Test jas

January 12, 2021

[1]:
```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from ortho_lib import *
from sklearn import neighbors, datasets
from sklearn.neighbors import KNeighborsClassifier
```

[2]:
```
data_dir = 'transformed_data'
cat = '1'
pat = '1'
exercise = 'AB1'
file= data_dir +'/Category_' + cat + '/' + pat + '/' + exercise + '.txt'

#df = exercise_to_df(file)

#df
```

[3]:
```
def verschil_bepalen(snrL, snrR, df, cat):
    global diff
    df_test_L = df[df['sensor'] == snrL]
    df_test_L.reset_index(drop=True)
    df_test_R = df[df['sensor'] == snrR]
    df_test_R.reset_index(drop=True)
    #diff_x = (df_test_L['x'] - df_test_R['x']).abs().max()
    #diff_y = (df_test_L['y'] - df_test_R['y']).abs().max()
    diff_z = (df_test_L['z'] - df_test_R['z']).abs().mean()
    #diff = diff_x + diff_y + diff_z
    diff=diff_z
    diff_list[cat].append(diff)
```

[4]:
```
dfnieuw = pd.DataFrame(columns = ['cat', 'diff'])
diffs = []
cats = []
dfnieuw
```

```
[4]: Empty DataFrame
     Columns: [cat, diff]
     Index: []
```

```
[5]: diff_list = {'1': [], '2': [], '3':[], '4':[]}
     cats_list = ['1', '2', '3', '4']

     pats1 = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13',␣
      ↪'14', '15', '16', '17', '18', '19', '20','21', '22', '23', '24', '25', '26',␣
      ↪'27', '28', '29', '30']
     pats2 = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13',␣
      ↪'14', '15', '16', '17', '18', '19', '20','21', '22', '23', '24', '25', '26',␣
      ↪'27', '28', '29', '30', '31', '32', '33', '34', '35', '36']
     pats3 = ['3', '4', '5', '6', '7', '8', '9', '11', '12', '13', '14', '15', '16',␣
      ↪'17', '18', '19', '20','21', '22', '23', '24', '25', '26', '27', '28', '29',␣
      ↪'30', '31', '32', '33', '34', '35', '36', '37', '38', '39', '40']
     pats4 = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13',␣
      ↪'14', '21', '22', '23', '24', '25', '26', '27', '28', '29', '30', '31',␣
      ↪'33', '34', '35', '36', '37', '38', '39', '41']



     #groep1
     for pat in pats1:
         data_dir = 'transformed_data'
         cat = '1'
         exercise = 'AB1'
         file= data_dir +'/Category_' + cat + '/' + pat + '/' + exercise + '.txt'

         df1 = exercise_to_df(file)

         verschil_bepalen('5', '8', df1, cat)
         diffs.append(diff)
         cats.append(cat)

     #groep2
     for pat in pats2:
         data_dir = 'transformed_data'
         cat = '2'
         exercise = 'AB1'
         file= data_dir +'/Category_' + cat + '/' + pat + '/' + exercise + '.txt'

         df2 = exercise_to_df(file)

         verschil_bepalen('5', '8', df2, cat)
         diffs.append(diff)
         cats.append(cat)
```

```
#groep3
for pat in pats3:
    data_dir = 'transformed_data'
    cat = '3'
    exercise = 'AB1'
    file= data_dir +'/Category_' + cat + '/' + pat + '/' + exercise + '.txt'

    df3 = exercise_to_df(file)

    verschil_bepalen('5', '8', df3, cat)
    diffs.append(diff)
    cats.append(cat)


# #groep4
# for pat in pats4:
#     data_dir = 'transformed_data'
#     cat = '4'
#     exercise = 'AB1'
#     file= data_dir +'/Category_' + cat + '/' + pat + '/' + exercise + '.txt'

#     df4 = exercise_to_df(file)

#     verschil_bepalen('5', '8', df4, cat)
#     diffs.append(diff)
#     cats.append(cat)
```

[6]:

```
dfnieuw['cat'] = cats
dfnieuw['diff']= diffs
dfnieuw
```

[6]:

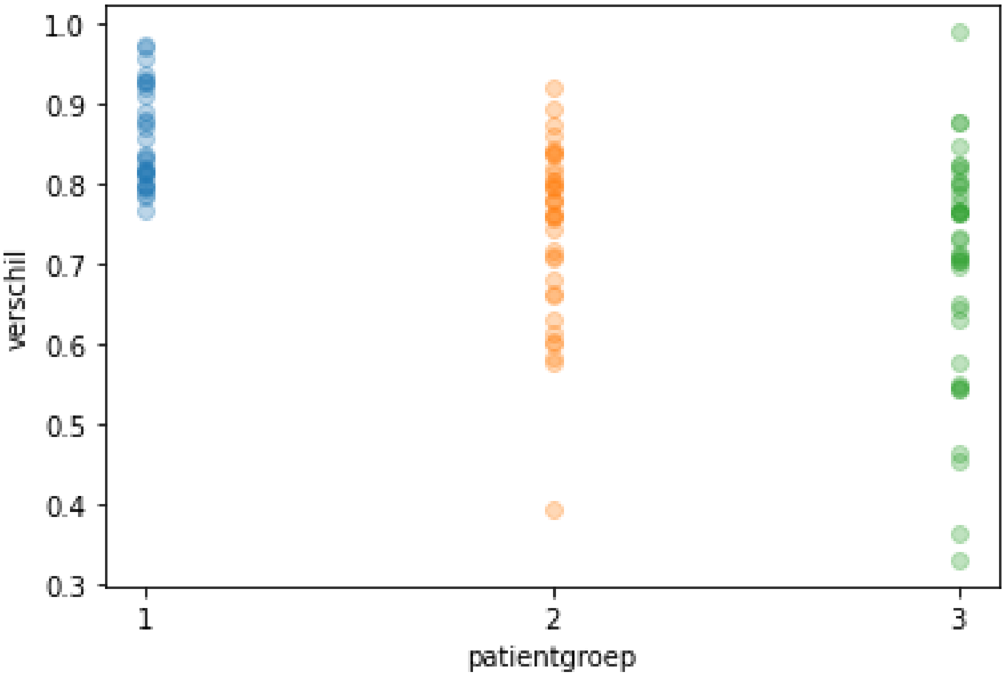
```
    cat      diff
0     1  0.967742
1     1  0.782144
2     1  0.811418
3     1  0.830475
4     1  0.856603
..   ..       …
98    3  0.815629
99    3  0.701190
100   3  0.574788
101   3  0.643605
102   3  0.704218

[103 rows x 2 columns]
```

[7]:

```
for key in diff_list:

    plt.scatter([key]*len(diff_list[key]), diff_list[key], label=key, alpha=0.3)
    plt.ylabel('verschil')
    plt.xlabel('patientgroep')
n_neighbors=5
#X = diff_list.keys().reshape(-1,1)
#y = diff_list.values().reshape(-1,1)
#clf = neighbors.KNeighborsClassifier(n_neighbors, weights='distance')
#clf.fit(X, y)
```



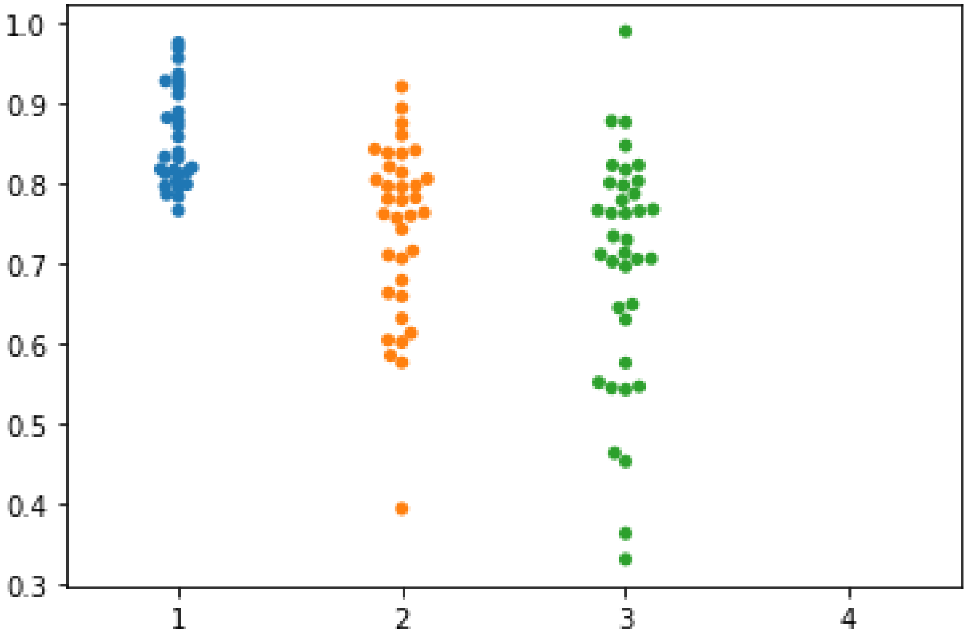
[8]:

```
for key in diff_list:

    sns.swarmplot([key]*len(diff_list[key]), diff_list[key],␣
 ↪order=['1','2','3','4'])
```

```
/opt/jupyterhub/anaconda/lib/python3.6/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
/opt/jupyterhub/anaconda/lib/python3.6/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
```

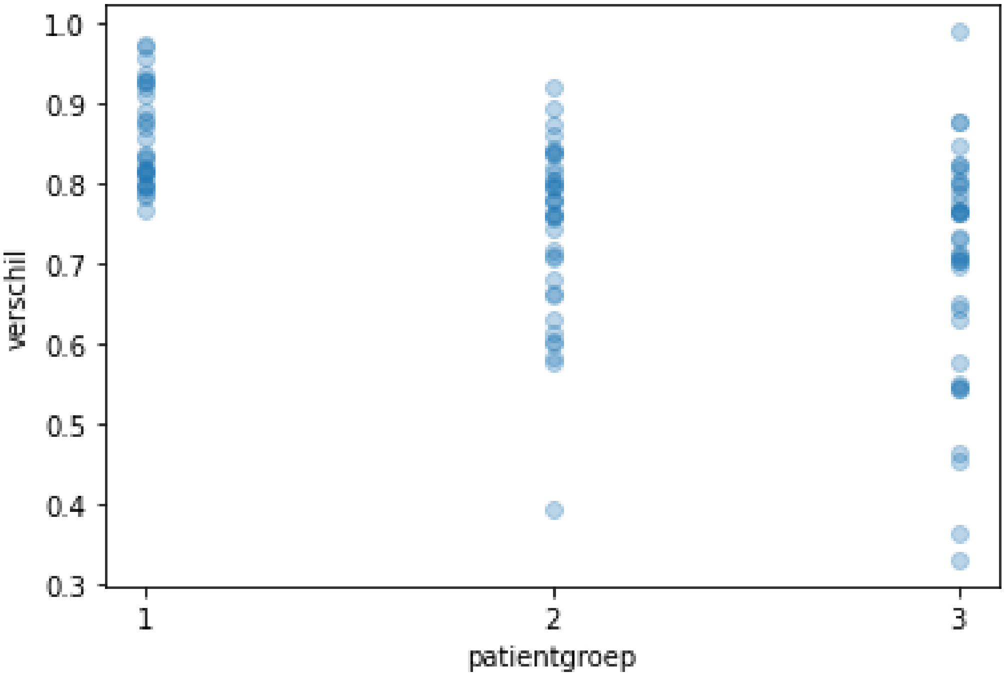
```
arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
/opt/jupyterhub/anaconda/lib/python3.6/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
/opt/jupyterhub/anaconda/lib/python3.6/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
```



[9]:

```
plt.scatter(x=dfnieuw['cat'], y=dfnieuw['diff'], alpha=0.3)
plt.xlabel('patientgroep')
plt.ylabel('verschil')
#n_neighbors=5
#X = dfnieuw['cat'].values.reshape(1,-1)
#y = dfnieuw['diff'].values.reshape(1,-1)
#knn = KNeighborsClassifier(n_neighbors=5)
#knn.fit(X,y)
#clf = neighbors.KNeighborsClassifier(n_neighbors, weights='distance')
#clf.fit(X, y)
```

```
[9]: Text(0, 0.5, 'verschil')
```



```
[ ]:
```