In [1]:
```
import pandas as pd
import re
import sys
```

In [5]:
```
#Replacing all underscores with spaces
df[0] = df[0].astype(str).str.replace('_', ' ').str.strip()
df
```

Out[5]:

	0
0	3 PM Mon 24th-Mar-2014 0.384 kwh
1	5AM 15-Aug-2014 1.201 kwh
2	8PM Thu 20-Mar-2014 1.523 kwh
3	6PM 23rd-Apr-2014 0.424 kwh
4	1AM Friday 19th-Dec-2014 0.209 kwh
...	...
8755	1AM Fri 07th-Nov-2014 0.084 kwh
8756	6AM 20-May-2014 1.027 kwh
8757	2 AM Tuesday 8th-Apr-2014 0.052 kwh
8758	9 PM 27th-Jan-2014 1.428 kwh
8759	12 PM 25th-Aug-2014 0.528 kwh

8760 rows × 1 columns

In [6]:
```

Time, Day, Date, Usage_kwh = [], [], [], []

for i in df[0]:
    time = re.findall("^\d+\ *[A-Z]+", i)
    Time.append(time)

    day = re.findall("[A-Z]+[a-z]+\s", i)
    Day.append(day)

    date = re.findall("\d+[a-z]*\-[A-Z]+[a-z]{2}\-[0-9]{4}", i)
    Date.append(date)

    usage = re.findall("[0-9]\.\d+", i)
    Usage_kwh.append(usage)

```

In [7]:

```
data_list = zip(Time, Day, Date, Usage_kwh)
column_names = ('time','day', 'date', 'usage')
electricity_df = pd.DataFrame(data_list, columns=column_names)

electricity_df
```

Out[7]:

	time	day	date	usage
0	[3 PM]	[Mon]	[24th-Mar-2014]	[0.384]
1	[5AM]	[]	[15-Aug-2014]	[1.201]
2	[8PM]	[Thu]	[20-Mar-2014]	[1.523]
3	[6PM]	[]	[23rd-Apr-2014]	[0.424]
4	[1AM]	[Friday]	[19th-Dec-2014]	[0.209]
...	...	...	...	...
8755	[1AM]	[Fri]	[07th-Nov-2014]	[0.084]
8756	[6AM]	[]	[20-May-2014]	[1.027]
8757	[2 AM]	[Tuesday]	[8th-Apr-2014]	[0.052]
8758	[9 PM]	[]	[27th-Jan-2014]	[1.428]
8759	[12 PM]	[]	[25th-Aug-2014]	[0.528]

8760 rows × 4 columns

In [8]:

```
time = electricity_df['time'].apply(pd.Series, index = ['time'])
day = electricity_df['day'].apply(pd.Series)
date =  electricity_df['date'].apply(pd.Series,  index = ['date'])
usage =  electricity_df['usage'].apply(pd.Series,  index = ['usage_kWh'])
```

In [9]:

```
df = pd.concat([time, day, date, usage], axis=1)
df.rename(columns={ df.columns[1]: "day" }, inplace = True)
df
```

Out[9]:

	time	day	date	usage_kWh
0	3 PM	Mon	24th-Mar-2014	0.384
1	5AM	NaN	15-Aug-2014	1.201
2	8PM	Thu	20-Mar-2014	1.523
3	6PM	NaN	23rd-Apr-2014	0.424
4	1AM	Friday	19th-Dec-2014	0.209
...	...	...	...	...
8755	1AM	Fri	07th-Nov-2014	0.084
8756	6AM	NaN	20-May-2014	1.027
8757	2 AM	Tuesday	8th-Apr-2014	0.052
8758	9 PM	NaN	27th-Jan-2014	1.428
8759	12 PM	NaN	25th-Aug-2014	0.528

8760 rows × 4 columns

In [10]:

```
df['time'] = df['time'].astype(str).str.replace(' ', '')
df
```

Out[10]:

	time	day	date	usage_kWh
0	3PM	Mon	24th-Mar-2014	0.384
1	5AM	NaN	15-Aug-2014	1.201
2	8PM	Thu	20-Mar-2014	1.523
3	6PM	NaN	23rd-Apr-2014	0.424
4	1AM	Friday	19th-Dec-2014	0.209
...	...	...	...	...
8755	1AM	Fri	07th-Nov-2014	0.084
8756	6AM	NaN	20-May-2014	1.027
8757	2AM	Tuesday	8th-Apr-2014	0.052
8758	9PM	NaN	27th-Jan-2014	1.428
8759	12PM	NaN	25th-Aug-2014	0.528

8760 rows × 4 columns

In [11]:

```
df[['Day','Months', 'Year']] = df.date.str.split("-",expand=True)
df
```

Out[11]:

	time	day	date	usage_kWh	Day	Months	Year
0	3PM	Mon	24th-Mar-2014	0.384	24th	Mar	2014
1	5AM	NaN	15-Aug-2014	1.201	15	Aug	2014
2	8PM	Thu	20-Mar-2014	1.523	20	Mar	2014
3	6PM	NaN	23rd-Apr-2014	0.424	23rd	Apr	2014
4	1AM	Friday	19th-Dec-2014	0.209	19th	Dec	2014
...	...	...	...	...	...	...	...
8755	1AM	Fri	07th-Nov-2014	0.084	07th	Nov	2014
8756	6AM	NaN	20-May-2014	1.027	20	May	2014
8757	2AM	Tuesday	8th-Apr-2014	0.052	8th	Apr	2014
8758	9PM	NaN	27th-Jan-2014	1.428	27th	Jan	2014
8759	12PM	NaN	25th-Aug-2014	0.528	25th	Aug	2014

8760 rows × 7 columns

In [12]:

```
df[['Time', 'AM/PM']] = df.time.str.split(r"[A-Z]{2}",expand=True)
df
```

Out[12]:

	time	day	date	usage_kWh	Day	Months	Year	Time	AM/PM
0	3PM	Mon	24th-Mar-2014	0.384	24th	Mar	2014	3	
1	5AM	NaN	15-Aug-2014	1.201	15	Aug	2014	5	
2	8PM	Thu	20-Mar-2014	1.523	20	Mar	2014	8	
3	6PM	NaN	23rd-Apr-2014	0.424	23rd	Apr	2014	6	
4	1AM	Friday	19th-Dec-2014	0.209	19th	Dec	2014	1	
...	...	...	...	...	...	...	...	...	...
8755	1AM	Fri	07th-Nov-2014	0.084	07th	Nov	2014	1	
8756	6AM	NaN	20-May-2014	1.027	20	May	2014	6	
8757	2AM	Tuesday	8th-Apr-2014	0.052	8th	Apr	2014	2	
8758	9PM	NaN	27th-Jan-2014	1.428	27th	Jan	2014	9	
8759	12PM	NaN	25th-Aug-2014	0.528	25th	Aug	2014	12	

8760 rows × 9 columns

In [13]:
```
df[[  'Time1', 'AM/PM']] = df.time.str.split(r"\d+",expand=True)
df
```

Out[13]:

	time	day	date	usage_kWh	Day	Months	Year	Time	AM/PM	Time1
0	3PM	Mon	24th-Mar-2014	0.384	24th	Mar	2014	3	PM	
1	5AM	NaN	15-Aug-2014	1.201	15	Aug	2014	5	AM	
2	8PM	Thu	20-Mar-2014	1.523	20	Mar	2014	8	PM	
3	6PM	NaN	23rd-Apr-2014	0.424	23rd	Apr	2014	6	PM	
4	1AM	Friday	19th-Dec-2014	0.209	19th	Dec	2014	1	AM	
...	...	...	...	...	...	...	...	...	...	...
8755	1AM	Fri	07th-Nov-2014	0.084	07th	Nov	2014	1	AM	
8756	6AM	NaN	20-May-2014	1.027	20	May	2014	6	AM	
8757	2AM	Tuesday	8th-Apr-2014	0.052	8th	Apr	2014	2	AM	
8758	9PM	NaN	27th-Jan-2014	1.428	27th	Jan	2014	9	PM	
8759	12PM	NaN	25th-Aug-2014	0.528	25th	Aug	2014	12	PM	

8760 rows × 10 columns

In [14]:
```
df.drop(columns =["time", "date", "Time1", "day"], inplace = True)
df
```

Out[14]:

	usage_kWh	Day	Months	Year	Time	AM/PM
0	0.384	24th	Mar	2014	3	PM
1	1.201	15	Aug	2014	5	AM
2	1.523	20	Mar	2014	8	PM
3	0.424	23rd	Apr	2014	6	PM
4	0.209	19th	Dec	2014	1	AM
...	...	...	...	...	...	...
8755	0.084	07th	Nov	2014	1	AM
8756	1.027	20	May	2014	6	AM
8757	0.052	8th	Apr	2014	2	AM
8758	1.428	27th	Jan	2014	9	PM
8759	0.528	25th	Aug	2014	12	PM

8760 rows × 6 columns

In [15]:

```
df.rename(columns={'Day': 'month_day', 'Months': 'months', 'Year': 'year', 'Ti
df
```

Out[15]:

	usage_kWh	month_day	months	year	time	am_pm
0	0.384	24th	Mar	2014	3	PM
1	1.201	15	Aug	2014	5	AM
2	1.523	20	Mar	2014	8	PM
3	0.424	23rd	Apr	2014	6	PM
4	0.209	19th	Dec	2014	1	AM
...	...	...	...	...	...	...
8755	0.084	07th	Nov	2014	1	AM
8756	1.027	20	May	2014	6	AM
8757	0.052	8th	Apr	2014	2	AM
8758	1.428	27th	Jan	2014	9	PM
8759	0.528	25th	Aug	2014	12	PM

8760 rows × 6 columns

In [16]:

```
df['month_day'] = df['month_day'].astype(str).str.replace(r'[a-z]{2}', '')
df
```

Out[16]:

	usage_kWh	month_day	months	year	time	am_pm
0	0.384	24	Mar	2014	3	PM
1	1.201	15	Aug	2014	5	AM
2	1.523	20	Mar	2014	8	PM
3	0.424	23	Apr	2014	6	PM
4	0.209	19	Dec	2014	1	AM
...	...	...	...	...	...	...
8755	0.084	07	Nov	2014	1	AM
8756	1.027	20	May	2014	6	AM
8757	0.052	8	Apr	2014	2	AM
8758	1.428	27	Jan	2014	9	PM
8759	0.528	25	Aug	2014	12	PM

8760 rows × 6 columns

In [17]:
```
df['month_day'] = df['month_day'].apply(lambda x: x.zfill(2))
df
```

Out[17]:

	usage_kWh	month_day	months	year	time	am_pm
0	0.384	24	Mar	2014	3	PM
1	1.201	15	Aug	2014	5	AM
2	1.523	20	Mar	2014	8	PM
3	0.424	23	Apr	2014	6	PM
4	0.209	19	Dec	2014	1	AM
...	...	...	...	...	...	...
8755	0.084	07	Nov	2014	1	AM
8756	1.027	20	May	2014	6	AM
8757	0.052	08	Apr	2014	2	AM
8758	1.428	27	Jan	2014	9	PM
8759	0.528	25	Aug	2014	12	PM

8760 rows × 6 columns

In [18]:
```
df = df[['time', 'am_pm', 'month_day', 'months', 'year', 'usage_kWh']]
df
```

Out[18]:

	time	am_pm	month_day	months	year	usage_kWh
0	3	PM	24	Mar	2014	0.384
1	5	AM	15	Aug	2014	1.201
2	8	PM	20	Mar	2014	1.523
3	6	PM	23	Apr	2014	0.424
4	1	AM	19	Dec	2014	0.209
...	...	...	...	...	...	...
8755	1	AM	07	Nov	2014	0.084
8756	6	AM	20	May	2014	1.027
8757	2	AM	08	Apr	2014	0.052
8758	9	PM	27	Jan	2014	1.428
8759	12	PM	25	Aug	2014	0.528

8760 rows × 6 columns

In [19]:

```
df['time'] = df['time'].apply(lambda x: x.zfill(2))
df
```

Out[19]:

```
      time  am_pm  month_day  months  year  usage_kWh
   0    03     PM         24     Mar  2014      0.384
   1    05     AM         15     Aug  2014      1.201
   2    08     PM         20     Mar  2014      1.523
   3    06     PM         23     Apr  2014      0.424
   4    01     AM         19     Dec  2014      0.209
 ...   ...    ...        ...     ...   ...        ...
8755    01     AM         07     Nov  2014      0.084
8756    06     AM         20     May  2014      1.027
8757    02     AM         08     Apr  2014      0.052
8758    09     PM         27     Jan  2014      1.428
8759    12     PM         25     Aug  2014      0.528

8760 rows × 6 columns
```

In [20]:

```
df['date']=df['month_day'].astype(str)+'-'+df['months']+'-'+df['year']
df
```

Out[20]:

```
      time  am_pm  month_day  months  year  usage_kWh         date
   0    03     PM         24     Mar  2014      0.384  24-Mar-2014
   1    05     AM         15     Aug  2014      1.201  15-Aug-2014
   2    08     PM         20     Mar  2014      1.523  20-Mar-2014
   3    06     PM         23     Apr  2014      0.424  23-Apr-2014
   4    01     AM         19     Dec  2014      0.209  19-Dec-2014
 ...   ...    ...        ...     ...   ...        ...          ...
8755    01     AM         07     Nov  2014      0.084  07-Nov-2014
8756    06     AM         20     May  2014      1.027  20-May-2014
8757    02     AM         08     Apr  2014      0.052  08-Apr-2014
8758    09     PM         27     Jan  2014      1.428  27-Jan-2014
8759    12     PM         25     Aug  2014      0.528  25-Aug-2014

8760 rows × 7 columns
```

