## Annex 3. Normalization of the measures that showed significant correlations and counting their means per session

```
"""
Author: Daria Alexander

We normalize the measures that showed good correlations and compute means of those measures per
session
"""

import pandas as pd
import numpy as np

"""
for the mean length of p-bursts (threshold 1000 ms) we 1) normalize the data 2) divide the data
into 3 sessions
"""

mlpb = pd.read_csv("C:/Users/Daria Alexander/PycharmProjects/fluency-an-
fin/norm/mlpb1000_corr.csv", encoding='utf-8', sep=';')

# normalize the data for the mean length of p-bursts
mlpb['norm'] = (mlpb['mean_l_p_bursts'] -
mlpb['mean_l_p_bursts'].min())/(mlpb['mean_l_p_bursts'].max()-mlpb['mean_l_p_bursts'].min())

# extract the data of the first session
mlpb_s1 = mlpb[mlpb['conv_init_time'].str.contains("2018-09-21") |
mlpb['conv_init_time'].str.contains("2018-11-06")|\
             mlpb['conv_init_time'].str.contains("2018-10-15")|
mlpb['conv_init_time'].str.contains("2018-10-16 06")| \
             mlpb['conv_init_time'].str.contains("2018-10-16 07")]

# compute the mean for the first session
ml_s1 = mlpb_s1['norm'].mean()
print("session1_mlpb",ml_s1)


# extract the data of the second session
mlpb_s2 = mlpb[mlpb['conv_init_time'].str.contains("2018-09-24") |\
             mlpb['conv_init_time'].str.contains("2018-09-
25")|mlpb['conv_init_time'].str.contains("2018-11-05")\
              | mlpb['conv_init_time'].str.contains("2018-11-07")|
mlpb['conv_init_time'].str.contains("2018-10-16 09")]

# compute the mean for the second session
ml_s2 = mlpb_s2['norm'].mean()
print("session2_mlpb",ml_s2)


# extract the data of the third session
mlpb_s3 = mlpb[mlpb['conv_init_time'].str.contains("2018-09-27") \
              | mlpb['conv_init_time'].str.contains("2018-09-26")|
mlpb['conv_init_time'].str.contains("2018-10-23")|\
             mlpb['conv_init_time'].str.contains("2018-11-
08")|mlpb['conv_init_time'].str.contains("2018-11-09")|\
             mlpb['conv_init_time'].str.contains("2018-11-12")]

# compute the mean of the third session
ml_s3 = mlpb_s3['norm'].mean()
print("session3_mlpb",ml_s3)

"""
for the mean duration of p-bursts (threshold 300 ms) we 1) normalize the data 2) divide the data
into 3 sessions
"""


mdpb = pd.read_csv("C:/Users/Daria Alexander/PycharmProjects/fluency-an-
fin/norm/mdpb300_corr.csv", encoding='utf-8', sep=';')

#normalize the data for the mean duration of p-bursts
mdpb['norm'] = (mdpb['mean_p_bursts'] -
```

```
mdpb['mean_p_bursts'].min())/(mdpb['mean_p_bursts'].max()-mdpb['mean_p_bursts'].min())

#extract the data of the first session
mdpb_s1 = mdpb[mdpb['conv_init_time'].str.contains("2018-09-21") |\
            mdpb['conv_init_time'].str.contains("2018-10-15")|
mdpb['conv_init_time'].str.contains("2018-10-16 06")|\
            mdpb['conv_init_time'].str.contains("2018-11-06")|
            mdpb['conv_init_time'].str.contains("2018-10-16 07")]

# compute the mean of the first session
md_s1 = mdpb_s1['norm'].mean()
print("session1_mdpb",md_s1)

# extract the data of the second session
mdpb_s2 = mdpb[mdpb['conv_init_time'].str.contains("2018-09-24") |\
            mdpb['conv_init_time'].str.contains("2018-09-25")|\
            mdpb['conv_init_time'].str.contains("2018-11-05")|
mdpb['conv_init_time'].str.contains("2018-11-07")|\
            mdpb['conv_init_time'].str.contains("2018-10-16 09")]

# compute the mean of the second session
md_s2 = mdpb_s2['norm'].mean()
print("session2_mdpb",md_s2)

# extract the data of the third session
mdpb_s3 = mdpb[mdpb['conv_init_time'].str.contains("2018-09-27") |\
            mdpb['conv_init_time'].str.contains("2018-09-26")|\
            mdpb['conv_init_time'].str.contains("2018-10-
23")|mdpb['conv_init_time'].str.contains("2018-11-08")|\
            mdpb['conv_init_time'].str.contains("2018-11-09")|\
            mdpb['conv_init_time'].str.contains("2018-11-12")]

# compute the mean of the third session
md_s3 = mdpb_s3['norm'].mean()
print("session3_mdpb",md_s3)
```