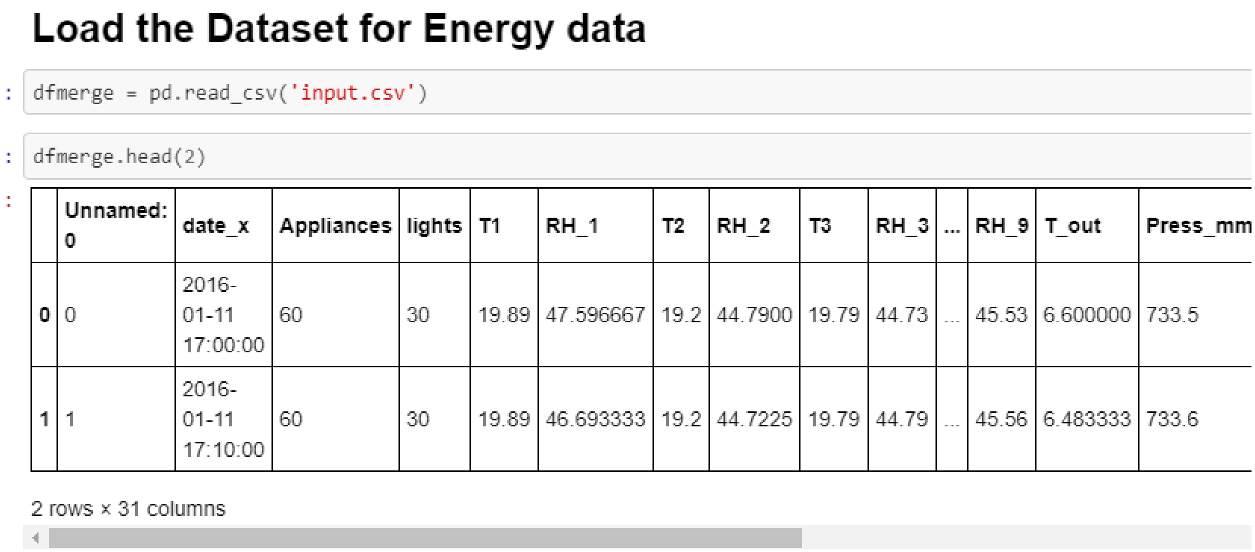
Capstone Project 1: Data Visualization

Appliances Energy Prediction –

Business Problem Description – Dataset contains the house temperature and humidity conditions were monitored with a ZigBee wireless sensor network. As per the description on UCI website, each wireless node transmitted the temperature and humidity conditions around 3.3 min, then, the wireless data was averaged for 10 minutes periods. The energy data was logged every 10 minutes with m-bus energy meters. Combining this data with the weather data based on the date time columns

Link for the Jupyter file -
https://github.com/arijitsinha80/Springboard/blob/master/Project/Capstoneproject_Phase2.ipynb

From the Data Wrangling activity, we created the **input.csv** as the final dataset.



Load the Dataset for Energy data

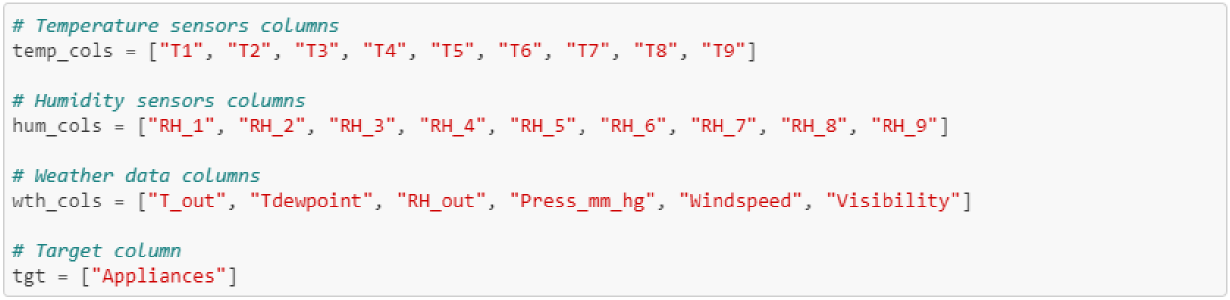
```
dfmerge = pd.read_csv('input.csv')
```

```
dfmerge.head(2)
```

	Unnamed: 0	date_x	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	...	RH_9	T_out	Press_mm
0	0	2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.7900	19.79	44.73	...	45.53	6.600000	733.5
1	1	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.7225	19.79	44.79	...	45.56	6.483333	733.6

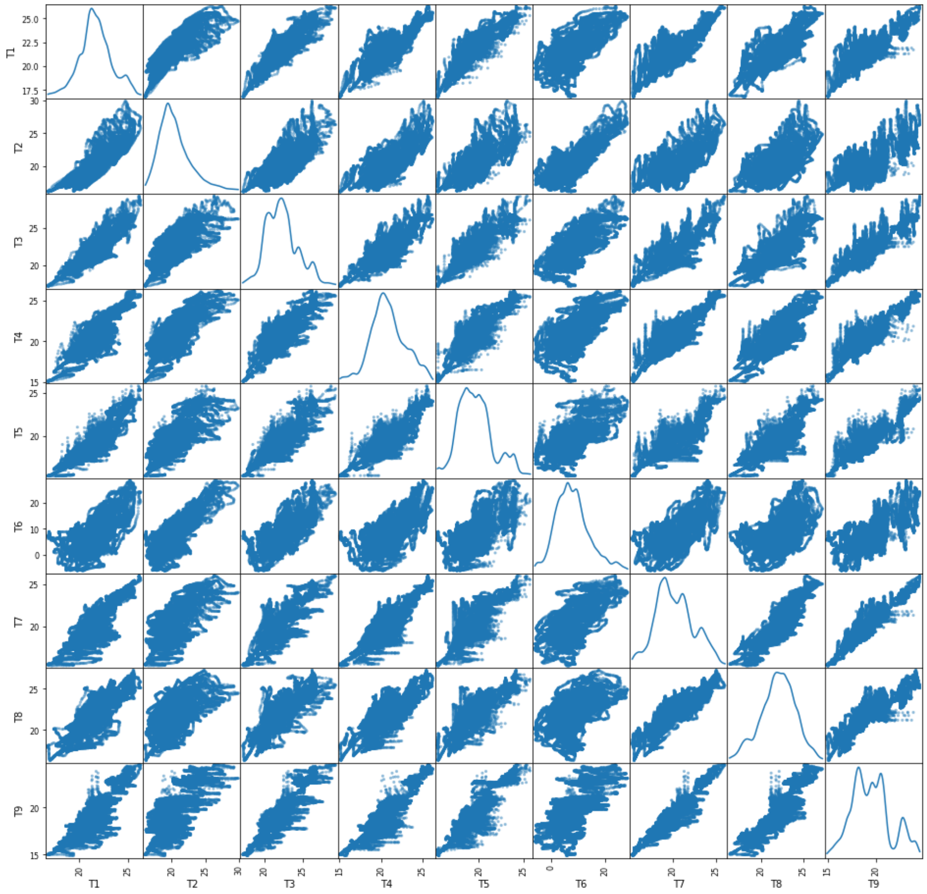
2 rows × 31 columns

Divide the data in dimension wise to explore –

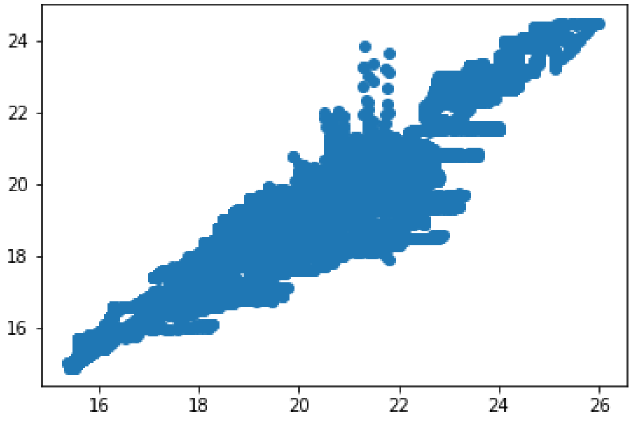


```
# Temperature sensors columns
temp_cols = ["T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"]

# Humidity sensors columns
hum_cols = ["RH_1", "RH_2", "RH_3", "RH_4", "RH_5", "RH_6", "RH_7", "RH_8", "RH_9"]

# Weather data columns
wth_cols = ["T_out", "Tdewpoint", "RH_out", "Press_mm_hg", "Windspeed", "Visibility"]

# Target column
tgt = ["Appliances"]
```

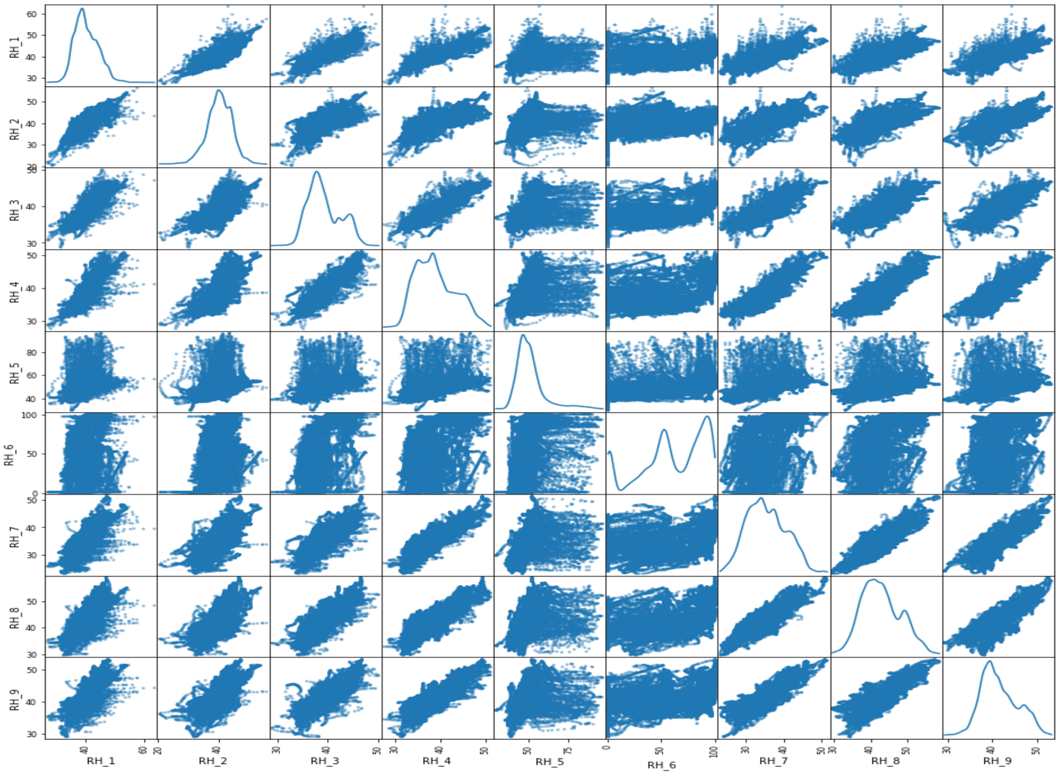
From the above, we can see that there is some linear relation between T7 and T9. Others are having the shape but are not exactly linear.



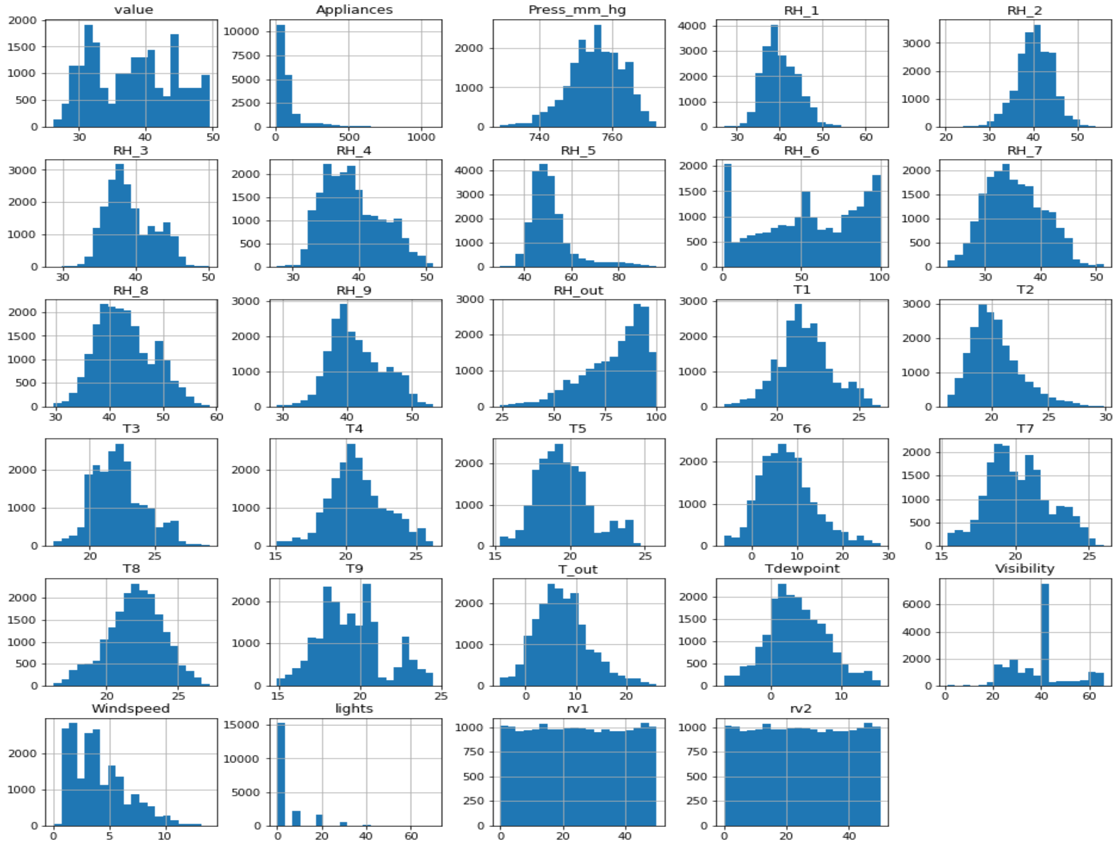
There is a definetly a relation between these two attributes but also have some outliers



Explore the data using the Weather data



Lets explore the distribution using the histogram -

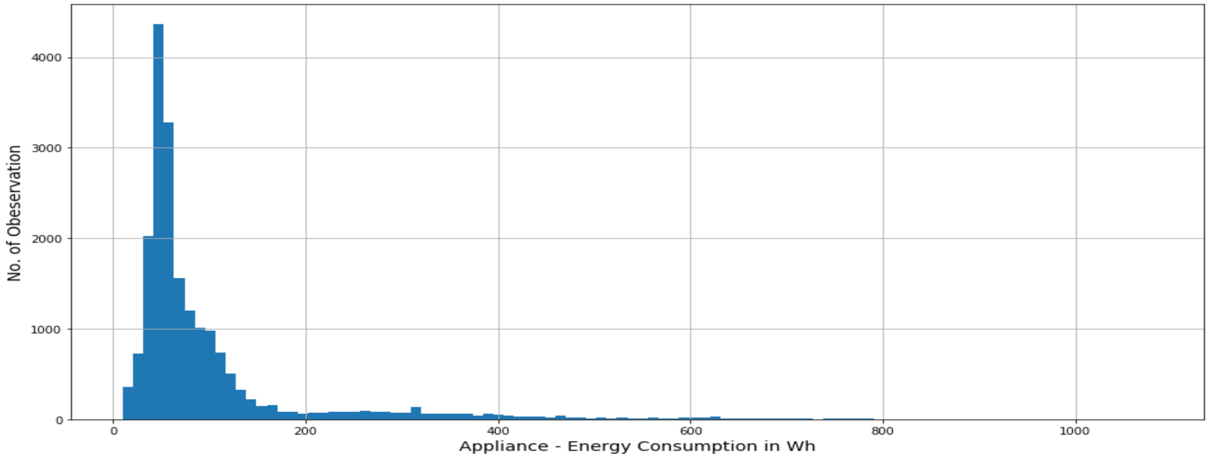


All humidity values are almost having normal distribution except RH_6 and RH_out. In other words the reading from inside the home is having normal distribution.

All temperature readings follow a Normal distribution except for T9.

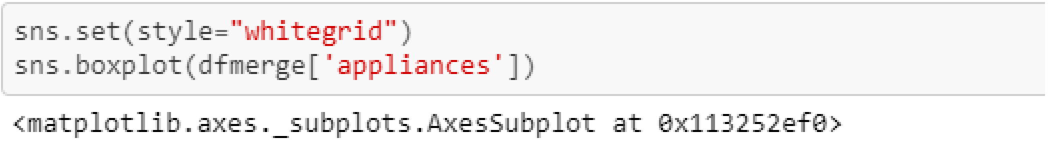
Visibility, Windspeed and Appliances are having skewed data.

On the Target –



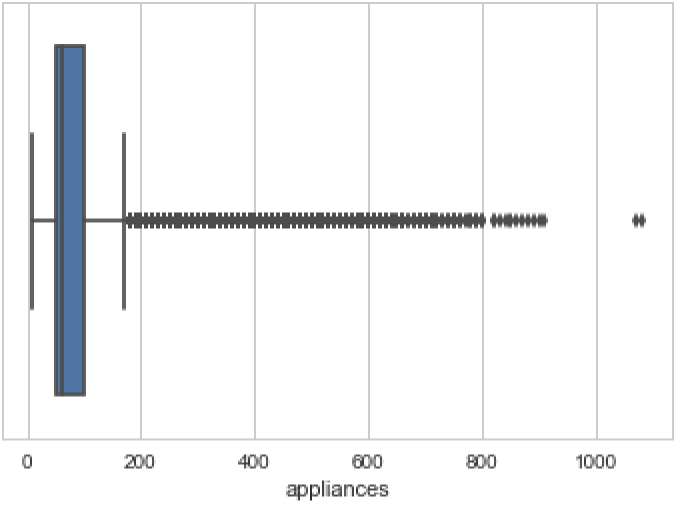
Target variable, Appliances is highly right skewed.

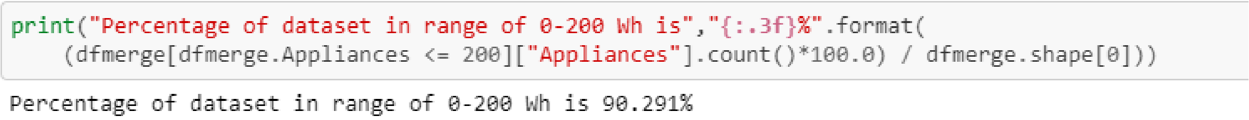
Checking with Boxplot –



```
sns.set(style="whitegrid")
sns.boxplot(dfmerge['appliances'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x113252ef0>
```



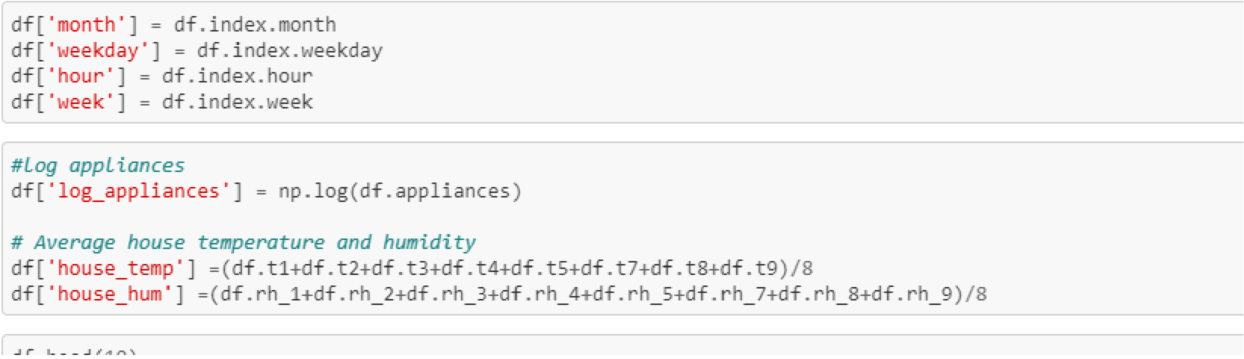


```python
print("Percentage of dataset in range of 0-200 Wh is","{:.3f}%".format(
    (dfmerge[dfmerge.Appliances <= 200]["Appliances"].count()*100.0) / dfmerge.shape[0]))
```

```
Percentage of dataset in range of 0-200 Wh is 90.291%
```

Percentage of dataset in range of 0-200 Wh is 90.291%

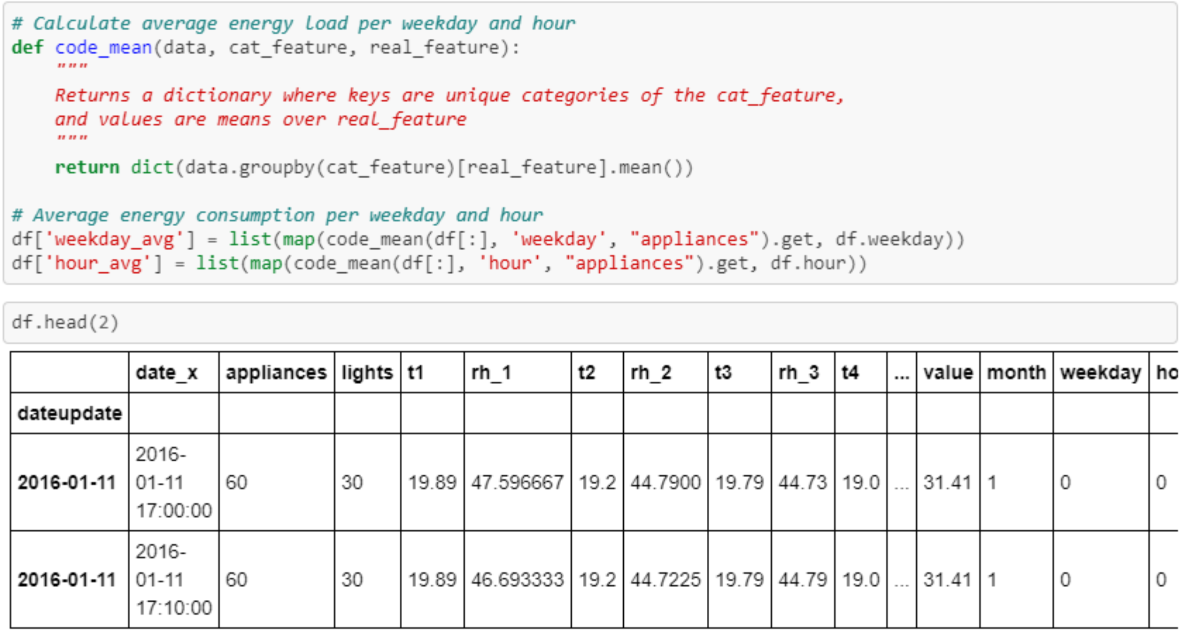
Create new columns for Month and Weeks



```python
df['month'] = df.index.month
df['weekday'] = df.index.weekday
df['hour'] = df.index.hour
df['week'] = df.index.week
```

```python
#log appliances
df['log_appliances'] = np.log(df.appliances)

# Average house temperature and humidity
df['house_temp'] =(df.t1+df.t2+df.t3+df.t4+df.t5+df.t7+df.t8+df.t9)/8
df['house_hum'] =(df.rh_1+df.rh_2+df.rh_3+df.rh_4+df.rh_5+df.rh_7+df.rh_8+df.rh_9)/8
```

```python
df.head(10)
```

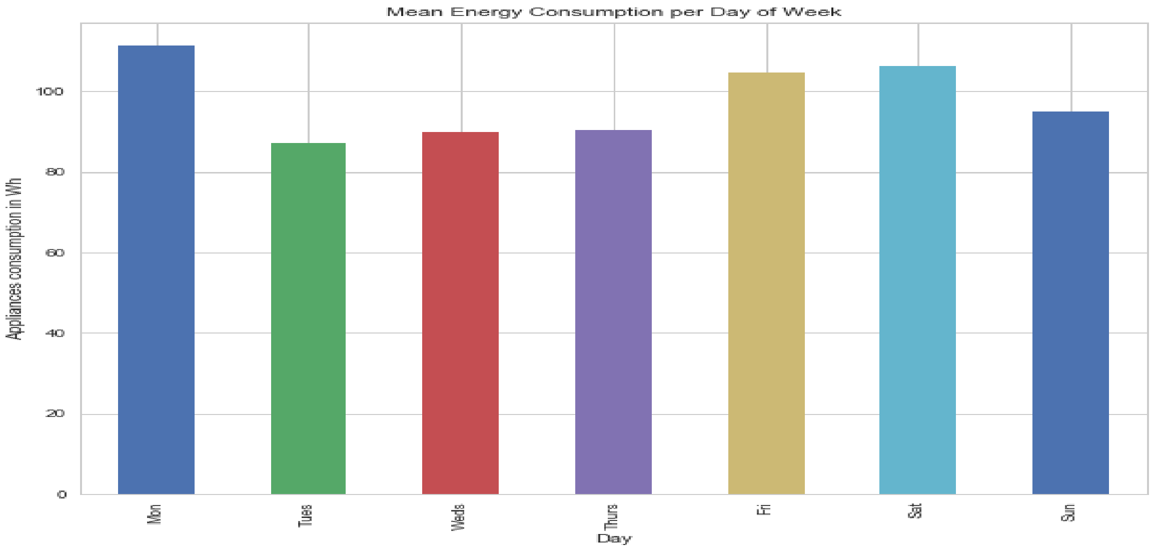


```python
# Calculate average energy load per weekday and hour
def code_mean(data, cat_feature, real_feature):
    """
    Returns a dictionary where keys are unique categories of the cat_feature,
    and values are means over real_feature
    """
    return dict(data.groupby(cat_feature)[real_feature].mean())

# Average energy consumption per weekday and hour
df['weekday_avg'] = list(map(code_mean(df[:], 'weekday', "appliances").get, df.weekday))
df['hour_avg'] = list(map(code_mean(df[:], 'hour', "appliances").get, df.hour))
```

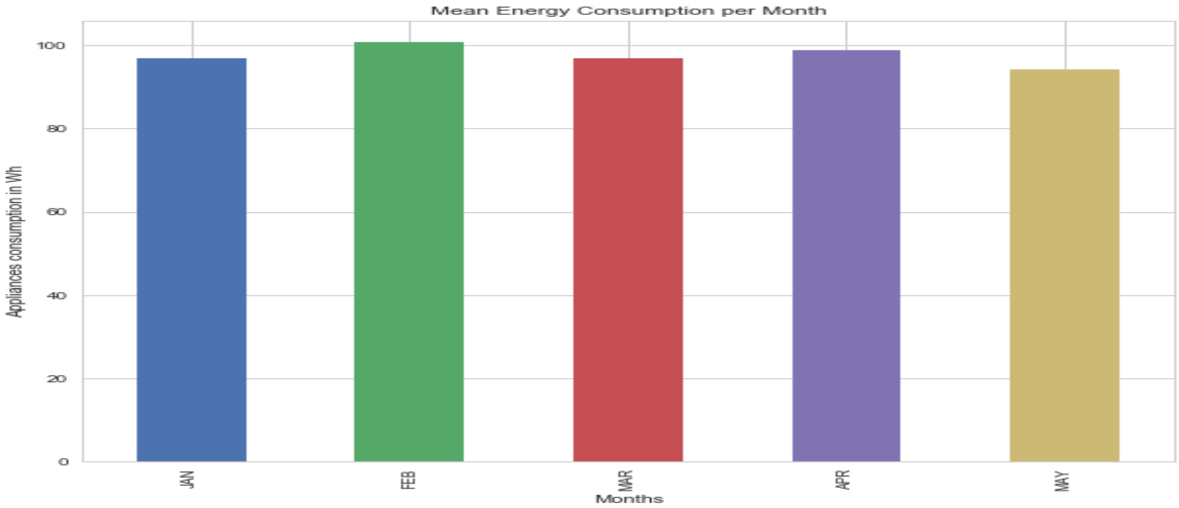
```python
df.head(2)
```

	date_x	appliances	lights	t1	rh_1	t2	rh_2	t3	rh_3	t4	...	value	month	weekday	ho
dateupdate															
2016-01-11	2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.7900	19.79	44.73	19.0	...	31.41	1	0	0
2016-01-11	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.7225	19.79	44.79	19.0	...	31.41	1	0	0

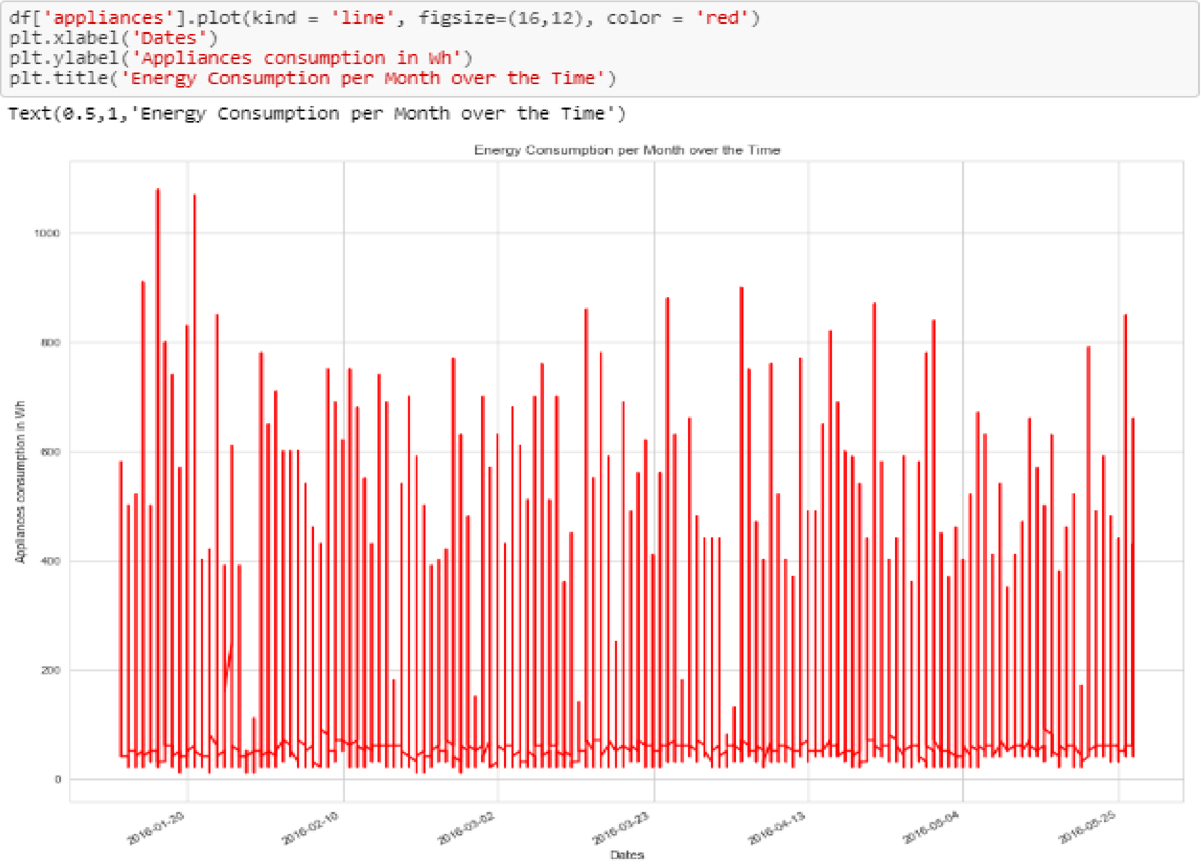
With taking the average on week –



For Monthly Average –



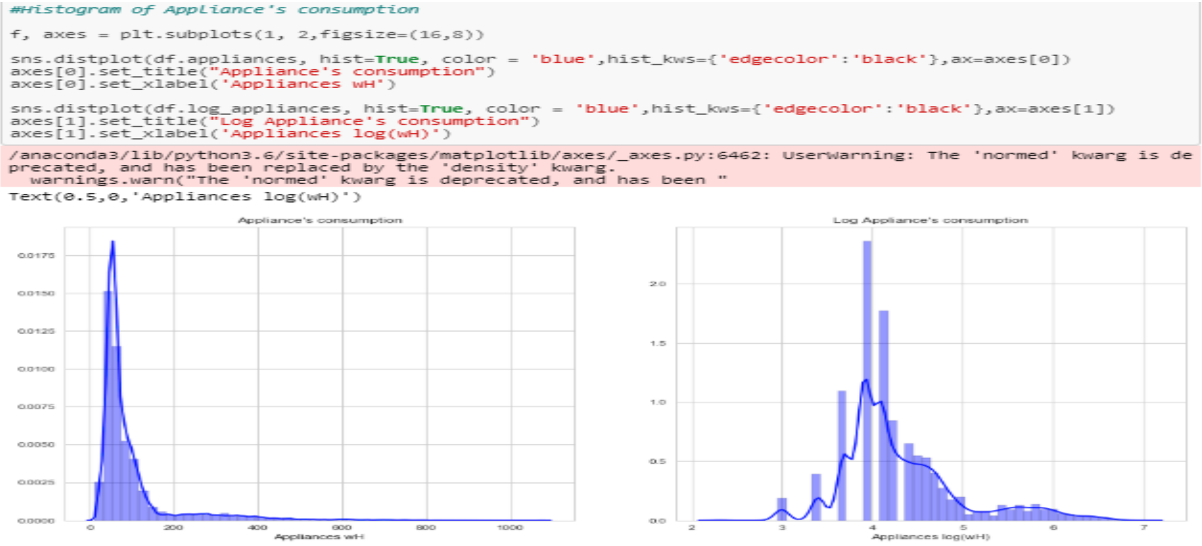
For Day wise comsumption –



```
df['appliances'].plot(kind = 'line', figsize=(16,12), color = 'red')
plt.xlabel('Dates')
plt.ylabel('Appliances consumption in Wh')
plt.title('Energy Consumption per Month over the Time')
```

```
Text(0.5,1,'Energy Consumption per Month over the Time')
```

