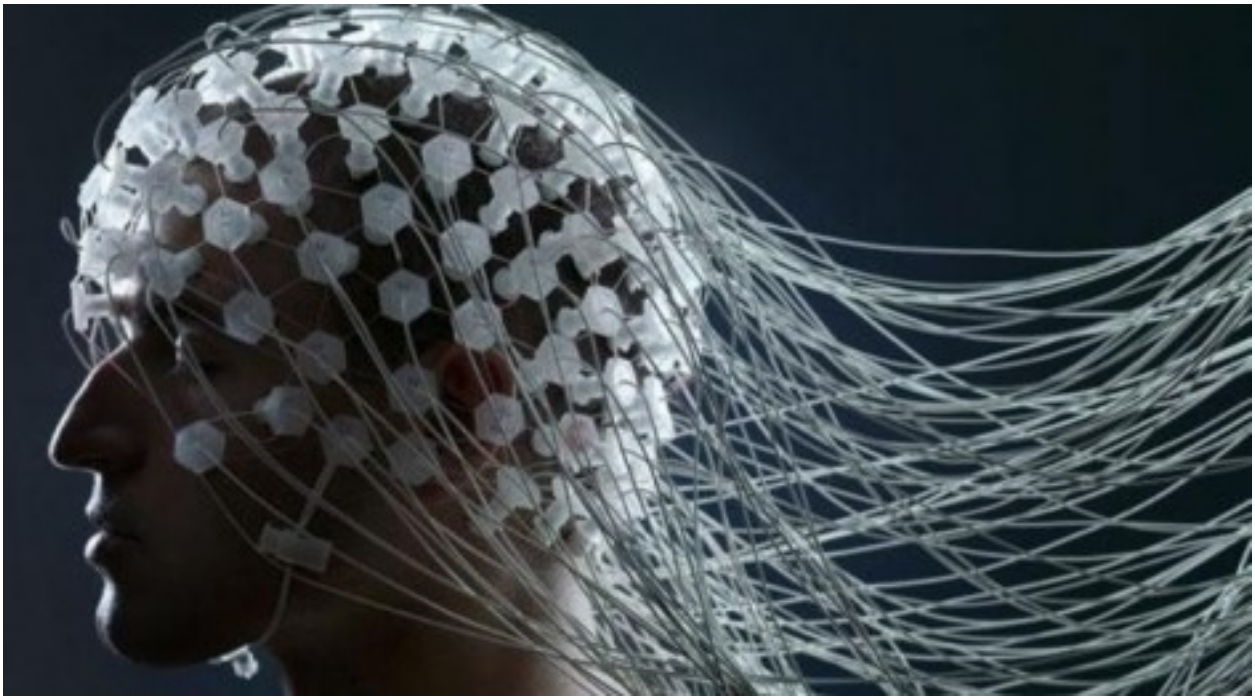
pyseries Documentation

Release 1.0.19

Ryszard Cetnarski

August 10, 2016

Contents



CHAPTER 1

Overwiev

pySeries is a package for statistical analysis of EEG data. Developed for neuro and cognitive science academics looking for a quick start into EEG data analysis with python.

This package is an initial release. Testing in progress

CHAPTER 2

Yalla

2.1 Analysis

2.1.1 Explore

Plots for common EEG analysis methods

- Event related potentials (ERP's)
- Periodogram and Welch power spectrum
- Spectrogram

Requires the signal to be epoched by Preprocessing.MakeSlices method. Epochs are grouped and colored by conditions.

pyseries.Analysis.Explore.**PlotErp**(*electrode_slices*, *n_back*)

Plot average and individual traces of epoched signal per condition.

Parameters

- **electrode_slices** (*dict*) – Key is a condition label, contains array of signal epochs.
- **n_back** (*int*) – information about the time of event from the end of the epoch.

pyseries.Analysis.Explore.**PlotPowerSpectrum**(*electrode_slices*, *exact_sr=498*, *freq_min=0*, *freq_max=100*, *mode='period'*, *name=''*, *save_path=''*)

Plot average and individual traces of power spectrum for signal epochs.

Parameters

- **electrode_slices** (*dict*) – Key is an event name, values are signal epochs.
- **exact_sr** (*float*) – exact sampling rate info from the EEG amplifier
- **freq_max** (*freq_min,*) – lower and upper frequency limits for plotting (default 0, 50 Hz)
- **mode** (*{'period', 'welch'}*) – Default is period which produces periodogram. change to 'welch' for alternative method of power estimation.
- **name** (*str, optional*) – title of the figure

Returns **power_density** – keys are event names. Under each key there are two np.arrays. One array stores info about frequency bins, second has power densities for all trials separately.

Return type dict

pyseries.Analysis.Explore.**PlotSpectrogram**(*electrode_slices*, *n_back*, *n_forth*)

Plot average spectrogram (time by frequency). Uses n_back and n_forth to approximate the time bin where event occured.

Parameters

- **electrode_slices** (*dict*) – Key is a condition label, contains array of signal epochs.
- **n_back** (*int*) – information about the time of event from the end of the epoch.
- **n_forth** (*int*) – information about the time of event from the begining of the epoch.

2.1.2 Anova

Statistical tools for time-series analysis.

- One-way: Find time intervals where signals recorded under single conditions differ from the baseline.
- Two-way: Find interactions between varying conditions time intervals of the recorded signal.
- Repeated-measures: Find time intervals where the signal was systematically changing on a group level.

pyseries.Analysis.Anova.**one_way**(*groups*)

Run one way analysis of variance on n groups of equal length.

- Identify which groups significanlty deviate from the grand mean.
- Prints a table with a spss-style output.

Parameters **group** (*list or ndarray*) –

If list then each index represents a group,
If ndarray then each column represents a group.

Returns

- **F** (*double*) – F-value, ratio between effect and error sum of squares.
- **p** (*double*) – Probability of obtaining F-value by chance.
- **df_effect** (*int*) – degrees of freedom for the effect (n groups -1).
- **df_error** (*int*) – degrees of freedom for the error (n groups * (n samples - 1)).

pyseries.Analysis.Anova.**two_way**(*data*, *f1_name*, *f2_name*)

Run two way analysis of variance in a factor by factor design.

- Identify main effects for each factor.
- Identify interaction between factors.
- Print a table with a spss-style output.

Parameters **data** (*ndarray*) –

Each row represents a 1st factor level.
Each column respresents a 2nd factor level.
Each layer (depth dimension) is an observation.

CHAPTER 3

Work

CHAPTER 4

Examples

4.1 ssvep_analysis

```
import sys
sys.path.insert(0, '/Users/user/Desktop/repo_for_pyseries/pyseries')

import pyseries.LoadingData as loading
import pyseries.Preprocessing as prep
import pyseries.Analysis as analysis
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy import stats

# Notebook settings (i.e. this thing settings)
%matplotlib inline
#%matplotlib notebook
#Change to %matplotlib notebook to be able to zoom, pan, etc the figures,
#inline is only used so the notebook can be exported to .rst format and hosted on readthedocs
%load_ext autoreload
%autoreload 2
```

```
def calc_corr(path):
    #Read single subject data - eeg, info about eeg and info from experimental app (unity)
    recording = loading.Read_edf.Combine_EDF_XML(path, 3, 70)
    #Display markers over the whole signal
    prep.Epochs.mark_events(recording,['EEG O1'], subject_name = path)

    #Define epochs for analysis
    epochs_info= {"Please Count": [0, 500*10], "Only Look": [0, 500 *10]}
    #Create them by slicing the signal
    epochs = prep.Epochs.Make_Epochs_for_Channels(recording, ['EEG O1','EEG O2','EEG P3', 'EEG P4'],e
    #Re-reference, because only by subtracting P from O-electrodes ssvep response becomes visible
    new_ref = {}
    new_ref['Only Look'] = epochs ['EEG O2']['Only Look'] - epochs['EEG P4']['Only Look']
    new_ref['Please Count'] = epochs ['EEG O2']['Please Count'] - epochs ['EEG P4']['Please Count']
    new_epochs  = {"O-P":new_ref}

    #Get the accuracy in counting condition
    responses = recording['events'][recording['events']["code"] == "responded"]
    accuracy = responses['response'] / responses['expected']

    #Get the power spectra in two conditions
```

```
    power_density= analysis.Explore.PlotPowerSpectrum(new_epochs['O-P'], 498, mode = 'period', name =

    ssvep = analysis.Normalize.Z_score( power_density['Please Count'][1][:,49] )
    accuracy = analysis.Normalize.Z_score( accuracy )

    return ssvep, accuracy, power_density
```

```
def plot_slow_ssvep():
    #Slow is 5Hz flicker
    paths = ['/Users/user/Desktop/nagrania_eeg/ssvep/Blazej_13_06_16/',
             '/Users/user/Desktop/nagrania_eeg/ssvep/Ania_14_06_16/',
             '/Users/user/Desktop/nagrania_eeg/ssvep/Karen_14_06_16/',
             '/Users/user/Desktop/nagrania_eeg/ssvep/Agnieszka_03_06/',
             '/Users/user/Desktop/nagrania_eeg/ssvep/Kuba_14_06_16/',
             '/Users/user/Desktop/nagrania_eeg/ssvep/Rysiek_03_06/'
             ]

    all_ssvep = []
    all_acc = []

    saving = {}
    for p in paths:
        ssvep, acc, pxx = calc_corr(p)
        all_ssvep.extend(ssvep)
        all_acc.extend(acc)

        saving[p] = ssvep

    sns.jointplot(x = np.array(all_ssvep),y =  np.array(all_acc), kind="reg")
    return saving
```

```
def plot_fast_ssvep():
    #fast is 20 Hz flicker
    paths = ['/Users/user/Desktop/nagrania_eeg/ssvep_20hz/Agnieszka_03_06/',
             '/Users/user/Desktop/nagrania_eeg/ssvep_20hz/Rysiek_03_06/']

    for path in paths:

        #Read single subject data - eeg, info about eeg and info from experimental app (unity)
        recording = loading.Read_edf.Combine_EDF_XML(path, 0, 70)
        #Define epochs for analysis
        epochs_info= {"Only Look": [0, 500 *10]}
        #Create them by slicing the signal
        epochs = prep.Epochs.Make_Epochs_for_Channels(recording, ['EEG O1','EEG O2','EEG P3','EEG P4
        #Re-reference, because oonly then ssvep response becomes visible
        new_ref = {}
        new_ref['Only Look'] = epochs ['EEG O2']['Only Look'] - epochs['EEG P4']['Only Look']
        new_epochs  = {"O-P":new_ref}

        #Get the power spectra in two conditions
        power_density= analysis.Explore.PlotPowerSpectrum(new_epochs['O-P'], 498, mode = 'period', na
```

```
plot_slow_ssvep()
```

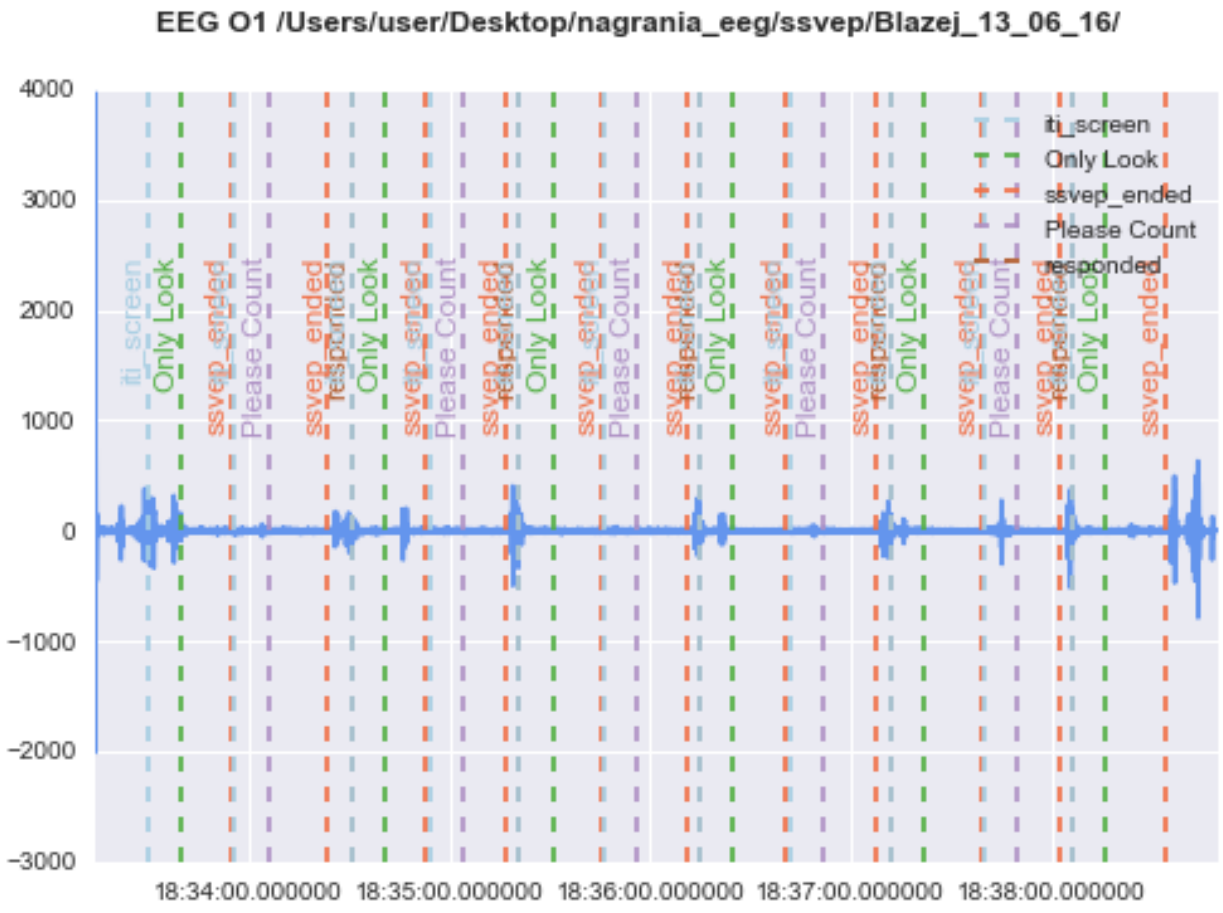
```
Channels:
EEG F3
EEG F4
EEG P3
```

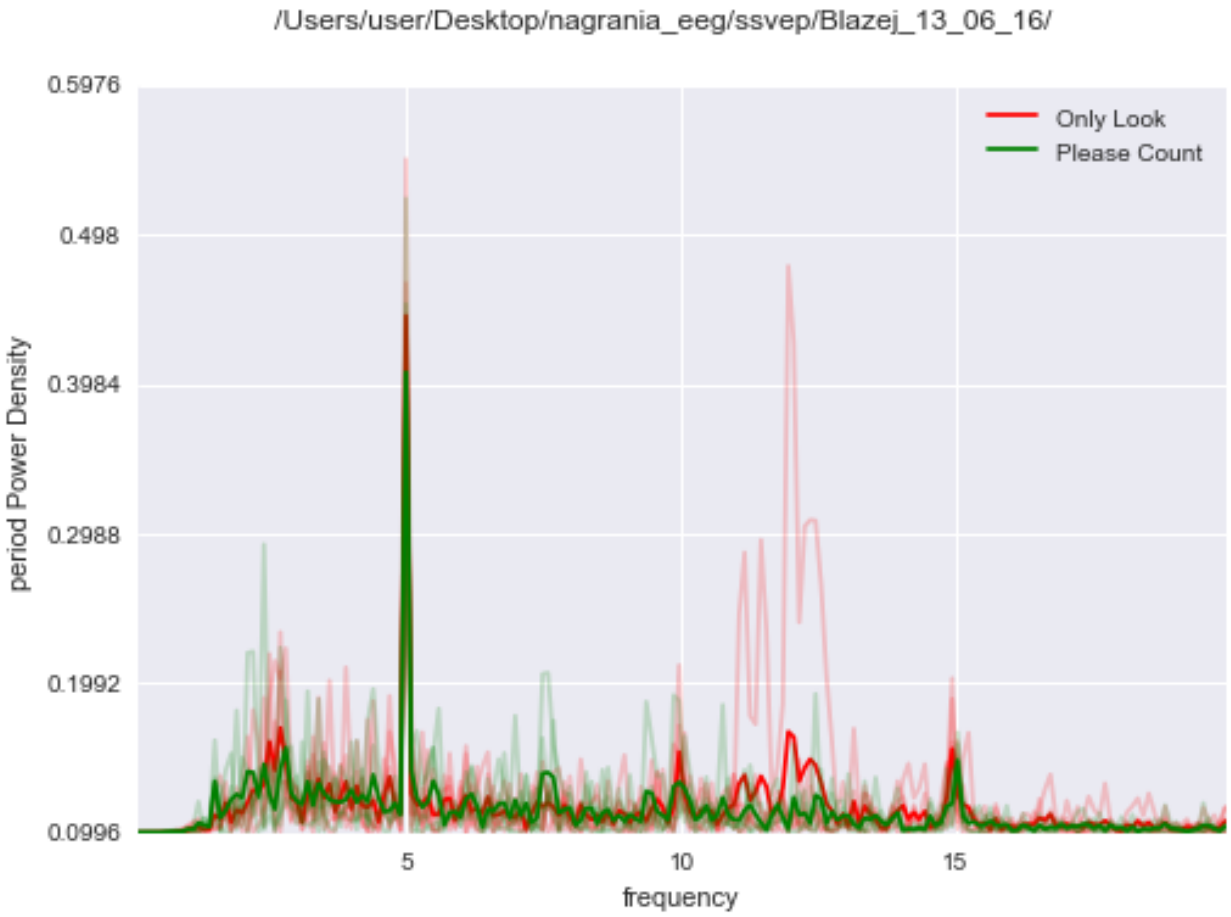
```
EEG P4
EEG O1
EEG O2
EEG T6
EEG A2
EEG Pz
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep/Blazej_13_06_16/
Channels:
EEG Fp1
EEG F3
EEG F4
EEG C3
EEG C4
EEG P3
EEG P4
EEG O1
EEG O2
EEG T5
EEG T6
EEG Pz
S1
S2
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep/Ania_14_06_16/
Channels:
EEG Fp1
EEG F3
EEG F4
EEG C3
EEG C4
EEG P3
EEG P4
EEG O1
EEG O2
EEG T3
EEG T4
EEG Pz
S1
S2
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep/Karen_14_06_16/
Channels:
EEG F3
EEG F4
EEG C3
EEG C4
EEG P3
EEG P4
EEG O1
EEG O2
EEG A2
EEG Cz
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep/Agnieszka_03_06/
Channels:
EEG Fp2
EEG F3
```

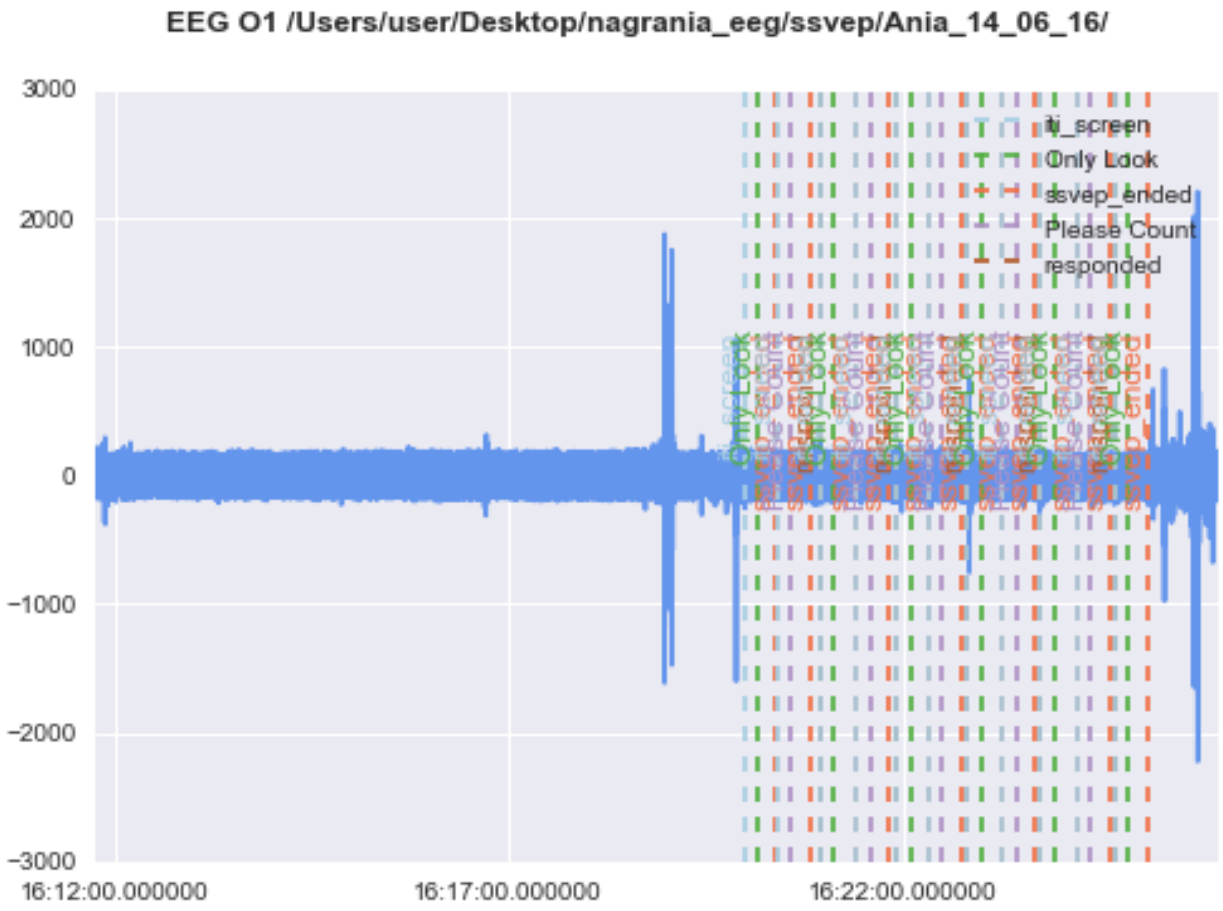
```
EEG F4
EEG C3
EEG C4
EEG P3
EEG P4
EEG O1
EEG O2
EEG T3
EEG Fz
EEG Cz
S1
S2
S3
S4
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep/Kuba_14_06_16/
Channels:
EEG F3
EEG F4
EEG C3
EEG C4
EEG P3
EEG P4
EEG O1
EEG O2
EEG A2
EEG Cz
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep/Rysiek_03_06/
```

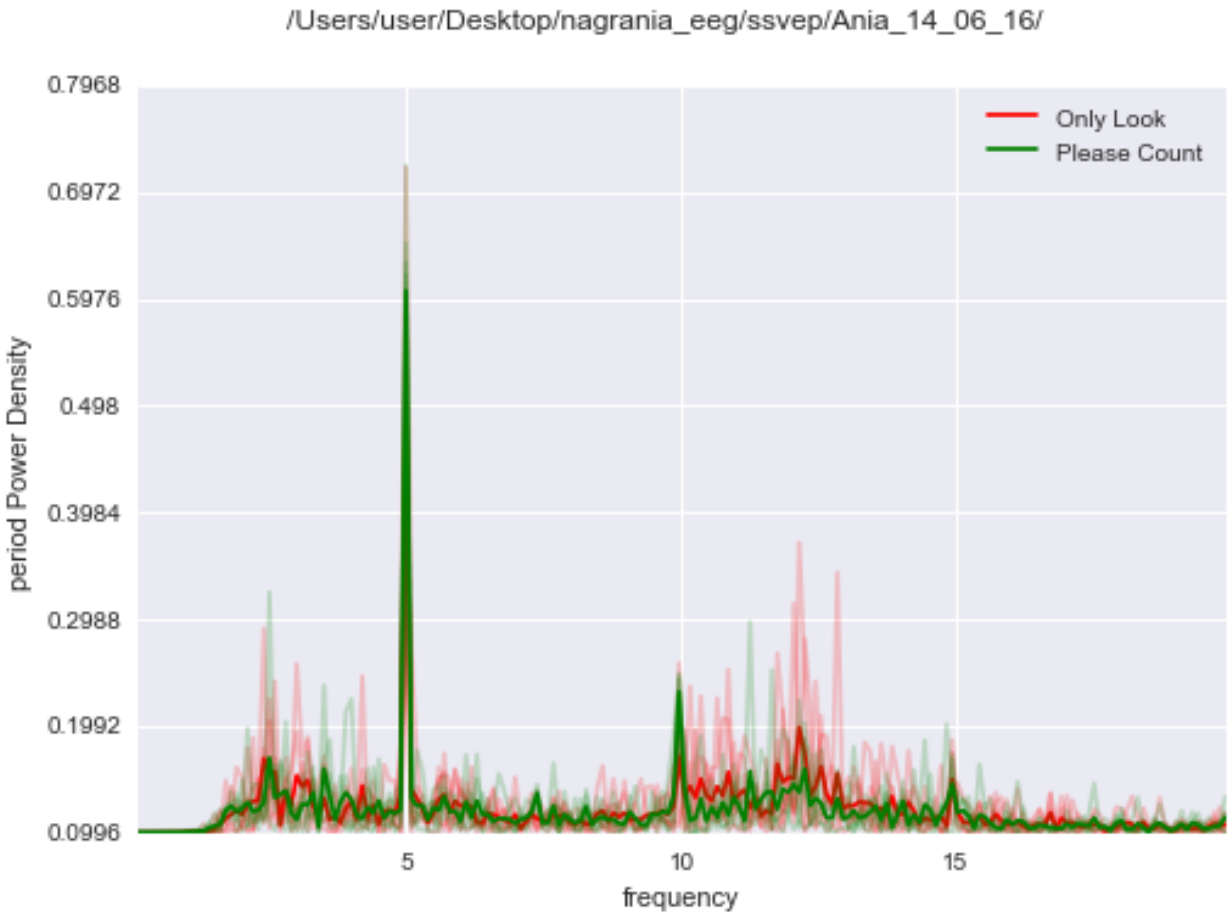
```
/Users/user/anaconda/lib/python3.5/site-packages/statsmodels/nonparametric/kdetools.py:20: VisibleDep
  y = X[:m/2+1] + np.r_[0,X[m/2+1:],0]*1j
```

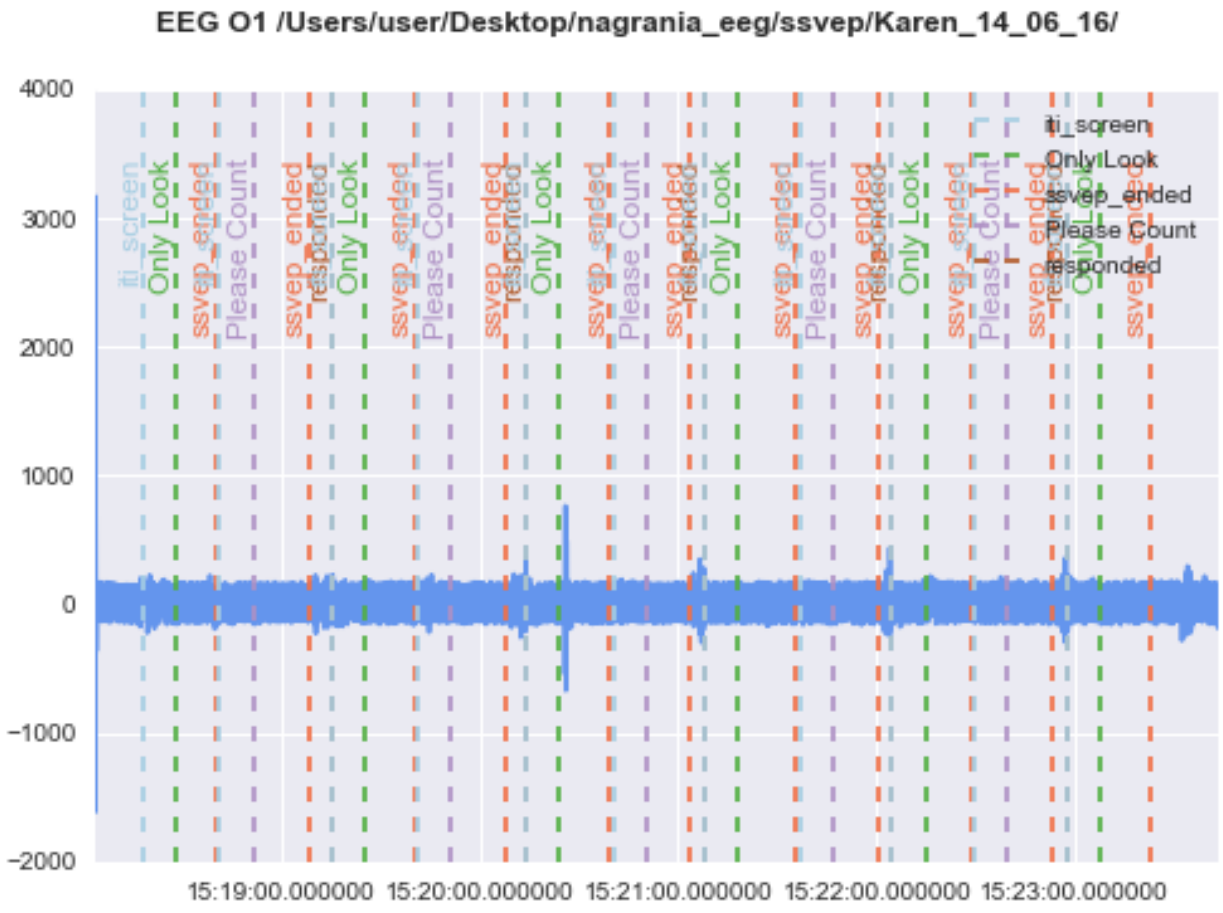
```
{'/Users/user/Desktop/nagrania_eeg/ssvep/Agnieszka_03_06/': array([ 1.62312664, -0.64587493, -1.07742
 '/Users/user/Desktop/nagrania_eeg/ssvep/Ania_14_06_16/': array([-1.88734758,  0.41555267,  1.089259
 '/Users/user/Desktop/nagrania_eeg/ssvep/Blazej_13_06_16/': array([-1.51988438,  1.45266728, -0.5048
 '/Users/user/Desktop/nagrania_eeg/ssvep/Karen_14_06_16/': array([-0.52336364, -0.10604935,  1.53643
 '/Users/user/Desktop/nagrania_eeg/ssvep/Kuba_14_06_16/': array([-1.1445404 , -0.02263413, -1.079133
 '/Users/user/Desktop/nagrania_eeg/ssvep/Rysiek_03_06/': array([ 0.57460658,  1.19457484, -1.4877330
```


EEG O1 /Users/user/Desktop/nagrania_eeg/ssvep/Blazej_13_06_16/
4000
3000
2000
1000
0
−1000
−2000
−3000
18:34:00.000000
18:35:00.000000
18:36:00.000000
18:37:00.000000
18:38:00.000000
iti_screen
Only Look
ssvep_ended
Please Count
responded

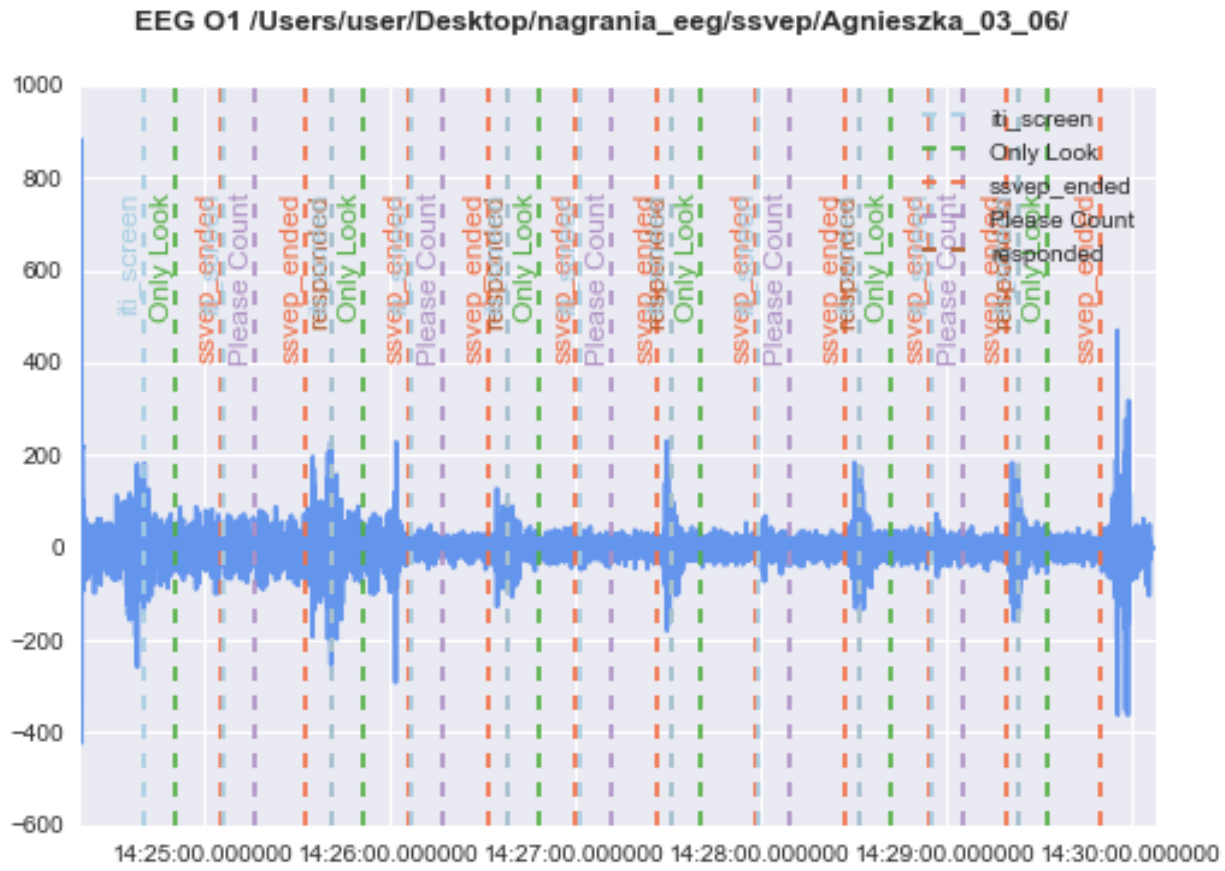

/Users/user/Desktop/nagrania_eeg/ssvep/Blazej_13_06_16/
0.5976
0.498
0.3984
0.2988
0.1992
0.0996
period Power Density
5
10
15
frequency
Only Look
Please Count

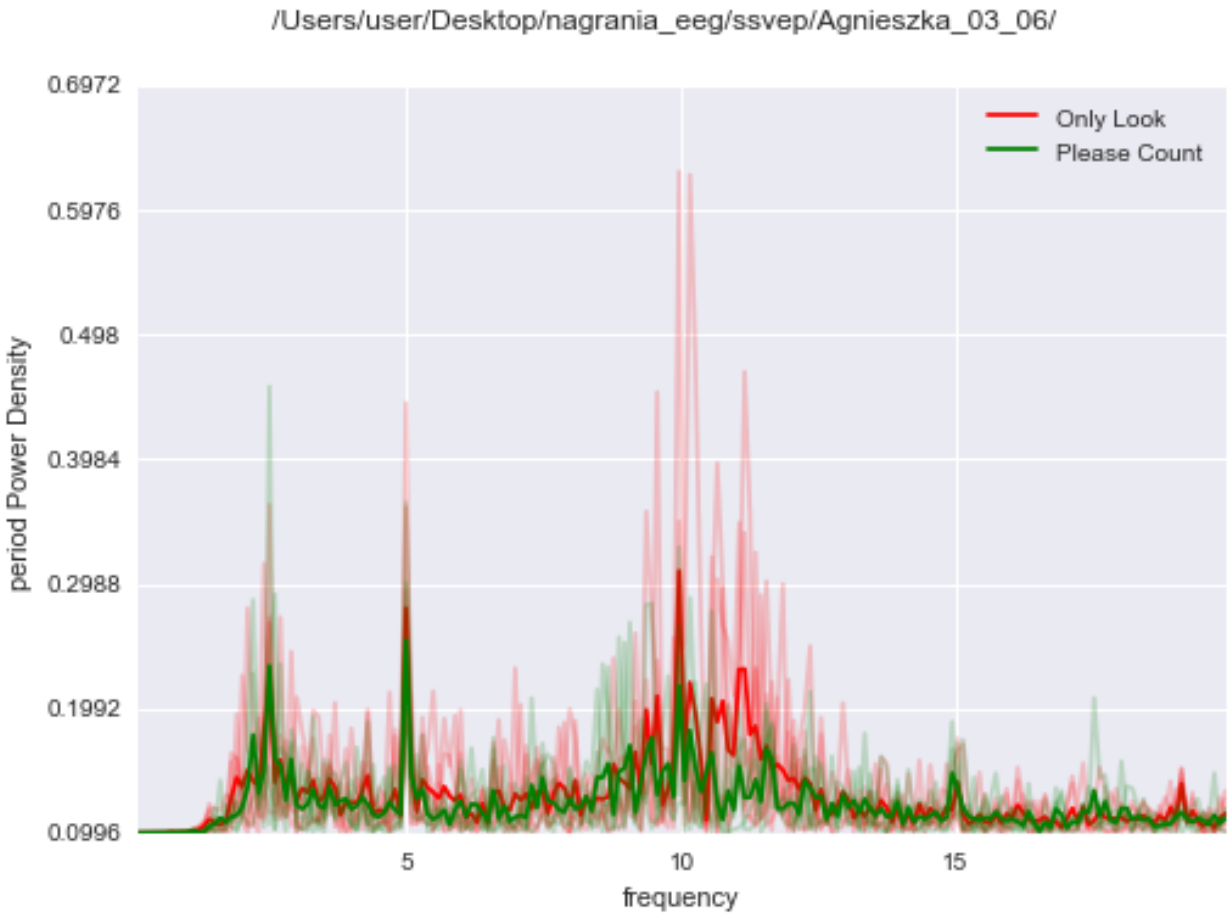

EEG O1 /Users/user/Desktop/nagrania_eeg/ssvep/Ania_14_06_16/
3000
2000
1000
0
−1000
−2000
−3000
16:12:00.000000
16:17:00.000000
16:22:00.000000
ti_screen
Only Look
ssvep_ended
Please Count
responded

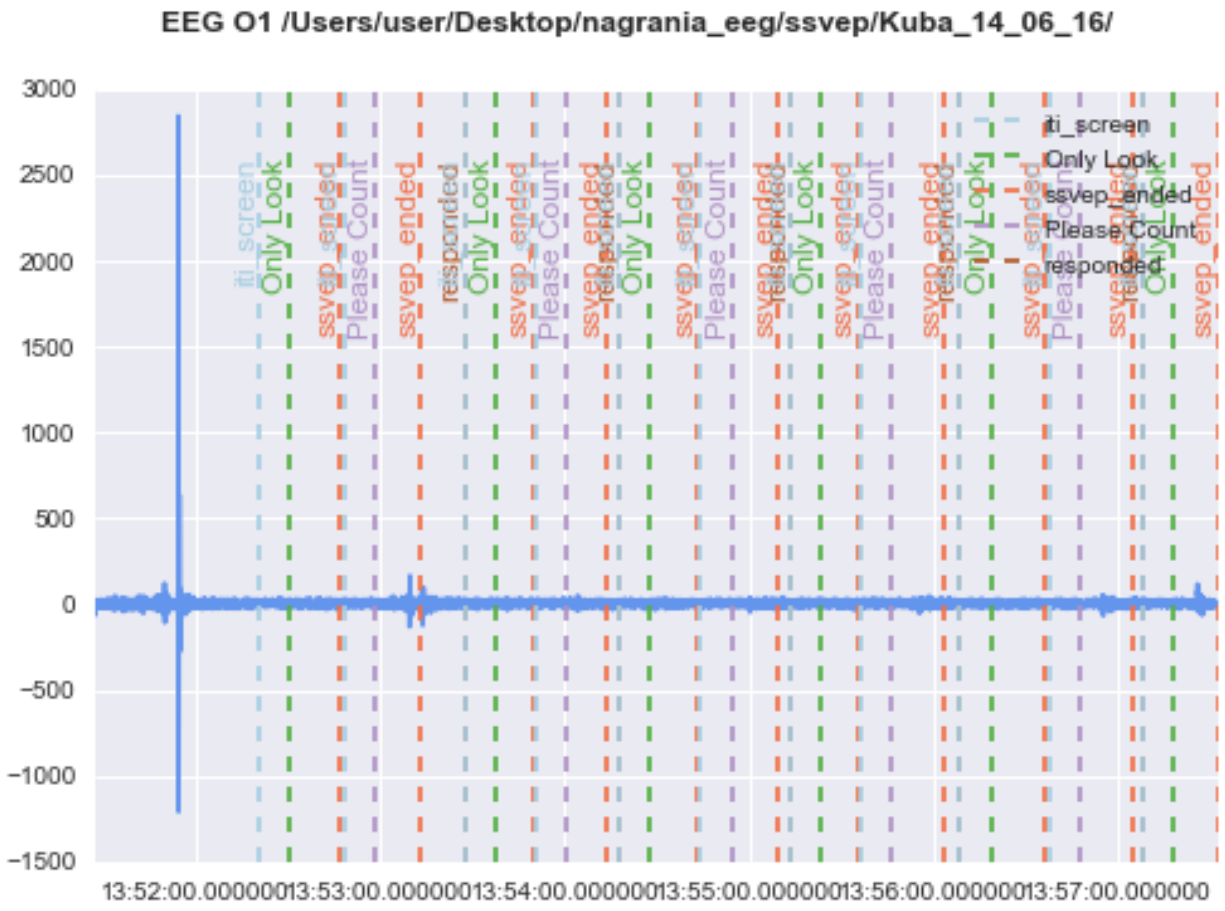

/Users/user/Desktop/nagrania_eeg/ssvep/Ania_14_06_16/
0.7968
0.6972
0.5976
0.498
0.3984
0.2988
0.1992
0.0996
period Power Density
5
10
15
frequency
Only Look
Please Count

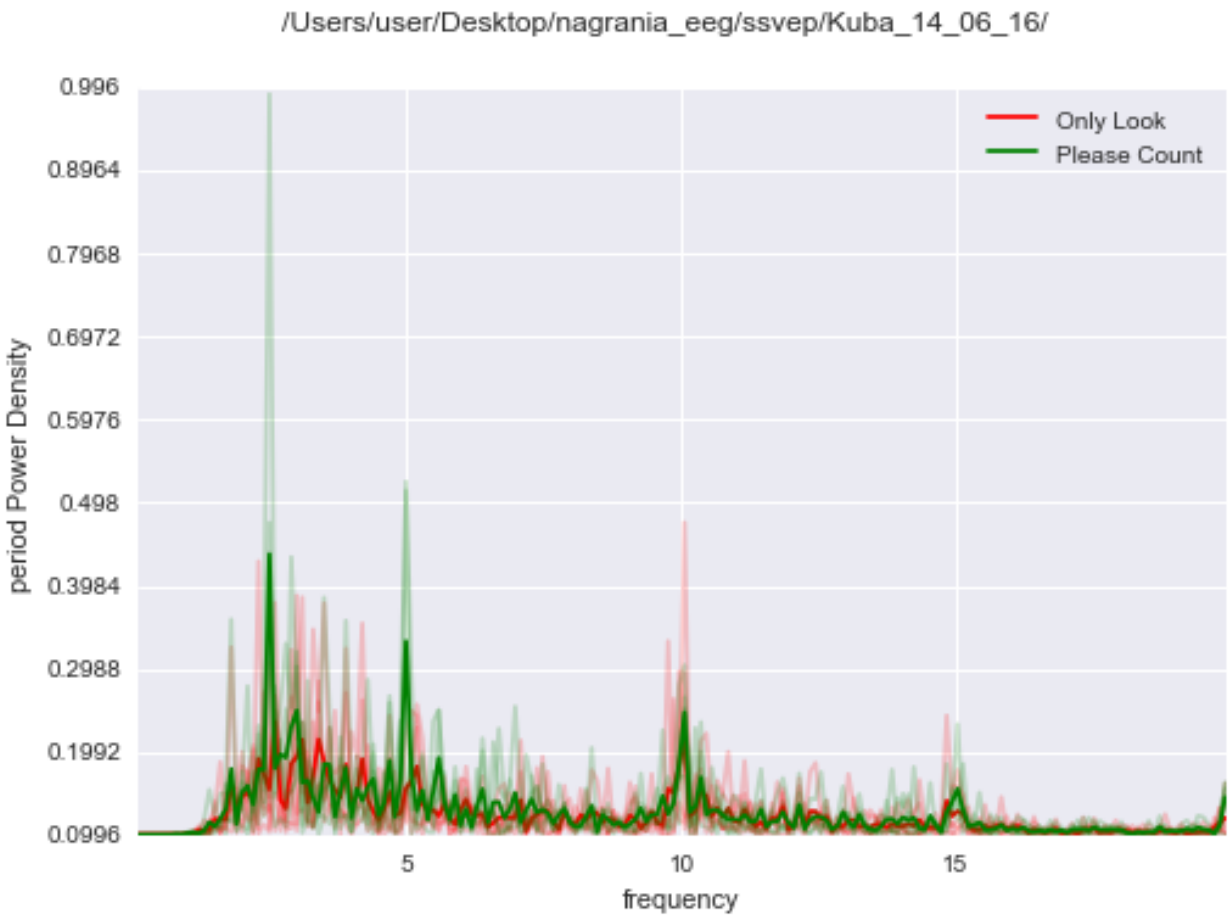

EEG O1 /Users/user/Desktop/nagrania_eeg/ssvep/Karen_14_06_16/
4000
3000
2000
1000
0
−1000
−2000
15:19:00.000000
15:20:00.000000
15:21:00.000000
15:22:00.000000
15:23:00.000000
ti_screen
Only Look
ssvep_ended
Please Count
responded

