In [ ]:

```

```

1. S1a Sensor and Activities Information

Input A `dsActivities = pd.read_csv('S1Activities.csv', index_col = None)`
Input B `dsS1Sensors = pd.read_csv('S1sensors.csv', index_col = None, header = None)`

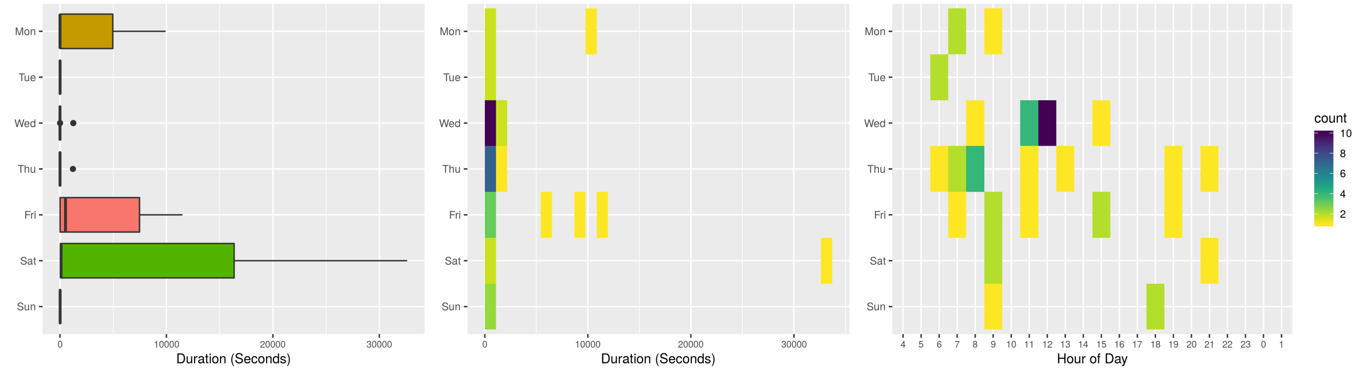
Checking the `S1Activities.csv` dataset. Importing the Sensor, `S1sensors.csv` data. Creating concatenated string values, e.g., Foyer | Light Switch becomes foyer_lightswitch. Creating a boolean feature, `reqElectricity` , to indicate if the activity requires electricity or not. Creating the dictionary `subActKeyWithStringDict` & the dictionary `subActKeyWithEnergyDict` , checking for dupes in the concatenated string values.

- Input = `S1Activities.csv` (checking only)
- Input = `S1sensors.csv`
- Output = `S1sensors_preprocessed.csv`

Output `dsS1Sensors.to_csv('S1Sensors_preprocessed.csv',index = False)`

In [1]:

```
# Invoke notebook code
```



MENTION THIS FROM Huang et al.

2. S1a Activities Data Preprocessing

Input `dsS1 = pd.read_csv('S1activities_data.csv', sep = 'delimiter', header = None)`

Importing `S1Activities_data.csv` , convert df to an array (list?), flatten to a 1D array (list?), chunk the array [5], extract activity, time & date. Merge time and date into datetime elements, determine start and end time.

Example preprocessed output:

Index (a[i])	activity	start	end
0	Bathing	2003-04-01 20:41:35	2003-04-01 21:32:50
1	Toileting	2003-04-01 17:30:36	2003-04-01 17:46:41
2	Toileting	2003-04-01 18:04:43	2003-04-01 18:18:02

- Input = `S1Activities_data.csv`
- Output = `S1Activities_preprocessed.csv`

Output `ds.to_csv('S1Activities_preprocessed.csv',index = False)`

```
In [2]:
```

```
# Invoke notebook code
```

3. S1a SubActivities Preprocessing

Input `dsS1 = pd.read_csv('S1activities_data.csv', sep = 'delimiter', header = None)`

Importing `S1Activities_data.csv` , convert df to an array (list?), flatten to a 1D array (list?), chunk the array [5], extract subActNum, subActivity, time & date. Merge time and date into datetime elements, determine start and end time.

Example preprocessed output:

idx	subActNum	subAct	start	end
0	100	Toilet Flush	2003-04-01 20:51:52	2003-04-01 21:05:20
1	68	Sink faucet - hot	2003-04-01 20:51:58	2003-04-01 20:52:05
2	81	Closet	2003-04-01 20:53:36	2003-04-01 20:53:43

- Input = `S1Activities.csv`
- Output = `S1SubActivities_preprocessed.csv`

Output `ds.to_csv('S1SubActivities_preprocessed.csv',index = False)`

In [ ]:

```
# Invoke notebook code
```

4. S1a SubActivities Added Time Range

Input `ds = pd.read_csv('S1SubActivities_preprocessed.csv', index_col = None)`

Describe

Example preprocessed output:

inx	subActNum	subAct	start	end	actDuration	timeStampList	timeStampArrayList
0	100	Toilet Flush	2003-04-01 20:51:52	2003-04-01 21:05:20	809	DatetimeIndex(['2003-04-01 20:51:58',]	[2003-04-01 20:51:52, 2003-04-01 20:51:53,]
1	68	Sink faucet - hot	2003-04-01 20:51:58	2003-04-01 20:52:05	8	DatetimeIndex(['2003-04-01 20:51:58',]	[2003-04-01 20:51:58, 2003-04-01 20:51:59,]
2	81	Closet	2003-04-01 20:53:36	2003-04-01 20:53:43	8	DatetimeIndex(['2003-04-01 20:53:36',]	[2003-04-01 20:53:36, 2003-04-01 20:53:37,]

Features

- [subActNum]
- [subAct]
- [start]
- [end]
- [actDuration]
- [timeStampList]
- [timeStampArrayList]

> Contains numeric duration value, may be used later to explore temporal relationships between events

- Input = `S1SubActivities_preprocessed.csv`
- Output = `S1SubActivities_timeStampRanges.csv`

Output `ds.to_csv('S1SubActivities_timeStampRanges.csv',index=False)`

In [3]:

```
# Invoke notebook code
```

5. S1a SubActivities Time Range Melt

Input `ds = pd.read_csv('S1SubActivities_timeStampRanges.csv', index_col = None)`

Describe

Example preprocessed output:

idx (start)	subActNum	actDuration	duration
2003-03-27 06:43:40	67	4	2003-03-27 06:43:40
2003-03-27 06:43:40	67	4	2003-03-27 06:43:41
2003-03-27 06:43:40	67	4	2003-03-27 06:43:42
2003-03-27 06:43:40	67	4	2003-03-27 06:43:43
2003-03-27 06:44:06	100	1716	2003-03-27 06:44:06

Features

- idx [start]
- [subActNum]
- [actDuration]
- [duration]

> Quote

- Input = `S1SubActivities_timeStampRanges.csv`
- Output = `S1SubActivities_timeRangeMelt.csv`

Output `ds.to_csv('S1SubActivities_timeRangeMelt.csv',index=False)`

In [ ]:

```
# Invoke notebook code
```

6. S1a SubActivities Time Range Boolean

Input `ds = pd.read_csv('S1SubActivities_timeRangeMelt.csv', index_col = None)`

Example preprocessed output:

ADD DIM

idx (duration)	subActNum_100	subActNum_101	subActNum_104	subActNum_105	subActNum_106
2003-03-27 06:43:40	0	0	0	0	0
2003-03-27 06:43:41	0	0	0	0	0
2003-03-27 06:43:42	0	0	0	0	0
2003-03-27 06:43:43	0	0	0	0	0
2003-03-27 06:44:06	1	0	0	0	0

- Input = 'S1SubActivities_timeRangeMelt.csv'
- Output = 'S1SubActivities_timeRangeBoolean_DuplicateIndex.csv' -- Features [idx(Timestamp), subActNumi, ..., subActNumf]

Example preprocessed output:

ADD DIM

idx (duration)	subActNum_100	subActNum_101	subActNum_104	subActNum_105	subActNum_106
2003-03-27 06:43:40	0	0	0	0	0
2003-03-27 06:43:41	0	0	0	0	0
2003-03-27 06:43:42	0	0	0	0	0
2003-03-27 06:43:43	0	0	0	0	0
2003-03-27 06:44:06	1	0	0	0	0

- Output = 'S1SubActivities_timeRangeBoolean.csv'
- Index collapsed -- Features [idx(Timestamp), subActNumi, ..., subActNumf]

Output `ds.to_csv('S1SubActivities_timeRangeBoolean_DuplicateIndex.csv',index='duration')`
Output `ds.to_csv('S1SubActivities_timeRangeBoolean.csv',index='duration')`

```
In [ ]:
# Invoke notebook code
```

7. S1a SubActivities Collapse into Minutes

Input `ds = pd.read_csv('S1SubActivities_timeRangeBoolean.csv', index_col = 'duration')`
Input pt II `ds.index = pd.to_datetime(ds.index)`

Example preprocessed output:

ADD DIM

idx (duration)	subActNum_100	subActNum_101	subActNum_104	subActNum_105	subActNum_106
2003-03-27 06:43:00	0.0	0.0	0.0	0.0	0.0
2003-03-27 06:44:00	1.0	1.0	0.0	0.0	0.0
2003-03-27 06:45:00	1.0	1.0	0.0	0.0	0.0
2003-03-27 06:46:00	1.0	1.0	0.0	0.0	0.0
2003-03-27 06:47:00	1.0	1.0	0.0	0.0	0.0

Output `ds.to_csv('S1SubActivities_timeRangeBooleanMinutes.csv', index = 'duration')`
Output `ds.to_csv('S1SubActivities_timeRangeBooleanMinutesDropNA.csv', index = 'duration')`

```
In [ ]:
# Invoke notebook code
```

8. S1a SubActivities Remove Duplicate Attributes

Input `ds = pd.read_csv('S1SubActivities_timeRangeBooleanMinutesDropNA.csv', index_col = 'duration')`
Input pt II `ds.index = pd.to_datetime(ds.index)`
Input B `dsS1Sensors = pd.read_csv('S1Sensors_preprocessed.csv', index_col = None)`

Example preprocessed output:

ADD DIM

idx (duration)	bathroom_cabinet	bathroom_door	bathroom_exhaustfan	bathroom_lightswitch	bathroom_medicineca
2003-03-27 06:43:00	1.0	0.0	0.0	0.0	
2003-03-27 06:44:00	1.0	0.0	0.0	1.0	
2003-03-27 06:45:00	0.0	0.0	0.0	1.0	
2003-03-27 06:46:00	0.0	0.0	0.0	1.0	
2003-03-27 06:47:00	0.0	0.0	0.0	1.0	

Example preprocessed output:

ADD DIM

idx (duration)	bathroom_cabinet	bathroom_door	bathroom_exhaustfan	bathroom_lightswitch	bathroom_medicineca
2003-03-27 06:43:00	1.0	0.0	0.0	0.0	
2003-03-27 06:44:00	1.0	0.0	0.0	1.0	
2003-03-27 06:45:00	0.0	0.0	0.0	1.0	
2003-03-27 06:46:00	0.0	0.0	0.0	1.0	
2003-03-27 06:47:00	0.0	0.0	0.0	1.0	

Output `ds.to_csv('S1Act_B_m_NoDupes.csv',index='duration')`

- 2019-08-25 Talk about abstracting away methods using `%run -i 'script.py'`