A.DataCheck

November 17, 2020

1 Check The data

Check if the Numpy data is the same as the excel data

[2]:

```
#Import modules:
import numpy as np
import pandas as pd
from tqdm import tqdm
import matplotlib.pyplot as plt
import glob
import seaborn as sns
```

[3]:

```
#define all variables:
    #Paths
loadpath_ex = '/home/18005152/notebooks/zero/Data(xlsx)/'
loadpath_np = '/home/18005152/notebooks/zero/A/'
paths_ex = glob.glob(loadpath_ex+"*.xlsx")
paths_np = glob.glob(loadpath_np+"*.npy")

    #vars:
tol_dif = 1
```

[19]:

```
#functions
def nParse3(n):
    """output a string that is 3 decimals long. Levy en Jefry kunnen uitleg␣
 ↪geven"""
    number = str(n)
    if len(number) == 1:
        number = "00" + number
    elif len(number) == 2:
        number = "0" + number
    elif len(number) == 3:
        number = number
    return str(number)

def check_eq_w_range(Numpy_Sum,Excel_Sum,abs_diff):
    """Check if a value is equal within a certain range."""
    if abs(Numpy_Sum-Excel_Sum) < abs_diff:
        return True
    else:
        return False

def open_file_append_sentence(msg):
    """open the file, append the sentence, and then close the file."""
    with open("DataCheck_info_NEW.txt",'a') as f:
        f.write(str(msg))
        f.close()
```

this is where the program will be ran.

[39]:

```
#Main loop:
for path in tqdm(paths_ex):
    #what is the house number?:
    house_number = path[-8:-5]

    #load the Excel data and get the sheet names:
    ex_df = pd.ExcelFile(loadpath_ex+house_number+'.xlsx', engine="openpyxl")
    sheets = ex_df.sheet_names

    #loop over all the sheets:
    for sheet in sheets:
        #load the excel sheet and the numpy sheet:
        try:
            np_data = np.nan_to_num(np.
 ↪load(loadpath_np+sheet+'_'+house_number+'.npy'),np.nan)
        except:
            open_file_append_sentence(sheet + ',' + house_number + "," +␣
 ↪"Pickle problem,nan\n")
            continue
        ex_data = ex_df.parse(sheet)

        #compute the sum
        np_sum = np_data.sum()
        try:
            ex_sum = ex_data.sum().sum()
        except:
            open_file_append_sentence(sheet + ',' + house_number + "," + "Excel␣
 ↪file NOK,nan\n")
            continue


        #1. check if the table is empty:
        if np_sum == 0 or ex_sum == 0:
            open_file_append_sentence(sheet + ',' + house_number + "," + "Empty␣
↪Table,"+str(abs(np_sum-ex_sum))+"\n")
            continue

        #2. check if the sum of the rows is equal:
        if check_eq_w_range(np_sum,ex_sum,tol_dif):
            open_file_append_sentence(sheet + ',' + house_number + "," + "Is␣
↪okay,"+str(abs(np_sum-ex_sum))+"\n")
            continue

        #3. checkif the the rows are equivalent:
        open_file_append_sentence(sheet + ',' + house_number + "," + "An␣
↪mistake(s),"+str(abs(np_sum-ex_sum))+"\n")
```

```
  0%|          | 0/120 [00:00<?, ?it/s]
  1%|          | 1/120 [01:19<2:37:11, 79.25s/it]
  2%|         | 2/120 [02:38<2:35:40, 79.16s/it]
  2%|         | 3/120 [03:54<2:32:38, 78.28s/it]
  3%|         | 4/120 [05:09<2:29:40, 77.42s/it]
  4%|         | 5/120 [05:45<2:04:22, 64.89s/it]
  5%|         | 6/120 [06:53<2:04:51, 65.72s/it]
  6%|         | 7/120 [08:04<2:06:48, 67.33s/it]
  7%|         | 8/120 [09:12<2:05:57, 67.48s/it]
  8%|         | 9/120 [10:30<2:11:03, 70.84s/it]
  8%|         | 10/120 [11:49<2:14:09, 73.17s/it]
  9%|         | 11/120 [13:07<2:15:44, 74.72s/it]
 10%|         | 12/120 [14:26<2:16:39, 75.93s/it]
 11%|         | 13/120 [15:45<2:17:02, 76.84s/it]
 12%|        | 14/120 [17:04<2:16:57, 77.52s/it]
 12%|        | 15/120 [18:14<2:11:38, 75.22s/it]
 13%|        | 16/120 [19:25<2:08:03, 73.88s/it]
 14%|        | 17/120 [20:44<2:09:42, 75.55s/it]
 15%|        | 18/120 [21:41<1:58:42, 69.83s/it]
 16%|        | 19/120 [22:53<1:58:55, 70.65s/it]
 17%|        | 20/120 [24:12<2:01:51, 73.12s/it]
 18%|        | 21/120 [25:21<1:58:48, 72.01s/it]
 18%|        | 22/120 [26:37<1:59:14, 73.01s/it]
 19%|        | 23/120 [27:56<2:01:02, 74.87s/it]
 20%|        | 24/120 [29:19<2:03:37, 77.27s/it]
 21%|        | 25/120 [30:41<2:04:38, 78.72s/it]
 22%|       | 26/120 [32:05<2:05:36, 80.18s/it]
 22%|       | 27/120 [33:24<2:04:10, 80.11s/it]
 23%|       | 28/120 [34:19<1:51:05, 72.45s/it]
 24%|       | 29/120 [35:25<1:46:57, 70.53s/it]
 25%|       | 30/120 [36:15<1:36:30, 64.34s/it]
 26%|       | 31/120 [37:33<1:41:27, 68.40s/it]
```

```
27%|          | 32/120 [38:51<1:44:36, 71.33s/it]
28%|          | 33/120 [40:10<1:46:42, 73.60s/it]
28%|          | 34/120 [41:06<1:37:51, 68.27s/it]
29%|          | 35/120 [42:24<1:40:46, 71.14s/it]
30%|          | 36/120 [43:43<1:42:54, 73.51s/it]
31%|          | 37/120 [44:55<1:41:23, 73.30s/it]
32%|         | 38/120 [46:14<1:42:11, 74.77s/it]
32%|         | 39/120 [47:25<1:39:44, 73.88s/it]
33%|         | 40/120 [48:44<1:40:13, 75.17s/it]
34%|         | 41/120 [49:57<1:38:12, 74.59s/it]
35%|         | 42/120 [50:58<1:31:52, 70.67s/it]
36%|         | 43/120 [52:17<1:33:39, 72.98s/it]
37%|         | 44/120 [53:34<1:34:13, 74.39s/it]
38%|         | 45/120 [54:35<1:27:55, 70.34s/it]
38%|         | 46/120 [55:44<1:26:15, 69.93s/it]
39%|         | 47/120 [56:56<1:25:49, 70.54s/it]
40%|         | 48/120 [58:00<1:22:03, 68.38s/it]
41%|         | 49/120 [59:16<1:23:49, 70.84s/it]
42%|        | 50/120 [1:00:30<1:23:33, 71.62s/it]
42%|        | 51/120 [1:01:51<1:25:42, 74.53s/it]
43%|        | 52/120 [1:03:03<1:23:44, 73.89s/it]
44%|        | 53/120 [1:04:22<1:24:06, 75.33s/it]
45%|        | 54/120 [1:05:37<1:22:41, 75.18s/it]
46%|        | 55/120 [1:06:55<1:22:14, 75.92s/it]
47%|        | 56/120 [1:08:11<1:21:18, 76.23s/it]
48%|        | 57/120 [1:10:13<1:34:11, 89.71s/it]
48%|        | 58/120 [1:11:31<1:29:11, 86.32s/it]
49%|        | 59/120 [1:12:49<1:25:07, 83.73s/it]
50%|        | 60/120 [1:14:08<1:22:20, 82.34s/it]
51%|        | 61/120 [1:15:28<1:20:22, 81.74s/it]
52%|       | 62/120 [1:16:30<1:13:16, 75.81s/it]
52%|       | 63/120 [1:17:48<1:12:39, 76.49s/it]
53%|       | 64/120 [1:19:05<1:11:36, 76.71s/it]
54%|       | 65/120 [1:20:19<1:09:23, 75.70s/it]
55%|       | 66/120 [1:21:38<1:09:02, 76.72s/it]
56%|       | 67/120 [1:22:45<1:05:19, 73.95s/it]
57%|       | 68/120 [1:23:53<1:02:26, 72.05s/it]
57%|       | 69/120 [1:25:11<1:02:48, 73.90s/it]
58%|       | 70/120 [1:26:22<1:00:51, 73.02s/it]
59%|       | 71/120 [1:27:30<58:19, 71.41s/it]
60%|       | 72/120 [1:28:46<58:15, 72.82s/it]
61%|       | 73/120 [1:30:02<57:45, 73.73s/it]
62%|      | 74/120 [1:31:21<57:48, 75.39s/it]
62%|      | 75/120 [1:32:17<52:08, 69.53s/it]
63%|      | 76/120 [1:33:35<52:55, 72.18s/it]
64%|      | 77/120 [1:34:54<53:04, 74.05s/it]
65%|      | 78/120 [1:36:03<50:46, 72.53s/it]
66%|      | 79/120 [1:37:21<50:44, 74.25s/it]
```

```
 67%|       | 80/120 [1:38:40<50:22, 75.57s/it]
 68%|       | 81/120 [1:39:49<47:53, 73.67s/it]
 68%|       | 82/120 [1:41:10<48:00, 75.80s/it]
 69%|       | 83/120 [1:42:25<46:38, 75.65s/it]
 70%|       | 84/120 [1:43:41<45:24, 75.69s/it]
 71%|       | 85/120 [1:44:59<44:34, 76.40s/it]
 72%|       | 86/120 [1:46:17<43:38, 77.02s/it]
 72%|       | 87/120 [1:47:34<42:24, 77.09s/it]
 73%|       | 88/120 [1:48:50<40:51, 76.62s/it]
 74%|       | 89/120 [1:50:01<38:39, 74.83s/it]
 75%|       | 90/120 [1:51:07<36:10, 72.36s/it]
 76%|       | 91/120 [1:52:26<35:53, 74.25s/it]
 77%|       | 92/120 [1:53:43<35:06, 75.22s/it]
 78%|       | 93/120 [1:55:02<34:15, 76.15s/it]
 78%|       | 94/120 [1:56:20<33:14, 76.73s/it]
 79%|       | 95/120 [1:57:14<29:11, 70.06s/it]
 80%|       | 96/120 [1:58:31<28:50, 72.10s/it]
 81%|       | 97/120 [1:59:48<28:14, 73.66s/it]
 82%|      | 98/120 [2:01:07<27:35, 75.26s/it]
 82%|      | 99/120 [2:02:16<25:41, 73.40s/it]
 83%|      | 100/120 [2:03:35<24:56, 74.82s/it]
 84%|      | 101/120 [2:04:53<24:03, 75.98s/it]
 85%|      | 102/120 [2:06:08<22:39, 75.50s/it]
 86%|      | 103/120 [2:07:12<20:26, 72.14s/it]
 87%|      | 104/120 [2:08:51<21:24, 80.29s/it]
 88%|      | 105/120 [2:10:10<19:56, 79.79s/it]
 88%|      | 106/120 [2:11:33<18:49, 80.67s/it]
 89%|      | 107/120 [2:12:55<17:36, 81.26s/it]
 90%|      | 108/120 [2:14:17<16:17, 81.44s/it]
 91%|      | 109/120 [2:15:23<14:04, 76.81s/it]
 92%|      | 110/120 [2:16:45<13:02, 78.26s/it]
 92%|      | 111/120 [2:18:03<11:45, 78.35s/it]
 93%|      | 112/120 [2:19:25<10:35, 79.41s/it]
 94%|      | 113/120 [2:20:39<09:04, 77.79s/it]
 95%|      | 114/120 [2:22:02<07:56, 79.34s/it]
 96%|      | 115/120 [2:23:25<06:42, 80.50s/it]
 97%|      | 116/120 [2:24:48<05:24, 81.05s/it]
 98%|      | 117/120 [2:26:10<04:04, 81.45s/it]
 98%|      | 118/120 [2:27:32<02:43, 81.69s/it]
 99%|      | 119/120 [2:28:54<01:21, 81.75s/it]
100%|      | 120/120 [2:30:10<00:00, 75.08s/it]
```

2 Check Statistics

- checks the program statistics

[8]:

```
st_df = pd.read_csv("DataCheck_info_NEW.txt",header=None)
values = st_df[[2]].value_counts()

%matplotlib notebook
plt.subplots(figsize=(30,7))
st_df = st_df.pivot(0,1,3)#.fillna(0)
ax = sns.heatmap(st_df, vmin=0, vmax=0.25, xticklabels=np.arange(121))
ax.set_yticks(np.arange(28))
ax.set_yticklabels(st_df.index)
ax.set_xticks(np.arange(121))
ax.set_xlim(0,120)
plt.title('HEATMAP SHOWING HOW THE SUM OF THE EXCEL & NUMPY DATASETS DIFFER PER␣
 ↪SHEET FOR EVERY HOUSE')
ax.set_ylabel(None)
ax.set_xlabel(None)
plt.savefig('heetmapje_week5_v1.0.png', dpi=1200)
```

```
                2
0          Is okay
1          Is okay
2          Is okay
3          Is okay
4          Is okay
…            …
2343   Empty Table
2344   Empty Table
2345   Empty Table
2346   Empty Table
2347   Empty Table

[2348 rows x 1 columns]

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>
```

3 Check single file

[36]:

```
ex_df = pd.read_excel(loadpath_ex+'058'+'.xlsx', engine="openpyxl",␣
 ↪sheet_name='thermostat')
np_data = np.nan_to_num(np.load(loadpath_np+'thermostat_058.npy'),np.nan)

ex_sum = ex_df.sum().sum()
np_sum = np_data.sum()

print(abs(np_sum-ex_sum))
```

```
0.0
```