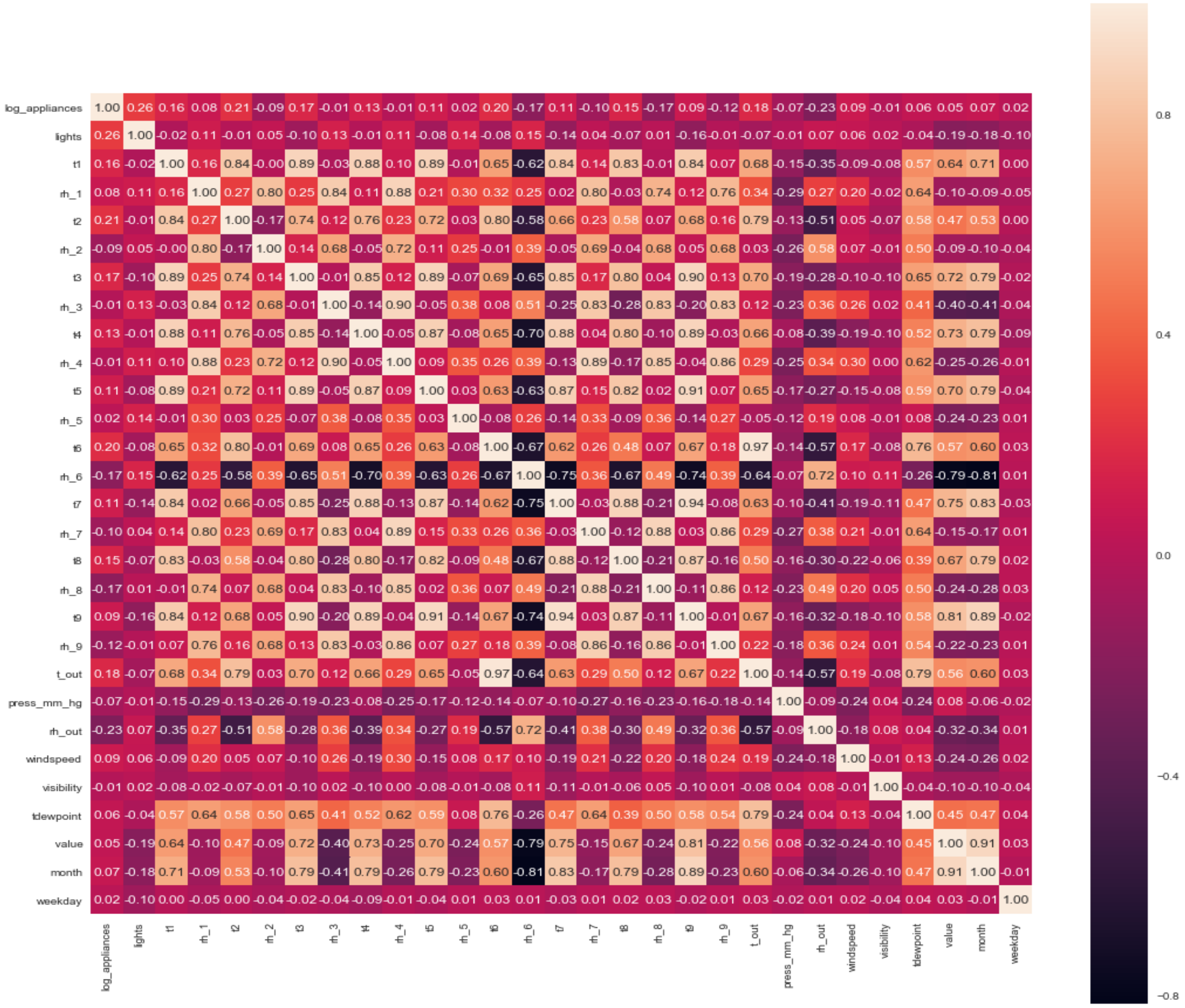
Since the Appliance data captured were rightly skewed, converting the column to Log values to see if it has the normal distribution



```
#Histogram of Appliance's consumption

f, axes = plt.subplots(1, 2,figsize=(16,8))

sns.distplot(df.appliances, hist=True, color = 'blue',hist_kws={'edgecolor':'black'},ax=axes[0])
axes[0].set_title("Appliance's consumption")
axes[0].set_xlabel('Appliances wH')

sns.distplot(df.log_appliances, hist=True, color = 'blue',hist_kws={'edgecolor':'black'},ax=axes[1])
axes[1].set_title("Log Appliance's consumption")
axes[1].set_xlabel('Appliances log(wH)')
```

```
/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: The 'normed' kwarg is de
precated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
Text(0.5,0,'Appliances log(wH)')
```

Let’s explore the Correlation plot –



The most correlated features with energy consumtion(log_appliances) are: lights=0.26, t6=0.20, t2=0.22, t3 = 0.17,t_out = 0.18, rh_out = -0.23, rh_8 = -0.17, rh_6 = -0.17, windspeed = 0.09.

In a linear regression problem only linear independent variables can be be used as features to explain energy consumption otherwise we will have multicolinearity issues.