In [21]:
```
df['date'] = pd.to_datetime(df['date'])

df['day_of_week'] = df['date'].dt.day_name()

df
```

Out[21]:

	time	am_pm	month_day	months	year	usage_kWh	date	day_of_week
0	03	PM	24	Mar	2014	0.384	2014-03-24	Monday
1	05	AM	15	Aug	2014	1.201	2014-08-15	Friday
2	08	PM	20	Mar	2014	1.523	2014-03-20	Thursday
3	06	PM	23	Apr	2014	0.424	2014-04-23	Wednesday
4	01	AM	19	Dec	2014	0.209	2014-12-19	Friday
...	...	...	...	...	...	...	...	...
8755	01	AM	07	Nov	2014	0.084	2014-11-07	Friday
8756	06	AM	20	May	2014	1.027	2014-05-20	Tuesday
8757	02	AM	08	Apr	2014	0.052	2014-04-08	Tuesday
8758	09	PM	27	Jan	2014	1.428	2014-01-27	Monday
8759	12	PM	25	Aug	2014	0.528	2014-08-25	Monday

8760 rows × 8 columns

In [22]:
```
df.sort_values(by = ['date', 'am_pm', 'time'], inplace = True)
```

In [23]:
```
df.to_csv('electricity.csv', index=False)
```

In [24]:
```
df = pd.read_csv('electricity.csv')
df
```

Out[24]:

	time	am_pm	month_day	months	year	usage_kWh	date	day_of_week
0	1	AM	1	Jan	2014	0.340	2014-01-01	Wednesday
1	2	AM	1	Jan	2014	0.332	2014-01-01	Wednesday
2	3	AM	1	Jan	2014	0.396	2014-01-01	Wednesday
3	4	AM	1	Jan	2014	0.388	2014-01-01	Wednesday
4	5	AM	1	Jan	2014	2.354	2014-01-01	Wednesday
...	...	...	...	...	...	...	...	...
8755	8	PM	31	Dec	2014	1.901	2014-12-31	Wednesday
8756	9	PM	31	Dec	2014	0.766	2014-12-31	Wednesday
8757	10	PM	31	Dec	2014	0.633	2014-12-31	Wednesday
8758	11	PM	31	Dec	2014	0.600	2014-12-31	Wednesday
8759	12	PM	31	Dec	2014	0.528	2014-12-31	Wednesday

8760 rows × 8 columns

Question 1

What is your average hourly electricity usage?

a. 0.641kWh

b. 0.782kWh

c. 0.884kWh

d. 0.937kWh

Answer: b

In [25]:
```
round(df["usage_kWh"].mean(), 3)
```

Out[25]:
```
0.782
```

Question 2

What is your average electricity usage per hour in February?

a. 0.760kWh

b. 0.784kWh

c. 0.808kWh

d. 0.833kWh

Answer: d

```
In [26]: df.groupby('months')['usage_kWh'].mean().sort_values()
```

```
Out[26]: months
         Apr    0.210058
         Nov    0.322806
         Oct    0.421665
         Sep    0.485675
         May    0.560695
         Mar    0.565522
         Dec    0.764434
         Feb    0.832915
         Aug    0.851331
         Jun    1.255899
         Jan    1.401425
         Jul    1.687532
         Name: usage_kWh, dtype: float64
```

Question 3

Which day of the week has the highest average usage?

a. Sunday

b. Monday

c. Tuesday

d. Wednesday

Answer: a

```
In [27]: df.groupby('day_of_week')['usage_kWh'].mean().sort_values()
```

```
Out[27]: day_of_week
         Monday       0.725945
         Tuesday      0.726058
         Wednesday    0.733943
         Thursday     0.745458
         Friday       0.753744
         Saturday     0.890764
         Sunday       0.898157
         Name: usage_kWh, dtype: float64
```

Question 4

What is the highest amount of electricity used in a continuous 4 hour period?

a. 17.237kWh

b. 17.327kWh

c. 17.422kWh

d. 17.487kWh

Answer: a

```
In [28]: df['usage_kWh'].rolling(4).sum().max()
```

```
Out[28]: 17.236999999999952
```

Question 5

Based on your historic electricity usage, what would your annual cost of electricity be under the "Monthly Flex" contract?

a. 1350.73

b. 1421.82

c. 1450.26

d. 1493.77

Answer: b

Question 6

Based on your historic electricity usage, which of the three contracts would produce the lowest annual cost?

a. The No Flex plan

b. The Monthly Flex plan

c. The Hourly Flex plan

d. Impossible to Determine

Answer: c