check_behavioral_data

September 6, 2019

1 Imports

[1]:
```
import pandas as pd; import seaborn as sb; import warnings; import scipy; import␣
 ↪re;
import os; from analysis_helpers import *; import itertools; from scipy import␣
 ↪stats
import random; import pandas as pd; import numpy as np; from sklearn import␣
 ↪datasets, linear_model;
from sklearn.linear_model import LinearRegression; import statsmodels.api as sm
from scipy import stats; from itertools import groupby; from operator import␣
 ↪itemgetter
warnings.filterwarnings('ignore')
%matplotlib inline
```

2 Check A: loading behavioral data

[2]:
```
# for each experiment's data directory
for data_dir in ['../sustained_attention_experiment/data', '../
 ↪variable_attention_experiment/data']:

    sub_list = []

    # for each participant directory in the data directory
    for sub_dir in os.listdir(data_dir):

        # aggregate all the data from the participant into a dataframe, and␣
 ↪append to a list
        sub_list.append(sum_pd(data_dir + '/' + sub_dir))

    print(f'We have loaded data from {len(sub_list)} unique subjects')

    ################################################

    # Concatenate the list into single dataframe
```

```
    concatenated = pd.concat(sub_list)

    # obtain the number of unique runs for each participant for each trial type␣
↪(Presentation and Memory)
    unique_runs = concatenated.groupby(['Subject','Trial Type'])['Run'].nunique()

    print()
    print('Below, we can see the number of unique runs loaded for each subject,␣
↪for each trial type.')
    print()
    print('The set of all numbers of runs from all participants contains '
          + str(unique_runs.nunique()) + ' unique value: '+str(unique_runs.
↪unique()))
    print()
    print(str(unique_runs))
```

```
We have loaded data from 30 unique subjects

Below, we can see the number of unique runs loaded for each subject, for each
trial type.

The set of all numbers of runs from all participants contains 1 unique value:
[8]

Subject  Trial Type
0        Memory          8
         Presentation    8
2        Memory          8
         Presentation    8
6        Memory          8
         Presentation    8
7        Memory          8
         Presentation    8
8        Memory          8
         Presentation    8
9        Memory          8
         Presentation    8
10       Memory          8
         Presentation    8
11       Memory          8
         Presentation    8
12       Memory          8
         Presentation    8
13       Memory          8
         Presentation    8
14       Memory          8
         Presentation    8
```

```
15       Memory          8
         Presentation    8
16       Memory          8
         Presentation    8
17       Memory          8
         Presentation    8
18       Memory          8
         Presentation    8
19       Memory          8
         Presentation    8
20       Memory          8
         Presentation    8
21       Memory          8
         Presentation    8
22       Memory          8
         Presentation    8
23       Memory          8
         Presentation    8
24       Memory          8
         Presentation    8
25       Memory          8
         Presentation    8
26       Memory          8
         Presentation    8
27       Memory          8
         Presentation    8
30       Memory          8
         Presentation    8
31       Memory          8
         Presentation    8
32       Memory          8
         Presentation    8
33       Memory          8
         Presentation    8
34       Memory          8
         Presentation    8
36       Memory          8
         Presentation    8
Name: Run, dtype: int64
We have loaded data from 30 unique subjects

Below, we can see the number of unique runs loaded for each subject, for each
trial type.

The set of all numbers of runs from all participants contains 1 unique value:
[8]

Subject  Trial Type
```

```
0       Memory          8
        Presentation    8
1       Memory          8
        Presentation    8
2       Memory          8
        Presentation    8
3       Memory          8
        Presentation    8
4       Memory          8
        Presentation    8
5       Memory          8
        Presentation    8
7       Memory          8
        Presentation    8
8       Memory          8
        Presentation    8
10      Memory          8
        Presentation    8
12      Memory          8
        Presentation    8
13      Memory          8
        Presentation    8
14      Memory          8
        Presentation    8
15      Memory          8
        Presentation    8
16      Memory          8
        Presentation    8
17      Memory          8
        Presentation    8
18      Memory          8
        Presentation    8
19      Memory          8
        Presentation    8
20      Memory          8
        Presentation    8
21      Memory          8
        Presentation    8
22      Memory          8
        Presentation    8
23      Memory          8
        Presentation    8
24      Memory          8
        Presentation    8
25      Memory          8
        Presentation    8
26      Memory          8
        Presentation    8
```

```
27       Memory          8
         Presentation    8
28       Memory          8
         Presentation    8
29       Memory          8
         Presentation    8
33       Memory          8
         Presentation    8
34       Memory          8
         Presentation    8
327      Memory          8
         Presentation    8
Name: Run, dtype: int64
```

3 Check B1: check attention level assignments

[3]:
```
# load in labeled data
exp1 = pd.read_csv('../parsed_data/behavioral_data_sustained.csv')
exp2 = pd.read_csv('../parsed_data/behavioral_data_variable.csv')


# label rows by trial number
# see the check in issue #83, where we confirm that all rows are in the temporal␣
 ↪order from the experiment
# (early trials at the top, late trials at the bottom)
for exp in [exp1, exp2]:

    # Number all presentation and memory trials
    exp.loc[exp['Trial Type']=='Memory','Trial'] = list(range(0,40))*30*8
    exp.loc[exp['Trial Type']=='Presentation','Trial'] = list(range(0,10))*30*8
```