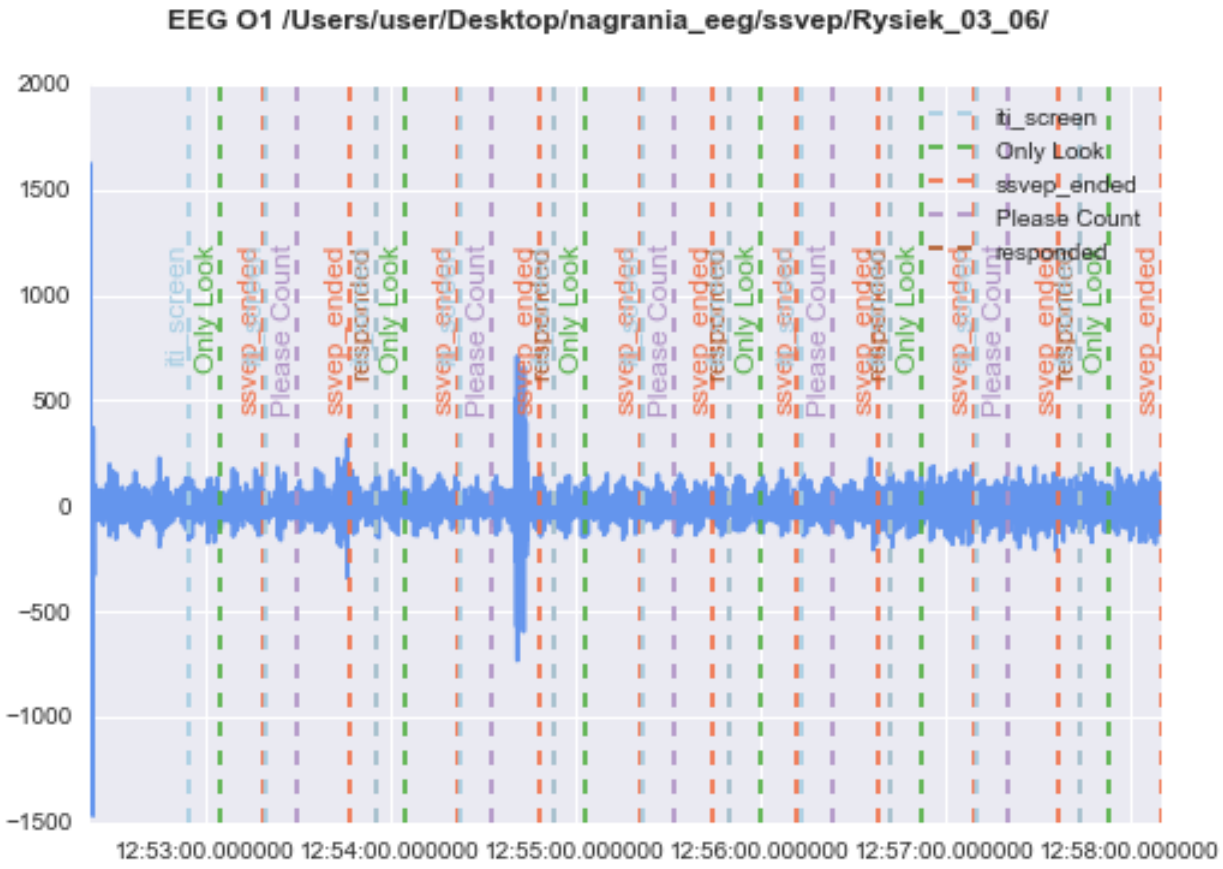

/Users/user/Desktop/nagrania_eeg/ssvep/Karen_14_06_16/
0.5976
0.498
0.3984
0.2988
0.1992
0.0996
period Power Density
5
10
15
frequency
Only Look
Please Count

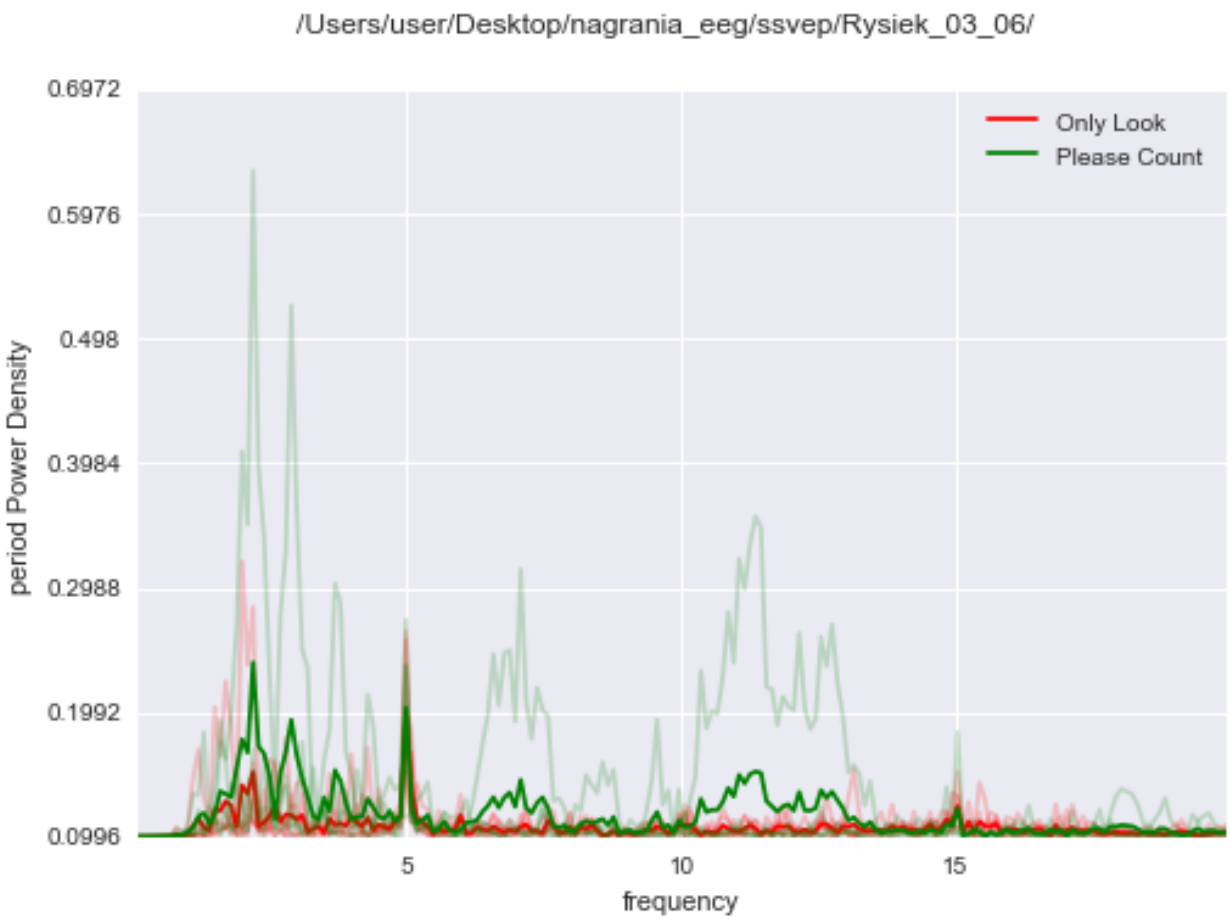

EEG O1 /Users/user/Desktop/nagrania_eeg/ssvep/Agnieszka_03_06/
1000
800
600
400
200
0
−200
−400
−600
iti_screen
Only Look
ssvep_ended
Please Count
responded
14:25:00.000000
14:26:00.000000
14:27:00.000000
14:28:00.000000
14:29:00.000000
14:30:00.000000

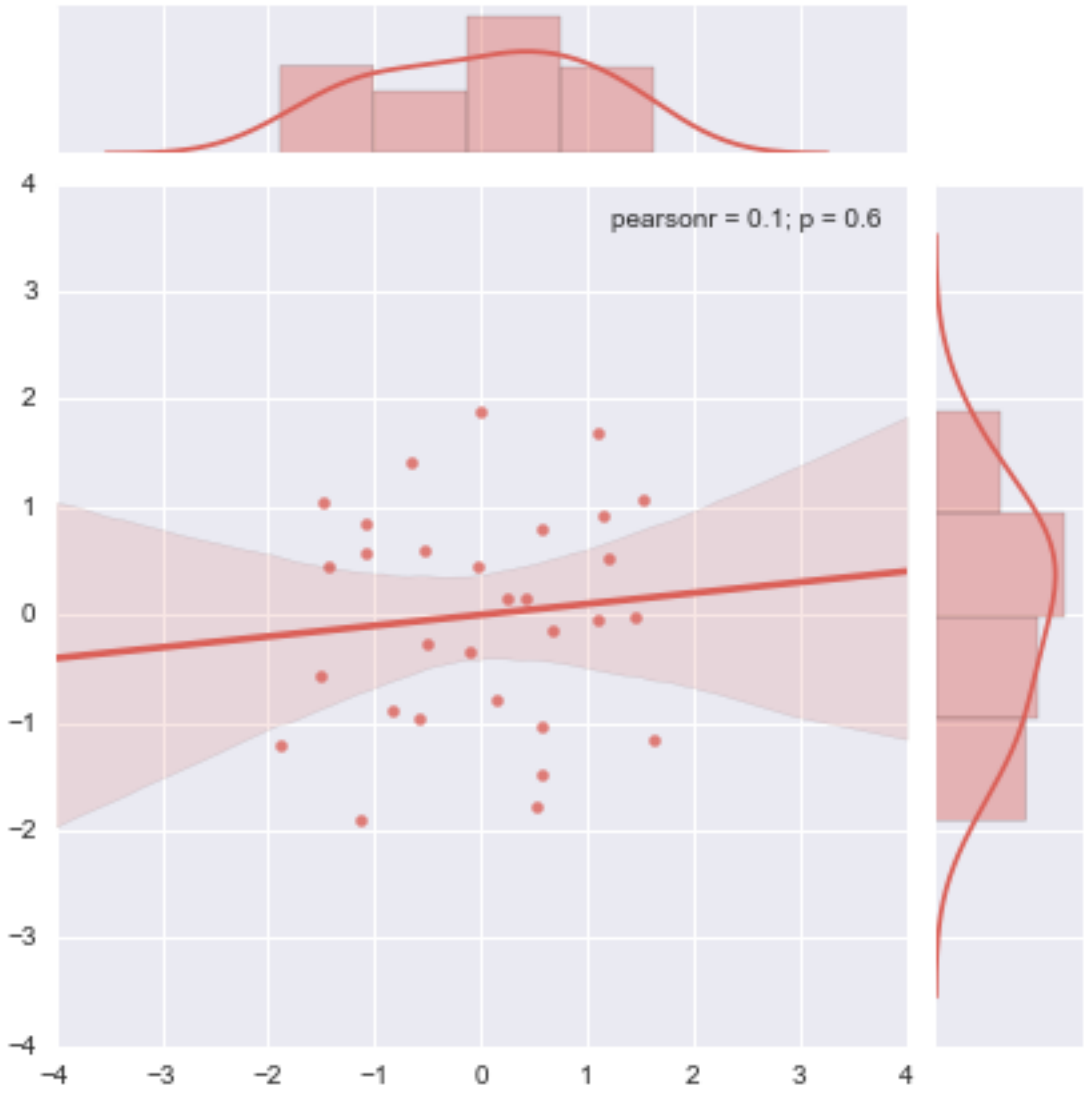

/Users/user/Desktop/nagrania_eeg/ssvep/Agnieszka_03_06/
0.6972
0.5976
0.498
0.3984
0.2988
0.1992
0.0996
period Power Density
5
10
15
frequency
Only Look
Please Count


EEG O1 /Users/user/Desktop/nagrania_eeg/ssvep/Kuba_14_06_16/
3000
2500
2000
1500
1000
500
0
−500
−1000
−1500
13:52:00.000000
13:53:00.000000
13:54:00.000000
13:55:00.000000
13:56:00.000000
13:57:00.000000
iti_screen
Only Look
ssvep_ended
Please Count
responded


/Users/user/Desktop/nagrania_eeg/ssvep/Kuba_14_06_16/
Only Look
Please Count
period Power Density
0.996
0.8964
0.7968
0.6972
0.5976
0.498
0.3984
0.2988
0.1992
0.0996
5
10
15
frequency


EEG O1 /Users/user/Desktop/nagrania_eeg/ssvep/Rysiek_03_06/
2000
1500
1000
500
0
−500
−1000
−1500
iti_screen
Only Look
ssvep_ended
Please Count
responded
12:53:00.000000
12:54:00.000000
12:55:00.000000
12:56:00.000000
12:57:00.000000
12:58:00.000000

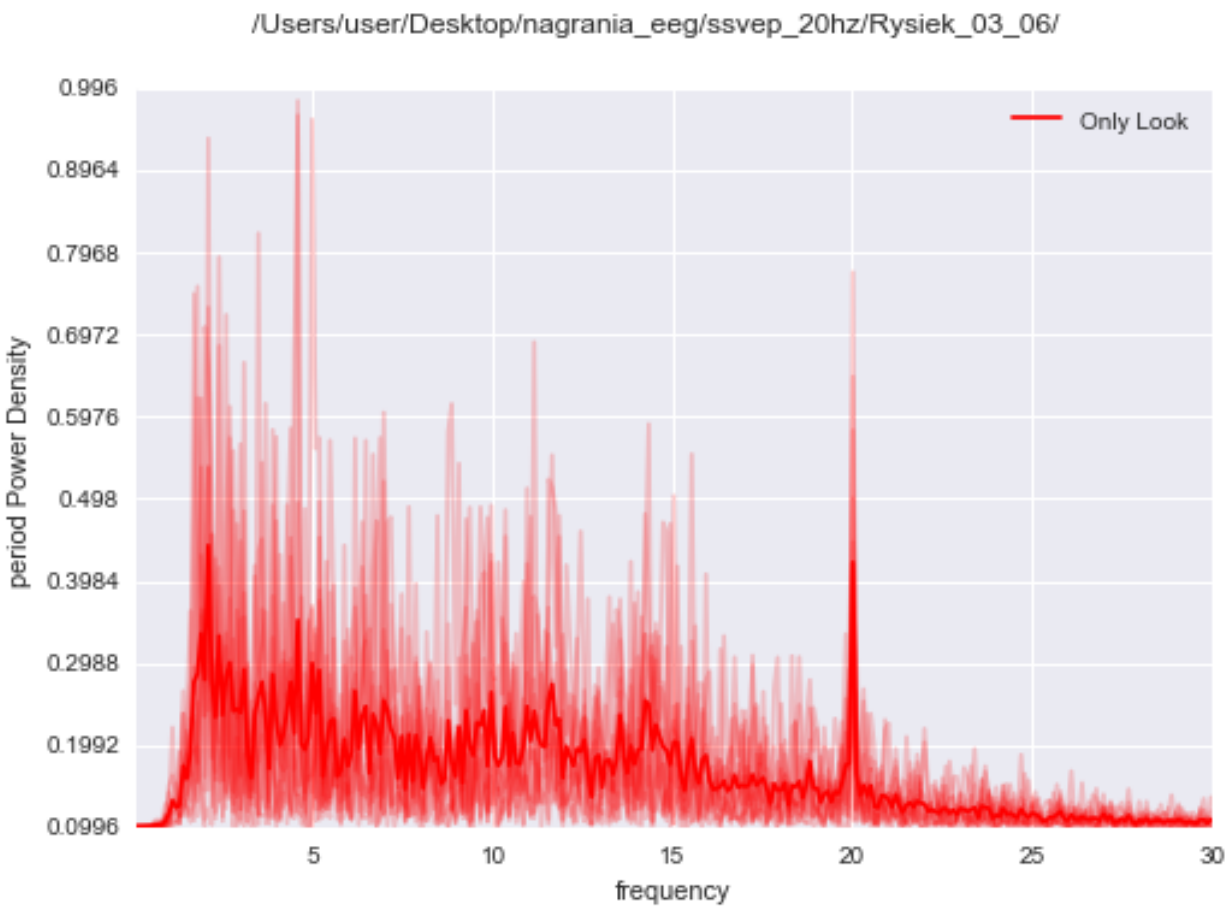

/Users/user/Desktop/nagrania_eeg/ssvep/Rysiek_03_06/
0.6972
0.5976
0.498
0.3984
0.2988
0.1992
0.0996
period Power Density
5
10
15
frequency
Only Look
Please Count




```
plot_fast_ssvep()
```

```
Channels:
EEG F3
EEG F4
EEG C3
EEG C4
EEG P3
EEG P4
EEG O1
EEG O2
EEG A2
EEG Cz
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep_20hz/Agnieszka_03_06/
Channels:
EEG F3
EEG F4
EEG C3
```

```
EEG C4
EEG P3
EEG P4
EEG O1
EEG O2
EEG A2
EEG Cz
(497.971446705165,)
/Users/user/Desktop/nagrania_eeg/ssvep_20hz/Rysiek_03_06/
```


/Users/user/Desktop/nagrania_eeg/ssvep_20hz/Rysiek_03_06/
Only Look
period Power Density
0.996
0.8964
0.7968
0.6972
0.5976
0.498
0.3984
0.2988
0.1992
0.0996
5
10
15
20
25
30
frequency

CHAPTER 5

Installation

You can install using pip:

```
>>> sudo pip install pyseries
```

p

O

P

T