Author: Yikang Li

As a Data Scientist, I want to explore my data set with visualizations and common statistical characteristics pertaining to time series in particular so that I can understand the nature and shape of the the data and get an idea of what algorithms will perform best.

To Do: Check each feature, extract characteristics. Check Correlation among the features and their extracted characteristics. Visualize time series data and explore trends and characteristics using Plotly and Cufflinks. Transform the data in the spectral domain using fast fourier transform and perform the same EDA to get more information. Study the seasonal decomposition of the time series of each column and check the trend and seasonality and other such characteristics.

Also, perform the same plotting for minute and 15 minute data.

In [1]:
```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import cufflinks as cf
import plotly
import plotly.plotly as py
import plotly.graph_objs as go
from  plotly.offline import plot

from datetime import datetime
import pandas_datareader.data as web
```

In [3]:
```
#load the data
house1 = pd.read_csv("./processed/house_01.csv")
house2 = pd.read_csv("./processed/house_02.csv")
house3 = pd.read_csv("./processed/house_03.csv")
house4 = pd.read_csv("./processed/house_04.csv")
house5 = pd.read_csv("./processed/house_05.csv")
house6 = pd.read_csv("./processed/house_06.csv")
```

House 1:

In [4]:
```
house1.head()
```

Out[4]:

	Unnamed: 0	date	occupancy	Fridge	Dryer	Coffee machine	Kettle	Washing machine	Freezer
0	0	2012-06-01 00:00:00	NaN	49.2516	830.508	NaN	0.0	4.39739	2.23178
1	1	2012-06-01 00:00:01	NaN	49.2516	834.774	NaN	0.0	4.39739	2.23178
2	2	2012-06-01 00:00:02	NaN	49.2516	834.774	NaN	0.0	4.39739	2.23178
3	3	2012-06-01 00:00:03	NaN	51.3899	832.641	NaN	0.0	4.39739	2.23178
4	4	2012-06-01 00:00:04	NaN	49.2516	832.641	NaN	0.0	6.53380	2.23178

The values are missing in columns "occupancy" and "Coffee machine".

Statistical Characteristics:

In [5]:
```
house1.describe()
```

Out[5]:

	Unnamed: 0	occupancy	Fridge	Dryer	Coffee machine	Kettle	
count	2.039040e+07	7.344000e+06	1.987200e+07	1.987200e+07	9.676800e+06	1.745280e+07	1.9
mean	1.019520e+07	7.859397e-01	2.070984e+01	2.246613e+01	4.421196e+00	4.402760e+00	2.3
std	5.886202e+06	4.101689e-01	2.497417e+01	1.298046e+02	7.318194e+01	9.016391e+01	1.8
min	0.000000e+00	0.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.0
25%	5.097600e+06	1.000000e+00	2.205780e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0
50%	1.019520e+07	1.000000e+00	4.344320e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0
75%	1.529280e+07	1.000000e+00	4.925160e+01	0.000000e+00	0.000000e+00	0.000000e+00	2.2
max	2.039040e+07	1.000000e+00	1.012680e+03	1.018200e+03	1.290880e+03	1.902460e+03	2.4

In [6]:
```
house1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20390400 entries, 0 to 20390399
Data columns (total 9 columns):
Unnamed: 0          int64
date                object
occupancy           float64
Fridge              float64
Dryer               float64
Coffee machine      float64
Kettle              float64
Washing machine     float64
Freezer             float64
dtypes: float64(7), int64(1), object(1)
memory usage: 1.4+ GB
```

Correlation:

In [7]:
```
pd.DataFrame.corr(house1)
```

Out[7]:

	Unnamed: 0	occupancy	Fridge	Dryer	Coffee machine	Kettle	Washing machine	Freezer
Unnamed: 0	1.000000	-0.124171	-0.035811	0.018758	-0.005077	0.015791	0.014661	-0.070602
occupancy	-0.124171	1.000000	0.027438	0.065237	0.034402	0.026767	0.039345	0.004867
Fridge	-0.035811	0.027438	1.000000	0.011183	0.006161	0.002722	0.010244	0.018268
Dryer	0.018758	0.065237	0.011183	1.000000	-0.001826	0.002089	0.048201	0.001266
Coffee machine	-0.005077	0.034402	0.006161	-0.001826	1.000000	0.034834	0.003753	0.001518
Kettle	0.015791	0.026767	0.002722	0.002089	0.034834	1.000000	0.004929	-0.005632
Washing machine	0.014661	0.039345	0.010244	0.048201	0.003753	0.004929	1.000000	0.006294
Freezer	-0.070602	0.004867	0.018268	0.001266	0.001518	-0.005632	0.006294	1.000000

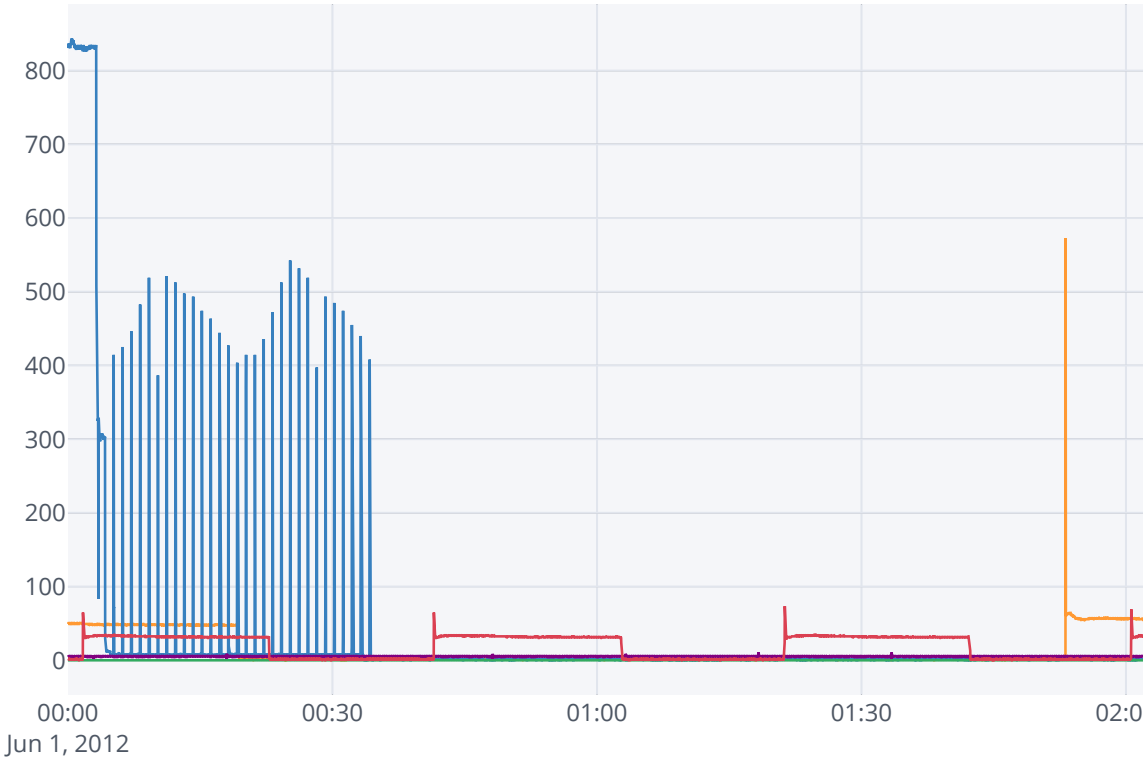
Visualize time series data:

In [8]:
```
plotly.tools.set_credentials_file('liyikang', 'gFn5H8Cy4VbGagtM2IUR')
```

In [9]:
```
house1_dt = house1[['date','Fridge', 'Dryer', 'Kettle', 'Washing machine',
house1_dt.iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for clients without much RAM.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Slow
```

Out[9]:



minute and 15 minute data:

In [10]:
```
group_mins = house1.groupby(np.arange(len(house1))//60)
house1_by_mins = pd.concat((group_mins['date'].first(), group_mins[[c for c
house1_by_mins.head()
```

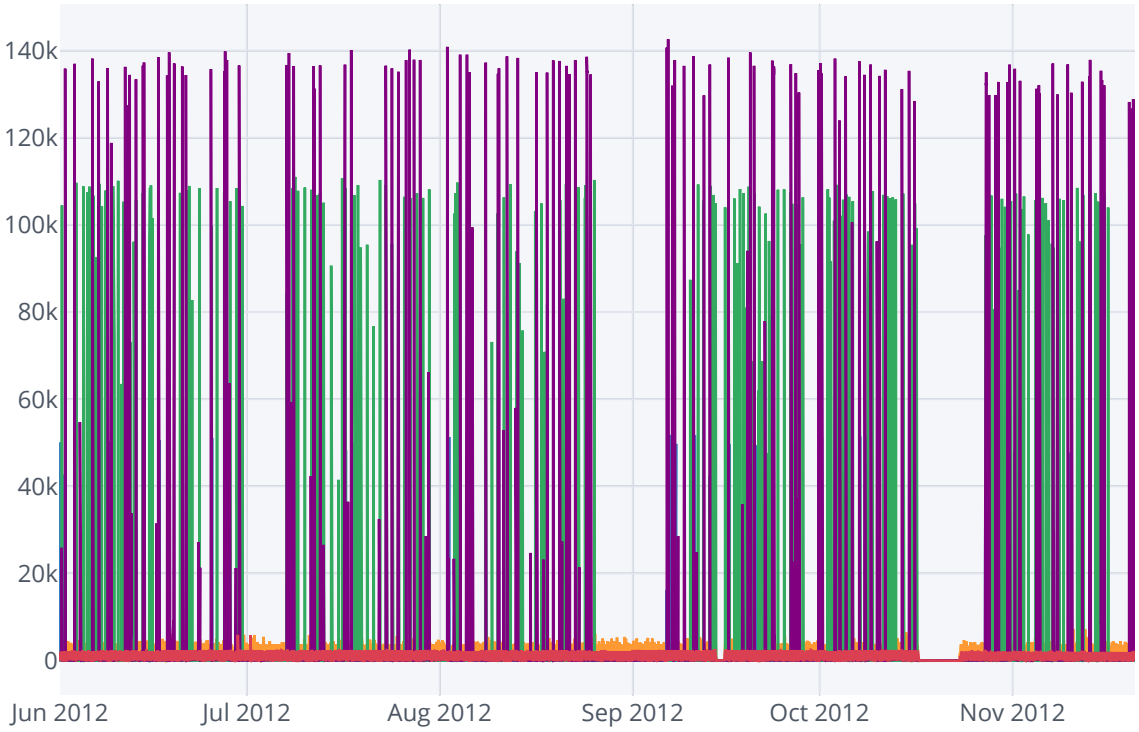
Out[10]:

	date	Unnamed: 0	occupancy	Fridge	Dryer	Coffee machine	Kettle	Washing machine	Freezer
0	2012-06-01 00:00:00	1770	0.0	2976.4790	50107.77500	0.0	0.0	291.61673	102.66188
1	2012-06-01 00:01:00	5370	0.0	2963.6492	49883.80200	0.0	0.0	295.88955	706.65375
2	2012-06-01 00:02:00	8970	0.0	2946.5424	49873.13800	0.0	0.0	291.61672	1937.26730
3	2012-06-01 00:03:00	12570	0.0	2935.8504	24947.11340	0.0	0.0	300.16237	1986.87900
4	2012-06-01 00:04:00	16170	0.0	2918.7432	4482.35334	0.0	0.0	302.29878	1995.50700

In [11]:
```
house1_by_mins[['date','Fridge', 'Dryer', 'Kettle', 'Washing machine', 'Free
```

```
/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/plotly/plotly.
py:230: UserWarning:

Woah there! Look at all those points! Due to browser limitations, the Plo
tly SVG drawing functions have a hard time graphing more than 500k data p
oints for line charts, or 40k points for other types of charts. Here are
some suggestions:
(1) Use the `plotly.graph_objs.Scattergl` trace object to generate a WebG
l graph.
(2) Trying using the image API to return an image instead of a graph URL
(3) Use matplotlib
(4) See if you can create your visualization with fewer data points

If the visualization you're using aggregates points (e.g., box plot, hist
ogram, etc.) you can disregard this warning.


The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[11]:



In [12]:
```
group_15mins = house1.groupby(np.arange(len(house1))//900)
house1_by_15mins = pd.concat((group_15mins['date'].first(), group_15mins[[c
house1_by_15mins.head()
```

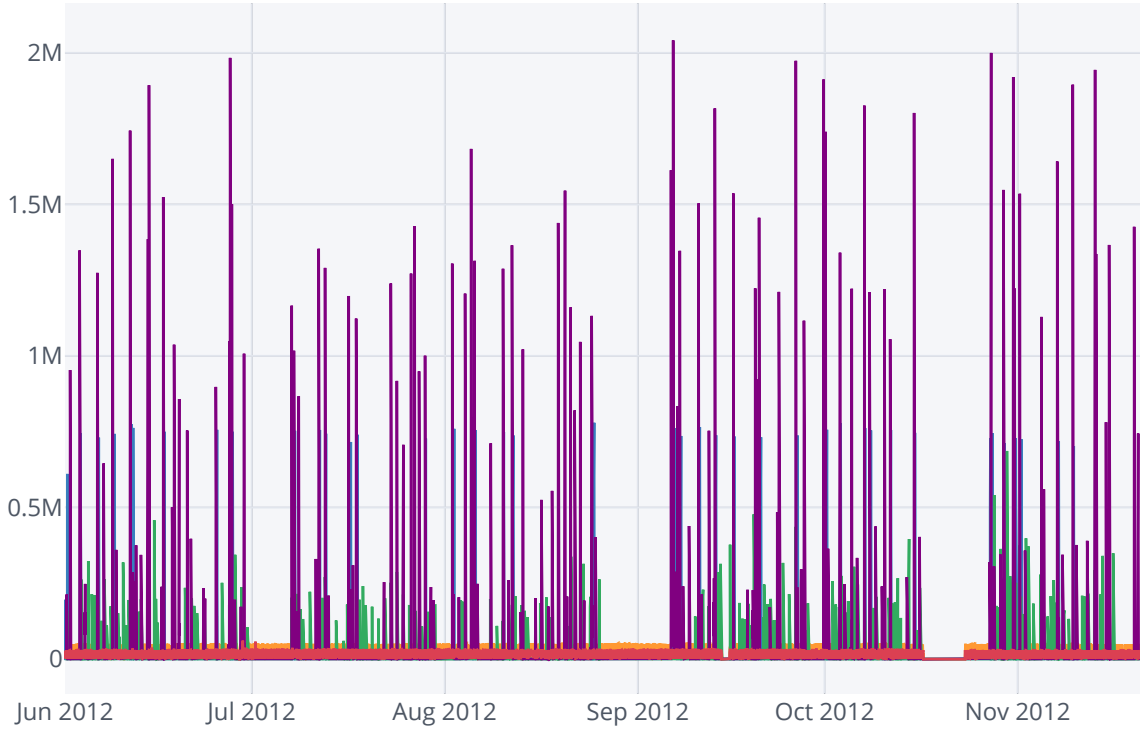
Out[12]:

	date	Unnamed: 0	occupancy	Fridge	Dryer	Coffee machine	Kettle	Washing machine	Fr
0	2012-06-01 00:00:00	404550	0.0	43507.43140	195942.03986	0.0	0.0	4459.70734	26138.
1	2012-06-01 00:15:00	1214550	0.0	12909.26128	24376.89486	0.0	0.0	4436.20682	15496.
2	2012-06-01 00:30:00	2024550	0.0	1166.85762	7796.92002	0.0	0.0	4466.11657	8083.
3	2012-06-01 00:45:00	2834550	0.0	1158.03450	453.45420	0.0	0.0	4425.52477	28938.
4	2012-06-01 01:00:00	3644550	0.0	1135.97670	527.93460	0.0	0.0	4434.07041	6333.

In [14]:
```
house1_by_15mins[['date','Fridge', 'Dryer', 'Kettle', 'Washing machine', 'Fr
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[14]:



House 2:

In [15]:
```
house2.head()
```

Out[15]:

	Unnamed: 0	date	occupancy	Tablet	Dishwasher	Air exhaust	Fridge	Entertainment	Freezer
0	0	2012-06-01 00:00:00	1.0	2.21504	0.0	0.0	2.21458	0.00000	53.6510
1	1	2012-06-01 00:00:01	1.0	4.32930	0.0	0.0	2.21458	2.17127	55.7929
2	2	2012-06-01 00:00:02	1.0	2.21504	0.0	0.0	0.00000	0.00000	53.6510
3	3	2012-06-01 00:00:03	1.0	2.21504	0.0	0.0	0.00000	0.00000	53.6510
4	4	2012-06-01 00:00:04	1.0	2.21504	0.0	0.0	0.00000	0.00000	55.7929

Statistical Characteristics:

In [16]:
```
house2.describe()
```

Out[16]:

	Unnamed: 0	occupancy	Tablet	Dishwasher	Air exhaust	Fridge	En
count	2.108160e+07	1.088640e+07	2.064960e+07	2.064960e+07	2.064960e+07	2.064960e+07	2.
mean	1.054080e+07	7.349464e-01	1.202025e+00	1.584929e+01	5.644951e-01	2.438221e+01	5.
std	6.085734e+06	4.413617e-01	1.397134e+00	1.801734e+02	7.394131e+00	4.086490e+01	8.
min	0.000000e+00	0.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.
25%	5.270400e+06	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.
50%	1.054080e+07	1.000000e+00	2.215040e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.
75%	1.581120e+07	1.000000e+00	2.215040e+00	0.000000e+00	0.000000e+00	6.817280e+01	5.
max	2.108160e+07	1.000000e+00	1.067200e+01	2.335920e+03	1.857060e+02	1.037750e+03	3.

In [17]:
```
house2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21081600 entries, 0 to 21081599
Data columns (total 14 columns):
Unnamed: 0       int64
date             object
occupancy        float64
Tablet           float64
Dishwasher       float64
Air exhaust      float64
Fridge           float64
Entertainment    float64
Freezer          float64
Kettle           float64
Lamp             float64
Laptops          float64
Stove            float64
Stereo           float64
dtypes: float64(12), int64(1), object(1)
memory usage: 2.2+ GB
```

Correlation:

In [18]:
```
pd.DataFrame.corr(house2)
```

Out[18]:

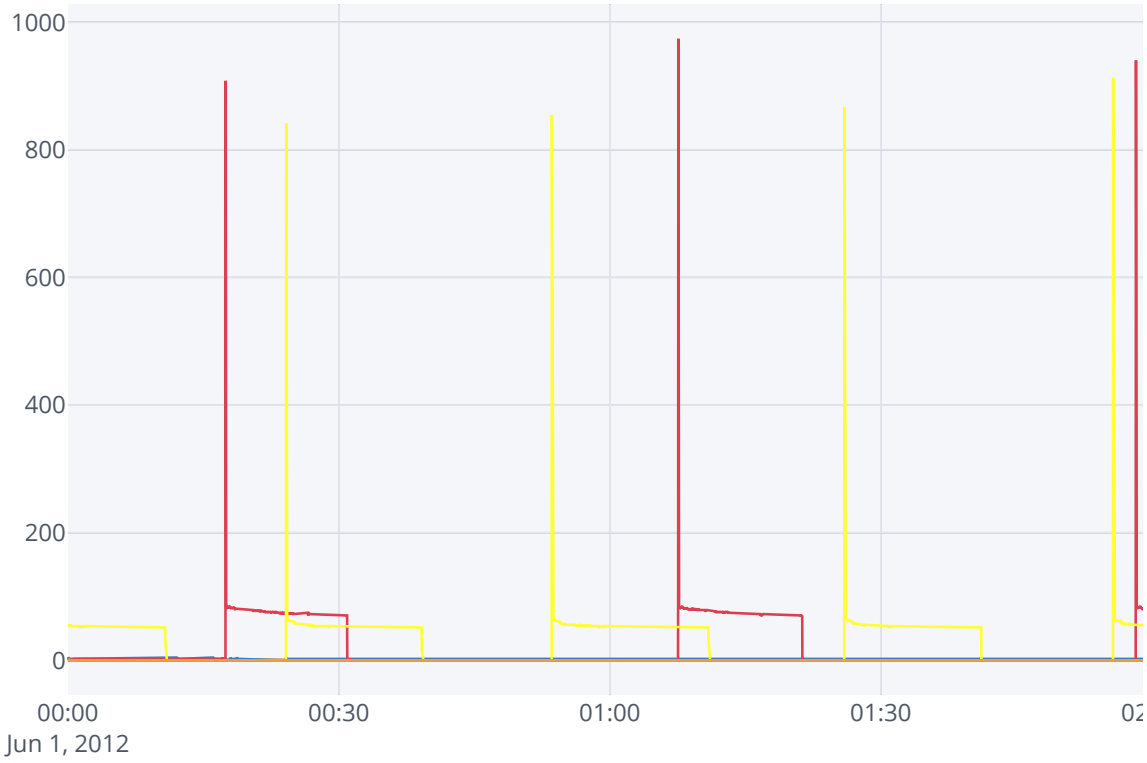
	Unnamed: 0	occupancy	Tablet	Dishwasher	Air exhaust	Fridge	Entertainment	
Unnamed: 0	1.000000	-0.035678	-0.042770	-0.000696	0.012986	-0.027235	0.051407	-0
occupancy	-0.035678	1.000000	-0.016264	0.030555	0.051676	0.026999	0.363817	-0
Tablet	-0.042770	-0.016264	1.000000	0.026378	0.011852	0.032669	0.029561	0
Dishwasher	-0.000696	0.030555	0.026378	1.000000	0.002779	0.004039	0.060516	0
Air exhaust	0.012986	0.051676	0.011852	0.002779	1.000000	0.015754	0.066469	0
Fridge	-0.027235	0.026999	0.032669	0.004039	0.015754	1.000000	0.029078	0
Entertainment	0.051407	0.363817	0.029561	0.060516	0.066469	0.029078	1.000000	0
Freezer	-0.067576	-0.002114	0.033417	0.001657	0.006188	0.022544	0.016083	
Kettle	-0.009343	0.034912	0.003525	-0.002181	0.045596	0.001513	0.031613	-0
Lamp	0.210775	0.165345	0.001212	0.023605	0.087837	0.003777	0.447210	-0
Laptops	-0.033360	0.217732	0.023116	0.047590	0.031006	0.010800	0.350369	0
Stove	-0.021155	0.047757	-0.001898	0.064955	0.328872	0.006518	0.045343	0
Stereo	0.031542	0.383578	0.050481	0.064952	0.085361	0.034642	0.839844	0

Visualize time series data:

In [19]:
```
house2_dt = house2[['date', 'occupancy', 'Tablet', 'Dishwasher',
       'Air exhaust', 'Fridge', 'Entertainment', 'Freezer', 'Kettle', 'Lamp'
       'Laptops', 'Stove', 'Stereo']].iloc[:10000, :]
house2_dt.iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[19]:



minute and 15 minute data:

In [20]:
```
group_mins = house2.groupby(np.arange(len(house2))//60)
house2_by_mins = pd.concat((group_mins['date'].first(), group_mins[[c for c
house2_by_mins.head()
```

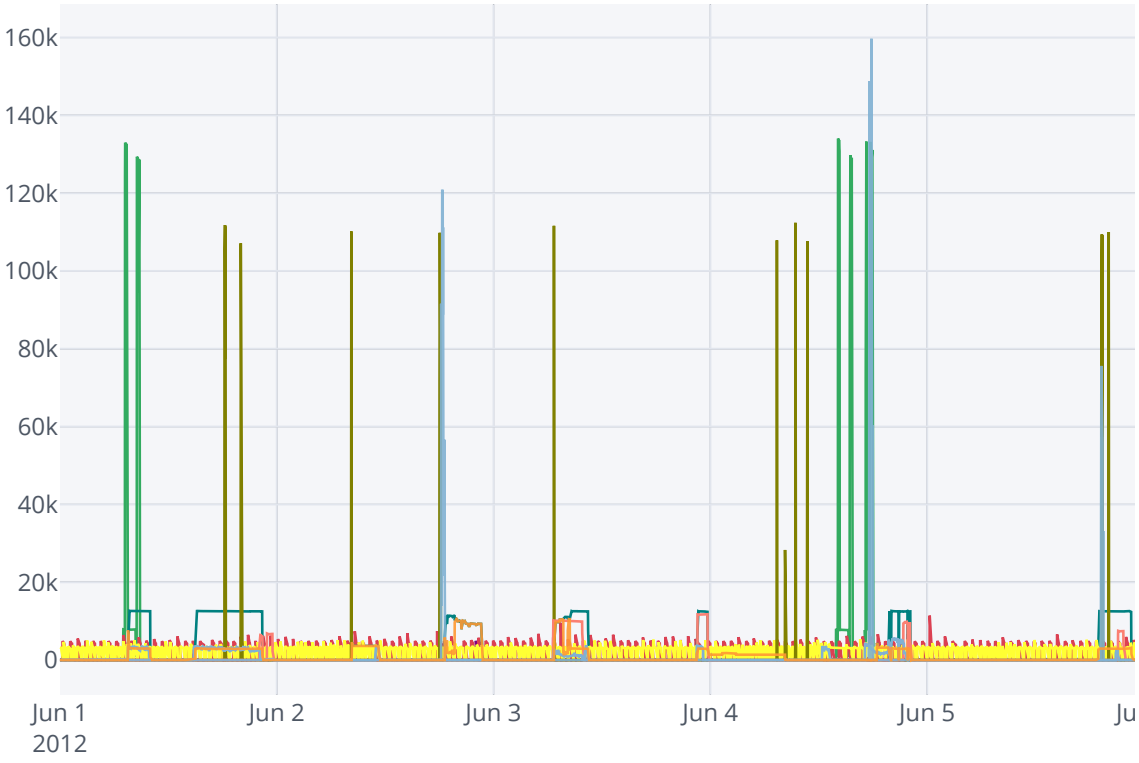
Out[20]:

	date	Unnamed: 0	occupancy	Tablet	Dishwasher	Air exhaust	Fridge	Entertainment	Fre
0	2012-06-01 00:00:00	1770	60.0	147.70222	0.0	0.00000	50.93534	21.71270	3240.
1	2012-06-01 00:01:00	5370	60.0	154.04500	0.0	2.23367	22.14580	26.05524	3204.
2	2012-06-01 00:02:00	8970	60.0	154.04500	0.0	0.00000	26.57496	26.05524	3169.
3	2012-06-01 00:03:00	12570	60.0	145.58796	0.0	0.00000	19.93122	30.39778	3141.
4	2012-06-01 00:04:00	16170	60.0	149.81648	0.0	0.00000	24.36038	21.71270	3150.

In [21]:
```
house2_by_mins[['date', 'occupancy', 'Tablet', 'Dishwasher',
       'Air exhaust', 'Fridge', 'Entertainment', 'Freezer', 'Kettle', 'Lamp'
       'Laptops', 'Stove', 'Stereo']].iloc[0:10000, :].iplot(kind = "scatter
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[21]:



In [22]:
```
group_15mins = house2.groupby(np.arange(len(house2))//900)
house2_by_15mins = pd.concat((group_15mins['date'].first(), group_15mins[[c
house2_by_15mins.head()
```

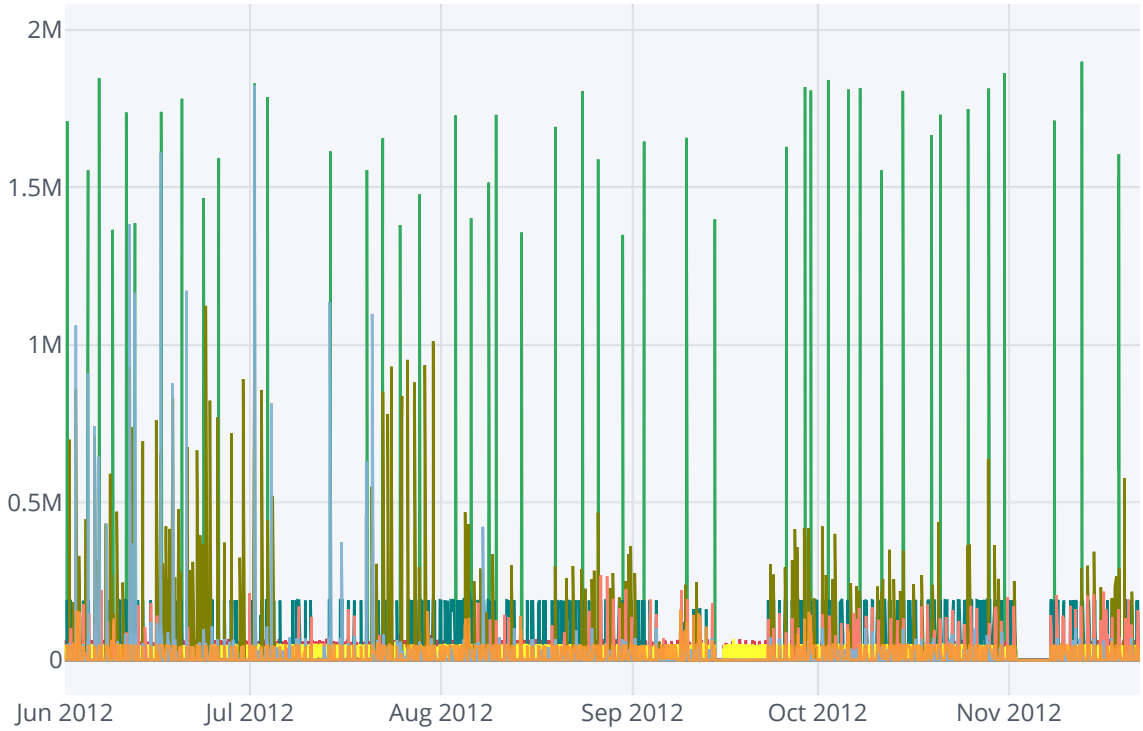
Out[22]:

	date	Unnamed: 0	occupancy	Tablet	Dishwasher	Air exhaust	Fridge	Entertainment	
0	2012-06-01 00:00:00	404550	900.0	2156.13246	0.0	6.70101	372.04944	384.31479	
1	2012-06-01 00:15:00	1214550	900.0	1795.28886	0.0	6.70101	58471.07208	410.37003	
2	2012-06-01 00:30:00	2024550	900.0	959.11232	0.0	4.46734	4215.81695	395.17114	
3	2012-06-01 00:45:00	2834550	900.0	945.82208	0.0	2.23367	367.62028	423.39765	
4	2012-06-01 01:00:00	3644550	900.0	1007.84320	0.0	4.46734	36376.85084	382.14352	

In [23]:
```
house2_by_15mins[['date', 'occupancy', 'Tablet', 'Dishwasher',
       'Air exhaust', 'Fridge', 'Entertainment', 'Freezer', 'Kettle', 'Lamp'
       'Laptops', 'Stove', 'Stereo']].iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[23]:



House 3:

In [24]:
```
house3.head()
```

Out[24]:

	Unnamed: 0	date	occupancy	Tablet	Freezer	Coffee machine	Fridge	Kettle	Entertainment
0	0	2012-10-23 00:00:00	NaN	-1.0	-1.0	-1.0	-1.0	NaN	NaN
1	1	2012-10-23 00:00:01	NaN	-1.0	-1.0	-1.0	-1.0	NaN	NaN
2	2	2012-10-23 00:00:02	NaN	-1.0	-1.0	-1.0	-1.0	NaN	NaN
3	3	2012-10-23 00:00:03	NaN	-1.0	-1.0	-1.0	-1.0	NaN	NaN
4	4	2012-10-23 00:00:04	NaN	-1.0	-1.0	-1.0	-1.0	NaN	NaN

Statistical Characteristics:

In [25]:
```
house3.describe()
```

Out[25]:

	Unnamed: 0	occupancy	Tablet	Freezer	Coffee machine	Fridge	
count	8.640000e+06	1.814400e+06	8.294400e+06	8.294400e+06	5.184000e+06	3.542400e+06	3.
mean	4.320000e+06	7.438040e-01	1.280892e+00	5.770378e+00	2.899241e-01	9.513641e+00	1.
std	2.494153e+06	4.365315e-01	1.605000e+00	9.244474e+00	2.159075e+01	2.689626e+01	1.
min	0.000000e+00	0.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.
25%	2.160000e+06	0.000000e+00	0.000000e+00	0.000000e+00	-1.000000e+00	-1.000000e+00	-1.
50%	4.320000e+06	1.000000e+00	0.000000e+00	2.225500e+00	0.000000e+00	0.000000e+00	0.
75%	6.479999e+06	1.000000e+00	2.238570e+00	2.225500e+00	0.000000e+00	0.000000e+00	0.
max	8.639999e+06	1.000000e+00	1.071860e+01	7.439030e+01	1.295990e+03	1.211630e+03	2.

In [26]:
```
house3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8640000 entries, 0 to 8639999
Data columns (total 9 columns):
Unnamed: 0        int64
date              object
occupancy         float64
Tablet            float64
Freezer           float64
Coffee machine    float64
Fridge            float64
Kettle            float64
Entertainment     float64
dtypes: float64(7), int64(1), object(1)
memory usage: 593.3+ MB
```

Correlation:

In [27]:
```
pd.DataFrame.corr(house3)
```

Out[27]:

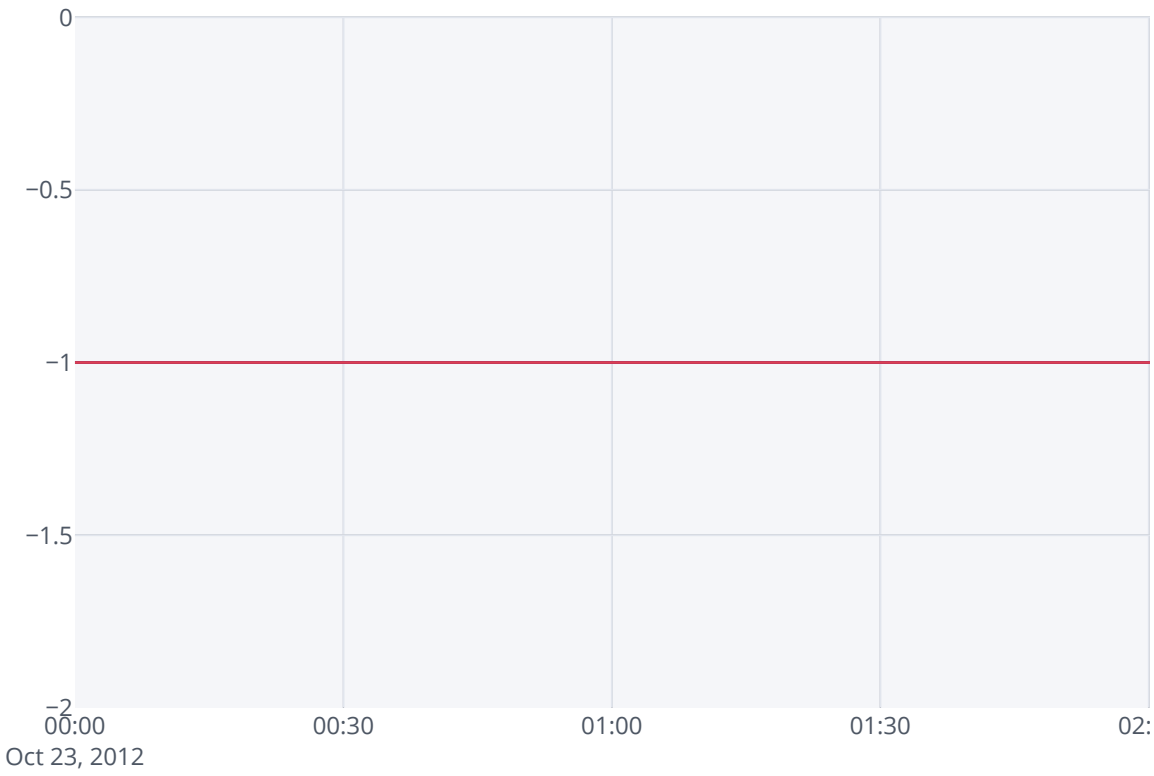
	Unnamed: 0	occupancy	Tablet	Freezer	Coffee machine	Fridge	Kettle	Enterta
Unnamed: 0	1.000000	0.246438	0.071583	0.028032	-0.009354	0.013647	-0.019516	0.
occupancy	0.246438	1.000000	-0.098691	0.011559	0.025263	0.092494	-0.038727	0.
Tablet	0.071583	-0.098691	1.000000	0.008209	0.001777	-0.030414	-0.011618	0.
Freezer	0.028032	0.011559	0.008209	1.000000	0.002975	-0.007825	0.000474	0.
Coffee machine	-0.009354	0.025263	0.001777	0.002975	1.000000	0.002145	-0.001492	0.
Fridge	0.013647	0.092494	-0.030414	-0.007825	0.002145	1.000000	0.023615	0.
Kettle	-0.019516	-0.038727	-0.011618	0.000474	-0.001492	0.023615	1.000000	0.
Entertainment	0.000505	0.117429	0.041807	0.006144	0.008450	0.017771	0.140681	1.

Visualize time series data:

In [67]:
```
house3_dt = house3[['date', 'occupancy', 'Tablet', 'Freezer',
       'Coffee machine', 'Fridge', 'Kettle', 'Entertainment']].iloc[:10000,
house3_dt.iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[67]:



minute and 15 minute data:

In [29]:
```
group_mins = house3.groupby(np.arange(len(house3))//60)
house3_by_mins = pd.concat((group_mins['date'].first(), group_mins[[c for c :
house3_by_mins.head()
```

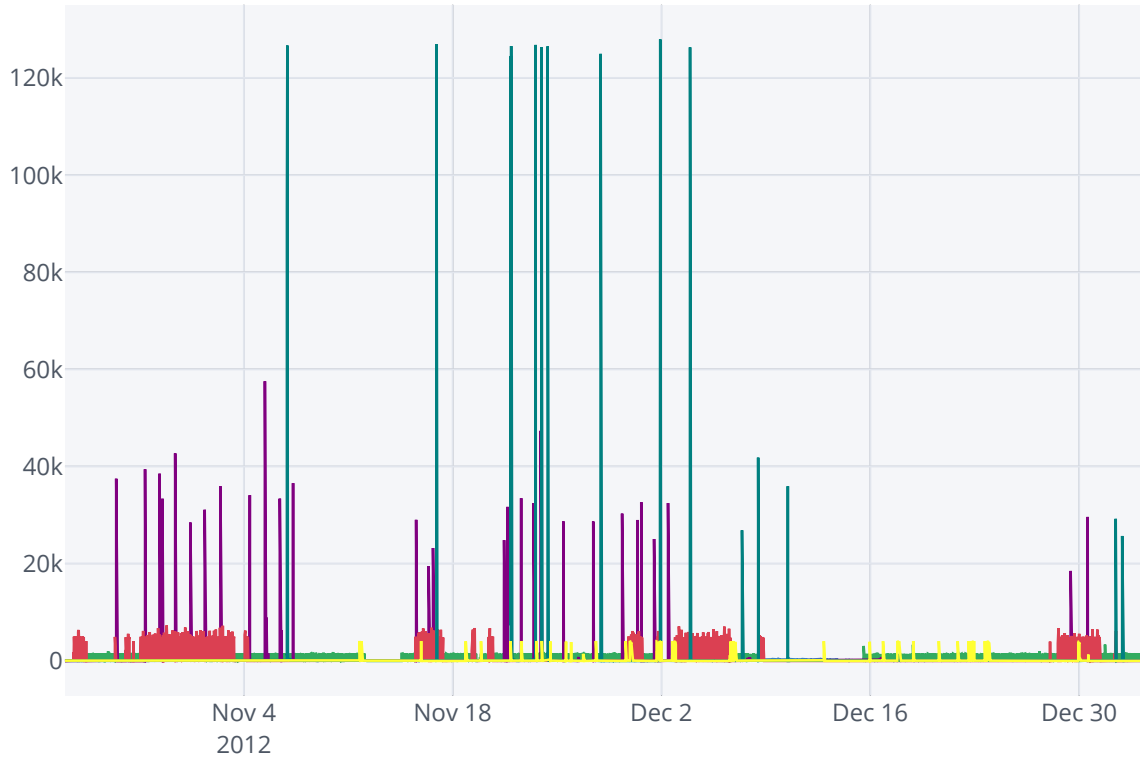
Out[29]:

	date	Unnamed: 0	occupancy	Tablet	Freezer	Coffee machine	Fridge	Kettle	Entertainment
0	2012-10-23 00:00:00	1770	0.0	-60.0	-60.0	-60.0	-60.0	0.0	0.0
1	2012-10-23 00:01:00	5370	0.0	-60.0	-60.0	-60.0	-60.0	0.0	0.0
2	2012-10-23 00:02:00	8970	0.0	-60.0	-60.0	-60.0	-60.0	0.0	0.0
3	2012-10-23 00:03:00	12570	0.0	-60.0	-60.0	-60.0	-60.0	0.0	0.0
4	2012-10-23 00:04:00	16170	0.0	-60.0	-60.0	-60.0	-60.0	0.0	0.0

In [30]:
```
house3_by_mins[['date', 'occupancy', 'Tablet', 'Freezer',
       'Coffee machine', 'Fridge', 'Kettle', 'Entertainment']].iplot(kind =
```

```
/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/plotly/plotly.
py:230: UserWarning:

Woah there! Look at all those points! Due to browser limitations, the Plo
tly SVG drawing functions have a hard time graphing more than 500k data p
oints for line charts, or 40k points for other types of charts. Here are
some suggestions:
(1) Use the `plotly.graph_objs.Scattergl` trace object to generate a WebG
l graph.
(2) Trying using the image API to return an image instead of a graph URL
(3) Use matplotlib
(4) See if you can create your visualization with fewer data points

If the visualization you're using aggregates points (e.g., box plot, hist
ogram, etc.) you can disregard this warning.


The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[30]:



In [31]:
```
group_15mins = house3.groupby(np.arange(len(house3))//900)
house3_by_15mins = pd.concat((group_15mins['date'].first(), group_15mins[[c
house3_by_15mins.head()
```

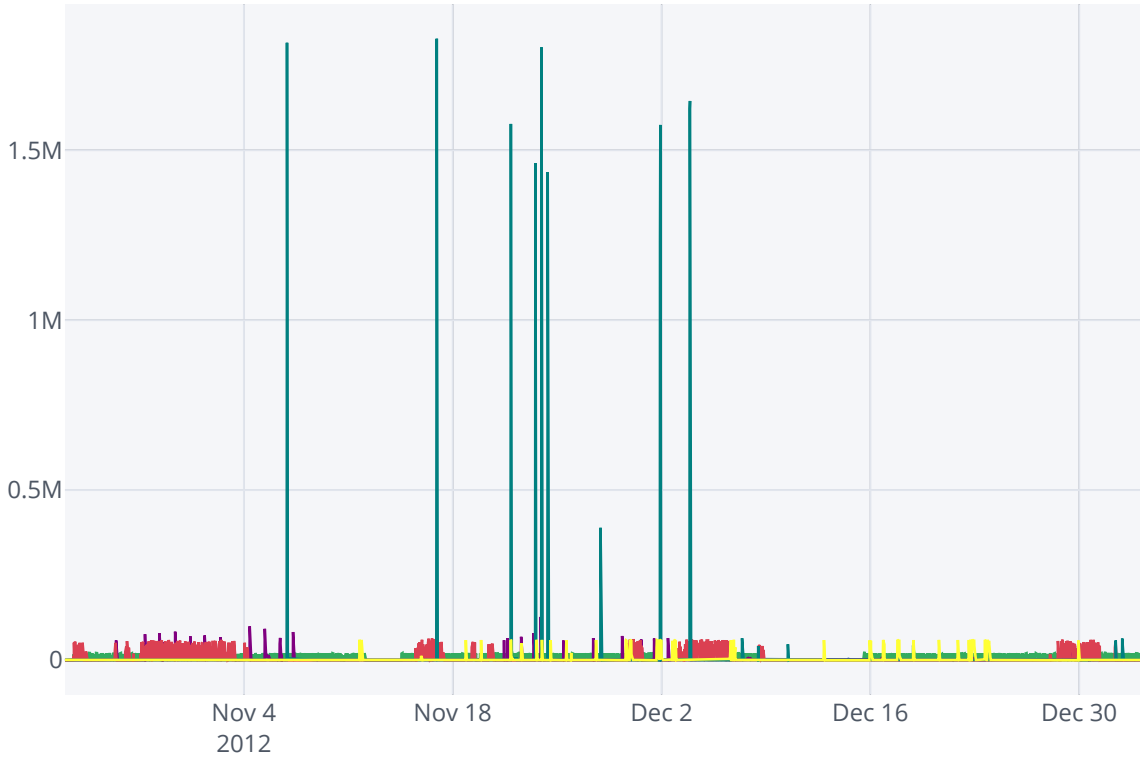
Out[31]:

	date	Unnamed: 0	occupancy	Tablet	Freezer	Coffee machine	Fridge	Kettle	Entertainment
0	2012-10-23 00:00:00	404550	0.0	-900.0	-900.0	-900.0	-900.0	0.0	0.0
1	2012-10-23 00:15:00	1214550	0.0	-900.0	-900.0	-900.0	-900.0	0.0	0.0
2	2012-10-23 00:30:00	2024550	0.0	-900.0	-900.0	-900.0	-900.0	0.0	0.0
3	2012-10-23 00:45:00	2834550	0.0	-900.0	-900.0	-900.0	-900.0	0.0	0.0
4	2012-10-23 01:00:00	3644550	0.0	-900.0	-900.0	-900.0	-900.0	0.0	0.0

In [32]:
```
house3_by_15mins[['date', 'occupancy', 'Tablet', 'Freezer',
       'Coffee machine', 'Fridge', 'Kettle', 'Entertainment']].iplot(kind =
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[32]:



House 4:

In [33]:
```
house4.head()
```

Out[33]:

	Unnamed: 0	date	occupancy	Fridge	Kitchen appliances	Lamp	Stereo and laptop	Freezer	Tablet	Enterta
0	0	2012-06-27 00:00:00	NaN	102.429	2.16516	2.23978	15.0524	172.720	0.00000	
1	1	2012-06-27 00:00:01	NaN	100.296	2.16516	2.23978	15.0524	170.589	0.00000	
2	2	2012-06-27 00:00:02	NaN	102.429	0.00000	0.00000	15.0524	172.720	0.00000	
3	3	2012-06-27 00:00:03	NaN	102.429	0.00000	0.00000	15.0524	172.720	2.22889	
4	4	2012-06-27 00:00:04	NaN	100.296	2.16516	2.23978	15.0524	172.720	0.00000	

Statistical Characteristics:

In [34]:
```
house4.describe()
```

Out[34]:

	Unnamed: 0	occupancy	Fridge	Kitchen appliances	Lamp	Stereo and laptop	
count	1.814400e+07	7.430400e+06	1.667520e+07	1.667520e+07	1.468800e+07	1.460160e+07	1.
mean	9.072000e+06	9.335496e-01	2.703878e+01	9.410700e+00	1.056615e+01	1.219372e+01	1.
std	5.237722e+06	2.490678e-01	4.481938e+01	9.932784e+01	2.793486e+01	1.428696e+01	1.
min	0.000000e+00	0.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.
25%	4.536000e+06	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	9.
50%	9.072000e+06	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.291450e+01	1.
75%	1.360800e+07	1.000000e+00	8.749980e+01	0.000000e+00	2.239780e+00	1.505240e+01	2.
max	1.814400e+07	1.000000e+00	1.174220e+03	2.331240e+03	8.675200e+02	1.497260e+02	3.

In [35]:
```
house4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18144000 entries, 0 to 18143999
Data columns (total 11 columns):
Unnamed: 0            int64
date                  object
occupancy             float64
Fridge                float64
Kitchen appliances    float64
Lamp                  float64
Stereo and laptop     float64
Freezer               float64
Tablet                float64
Entertainment         float64
Microwave             float64
dtypes: float64(9), int64(1), object(1)
memory usage: 1.5+ GB
```

Correlation:

In [36]:
```
pd.DataFrame.corr(house4)
```

Out[36]:

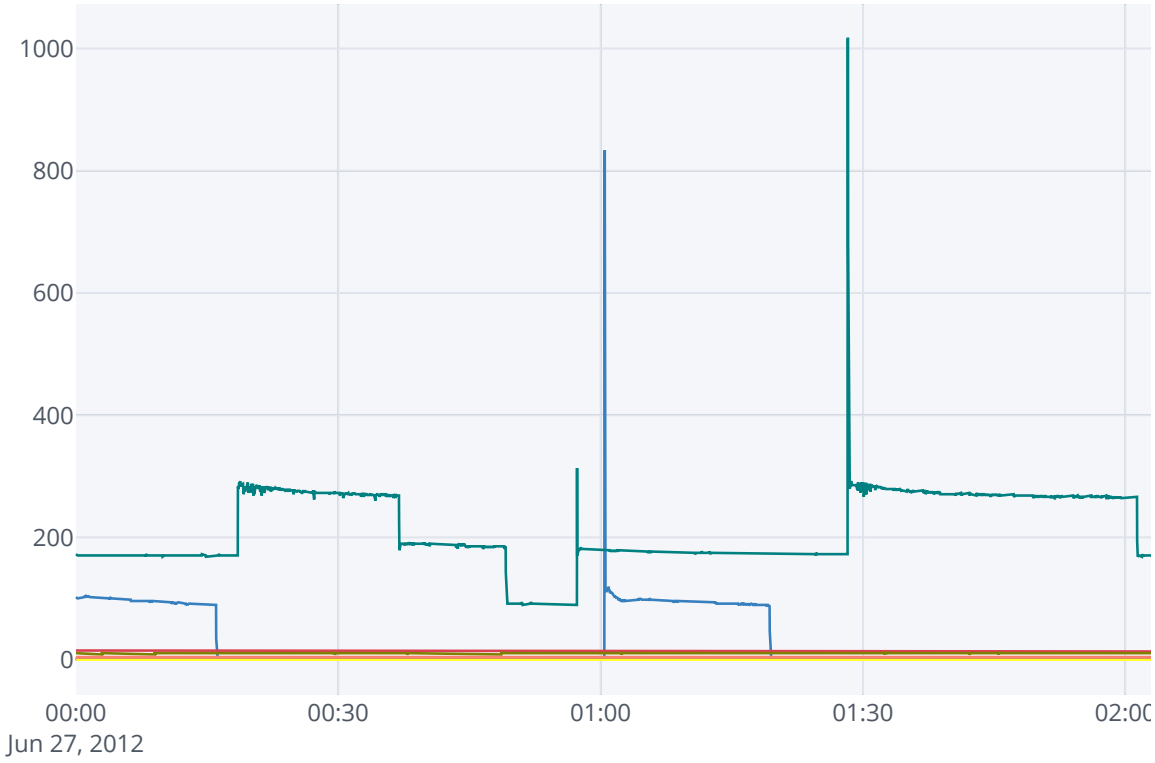
	Unnamed: 0	occupancy	Fridge	Kitchen appliances	Lamp	Stereo and laptop	Freezer	Tal
Unnamed: 0	1.000000	0.048192	-0.064277	0.015337	0.333668	-0.335850	-0.396736	0.059
occupancy	0.048192	1.000000	0.000025	0.024750	-0.003765	0.054727	-0.030660	-0.034
Fridge	-0.064277	0.000025	1.000000	0.014399	-0.003508	0.051860	0.056912	0.002
Kitchen appliances	0.015337	0.024750	0.014399	1.000000	0.018169	0.007180	0.002179	-0.001
Lamp	0.333668	-0.003765	-0.003508	0.018169	1.000000	-0.146481	0.048068	0.032
Stereo and laptop	-0.335850	0.054727	0.051860	0.007180	-0.146481	1.000000	0.075864	0.009
Freezer	-0.396736	-0.030660	0.056912	0.002179	0.048068	0.075864	1.000000	-0.023
Tablet	0.059188	-0.034001	0.002854	-0.001926	0.032655	0.009398	-0.023911	1.000
Entertainment	0.021430	0.121855	0.021206	0.009854	0.037990	0.050456	0.079761	0.003
Microwave	0.013133	0.026881	0.017328	0.090353	0.018050	0.011214	0.006758	0.002

Visualize time series data:

In [37]:
```
house4_dt = house4[['date', 'occupancy', 'Fridge', 'Kitchen appliances',
       'Lamp', 'Stereo and laptop', 'Freezer', 'Tablet', 'Entertainment',
       'Microwave']].iloc[:10000, :]
house4_dt.iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[37]:



minute and 15 minute data:

In [38]:
```
group_mins = house4.groupby(np.arange(len(house4))//60)
house4_by_mins = pd.concat((group_mins['date'].first(), group_mins[[c for c
house4_by_mins.head()
```

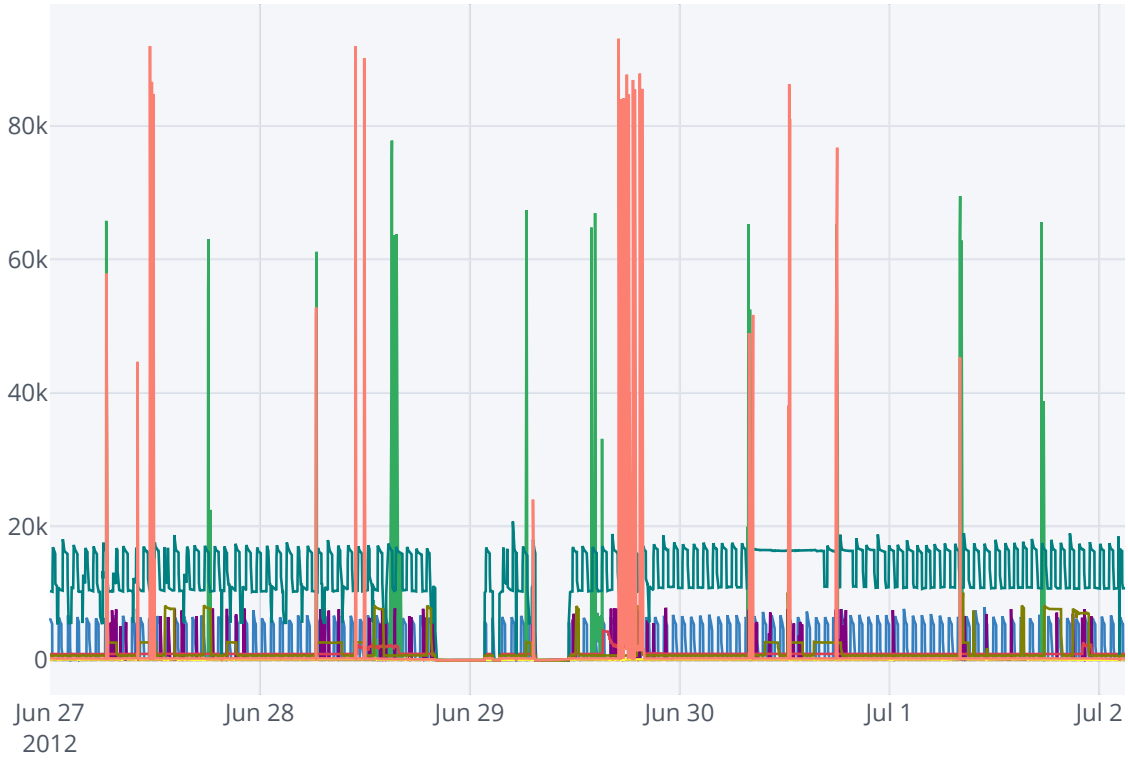
Out[38]:

	date	Unnamed: 0	occupancy	Fridge	Kitchen appliances	Lamp	Stereo and laptop	Freezer	Tablet
0	2012-06-27 00:00:00	1770	0.0	6090.2820	21.65160	47.03538	886.0408	10318.449	57.95114
1	2012-06-27 00:01:00	5370	0.0	6182.0010	54.12900	26.87736	853.9723	10307.794	78.01115
2	2012-06-27 00:02:00	8970	0.0	6135.0750	17.32128	58.23428	845.4207	10318.449	64.63781
3	2012-06-27 00:03:00	12570	0.0	6049.7560	41.13804	38.07626	862.5239	10260.912	69.09559
4	2012-06-27 00:04:00	16170	0.0	5977.2425	45.46836	20.15802	881.7650	10243.864	35.66224

In [39]:
```
house4_by_mins[['date', 'occupancy', 'Fridge', 'Kitchen appliances',
       'Lamp', 'Stereo and laptop', 'Freezer', 'Tablet', 'Entertainment',
       'Microwave']].iloc[0:10000, :].iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[39]:



In [40]:
```
group_15mins = house4.groupby(np.arange(len(house4))//900)
house4_by_15mins = pd.concat((group_15mins['date'].first(), group_15mins[[c
house4_by_15mins.head()
```

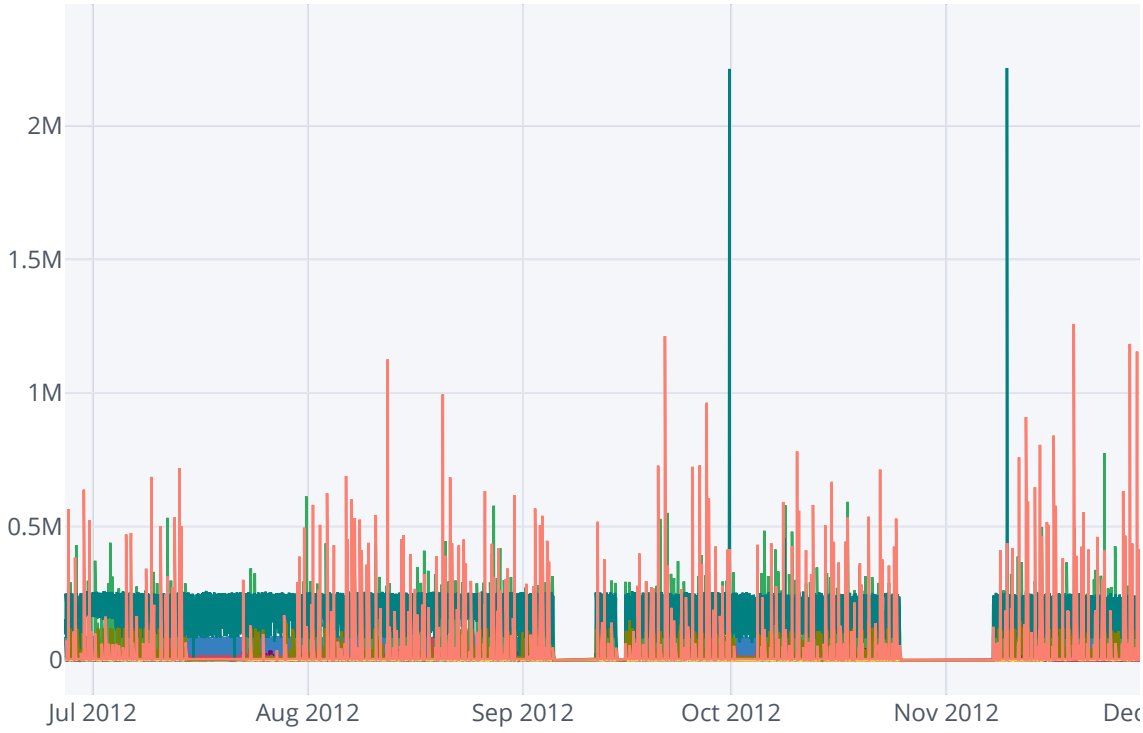
Out[40]:

	date	Unnamed: 0	occupancy	Fridge	Kitchen appliances	Lamp	Stereo and laptop	Freezer	
0	2012-06-27 00:00:00	404550	0.0	87041.93120	480.66552	598.02126	13057.5809	153738.9380	8
1	2012-06-27 00:15:00	1214550	0.0	5615.41787	452.51844	602.50082	13004.1334	227019.6110	7
2	2012-06-27 00:30:00	2024550	0.0	85.15533	474.17004	604.74060	13029.7882	203782.1050	5
3	2012-06-27 00:45:00	2834550	0.0	39.30246	441.69264	562.18478	13012.6850	120148.5577	9
4	2012-06-27 01:00:00	3644550	0.0	86007.65210	474.17004	582.34280	13066.1325	158572.0460	8

In [41]:
```
house4_by_15mins[['date', 'occupancy', 'Fridge', 'Kitchen appliances',
       'Lamp', 'Stereo and laptop', 'Freezer', 'Tablet', 'Entertainment',
       'Microwave']].iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[41]:



House 5:

In [42]:
```
house5.head()
```

Out[42]:

	Unnamed: 0	date	occupancy	Tablet	Coffee machine	Fountain	Microwave	Fridge	Entertainment
0	0	2012-06-27 00:00:00	NaN	2.20778	4.48706	8.72041	4.44332	4.44546	6.56679
1	1	2012-06-27 00:00:01	NaN	2.20778	2.34770	8.72041	4.44332	4.44546	8.69303
2	2	2012-06-27 00:00:02	NaN	4.33249	4.48706	8.72041	6.57853	4.44546	8.69303
3	3	2012-06-27 00:00:03	NaN	4.33249	4.48706	8.72041	4.44332	4.44546	6.56679
4	4	2012-06-27 00:00:04	NaN	2.20778	2.34770	8.72041	4.44332	4.44546	8.69303

Statistical Characteristics:

In [43]:
```
house5.describe()
```

Out[43]:

	Unnamed: 0	occupancy	Tablet	Coffee machine	Fountain	Microwave	
count	1.883520e+07	6.393600e+06	1.874880e+07	1.874880e+07	6.134400e+06	1.874880e+07	1.
mean	9.417600e+06	9.008222e-01	4.598005e+00	5.523087e+00	1.193648e+01	8.713093e+00	4.
std	5.437254e+06	2.989006e-01	1.267522e+00	8.333408e+01	9.892267e+00	8.431363e+01	5.
min	0.000000e+00	0.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.
25%	4.708800e+06	1.000000e+00	4.332490e+00	0.000000e+00	8.720410e+00	4.443320e+00	4.
50%	9.417600e+06	1.000000e+00	4.332490e+00	0.000000e+00	8.720410e+00	4.443320e+00	4.
75%	1.412640e+07	1.000000e+00	4.332490e+00	0.000000e+00	8.720410e+00	6.578530e+00	1.
max	1.883520e+07	1.000000e+00	1.495590e+01	1.581540e+03	4.509910e+01	2.680550e+03	1.

In [44]:
```
house5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18835200 entries, 0 to 18835199
Data columns (total 10 columns):
Unnamed: 0        int64
date              object
occupancy         float64
Tablet            float64
Coffee machine    float64
Fountain          float64
Microwave         float64
Fridge            float64
Entertainment     float64
Kettle            float64
dtypes: float64(8), int64(1), object(1)
memory usage: 1.4+ GB
```

Correlation:

In [45]:
```
pd.DataFrame.corr(house5)
```

Out[45]:

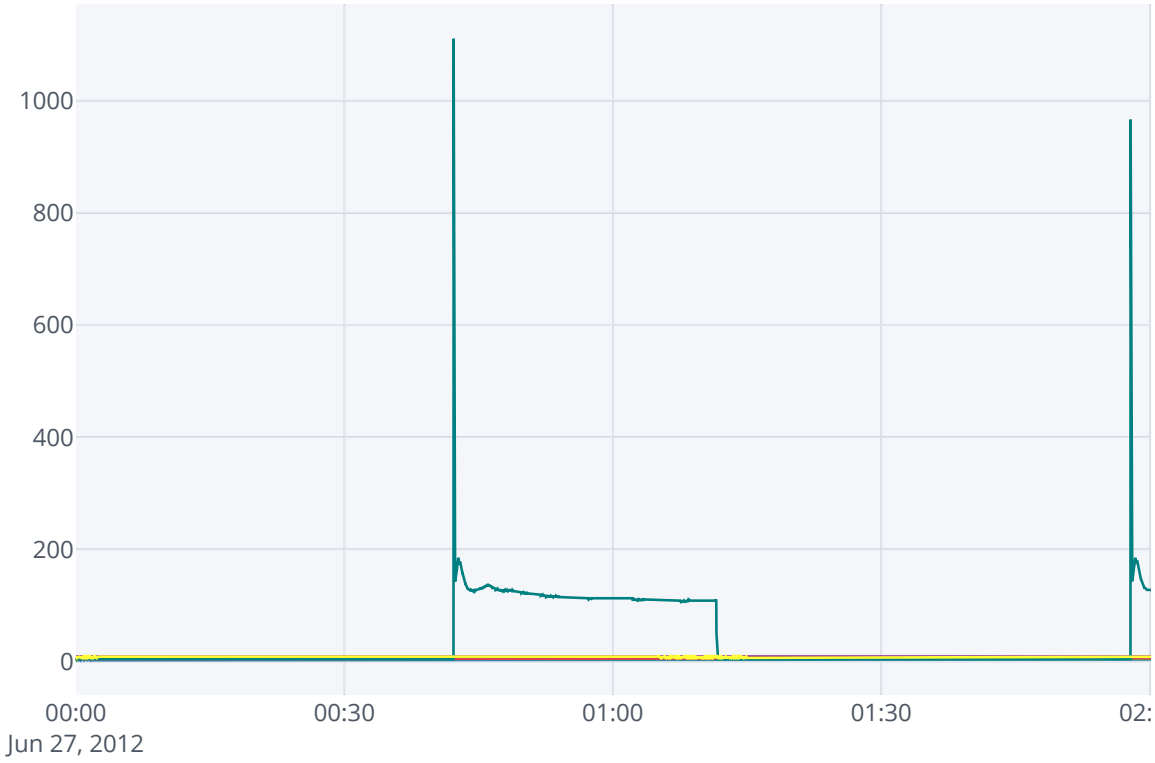
	Unnamed: 0	occupancy	Tablet	Coffee machine	Fountain	Microwave	Fridge	Entert
Unnamed: 0	1.000000	-0.095775	-0.034843	0.000702	0.017279	-0.015699	-0.050267	(
occupancy	-0.095775	1.000000	-0.048231	0.018819	0.081846	0.014140	0.010938	(
Tablet	-0.034843	-0.048231	1.000000	0.005634	0.031790	0.012375	0.066179	(
Coffee machine	0.000702	0.018819	0.005634	1.000000	-0.017290	0.003413	0.007602	-(
Fountain	0.017279	0.081846	0.031790	-0.017290	1.000000	-0.016810	0.008741	(
Microwave	-0.015699	0.014140	0.012375	0.003413	-0.016810	1.000000	0.006285	-(
Fridge	-0.050267	0.010938	0.066179	0.007602	0.008741	0.006285	1.000000	(
Entertainment	0.065159	0.076323	0.049081	-0.014424	0.116845	-0.009260	0.009198	-
Kettle	-0.006182	NaN	0.007990	0.000201	NaN	-0.000520	-0.003141	-(

Visualize time series data:

In [65]:
```
house5_dt = house5[['date', 'occupancy', 'Tablet', 'Coffee machine',
       'Fountain', 'Microwave', 'Fridge', 'Entertainment', 'Kettle']].iloc[:
house5_dt.iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[65]:



minute and 15 minute data:

In [49]:
```
group_mins = house5.groupby(np.arange(len(house5))//60)
house5_by_mins = pd.concat((group_mins['date'].first(), group_mins[[c for c i
house5_by_mins.head()
```

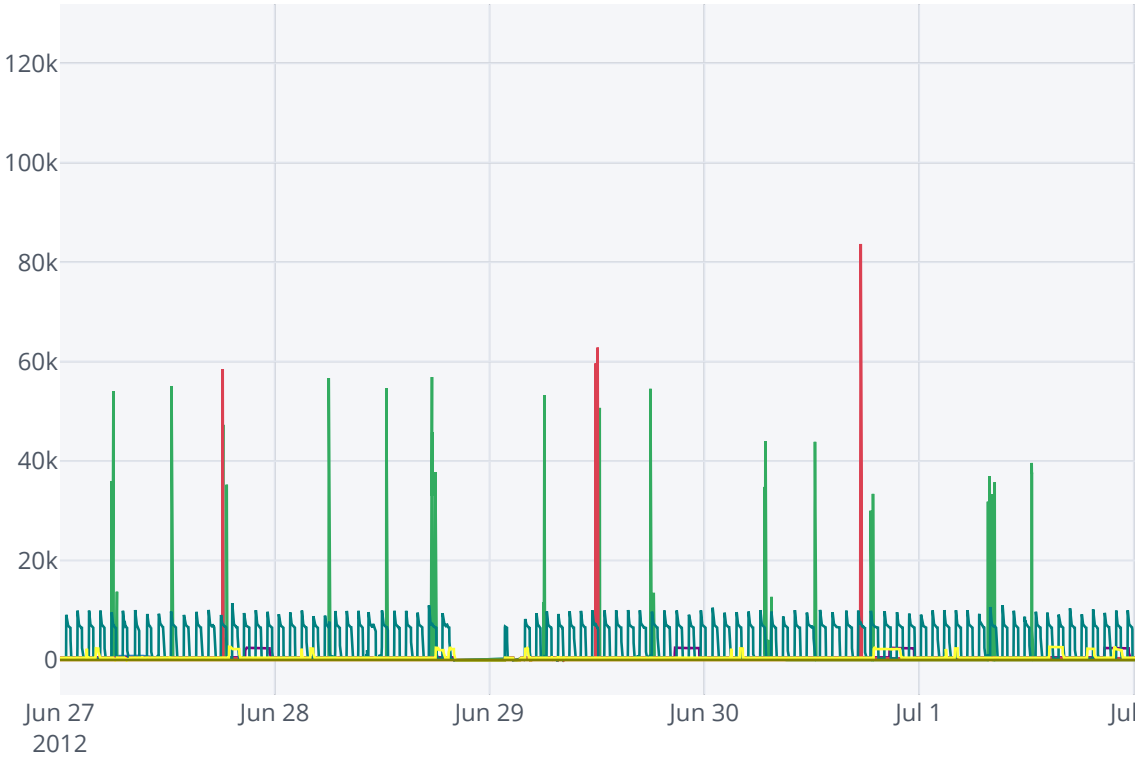
Out[49]:

	date	Unnamed: 0	occupancy	Tablet	Coffee machine	Fountain	Microwave	Fridge	Entertai
0	2012-06-27 00:00:00	1770	0.0	194.08339	245.69064	510.3846	319.97945	273.10203	464.
1	2012-06-27 00:01:00	5370	0.0	189.83397	239.27256	503.9646	322.11466	270.97722	472.
2	2012-06-27 00:02:00	8970	0.0	211.08107	249.96936	506.1046	309.30340	270.97722	470.
3	2012-06-27 00:03:00	12570	0.0	200.45752	239.27256	503.9646	319.97945	275.22684	457.
4	2012-06-27 00:04:00	16170	0.0	206.83165	239.27256	499.6846	319.97945	273.10203	455.

In [50]:
```
house5_by_mins[['date', 'occupancy', 'Tablet', 'Coffee machine',
       'Fountain', 'Microwave', 'Fridge', 'Entertainment', 'Kettle']].iloc[(
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[50]:



In [51]:
```
group_15mins = house5.groupby(np.arange(len(house5))//900)
house5_by_15mins = pd.concat((group_15mins['date'].first(), group_15mins[[c
house5_by_15mins.head()
```

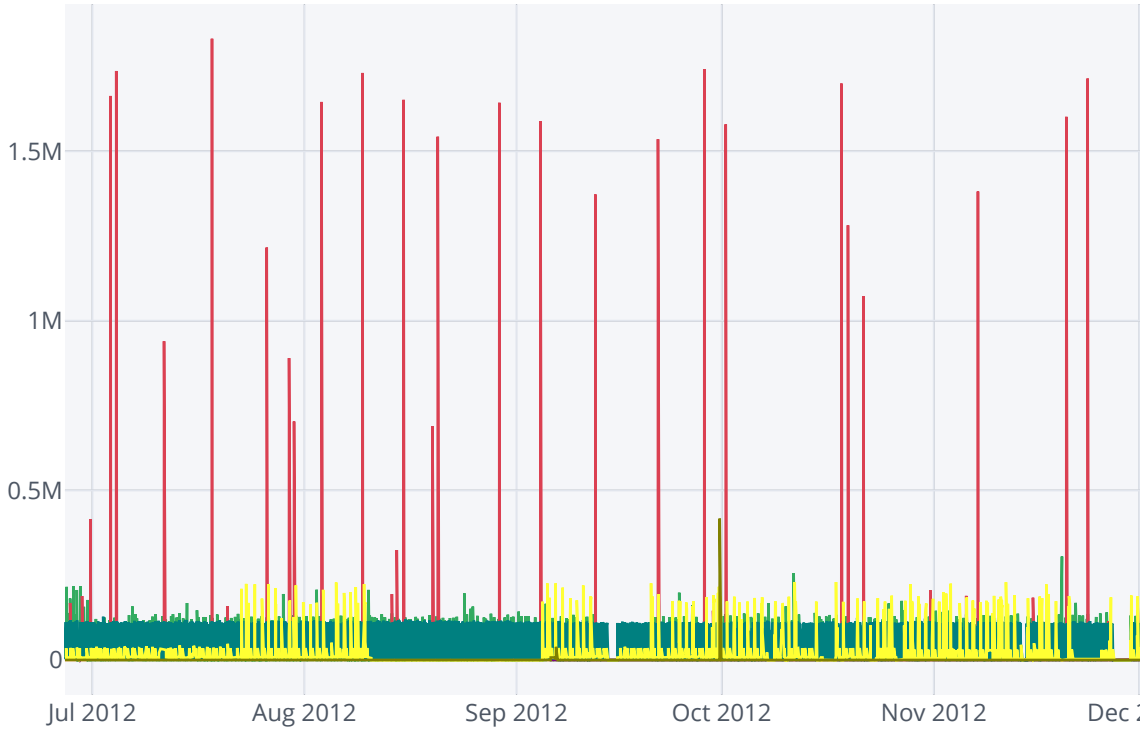
Out[51]:

	date	Unnamed: 0	occupancy	Tablet	Coffee machine	Fountain	Microwave	Fridge	Er
0	2012-06-27 00:00:00	404550	0.0	2996.23925	3621.17880	7578.729	4752.71713	4109.27931	
1	2012-06-27 00:15:00	1214550	0.0	2957.99447	3621.17880	7570.169	4767.66360	4143.27627	
2	2012-06-27 00:30:00	2024550	0.0	2943.12150	3621.17880	7625.809	4769.79881	28627.50803	
3	2012-06-27 00:45:00	2834550	0.0	2943.12150	3606.20328	7619.389	4816.77343	107712.70900	
4	2012-06-27 01:00:00	3644550	0.0	3019.61106	3648.99048	7619.389	4825.31427	76826.05290	

In [52]:
```
house5_by_15mins[['date', 'occupancy', 'Tablet', 'Coffee machine',
       'Fountain', 'Microwave', 'Fridge', 'Entertainment', 'Kettle']].iplot(
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[52]:



House 6:

In [53]:
```
house6.head()
```

Out[53]:

	Unnamed: 0	date	Lamp	Laptop	Router	Coffee machine	Entertainment	Fridge	Kettle
0	0	2012-06-27 00:00:00	0.0	4.35384	19.3387	0.0	15.0043	2.19884	0.0
1	1	2012-06-27 00:00:01	0.0	4.35384	19.3387	0.0	15.0043	2.19884	0.0
2	2	2012-06-27 00:00:02	0.0	6.47995	19.3387	0.0	15.0043	0.00000	0.0
3	3	2012-06-27 00:00:03	0.0	6.47995	19.3387	0.0	15.0043	0.00000	0.0
4	4	2012-06-27 00:00:04	0.0	4.35384	19.3387	0.0	15.0043	0.00000	0.0

Statistical Characteristics:

In [54]:
```
house6.describe()
```

Out[54]:

	Unnamed: 0	Lamp	Laptop	Router	Coffee machine	Entertainment	
count	1.883520e+07	1.425600e+07	1.589760e+07	7.603200e+06	1.537920e+07	1.555200e+07	1
mean	9.417600e+06	-8.376633e-02	6.586103e+00	1.917793e+01	3.643533e+00	2.341736e+01	1
std	5.437254e+06	2.567623e+00	6.320948e+00	3.825831e+00	4.375191e+01	2.186640e+01	3
min	0.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1
25%	4.708800e+06	-1.000000e+00	4.353840e+00	1.933870e+01	0.000000e+00	1.500430e+01	0
50%	9.417600e+06	0.000000e+00	6.479950e+00	1.933870e+01	0.000000e+00	1.500430e+01	0
75%	1.412640e+07	0.000000e+00	6.479950e+00	1.933870e+01	0.000000e+00	2.139790e+01	2
max	1.883520e+07	4.729930e+01	9.364450e+01	2.786790e+01	1.285580e+03	1.535160e+02	1

In [55]:
```
house6.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18835200 entries, 0 to 18835199
Data columns (total 9 columns):
Unnamed: 0        int64
date              object
Lamp              float64
Laptop            float64
Router            float64
Coffee machine    float64
Entertainment     float64
Fridge            float64
Kettle            float64
dtypes: float64(7), int64(1), object(1)
memory usage: 1.3+ GB
```

Correlation:

In [56]:
```
pd.DataFrame.corr(house6)
```

Out[56]:

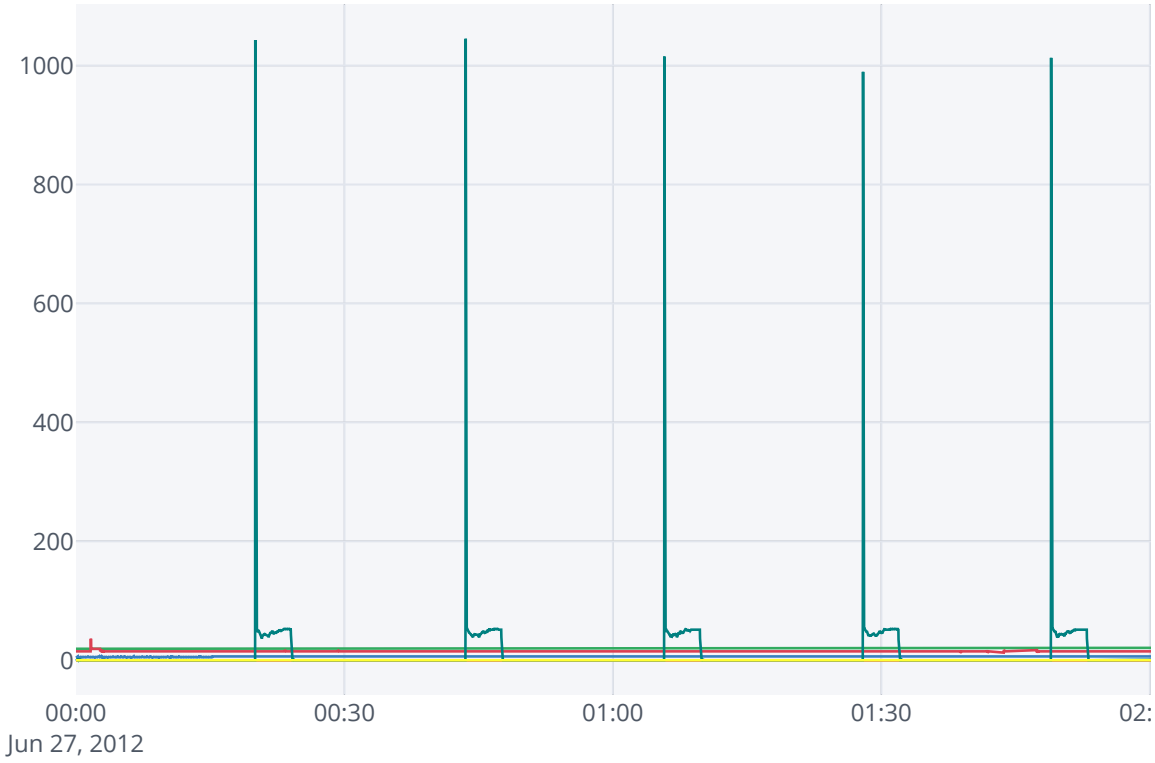
	Unnamed: 0	Lamp	Laptop	Router	Coffee machine	Entertainment	Fridge	K
Unnamed: 0	1.000000	0.072416	-0.033790	-0.055584	0.002007	0.023260	-0.028748	-0.00
Lamp	0.072416	1.000000	0.022549	0.076244	0.018302	0.089889	0.010271	0.02
Laptop	-0.033790	0.022549	1.000000	0.193302	0.027232	0.136001	0.021443	0.00
Router	-0.055584	0.076244	0.193302	1.000000	0.018071	0.205047	0.070357	0.00
Coffee machine	0.002007	0.018302	0.027232	0.018071	1.000000	-0.014171	0.000028	0.00
Entertainment	0.023260	0.089889	0.136001	0.205047	-0.014171	1.000000	0.020155	0.02
Fridge	-0.028748	0.010271	0.021443	0.070357	0.000028	0.020155	1.000000	0.00
Kettle	-0.002645	0.020491	0.007476	0.008394	0.009734	0.028553	0.001574	1.00

Visualize time series data:

In [57]:
```
house6_dt = house6[['date', 'Lamp', 'Laptop', 'Router', 'Coffee machine',
       'Entertainment', 'Fridge', 'Kettle']].iloc[:10000, :]
house6_dt.iplot(kind = "scatter", x = 'date')
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[57]:



minute and 15 minute data:

In [58]:
```
group_mins = house6.groupby(np.arange(len(house6))//60)
house6_by_mins = pd.concat((group_mins['date'].first(), group_mins[[c for c
house6_by_mins.head()
```

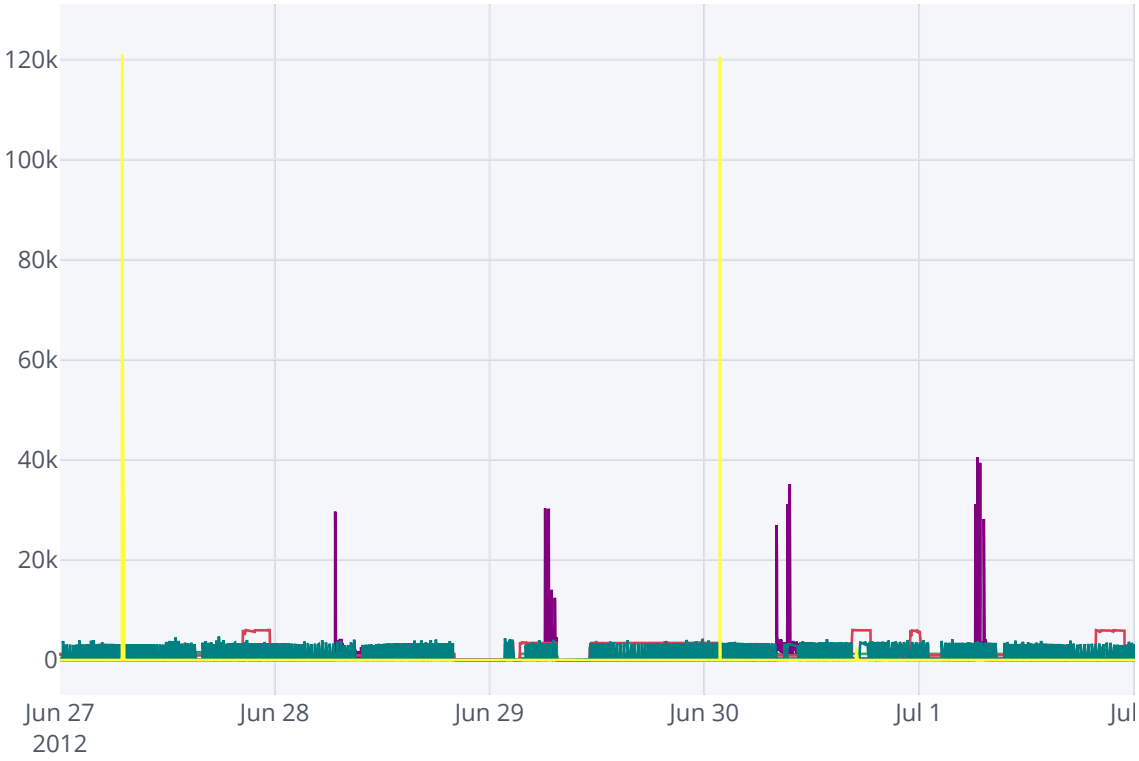
Out[58]:

	date	Unnamed: 0	Lamp	Laptop	Router	Coffee machine	Entertainment	Fridge	Kettle
0	2012-06-27 00:00:00	1770	0.0	325.01370	1190.1742	0.0	891.7332	24.18724	0.0
1	2012-06-27 00:01:00	5370	0.0	331.39203	1202.9680	0.0	1017.4737	19.78956	0.0
2	2012-06-27 00:02:00	8970	0.0	329.26592	1207.2326	0.0	1104.8532	24.18724	0.0
3	2012-06-27 00:03:00	12570	0.0	322.88759	1200.8357	0.0	889.6020	21.98840	0.0
4	2012-06-27 00:04:00	16170	0.0	339.89647	1205.1003	0.0	891.7332	13.19304	0.0

In [59]:
```
house6_by_mins[['date', 'Lamp', 'Laptop', 'Router', 'Coffee machine',
       'Entertainment', 'Fridge', 'Kettle']].iloc[0: 10000, :].iplot(kind =
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[59]:



In [60]:
```
group_15mins = house6.groupby(np.arange(len(house6))//900)
house6_by_15mins = pd.concat((group_15mins['date'].first(), group_15mins[[c
house6_by_15mins.head()
```

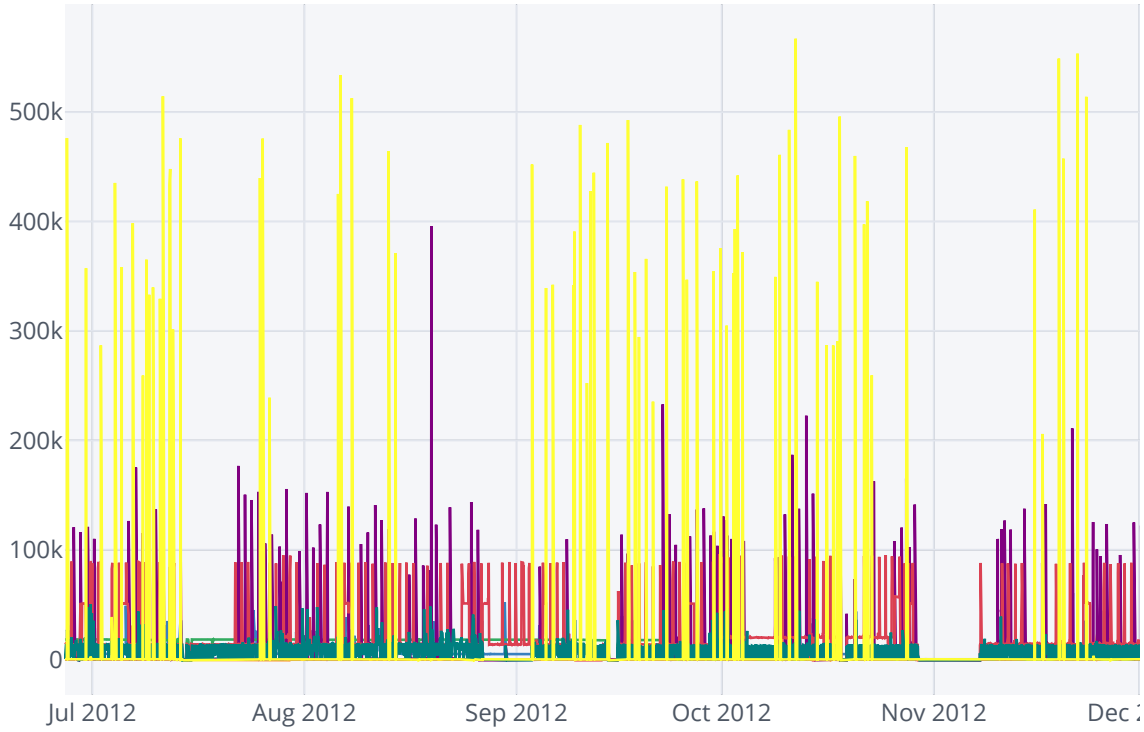
Out[60]:

	date	Unnamed: 0	Lamp	Laptop	Router	Coffee machine	Entertainment	Fridge	Kettle
0	2012-06-27 00:00:00	404550	0.0	4962.37601	18040.2554	0.0	13714.8585	294.64456	0.0
1	2012-06-27 00:15:00	1214550	0.0	4943.24102	18080.7691	0.0	13437.8028	12723.95104	0.0
2	2012-06-27 00:30:00	2024550	0.0	4924.10603	18061.5784	0.0	13576.3308	5344.43652	0.0
3	2012-06-27 00:45:00	2834550	0.0	4953.87157	18076.5045	0.0	13589.1180	7839.97874	0.0
4	2012-06-27 01:00:00	3644550	0.0	4921.97992	18114.8859	0.0	13616.8236	12930.33968	0.0

In [61]:
```
house6_by_15mins[['date', 'Lamp', 'Laptop', 'Router', 'Coffee machine',
       'Entertainment', 'Fridge', 'Kettle']].iplot(kind = "scatter", x = 'da
```

```
The draw time for this plot will be slow for all clients.

/Users/glance/anaconda3/lib/python3.6/site-packages/plotly/api/v1/clientr
esp.py:40: UserWarning:

Estimated Draw Time Too Long
```

Out[61]:



As a Data Scientist, I want to understand the time series algorithms and realize the differences between time series forecasting and traditional regression models.

To Do: Choose one feature from any house. Preferably the whole team should choose the same feature so as to have the same sample data to compare accuracy metrics. AR models MA Models ARIMA models Holt Winters exponential smoothing model

In [62]:
```
#import statsmodels as sm
#sm.tsa.ar_model.AR(np.asarray(house1), missing='drop')
#AR(house1['Kettle'], house1['date'])
```

In [63]:
```
#sm.tsa.arima_model.ARMA(house1, missing='drop', freq='D')
```

In [64]:
```
#sm.tsa.holtwinters.Holt(house1)
```

In [ ]:
```

```