

# Format Examples

May 8, 2020

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

## 0.1 Formatting Examples

Below, we show multiple examples of formatting to get our data into an easily interpreted form.

### Example incoming data

```
In [6]: xls = pd.ExcelFile(r'data\\191126P2_ROIAnnotationSummary_200218.xlsx')
df = pd.read_excel(xls, 'Annotation_Summary')
df
```

```
Out [6]:
```

	Flash	ROI	2P	ROI	total	ROI	C1:RBPMS	C2: Syt10	C3:Syt6	\
0	Flash_40	1	50		1		1	0	0	
1	Flash_40	2	6		2		1	0	0	
2	Flash_40	3	4		3		1	0	0	
3	Flash_40	4	53		4		1	0	0	
4	Flash_40	5	54		5		1	0	0	
...	...	...	...		...		...	...	...	
1511	Flash_68	179	NaN		1513		0	0	0	
1512	Flash_68	180	NaN		1514		0	0	0	
1513	Flash_68	181	NaN		1515		0	0	0	
1514	Flash_68	182	NaN		1516		1	0	0	
1515	Flash_68	183	NaN		1517		1	0	0	

  

	C4:CAVIII	C5:MEIS	C7:Parv	...	Satb2	MEIS	CalR	total	\
0	1	0	1	...	NaN	NaN	NaN	GC///C8+/////	
1	0	0	1	...	NaN	NaN	NaN	GC////////	
2	0	0	1	...	NaN	NaN	NaN	GC////////	
3	1	0	1	...	NaN	NaN	NaN	GC///C8+/////	
4	1	0	1	...	NaN	NaN	NaN	GC///C8+/////	
...	...	...	...	...	...	...	...	...	
1511	0	0	0	...	NaN	NaN	NaN	AC////////	
1512	0	0	0	...	NaN	NaN	NaN	AC////////	
1513	0	0	0	...	NaN	NaN	NaN	AC////////	

```

1514      0      0      1 ...   NaN   NaN   NaN   GC/////////
1515      0      0      0 ...   NaN   NaN   NaN   GC/////////

      ROI color  Unnamed: 26      Unnamed: 27 Unnamed: 28 Unnamed: 29 \
0      #ffff66      NaN      NaN      NaN      NaN
1      #ff66d9      NaN      Lookup Table      color      count
2      #ff66d9      NaN  AC//Syt6+////MEIS+//      #ff6666      73
3      #ffff66      NaN      AC////////MEIS+//      #66ff66      108
4      #ffff66      NaN  AC//Syt6+//ChAT+////      #b366ff      81
...      ...      ...      ...      ...      ...
1511     #66b3ff      NaN      NaN      NaN      NaN
1512     #66b3ff      NaN      NaN      NaN      NaN
1513     #66b3ff      NaN      NaN      NaN      NaN
1514     #ff66d9      NaN      NaN      NaN      NaN
1515     #ff66d9      NaN      NaN      NaN      NaN

      Unnamed: 30
0      NaN
1      NaN
2      light red
3      light green
4      purple
...      ...
1511     NaN
1512     NaN
1513     NaN
1514     NaN
1515     NaN

[1516 rows x 31 columns]

```

### Select Columns to Keep (Example 1)

```

In [7]: df = df[['Flash', '2P ROI', 'RBPMS', 'Syt10+', 'Syt6+', 'CAVIII', 'ChAT', 'Satb2', 'MEIS', 'CalR']]
df = df.dropna(axis = 0, subset = ["2P ROI"])
df = df[df['2P ROI'].apply(lambda x: str(x).isdigit())]
df = df.astype({"2P ROI": int})
for col in ['Syt10+', 'Syt6+', 'CAVIII', 'ChAT', 'Satb2', 'MEIS', 'CalR']:
    df[col] = df[col].apply(lambda x: int(not pd.isna(x)))
df

```

```

Out[7]:      Flash  2P ROI  RBPMS  Syt10+  Syt6+  CAVIII  ChAT  Satb2  MEIS  CalR
0    Flash_40      50    GC        0      0        1    0      0      0      0
1    Flash_40       6    GC        0      0        0    0      0      0      0
2    Flash_40       4    GC        0      0        0    0      0      0      0
3    Flash_40      53    GC        0      0        1    0      0      0      0
4    Flash_40      54    GC        0      0        1    0      0      0      0
...      ...      ...      ...      ...      ...      ...      ...      ...      ...

```

1412	Flash_68	17	GC	0	0	1	0	0	0	0
1432	Flash_68	55	GC	0	0	1	0	0	0	0
1433	Flash_68	97	GC	0	0	1	0	0	0	0
1458	Flash_68	63	GC	0	0	1	0	0	0	0
1476	Flash_68	12	AC	0	1	0	0	0	0	0

[607 rows x 10 columns]

### Optional: Look for all unique instances of data points

In [8]: # *OPTIONAL - MOST DATASETS WILL NOT USE THIS*

```
l = list(df.T)
def name_merge(x):
    p = [str(i[l[x]]) for _, i in df.loc[l[x]].to_dict().items()]
    return p[0] + '_wave_' + str(p[1])
name_merge(0)

def uniquer(x):
    return "".join([str(i[l[x]]) for _, i in df.loc[l[x]].to_dict().items()][2:])

d = {}
c = 0
z = []
for i in range(df.shape[0]):
    u = uniquer(i)
    if u not in d.keys():
        d[u] = c
        c += 1
    z.append(d[u])
df.insert(10, "Class", z)

s = []
for i in range(df.shape[0]):
    s.append(name_merge(i))
```

### Combine multiple sheets from the same file into the same df (Example 2)

In [16]: # *FILENAME*

```
xlsx_filename = "data\\191126P2PhysData_withlabels.xlsx"
excel = pd.ExcelFile(xlsx_filename)
```

```
def renamer(sheet, ind):
    l = lambda name: str(ind) + '_' + name
    sheet = sheet.rename(index = l)
    return sheet
```

```
i = 0
```

```
new_sheetnames = ['Flash_40', 'Flash_52', 'Flash_56', 'Flash_58', 'Flash_60', 'Flash_62', 'Flash_64', 'Flash_66', 'Flash_68', 'Flash_70', 'Flash_72', 'Flash_74', 'Flash_76', 'Flash_78', 'Flash_80', 'Flash_82', 'Flash_84', 'Flash_86', 'Flash_88', 'Flash_90', 'Flash_92', 'Flash_94', 'Flash_96', 'Flash_98', 'Flash_100']
```

```

total = renamer(pd.read_excel(xlsx_filename, sheet_name=excel.sheet_names[i], header=0))
for i in range(1, len(excel.sheet_names)):
    print('Working on sheet ' + str(i + 1) + ' of ' + str(len(excel.sheet_names)))
    total = total.append(renamer(pd.read_excel(xlsx_filename, sheet_name=excel.sheet_names[i], header=0)))
print("Sheet combination complete.")
n = total

def getClassByName(name):
    return z[s.index(name)]
n = n[[i in s for i in n.index]]
n_class = []
for name in list(n.index):
    n_class.append(getClassByName(name))
df = n

```

```

Working on sheet 2 of 8
Working on sheet 3 of 8
Working on sheet 4 of 8
Working on sheet 5 of 8
Working on sheet 6 of 8
Working on sheet 7 of 8
Working on sheet 8 of 8
Sheet combination complete.

```

In [15]: df

```

Out[15]:

```

	0	1	2	3	4	5	\
Flash_40_wave_1	-0.282811	0.832223	1.571895	1.511685	1.777528	2.218740	
Flash_40_wave_2	0.571081	0.494429	0.751718	0.580546	0.632446	0.855266	
Flash_40_wave_3	2.096727	3.333588	3.859820	4.164157	4.005630	4.153227	
Flash_40_wave_4	0.281441	0.346466	0.693288	0.877922	0.841007	0.955635	
Flash_40_wave_5	0.578530	1.201865	1.635306	1.709711	1.883825	1.952654	
...	...	...	...	...	...	...	
Flash_46_wave_59	3.514678	4.924971	5.835104	6.257974	6.559562	6.262323	
Flash_46_wave_65	2.334159	2.937608	2.372215	2.007245	1.645733	1.116064	
Flash_46_wave_76	5.914798	7.498043	7.846637	7.547990	7.341963	7.228784	
Flash_46_wave_86	0.569609	1.132857	1.210412	1.188244	1.100053	1.046596	
Flash_46_wave_94	-0.372969	-0.393594	0.042007	-0.065661	-0.007022	-0.026779	
	6	7	8	9	...	190	\
Flash_40_wave_1	2.128655	1.977111	2.087082	2.107779	...	2.047339	
Flash_40_wave_2	0.680370	0.764025	0.453674	0.620698	...	0.516698	
Flash_40_wave_3	4.078456	3.821792	3.855679	3.537391	...	4.618674	
Flash_40_wave_4	1.120212	1.092848	1.156230	1.256585	...	1.031648	
Flash_40_wave_5	1.961918	2.008558	1.999788	2.141422	...	1.150016	
...	...	...	...	...	...	...	
Flash_46_wave_59	6.273296	6.150227	6.417154	6.272162	...	4.959475	

Flash_46_wave_65	0.814676	1.469086	1.106885	1.502549	...	0.952743
Flash_46_wave_76	7.064824	7.246309	7.724882	7.847785	...	5.629679
Flash_46_wave_86	0.846080	0.704067	0.861538	0.743415	...	0.876272
Flash_46_wave_94	-0.182787	-0.428858	-0.207789	-0.179428	...	-0.550326

  

	191	192	193	194	195	196 \
Flash_40_wave_1	2.123366	1.872845	2.199981	2.028838	2.053396	2.005515
Flash_40_wave_2	0.785411	0.556864	0.525525	0.621531	0.685788	0.578357
Flash_40_wave_3	4.476266	4.738743	4.588993	4.546078	4.538284	4.452612
Flash_40_wave_4	0.785741	0.998500	0.974495	0.936173	0.851405	0.943852
Flash_40_wave_5	0.960056	1.104510	0.978815	0.958205	0.997115	1.030643
...	...	...	...	...	...	...
Flash_46_wave_59	5.125802	5.191646	5.029380	5.086942	5.264967	5.195457
Flash_46_wave_65	0.747290	0.905035	0.744348	0.933812	0.672914	0.545547
Flash_46_wave_76	5.542924	5.973587	5.648575	5.370740	5.610396	5.612535
Flash_46_wave_86	1.013606	1.181771	1.193820	0.839227	0.850917	0.906549
Flash_46_wave_94	-0.428076	-0.298865	-0.432138	-0.396407	-0.661898	-0.674112

  

	197	198	199
Flash_40_wave_1	2.070698	1.913456	1.785120
Flash_40_wave_2	0.565160	0.614412	0.469929
Flash_40_wave_3	4.481684	4.693524	4.886218
Flash_40_wave_4	0.935190	0.879609	0.934099
Flash_40_wave_5	1.107826	0.940766	1.002950
...	...	...	...
Flash_46_wave_59	5.340018	5.472021	5.607791
Flash_46_wave_65	0.528410	0.571942	0.361723
Flash_46_wave_76	5.962801	5.818900	5.763261
Flash_46_wave_86	0.899229	1.125085	1.173561
Flash_46_wave_94	-0.378286	-0.261541	-0.623124

[603 rows x 200 columns]

The final format should look like one of the two examples above.

In [ ]: