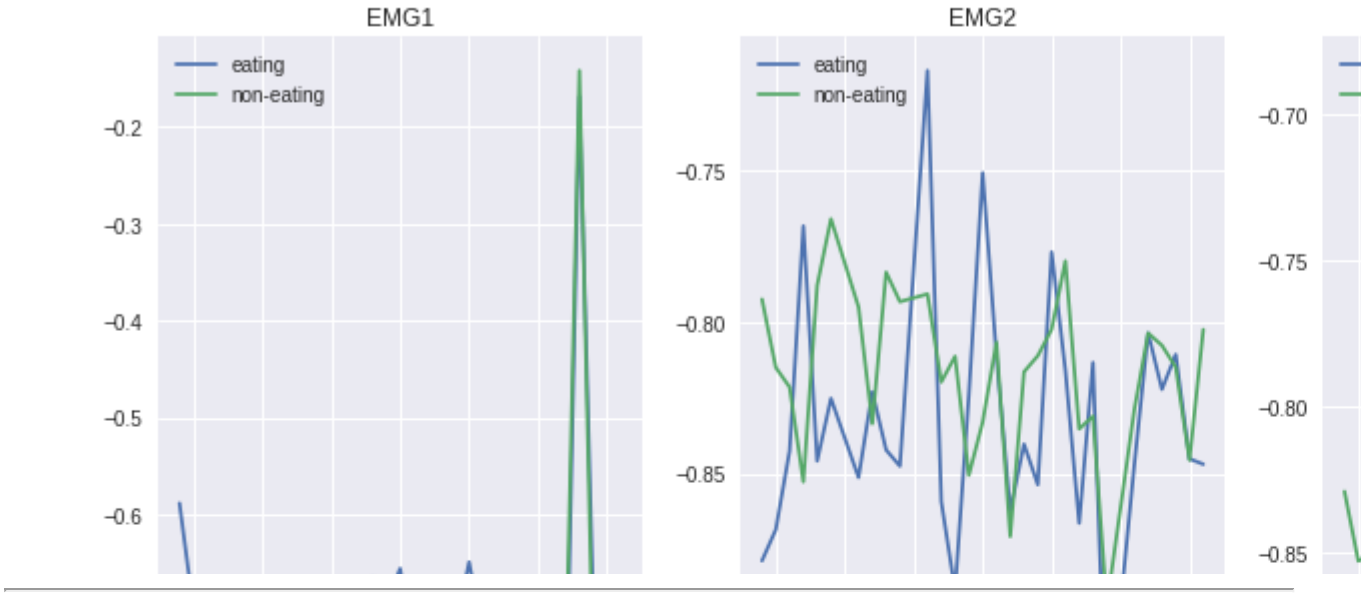



```
df = pd.read_csv("emgnoneatingfile.csv")
a = df['Number'].unique()
a.sort()
li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x2, y2 = {}, {}
# Mean of EMG1 for all Non-Eating
means_noneating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_noneating[i] = df[col].mean() # means_noneating[1] is mean of EMG1
    #print("Done for person ", i)
  lists = sorted(means_noneating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x2[j], y2[j] = u, v
  #print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x2[i], y2[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='upper le
plt.title("Feature #1: Mean (Non-eating)")
plt.show()
```



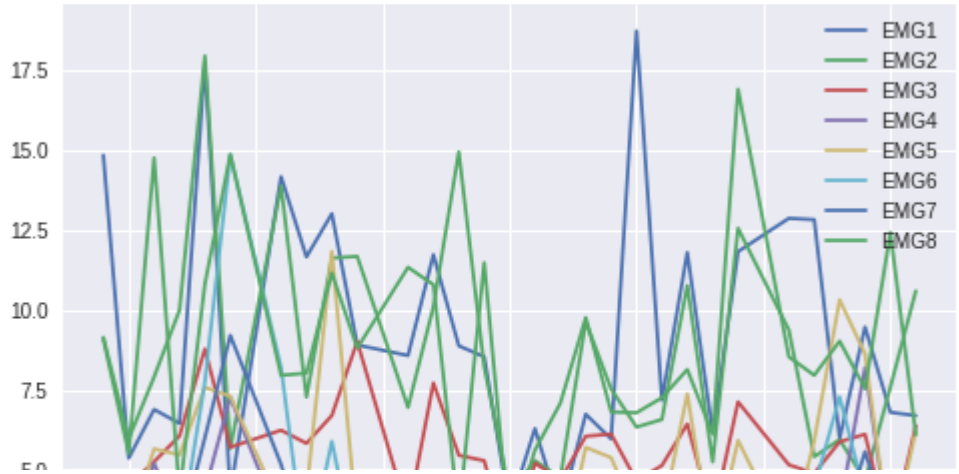



```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Feature #1: Mean', fontsize=30)
for i in range(1,9):
  plt.subplot(2, 4, i).set_title('EMG{}'.format(i))
  plt.plot(x1[i], y1[i])
  plt.plot(x2[i], y2[i])
  plt.legend(['eating', 'non-eating'], loc='upper left')

plt.show()
```

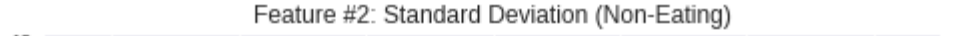


Feature 2: Standard Deviation

```
df = pd.read_csv("emgeatingfile.csv")
a = df['Number'].unique()
a.sort()

li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x1, y1 = {}, {}
# Mean of EMG1 for all Eating
means_eating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_eating[i] = df[col].std() # means_eating[1] is mean of EMG1
    #print("Done for person ", i)
  lists = sorted(means_eating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x1[j], y1[j] = u, v
  #print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x1[i], y1[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='upper ri
plt.title("Feature #2: Standard Deviation (Eating)")
plt.show()
```

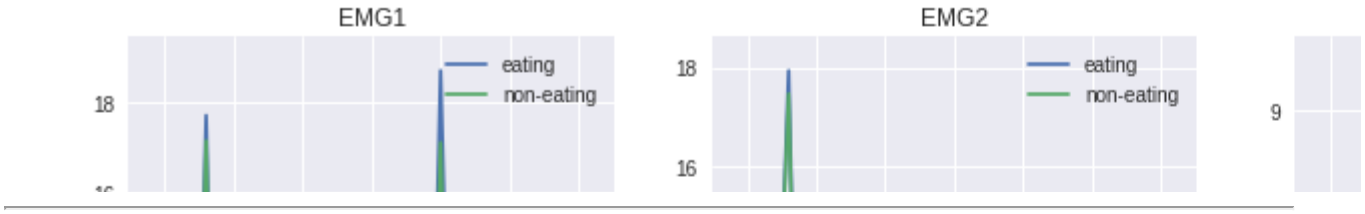



```
df = pd.read_csv("emgnoneatingfile.csv")
a = df['Number'].unique()
a.sort()
li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x2, y2 = {}, {}
# Mean of EMG1 for all Non-Eating
means_noneating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_noneating[i] = df[col].std() # means_noneating[1] is mean of EMG1
  #  print("Done for person ", i)
  lists = sorted(means_noneating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x2[j], y2[j] = u, v
 # print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x2[i], y2[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='upper ri
plt.title("Feature #2: Standard Deviation (Non-Eating)")
plt.show()
```



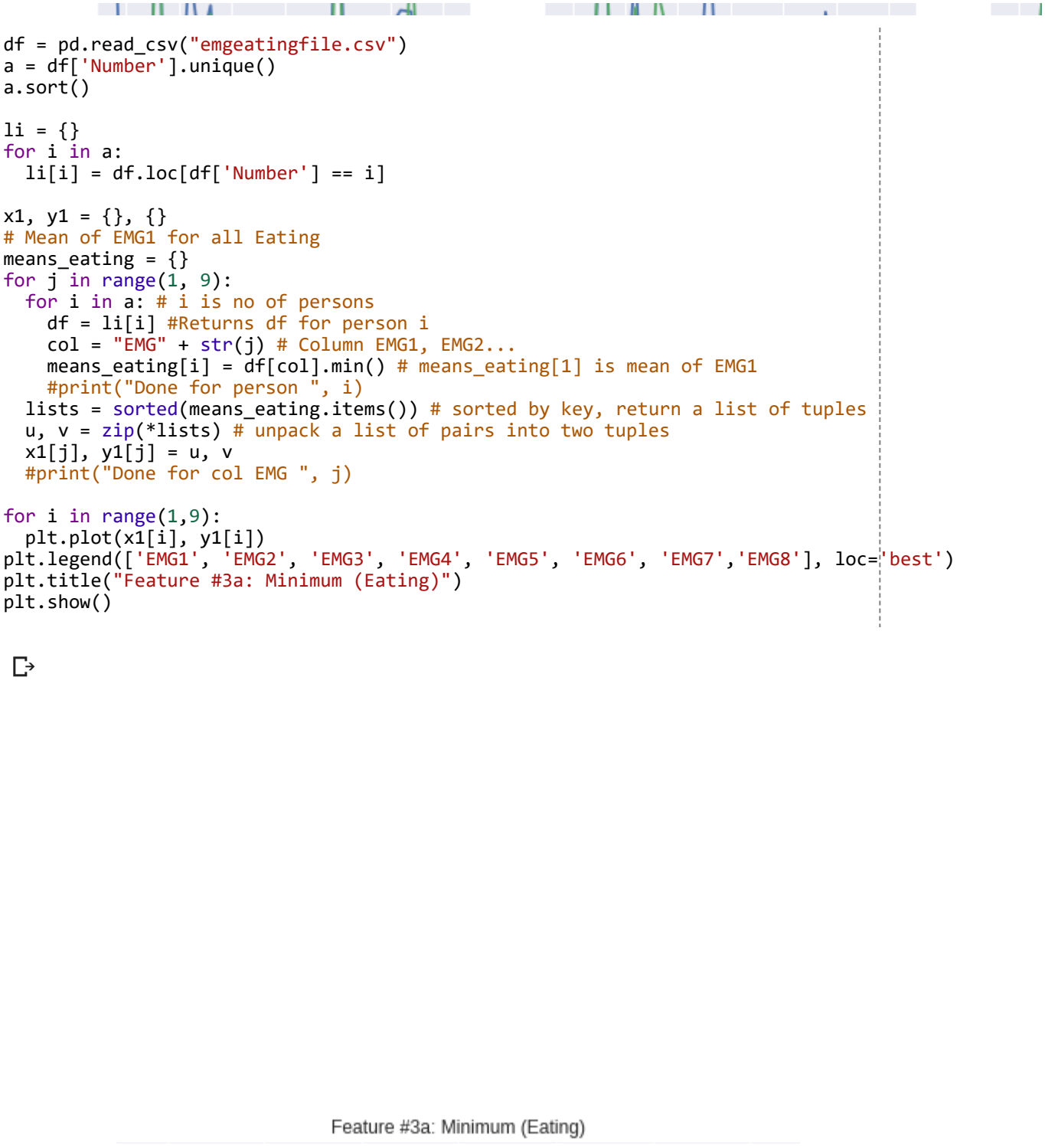
18 EMG1

```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Feature #2: Standard Deviation', fontsize=30)
for i in range(1,9):
  plt.subplot(2, 4, i).set_title('EMG{}'.format(i))
  plt.plot(x1[i], y1[i])
  plt.plot(x2[i], y2[i])
  plt.legend(['eating', 'non-eating'], loc='upper right')

plt.show()
```

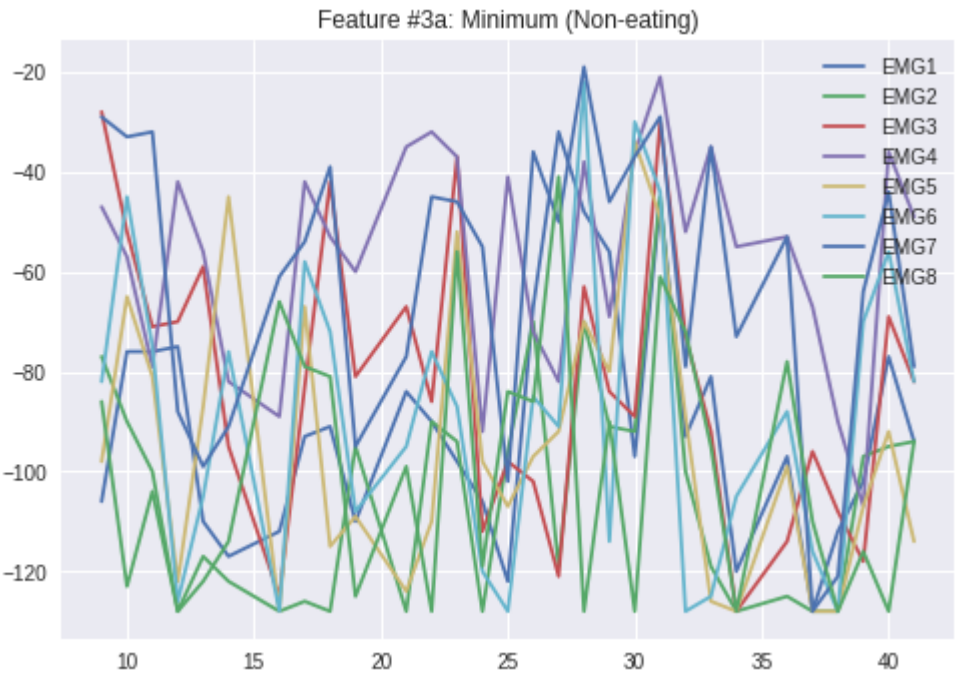


Feature 3a: Minimum

```
df = pd.read_csv("emgeatingfile.csv")
a = df['Number'].unique()
a.sort()

li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x1, y1 = {}, {}
# Mean of EMG1 for all Eating
means_eating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_eating[i] = df[col].min() # means_eating[1] is mean of EMG1
    #print("Done for person ", i)
  lists = sorted(means_eating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x1[j], y1[j] = u, v
  #print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x1[i], y1[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7','EMG8'], loc='best')
plt.title("Feature #3a: Minimum (Eating)")
plt.show()
```

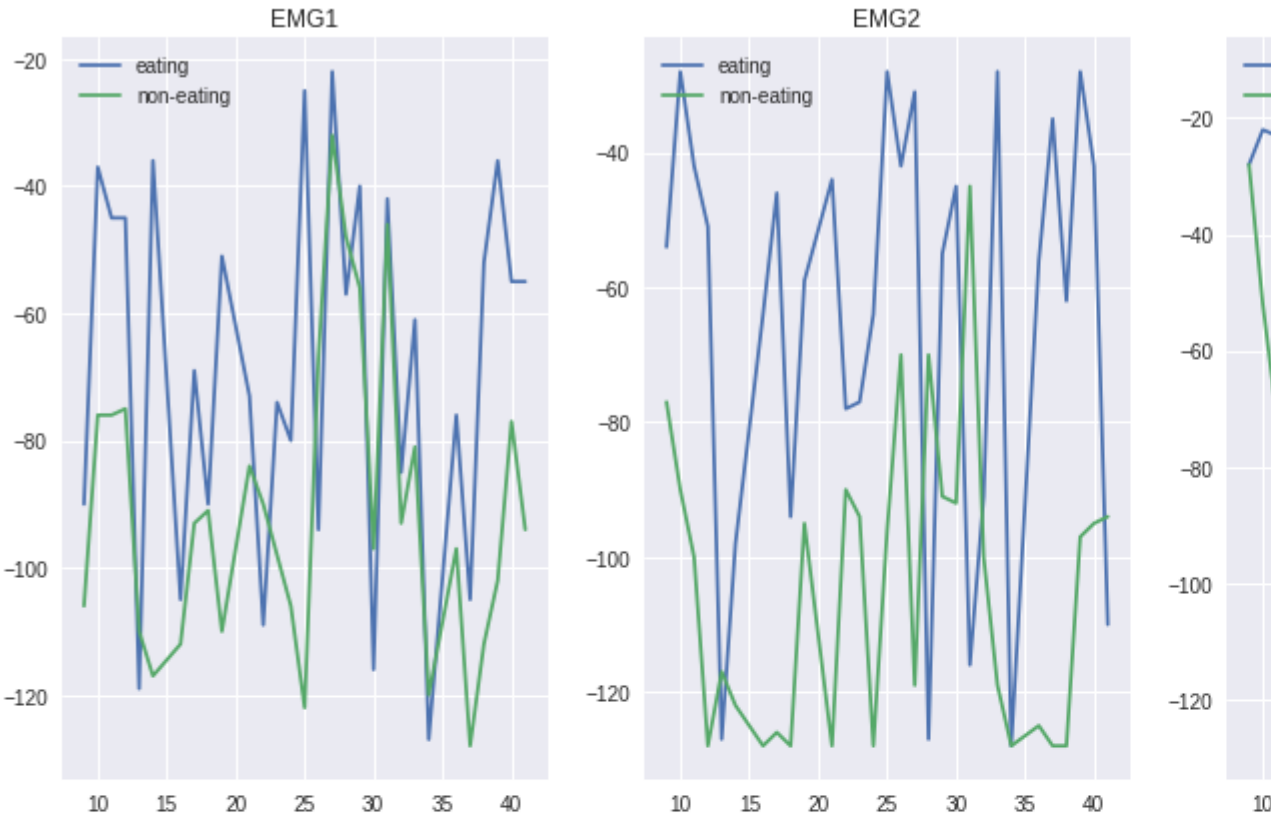


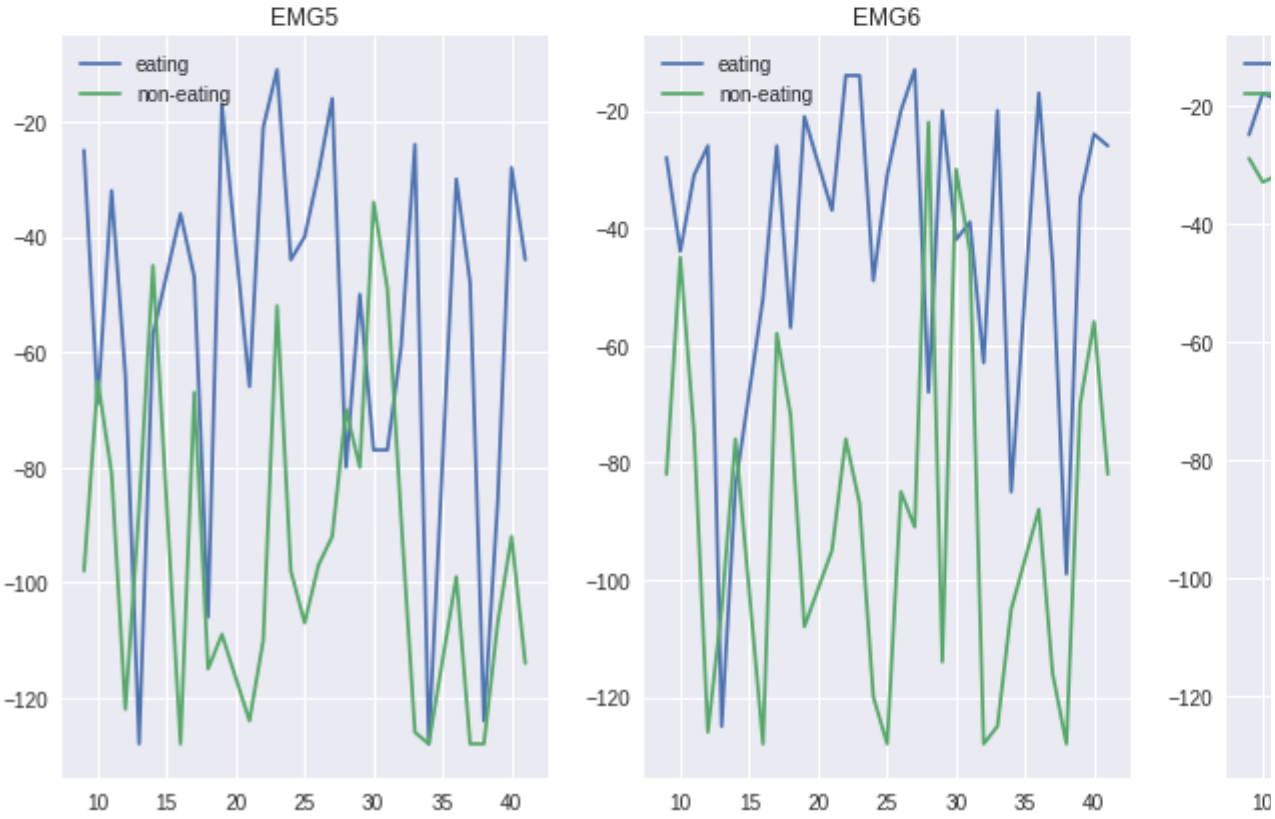

```
df = pd.read_csv("emgnoneatingfile.csv")
a = df['Number'].unique()
a.sort()
li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x2, y2 = {}, {}
# Mean of EMG1 for all Non-Eating
means_noneating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_noneating[i] = df[col].min() # means_noneating[1] is mean of EMG1
    #print("Done for person ", i)
  lists = sorted(means_noneating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x2[j], y2[j] = u, v
  #print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x2[i], y2[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='best')
plt.title("Feature #3a: Minimum (Non-eating)")
plt.show()
```




```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Feature #3a: Minimum', fontsize=30)
for i in range(1,9):
  plt.subplot(2, 4, i).set_title('EMG{}'.format(i))
  plt.plot(x1[i], y1[i])
  plt.plot(x2[i], y2[i])
  plt.legend(['eating', 'non-eating'], loc='upper left')

plt.show()
```

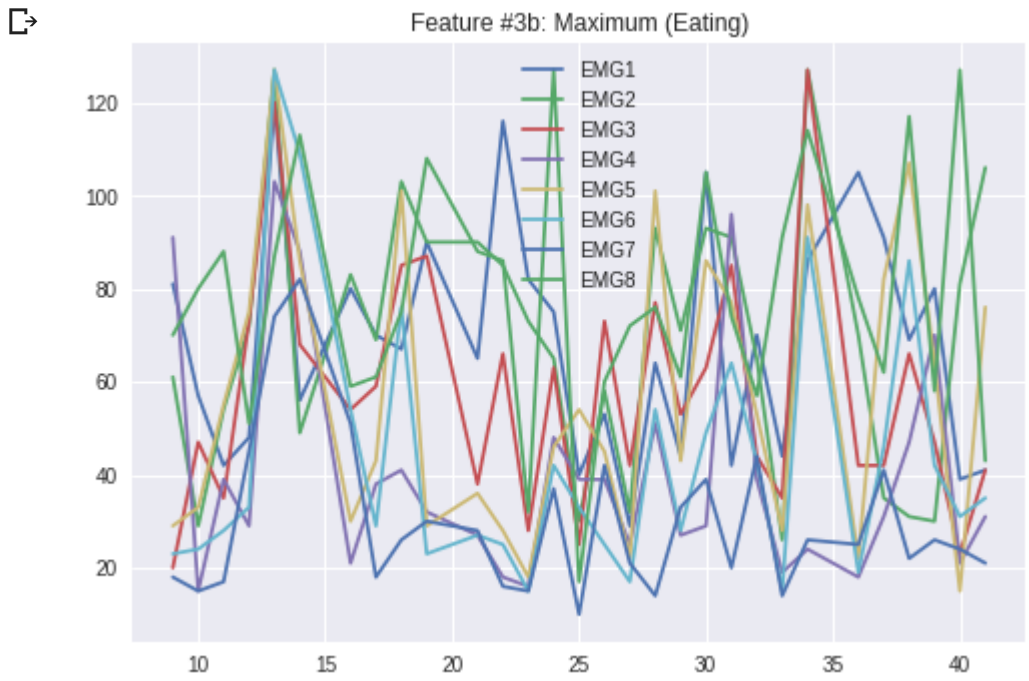
Feature #3a: Mi





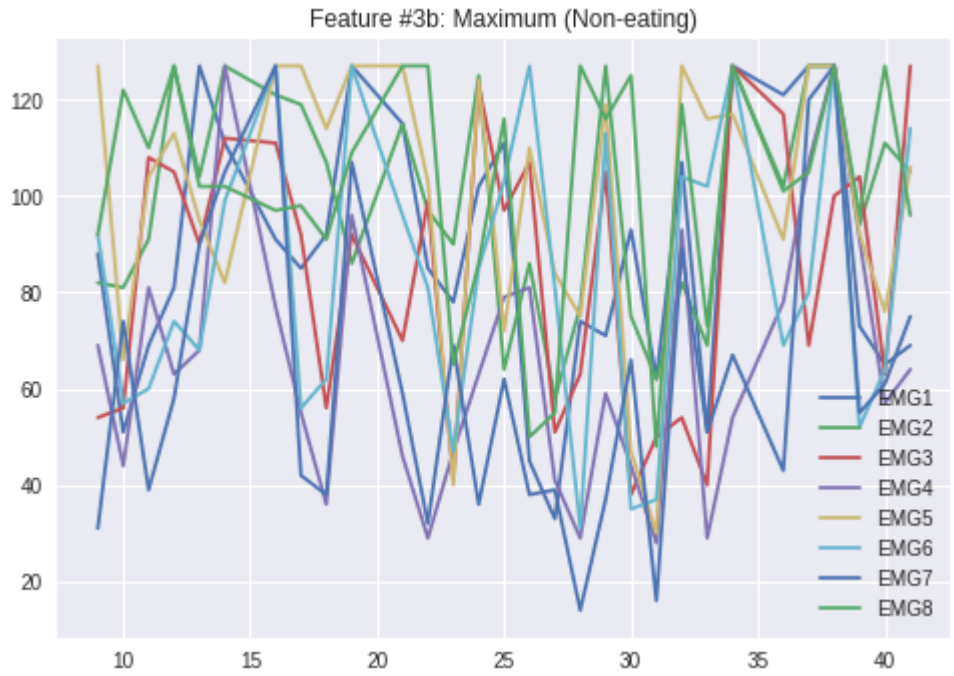
Feature 3b: Maximum

```
df = pd.read_csv("emgeatingfile.csv")
a = df['Number'].unique()
a.sort()

li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x1, y1 = {}, {}
# Mean of EMG1 for all Eating
means_eating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_eating[i] = df[col].max() # means_eating[1] is mean of EMG1
    #print("Done for person ", i)
  lists = sorted(means_eating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x1[j], y1[j] = u, v
  #print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x1[i], y1[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='best')
plt.title("Feature #3b: Maximum (Eating)")
plt.show()
```

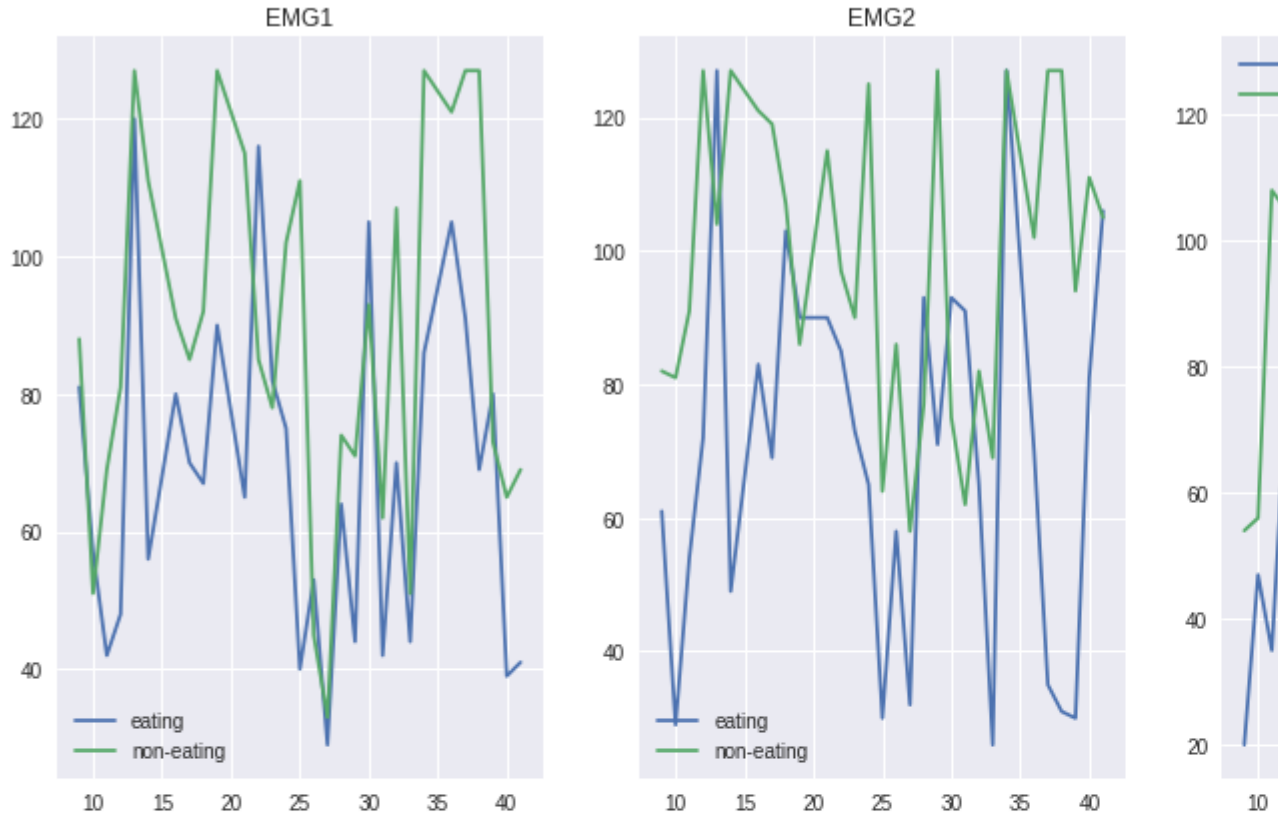


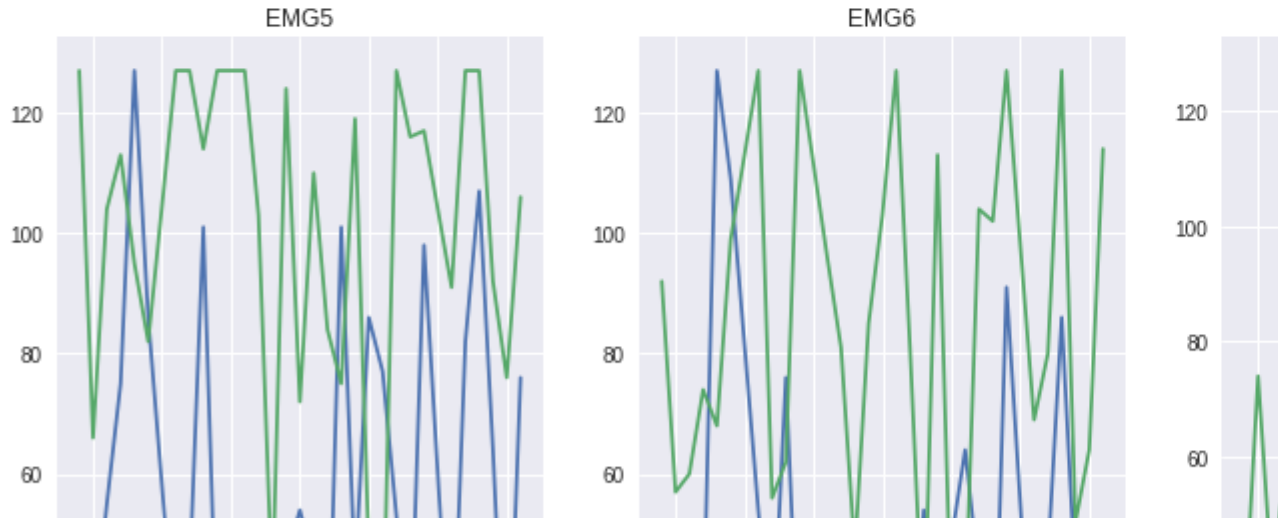

```
df = pd.read_csv("emgnoneatingfile.csv")
a = df['Number'].unique()
a.sort()
li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x2, y2 = {}, {}
# Mean of EMG1 for all Non-Eating
```

```
means_noneating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_noneating[i] = df[col].max() # means_noneating[1] is mean of EMG1
    #print("Done for person ", i)
  lists = sorted(means_noneating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x2[j], y2[j] = u, v
  #print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x2[i], y2[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='best')
plt.title("Feature #3b: Maximum (Non-eating)")
plt.show()
```




```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Feature #3b: Maximum', fontsize=30)
for i in range(1,9):
  plt.subplot(2, 4, i).set_title('EMG{}'.format(i))
  plt.plot(x1[i], y1[i])
  plt.plot(x2[i], y2[i])
  plt.legend(['eating', 'non-eating'], loc='best')

plt.show()
```

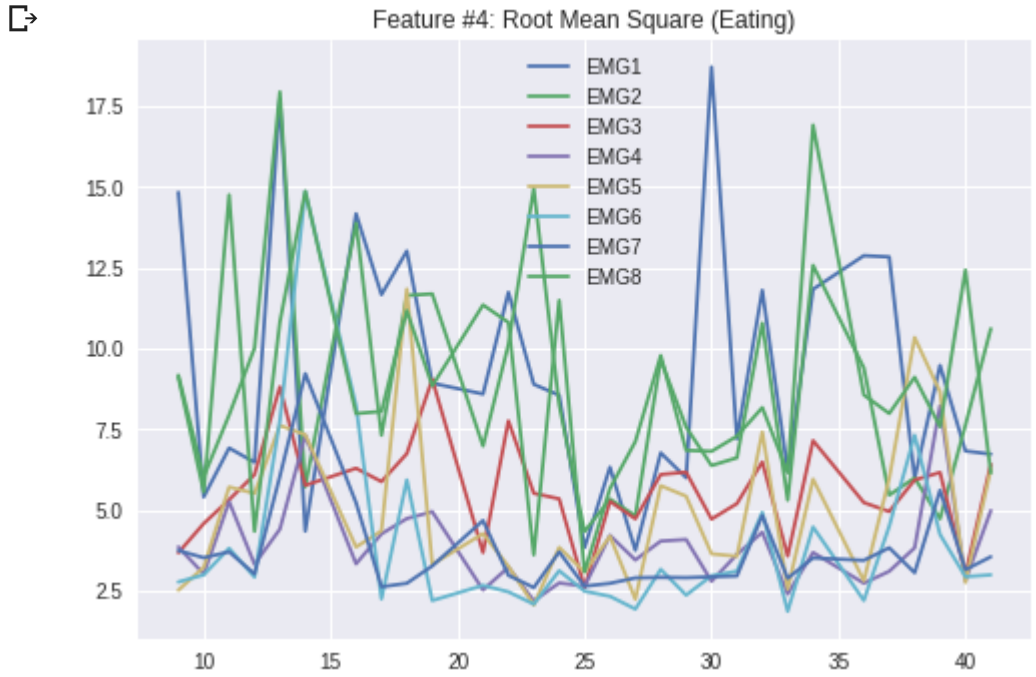




Feature 4: Root Mean Square

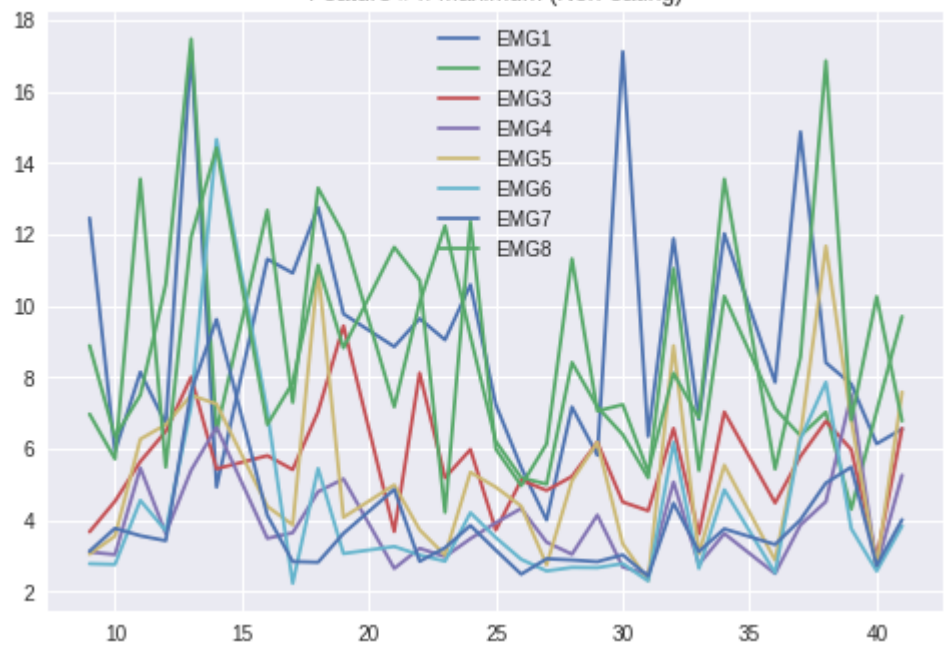
```
df = pd.read_csv("emgeatingfile.csv")
a = df['Number'].unique()
a.sort()


li = {}
for i in a:
```

```
  li[i] = df.loc[df['Number'] == i]

x1, y1 = {}, {}
# Mean of EMG1 for all Eating
means_eating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_eating[i] = np.sqrt(np.mean(df[col]**2)) # Root Mean Square
    #print("Done for person ", i)
  lists = sorted(means_eating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x1[j], y1[j] = u, v
  #print("Done for col EMG ", j)

for i in range(1,9):
  plt.plot(x1[i], y1[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='best')
plt.title("Feature #4: Root Mean Square (Eating)")
plt.show()
```

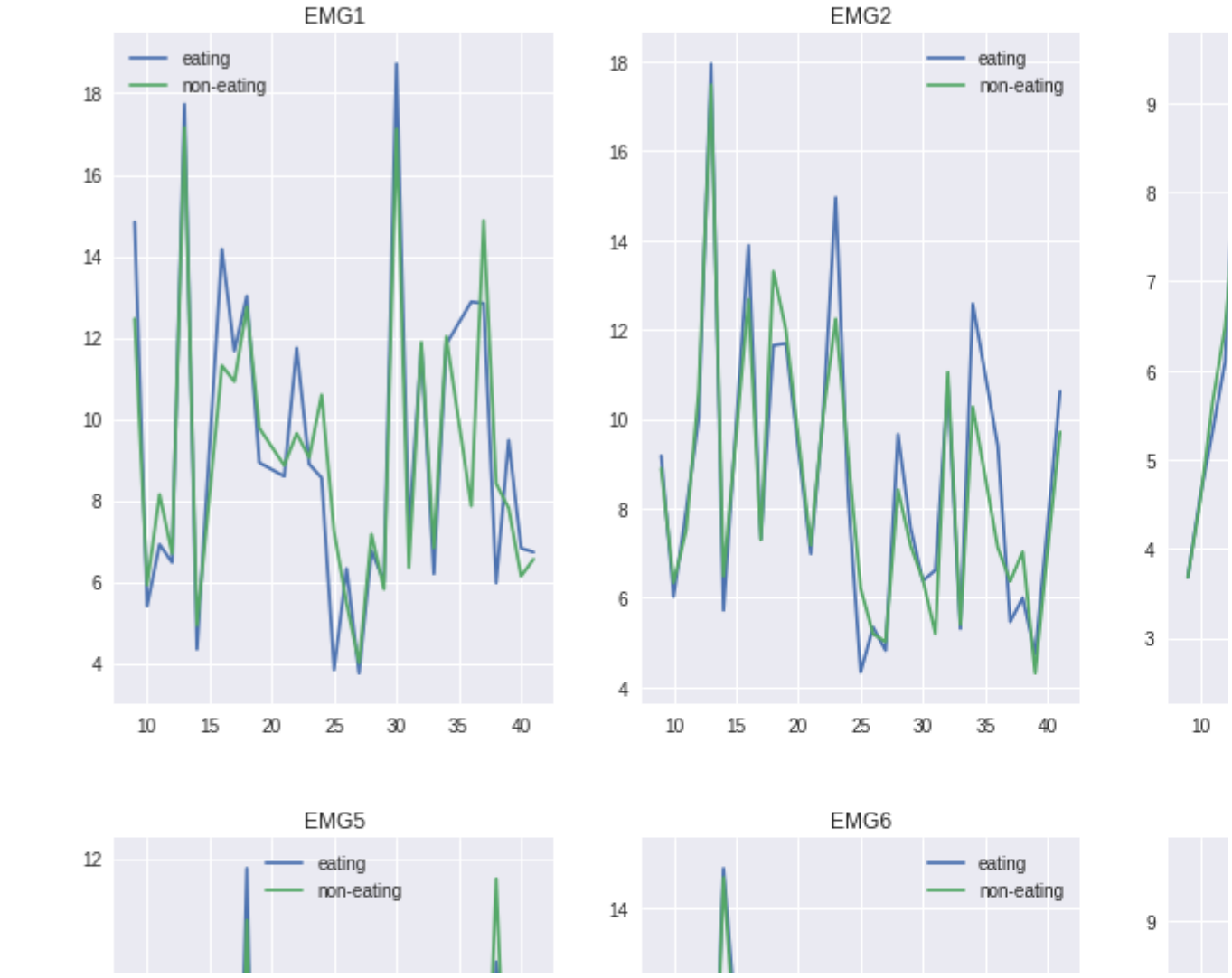



```
df = pd.read_csv("emgnoneatingfile.csv")
a = df['Number'].unique()
a.sort()
li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x2, y2 = {}, {}
# Mean of EMG1 for all Non-Eating
means_noneating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...
    means_noneating[i] = np.sqrt(np.mean(df[col]**2)) # Root Mean Square
    #print("Done for person ", i)
  lists = sorted(means_noneating.items()) # sorted by key, return a list of tuples
  u, v = zip(*lists) # unpack a list of pairs into two tuples
  x2[j], y2[j] = u, v
  #print("Done for col EMG ", j)
```

```
for i in range(1,9):
  plt.plot(x2[i], y2[i])
plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='best')
plt.title("Feature #4: Maximum (Non-eating)")
plt.show()
```



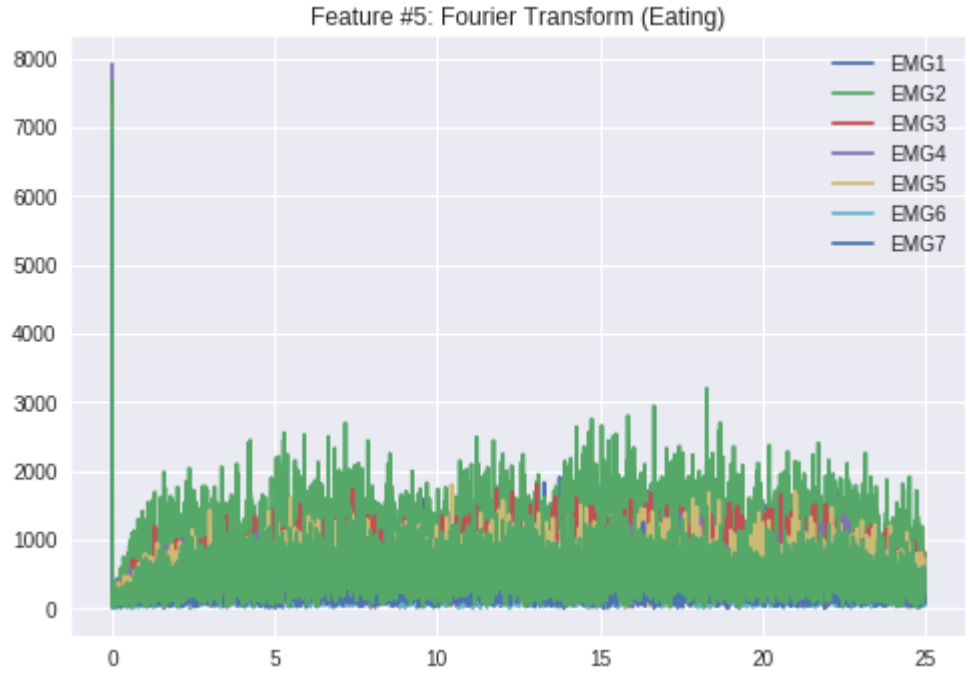

```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Feature #4: Root Mean Square', fontsize=30)
for i in range(1,9):
  plt.subplot(2, 4, i).set_title('EMG{}'.format(i))
  plt.plot(x1[i], y1[i])
  plt.plot(x2[i], y2[i])
  plt.legend(['eating', 'non-eating'], loc='best')

plt.show()
```



Feature 5: Fourier Transform

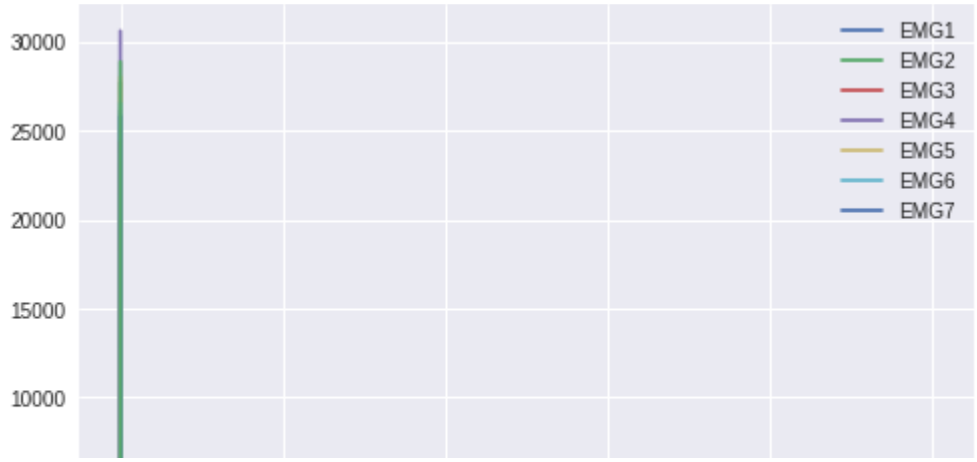
```
import scipy.fftpack

df = pd.read_csv("emgeatingfile.csv")
a = df['Number'].unique()
a.sort()

li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x1, y1 = {}, {}
# Mean of EMG1 for all Eating
means_eating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...

    yf = scipy.fft(df[col].values)
    x = scipy.fftpack.fftfreq(yf.size, 1 / 50) # inverse frequency
```

```
  x1[j], y1[j] = x, yf
  plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='best')
  plt.title("Feature #5: Fourier Transform (Eating)")
  plt.plot(x[:x.size//2], abs(yf)[:yf.size//2])
```

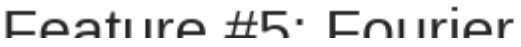


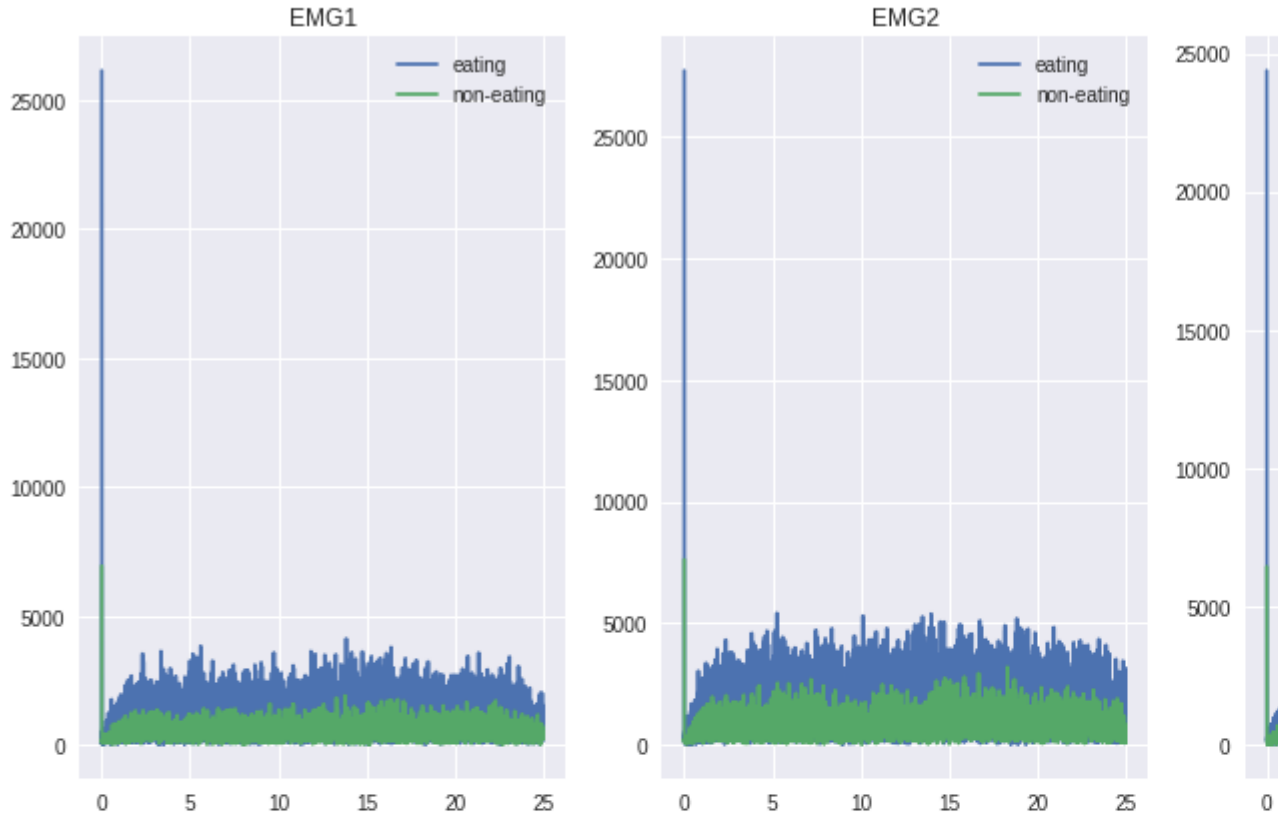

```
df = pd.read_csv("emgnoneatingfile.csv")
a = df['Number'].unique()
a.sort()

li = {}
for i in a:
  li[i] = df.loc[df['Number'] == i]

x2, y2 = {}, {}
# Mean of EMG1 for all Eating
means_eating = {}
for j in range(1, 9):
  for i in a: # i is no of persons
    df = li[i] #Returns df for person i
    col = "EMG" + str(j) # Column EMG1, EMG2...

    yf = scipy.fft(df[col].values)
    x = scipy.fftpack.fftfreq(yf.size, 1 / 50) # inverse frequency
  x2[j], y2[j] = x, yf

  plt.title("Feature #5: Fourier Transform (Non-Eating)")
  plt.legend(['EMG1', 'EMG2', 'EMG3', 'EMG4', 'EMG5', 'EMG6', 'EMG7', 'EMG8'], loc='best')
  plt.plot(x[:x.size//2], abs(yf)[:yf.size//2])
```

Feature #5: Fourier Transform (Non-Eating)

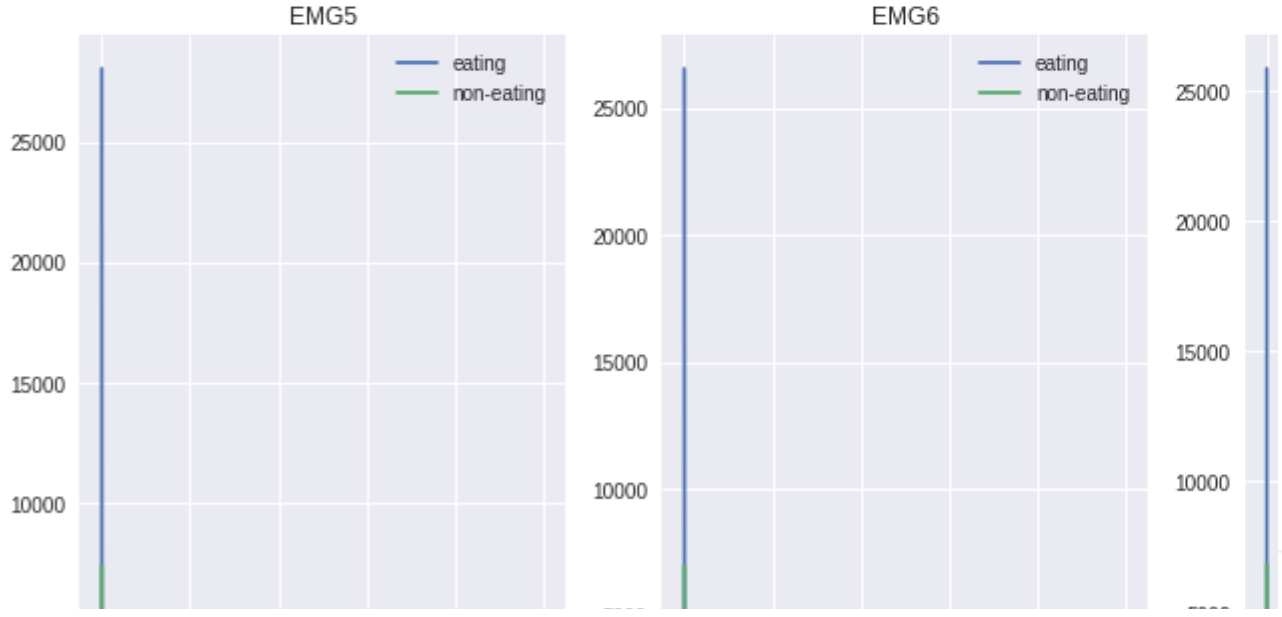



```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Feature #5: Fourier Transform', fontsize=30)
for i in range(1,9):
  plt.subplot(2, 4, i).set_title('EMG{}'.format(i))
  plt.plot(x2[i][:x2[i].size//2], abs(y2[i])[:y2[i].size//2])
  plt.plot(x1[i][:x1[i].size//2], abs(y1[i])[:y1[i].size//2])
  plt.legend(['eating', 'non-eating'], loc='best')

plt.show()
```




EMG1
eating
non-eating
25000
20000
15000
10000
5000
0
0
5
10
15
20
25

EMG2
eating
non-eating
25000
20000
15000
10000
5000
0
0
5
10
15
20
25


EMG5
eating
non-eating
25000
20000
15000
10000

EMG6
eating
non-eating
25000
20000
15000
10000