3.0.1 Check B1a: Number of images at each attention level, in each memory run

[4]:
```
# for each experiment
for exp in [exp1, exp2]:

    # number of unique images at each attention level (prev-seen images and␣
 ↪Novel images)
    unique_seen  = exp[exp['Attention Level']!='Novel'].
 ↪groupby(['Subject','Run', 'Attention Level'])['Trial'].nunique()
    unique_novel = exp[exp['Attention Level']=='Novel'].
 ↪groupby(['Subject','Run', 'Attention Level'])['Trial'].nunique()
```

```
   for name,data in zip(['novel','previously seen'],[unique_novel,␣
↪unique_seen]):

       print('The set of the numbers of ' + name + ' images from each attention␣
↪level, displayed to each participant, in each run, contains '+
             str(data.nunique())+' unique value : '+str(data.unique()))
       print()
       print(data)
       print()
```

```
The set of the numbers of novel images from each attention level, displayed to
each participant, in each run, contains 1 unique value : [20]

Subject  Run  Attention Level
0        0    Novel              20
         1    Novel              20
         2    Novel              20
         3    Novel              20
         4    Novel              20
                                  ..
36       3    Novel              20
         4    Novel              20
         5    Novel              20
         6    Novel              20
         7    Novel              20
Name: Trial, Length: 240, dtype: int64

The set of the numbers of previously seen images from each attention level,
displayed to each participant, in each run, contains 1 unique value : [5]

Subject  Run  Attention Level
0        0    Category           5
              Full               5
              None               5
              Side               5
         1    Category           5
                                 ..
36       6    Side               5
         7    Category           5
              Full               5
              None               5
              Side               5
Name: Trial, Length: 960, dtype: int64

The set of the numbers of novel images from each attention level, displayed to
each participant, in each run, contains 1 unique value : [20]
```

```
Subject  Run  Attention Level
0        0    Novel              20
         1    Novel              20
         2    Novel              20
         3    Novel              20
         4    Novel              20
                                  ..
327      3    Novel              20
         4    Novel              20
         5    Novel              20
         6    Novel              20
         7    Novel              20
Name: Trial, Length: 240, dtype: int64

The set of the numbers of previously seen images from each attention level,
displayed to each participant, in each run, contains 1 unique value : [5]

Subject  Run  Attention Level
0        0    Category           5
              Full               5
              None               5
              Side               5
         1    Category           5
                                 ..
327      6    Side               5
         7    Category           5
              Full               5
              None               5
              Side               5
Name: Trial, Length: 960, dtype: int64
```

3.0.2 Check B1b: Novel images equal proportion faces and places

[5]:
```
# Novel Faces and Places
for exp in [exp1, exp2]:

    # number of unique images at each attention level (prev-seen images and␣
↪Novel images)
    unique_novel = exp[exp['Attention Level']=='Novel'].
↪groupby(['Subject','Run', 'Attention Level','Category'])['Trial'].nunique()

    print('The set of the numbers of Novel face images and Novel place images␣
↪displayed to each participant, in each run, contains '+
    str(unique_novel.nunique())+' unique value(s) : '+str(unique_novel.unique()))
```

```
    print()
    print(unique_novel)
    print()
```

```
The set of the numbers of Novel face images and Novel place images displayed to
each participant, in each run, contains 1 unique value(s) : [10]

Subject  Run  Attention Level  Category
0        0    Novel            Face        10
                               Place       10
         1    Novel            Face        10
                               Place       10
         2    Novel            Face        10
                                           ..
36       5    Novel            Place       10
         6    Novel            Face        10
                               Place       10
         7    Novel            Face        10
                               Place       10
Name: Trial, Length: 480, dtype: int64

The set of the numbers of Novel face images and Novel place images displayed to
each participant, in each run, contains 1 unique value(s) : [10]

Subject  Run  Attention Level  Category
0        0    Novel            Face        10
                               Place       10
         1    Novel            Face        10
                               Place       10
         2    Novel            Face        10
                                           ..
327      5    Novel            Place       10
         6    Novel            Face        10
                               Place       10
         7    Novel            Face        10
                               Place       10
Name: Trial, Length: 480, dtype: int64
```

4 Check B2: Randomly select runs from random participants to spot check manually

[6]:
```
# randomly select two participants
# randomly select a run from each participant's data
# output randomly selected runs to html file for manual check
```

[7]:
```
for exp_name,exp,seed in zip(['exp1','exp2'],[exp1,exp2],[1,2]):

    random.seed(seed)
    sub = random.sample(list(exp['Subject'].unique()), 2)
    run = random.sample(list(exp['Run'].unique()), 2)

    for s,r in zip(sub,run):
        exp[(exp['Subject']==s) & (exp['Run']==r)].
↪to_csv(exp_name+'_'+str(s)+'_'+str(r)+'_test.csv')
```

[ ]:

[ ]:

[ ]:

[ ]: