Analysis of UG01 consumption characteristics

By: Mitchell Lee

Date: November 4, 2013

Import Necessary Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Open SharedSolar data analysis toolkit

In [2]:
```
run ../python/sd_data_stats.py
```

Import SharedSolar SD card data and remove obvious outliers

In [3]:
```
SD4 = pd.read_csv('../demand_data/drop_00_08_2013DF.csv', parse_dates = True, index_col = 0)
SD4[SD4>= 800] = np.nan
```

Make circuit to site dictionary

In [4]:
```
# add in ug02 mains so that script runs
SD4['ug02_0'] = np.nan

# make dictionary
site_dict = make_site_dict(SD4)
```

Isolate UG01 data and remove more outliers

- In rows where inverter limilations are exceeded, replace values with zero
- Only use most recent year of data (09-01-2012 to 08-31-2013)

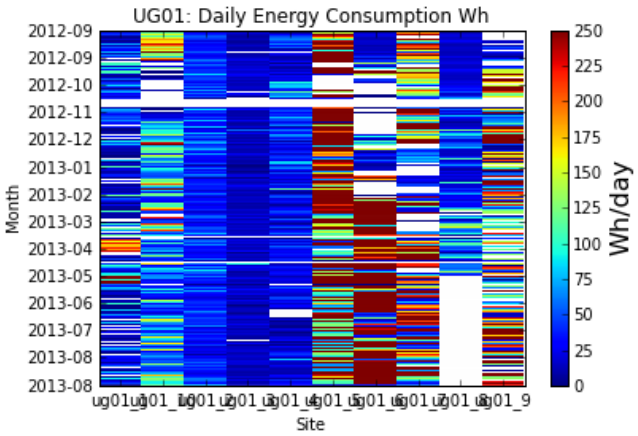
In [5]:
```
# isolate UG01 data
ug01 = SD4[site_dict['ug01']]['2012-09-01':'2013-08-31']
#Series of sum of circuits data
ug01_sum = ug01.sum(axis = 1)

# remove data that exceeds inverter limits
invert_lim = 750 # 750 W
[rows_lost, junk] = np.shape(ug01[ug01_sum >= invert_lim])
per_rows_lost = rows_lost/8760.
ug01.ix[ug01_sum >= invert_lim] =  0
```

Map UG01 Data Availability and Magnituded

Note x-axis labels are [ug01_1, ug01_10, ug01_2, ug01_3,..., ug01_9]

In [25]:
```
data_map_mag(ug01.resample('D',how = 'sum'),0,250)
plt.title('UG01: Daily Energy Consumption Wh')
# Note x-axis labels are [ug01_1, ug01_10, ug01_2, ug01_3,..., ug01_9]
```

Out[25]: <matplotlib.text.Text at 0xb666dcc>



Mean, standard deviation, and max daily energy consumption of each customer

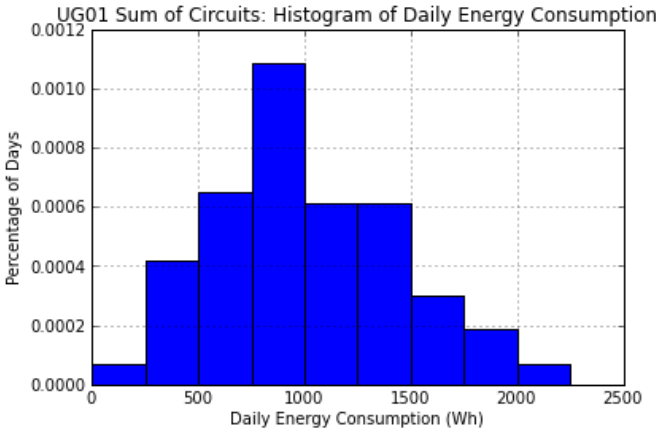
In [6]:
```
# mean daily energy consumption
ug01_mean = ug01.resample('D', how = 'sum').mean()
print "mean"
print ug01_mean
# standard deviation of daily energy consumption
ug01_std = ug01.resample('D', how = 'sum').std()
print "standard deviation"
print ug01_std
# max daily energy consumption
ug01_max =  ug01.resample('D', how = 'sum').max()
print "max"
print ug01_max
```

```
mean
ug01_1      47.414603
ug01_10     82.941824
ug01_2      39.426781
ug01_3      14.918841
ug01_4      30.601471
ug01_5     293.934650
ug01_6     301.489161
ug01_7     158.730100
ug01_8      37.466383
ug01_9     157.158885
standard deviation
ug01_1      59.289526
ug01_10     56.725256
ug01_2      15.956612
ug01_3       7.978078
ug01_4      22.979688
ug01_5     193.331283
ug01_6     306.139994
ug01_7     135.033899
ug01_8      30.906453
ug01_9     135.134309
max
ug01_1      374.3
ug01_10     292.4
ug01_2       95.4
ug01_3       95.6
ug01_4      133.4
ug01_5      889.7
ug01_6     1614.1
ug01_7      599.7
ug01_8      155.4
ug01_9      899.1
```

UG01: Sum of Circuits Historgram

In [16]:
```
ug01_sum.resample('D',how= 'sum').hist(bins = [0,250,500,750,1000,1250,1500,1750,2000,2250], normed =
plt.title('UG01 Sum of Circuits: Histogram of Daily Energy Consumption')
plt.xlabel('Daily Energy Consumption (Wh)')
plt.ylabel('Percentage of Days')
```

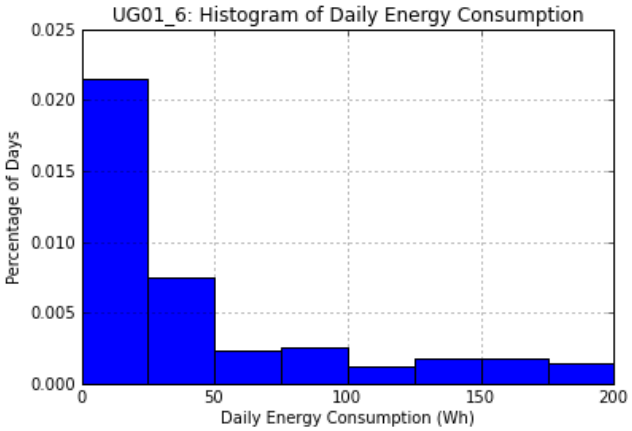
Out[16]: <matplotlib.text.Text at 0xb15bccc>



UG01_6 (Highest Consumption Consumer) Historgram

In [18]:
```
ug01['ug01_6'].resample('D',how= 'sum').hist(bins = [0,25,50,75,100,125,150,175,200], normed = True)
plt.title('UG01_6: Histogram of Daily Energy Consumption')
plt.xlabel('Daily Energy Consumption (Wh)')
plt.ylabel('Percentage of Days')
```

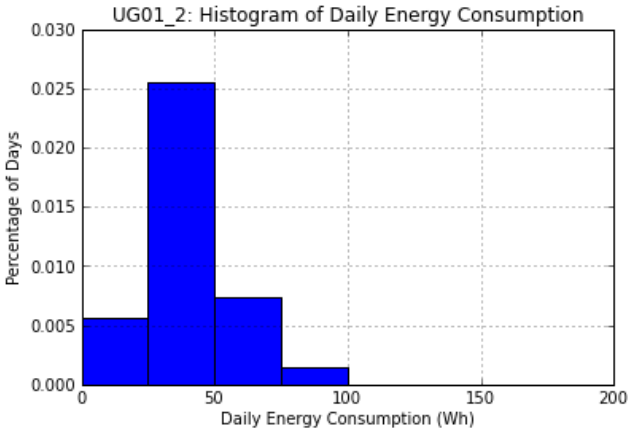
Out[18]: <matplotlib.text.Text at 0xb2dd10c>



UG01_2 (Lower-tier Consumption Consumer) Historgram

In [27]:
```
ug01['ug01_2'].resample('D',how= 'sum').hist(bins = [0,25,50,75,100,125,150,175,200], normed = True)
plt.title('UG01_2: Histogram of Daily Energy Consumption')
plt.xlabel('Daily Energy Consumption (Wh)')
plt.ylabel('Percentage of Days')
```

Out[27]: <matplotlib.text.Text at 0xc62956c>



In [ ]: