In [122]:
```
import pandas as pd
import numpy as np
```

In [123]:
```
df = pd.read_csv('household_power_consumption.txt', sep=';',
                 parse_dates={'dt' : ['Date', 'Time']}, infer_datetime_format=True,
                 low_memory=False, na_values=['nan','?'], index_col='dt')
```

In [124]:
```
df.head()
```

Out[124]:

dt	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2006-12-16 17:24:00	4.216	0.418	234.84	18.4	0.0	1.0	17.0
2006-12-16 17:25:00	5.360	0.436	233.63	23.0	0.0	1.0	16.0
2006-12-16 17:26:00	5.374	0.498	233.29	23.0	0.0	2.0	17.0
2006-12-16 17:27:00	5.388	0.502	233.74	23.0	0.0	1.0	17.0
2006-12-16 17:28:00	3.666	0.528	235.68	15.8	0.0	1.0	17.0

In [125]:
```
df.dtypes
```

Out[125]:
```
Global_active_power      float64
Global_reactive_power    float64
Voltage                  float64
Global_intensity         float64
Sub_metering_1           float64
Sub_metering_2           float64
Sub_metering_3           float64
dtype: object
```

In [126]:
```
for j in range(0,7):
        df.iloc[:,j]=df.iloc[:,j].fillna(df.iloc[:,j].mean())
```

In [127]:
```
df.isnull().sum()
```

Out[127]:
```
Global_active_power      0
Global_reactive_power    0
Voltage                  0
Global_intensity         0
Sub_metering_1           0
Sub_metering_2           0
Sub_metering_3           0
dtype: int64
```

In [128]:
```
df['Global_active_power'].resample('M').sum() # here we are caluclating the sum of global active power for each
```

Out[128]:

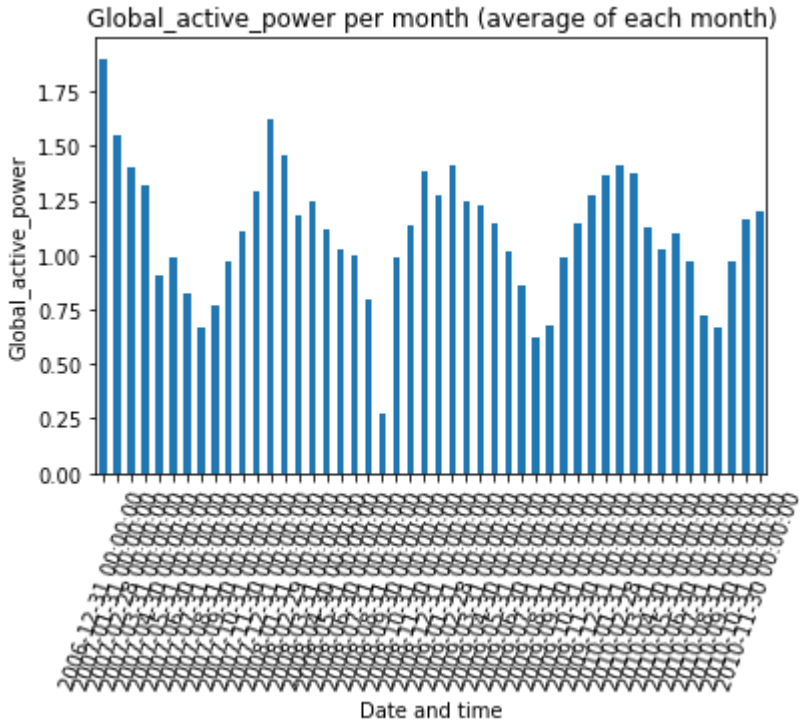
dt	
2006-12-31	41817.648460
2007-01-31	69014.045230
2007-02-28	56491.069230
2007-03-31	58863.283615
2007-04-30	39245.548781
2007-05-31	44008.872000
2007-06-30	35729.767447
2007-07-31	29846.831570
2007-08-31	34120.475531
2007-09-30	41874.789230
2007-10-31	49278.553230
2007-11-30	55920.827230
2007-12-31	72605.261615
2008-01-31	65170.473615
2008-02-29	49334.346845
2008-03-31	55591.685615
2008-04-30	48209.992000
2008-05-31	45724.043230
2008-06-30	42945.063615
2008-07-31	35479.601230
2008-08-31	12344.063230
2008-09-30	42667.792000
2008-10-31	50743.399447
2008-11-30	59918.584535
2008-12-31	56911.416668
2009-01-31	62951.099615
2009-02-28	50291.953362
2009-03-31	54761.169230
2009-04-30	49277.707230
2009-05-31	45214.196460
2009-06-30	37149.767696
2009-07-31	27594.810460
2009-08-31	30049.032998
2009-09-30	42631.838845
2009-10-31	51089.811615
2009-11-30	55068.733615
2009-12-31	60907.189230
2010-01-31	62797.504679

```
2010-02-28    55473.889230
2010-03-31    50368.601679
2010-04-30    44379.215615
2010-05-31    48893.491615
2010-06-30    41887.607230
2010-07-31    32188.843615
2010-08-31    29991.384254
2010-09-30    42026.211946
2010-10-31    51934.045615
2010-11-30    44598.388000
Freq: M, Name: Global_active_power, dtype: float64
```

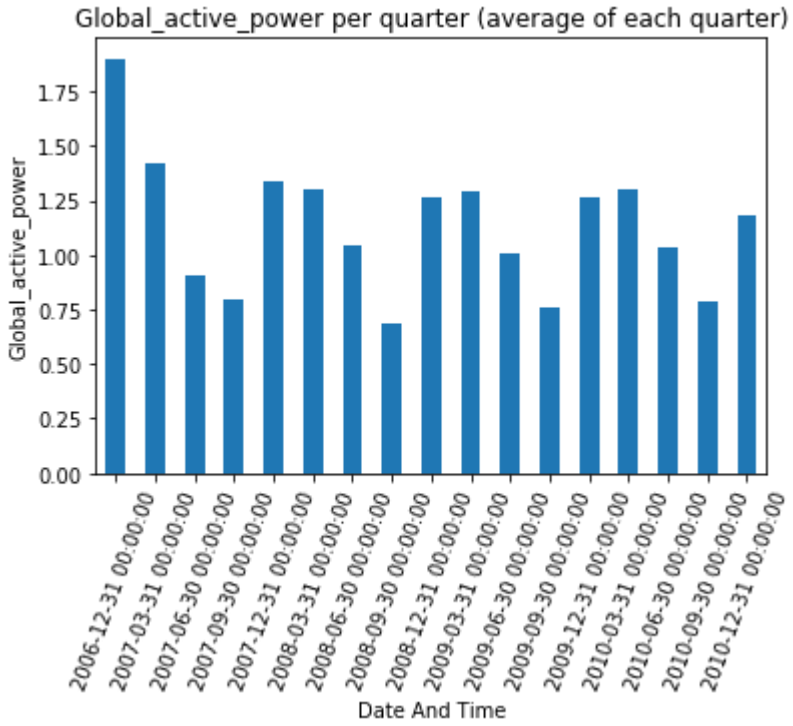
DATA VISULIZATION

Analysing the Global active power over a month and quarter

In [129]:
```
import matplotlib.pyplot as plt
df['Global_active_power'].resample('M').mean().plot(kind='bar')
plt.xticks(rotation=70)
plt.xlabel('Date and time')
plt.ylabel('Global_active_power')
plt.title('Global_active_power per month (average of each month)')
plt.show()
```

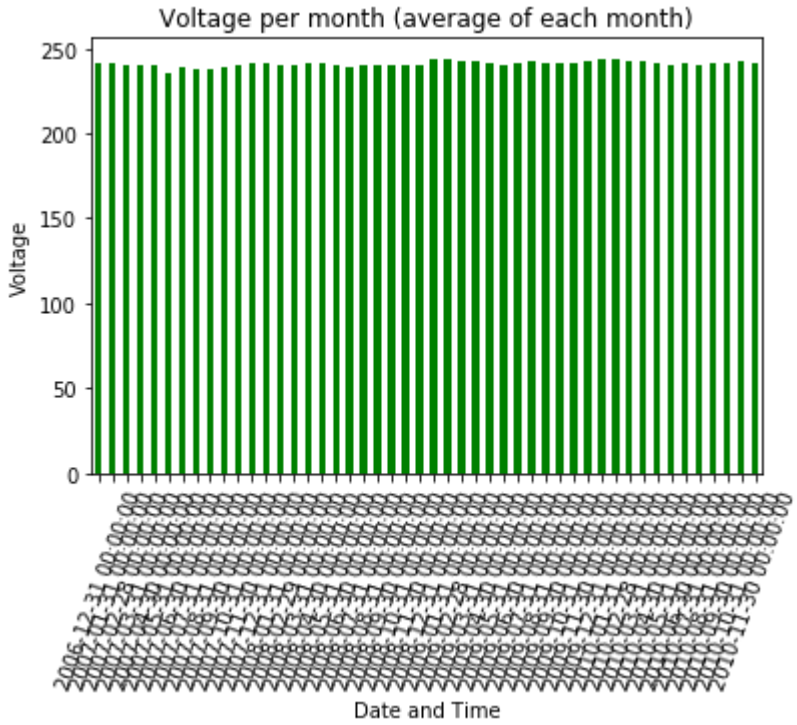


In [130]:
```
df['Global_active_power'].resample('Q').mean().plot(kind='bar')
plt.xticks(rotation=70)
plt.xlabel('Date And Time')
plt.ylabel('Global_active_power')
plt.title('Global_active_power per quarter (average of each quarter)')
plt.show()
```

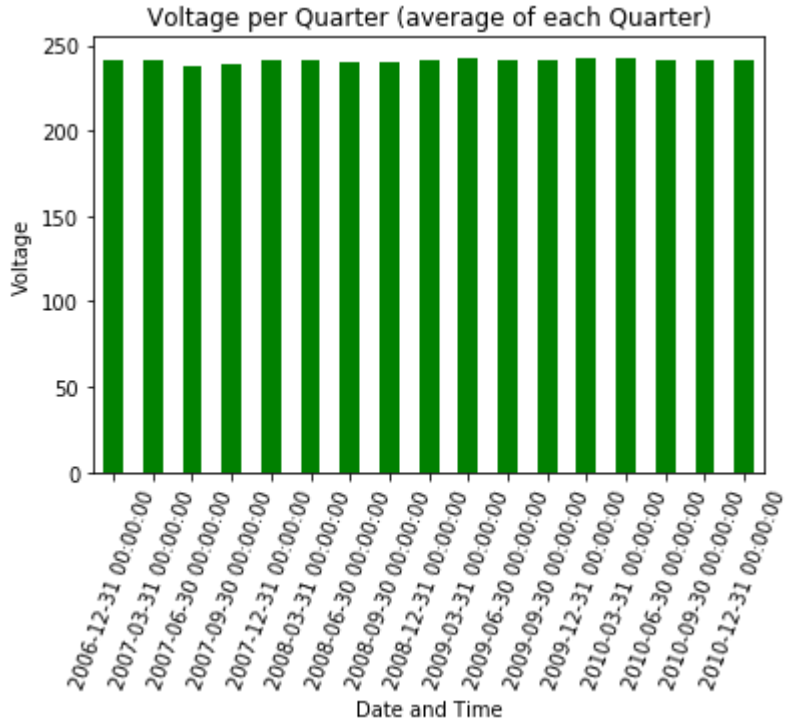


Analysing the voltage over a Month and quarter

In [131]:
```
df['Voltage'].resample('M').mean().plot(kind='bar', color='green')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Voltage')
plt.title('Voltage per month (average of each month)')
plt.show()
```



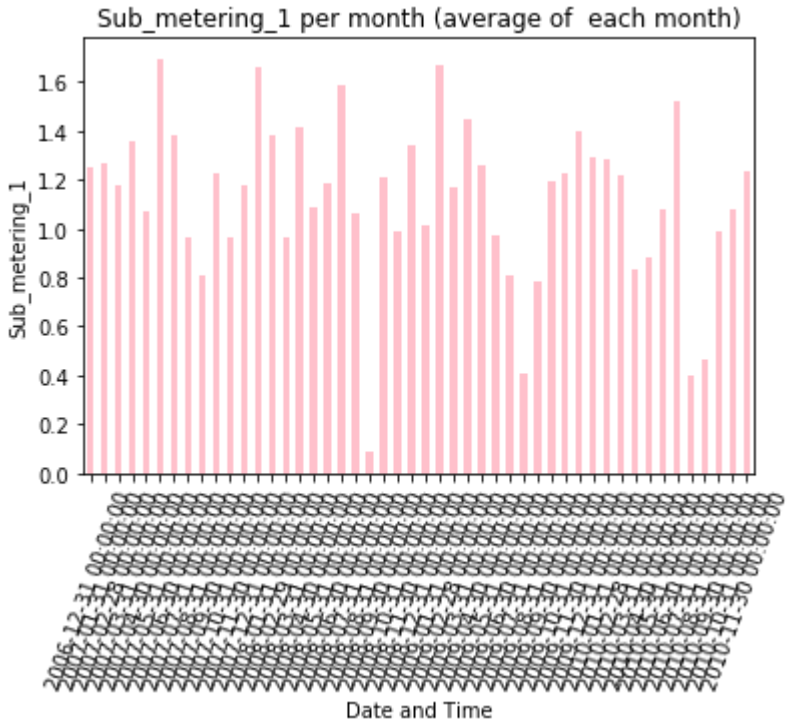
In [132]:
```
df['Voltage'].resample('Q').mean().plot(kind='bar', color='green')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Voltage')
plt.title('Voltage per Quarter (average of each Quarter)')
plt.show()
```



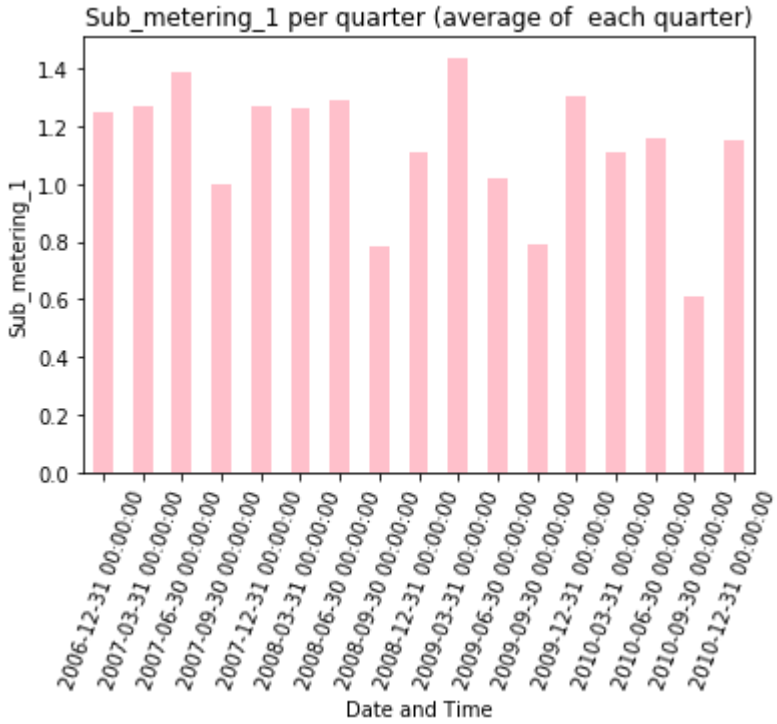
Analysing the Sub_metering 1 (power consumption in kitchen) over month

and quarter

In [133]:
```
df['Sub_metering_1'].resample('M').mean().plot(kind='bar', color='pink')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Sub_metering_1')
plt.title('Sub_metering_1 per month (average of  each month)')
plt.show()
```



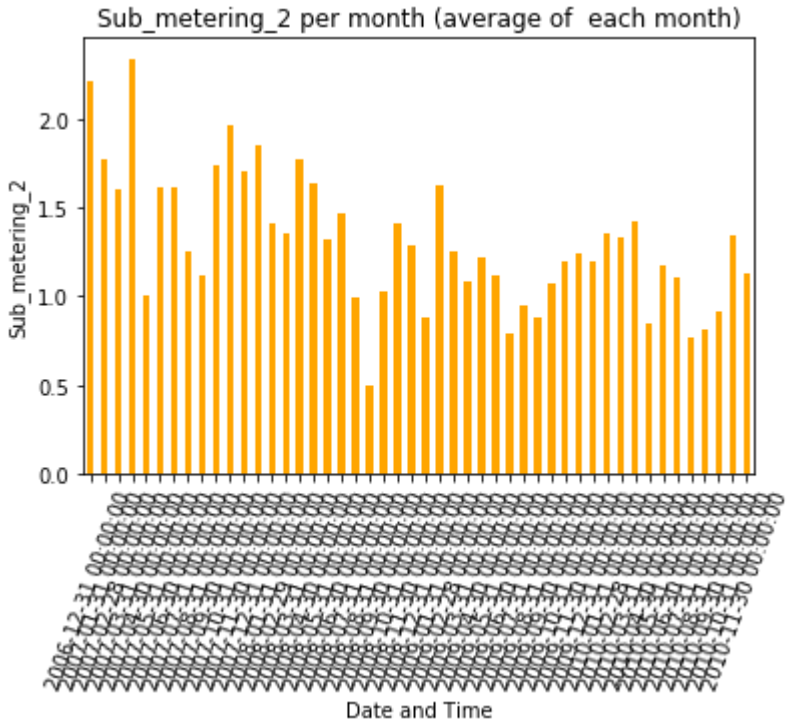
In [134]:
```
df['Sub_metering_1'].resample('Q').mean().plot(kind='bar', color='pink')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Sub_metering_1')
plt.title('Sub_metering_1 per quarter (average of  each quarter)')
plt.show()
```



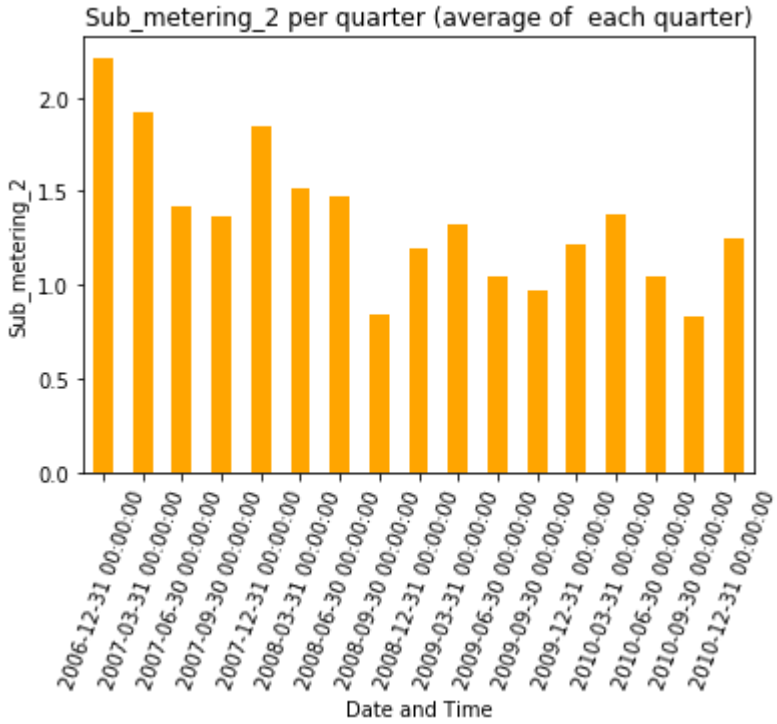
Analysis of sub metering2(power consumption in laundry room)over

month and quarter

In [135]:
```
df['Sub_metering_2'].resample('M').mean().plot(kind='bar', color= 'orange')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Sub_metering_2')
plt.title('Sub_metering_2 per month (average of  each month)')
plt.show()
```



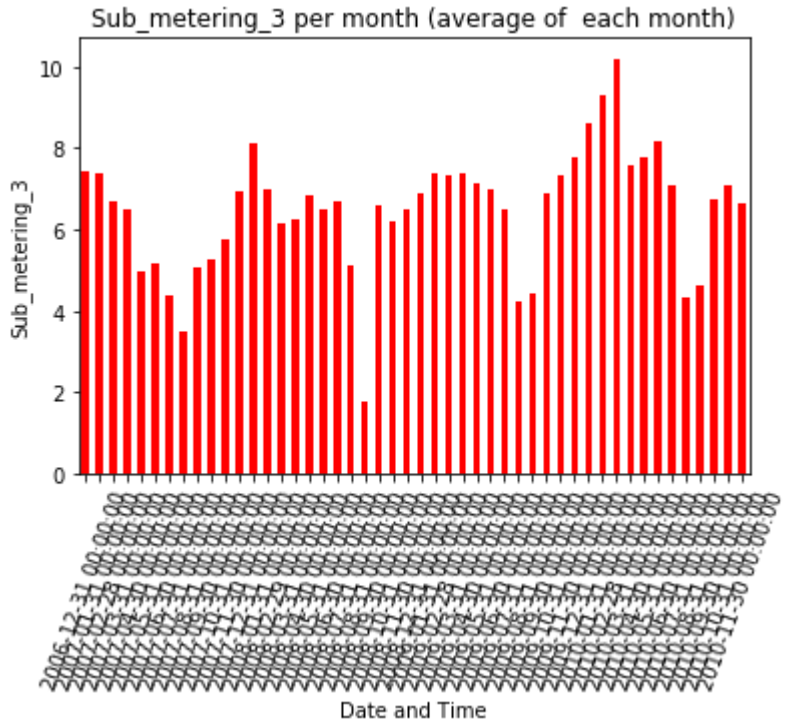
In [136]:
```
df['Sub_metering_2'].resample('Q').mean().plot(kind='bar', color='orange')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Sub_metering_2')
plt.title('Sub_metering_2 per quarter (average of  each quarter)')
plt.show()
```



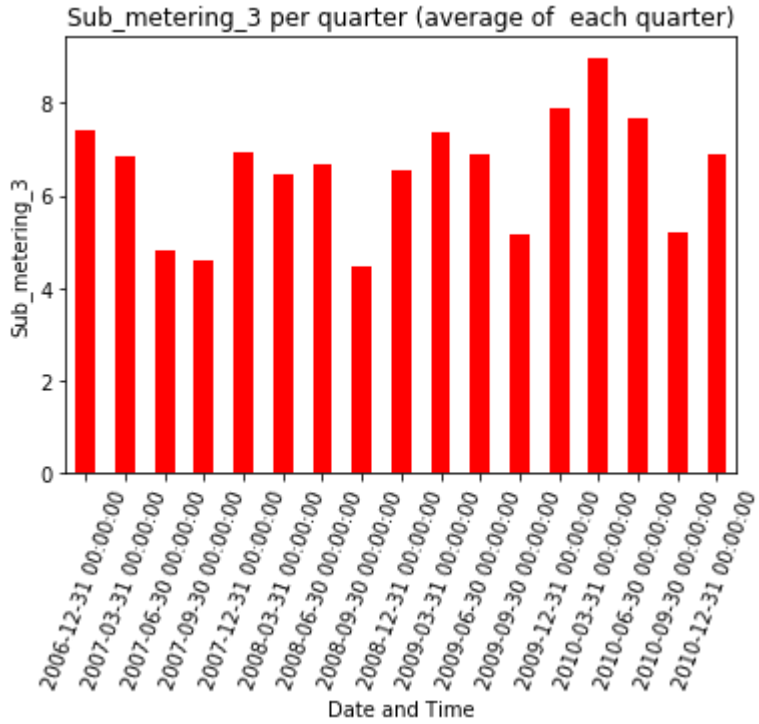
Analysis of Sub metering3(Power consumption of electric heater and

AC)over a month and quarter

In [137]:
```
df['Sub_metering_3'].resample('M').mean().plot(kind='bar', color='red')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Sub_metering_3')
plt.title('Sub_metering_3 per month (average of  each month)')
plt.show()
```

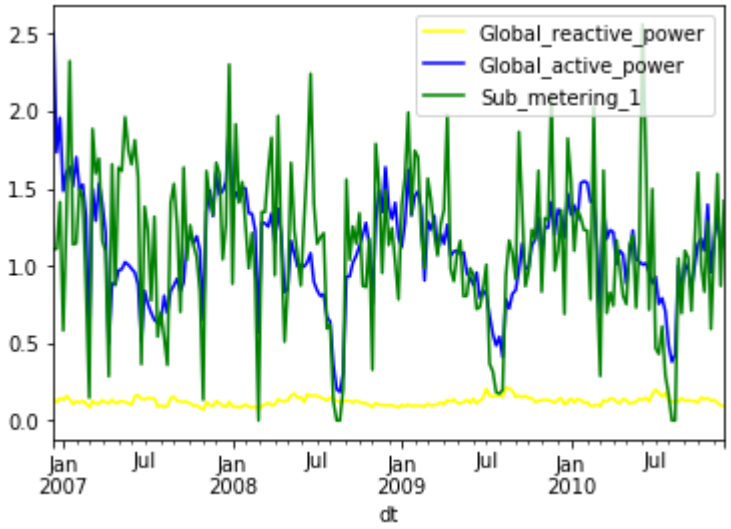


In [138]:
```
df['Sub_metering_3'].resample('Q').mean().plot(kind='bar', color='red')
plt.xticks(rotation=70)
plt.xlabel('Date and Time')
plt.ylabel('Sub_metering_3')
plt.title('Sub_metering_3 per quarter (average of  each quarter)')
plt.show()
```

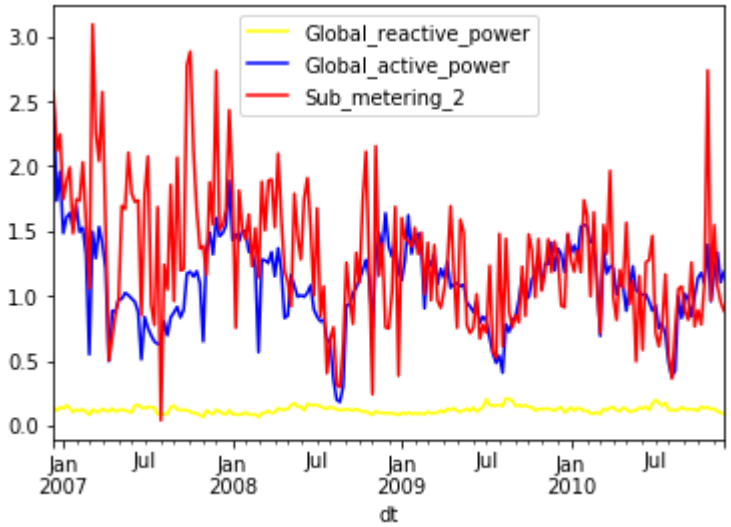


Analysing all the parameters over a week

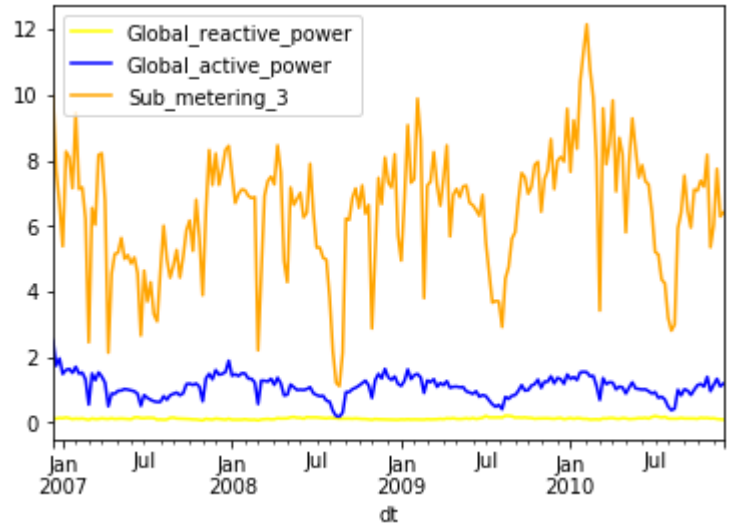
In [139]:
```
df.Global_reactive_power.resample('W').mean().plot(color='yellow', legend=True)
df.Global_active_power.resample('W').mean().plot(color='blue', legend=True)
df.Sub_metering_1.resample('W').mean().plot(color='green', legend=True)
plt.show()
```



In [140]:
```
df.Global_reactive_power.resample('W').mean().plot(color='yellow', legend=True)
df.Global_active_power.resample('W').mean().plot(color='blue', legend=True)
df.Sub_metering_2.resample('W').mean().plot(color='red', legend=True)
plt.show()
```



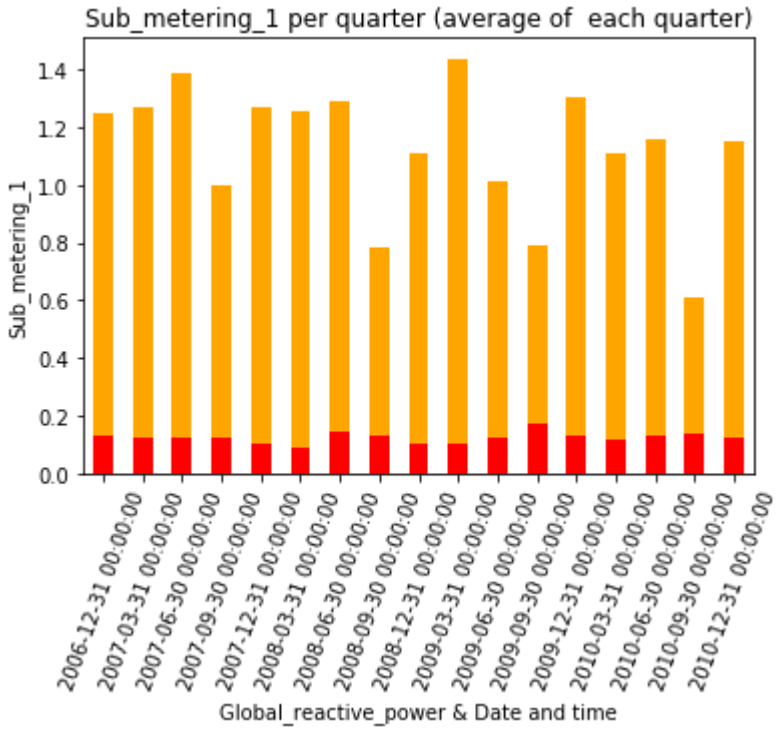
In [141]:
```
df.Global_reactive_power.resample('W').mean().plot(color='yellow', legend=True)
df.Global_active_power.resample('W').mean().plot(color='blue', legend=True)
df.Sub_metering_3.resample('W').mean().plot(color='orange', legend=True)
plt.show()
```



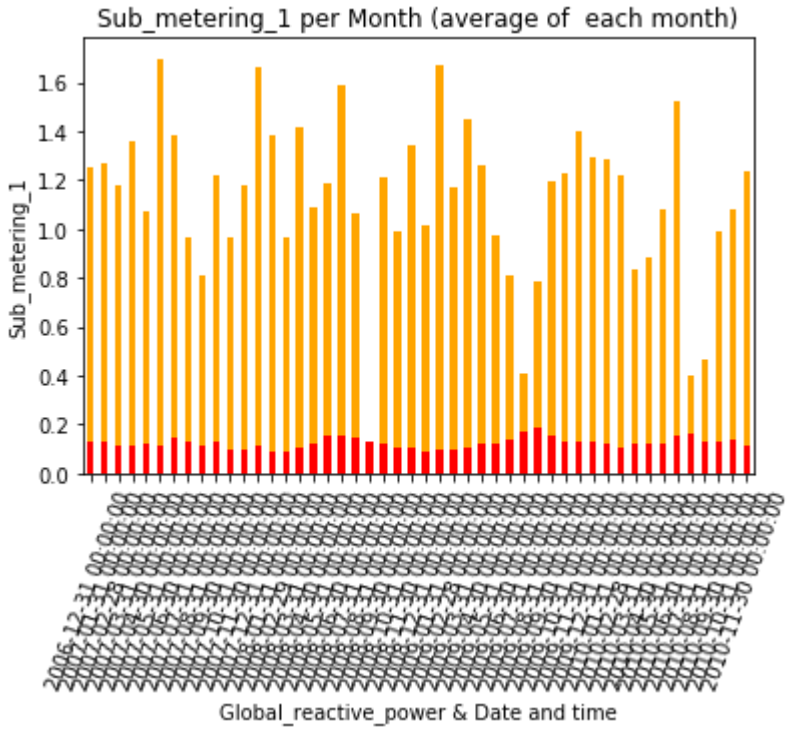
Analysing the Global reactive power and submetering1 over a month and quarter i.e.,

#the wasteage of power and the power consumed by submetering_1

In [142]:
```
df['Sub_metering_1'].resample('Q').mean().plot(kind='bar', color='orange')
df['Global_reactive_power'].resample('Q').mean().plot(kind='bar', color='red')
plt.xticks(rotation=70)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Sub_metering_1')
plt.title('Sub_metering_1 per quarter (average of  each quarter)')
plt.show()
```



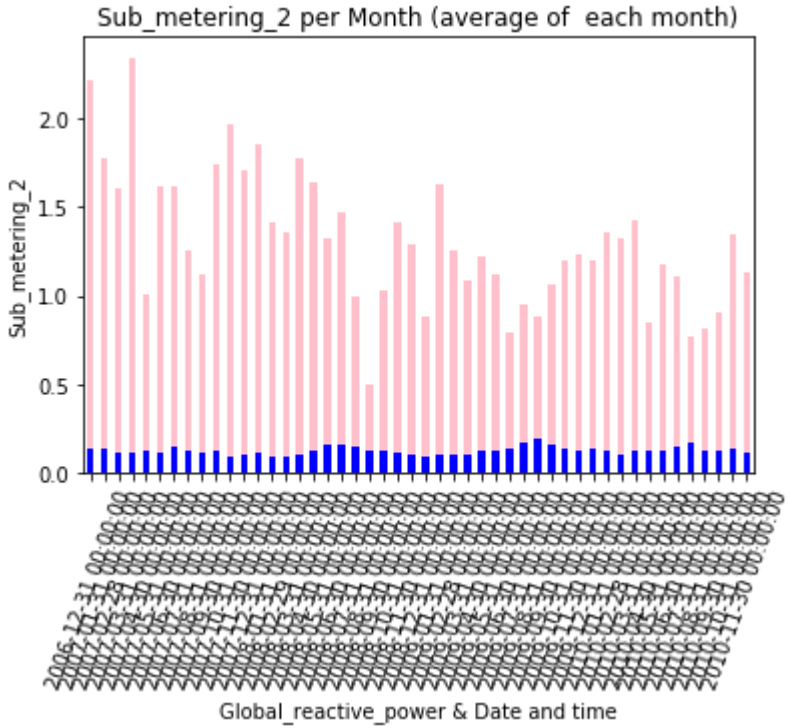
In [143]:
```
df['Sub_metering_1'].resample('M').mean().plot(kind='bar', color='orange')
df['Global_reactive_power'].resample('M').mean().plot(kind='bar', color='red')
plt.xticks(rotation=70)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Sub_metering_1')
plt.title('Sub_metering_1 per Month (average of  each month)')
plt.show()
```



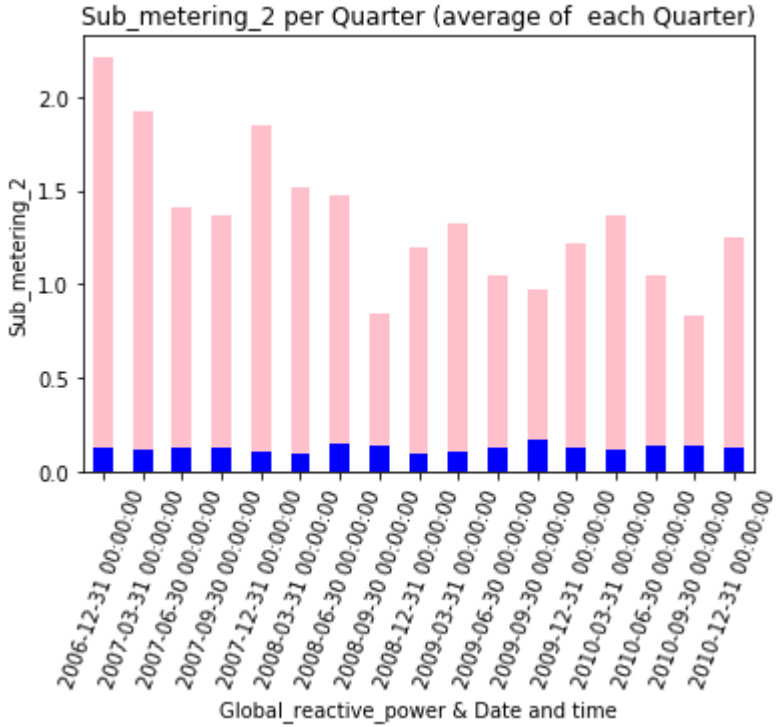
Analysing the Global reactive power and submetering2 over a month and quarter i.e.,

#the wasteage of power and the power consumed by submetering_2

In [144]:
```
df['Sub_metering_2'].resample('M').mean().plot(kind='bar', color='pink')
df['Global_reactive_power'].resample('M').mean().plot(kind='bar', color='blue')
plt.xticks(rotation=70)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Sub_metering_2')
plt.title('Sub_metering_2 per Month (average of  each month)')
plt.show()
```



In [145]:
```
df['Sub_metering_2'].resample('Q').mean().plot(kind='bar', color='pink')
df['Global_reactive_power'].resample('Q').mean().plot(kind='bar', color='blue')
plt.xticks(rotation=70)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Sub_metering_2')
plt.title('Sub_metering_2 per Quarter (average of  each Quarter)')
plt.show()
```

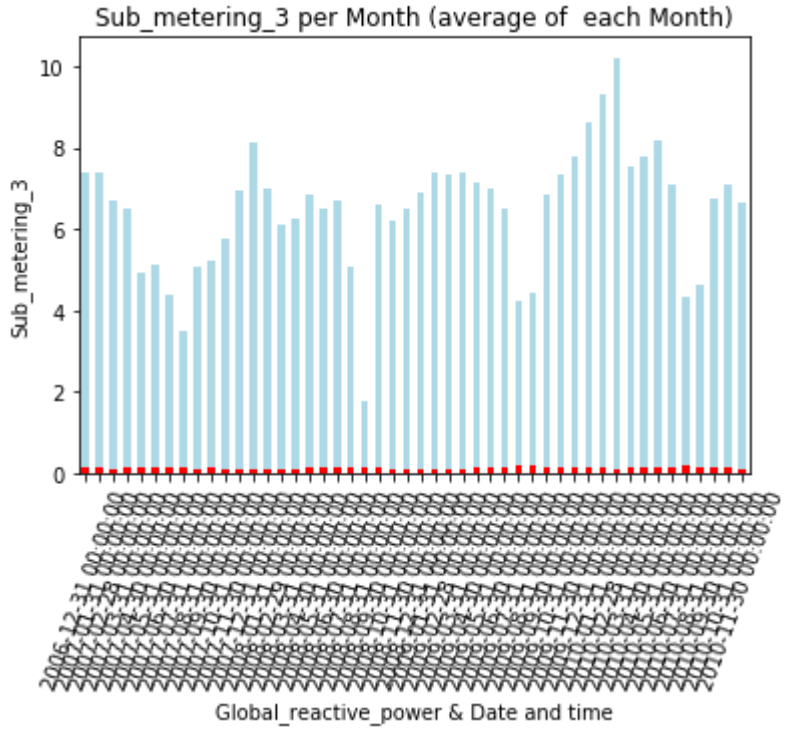


Analysing the Global reactive power and submetering3 over a month and

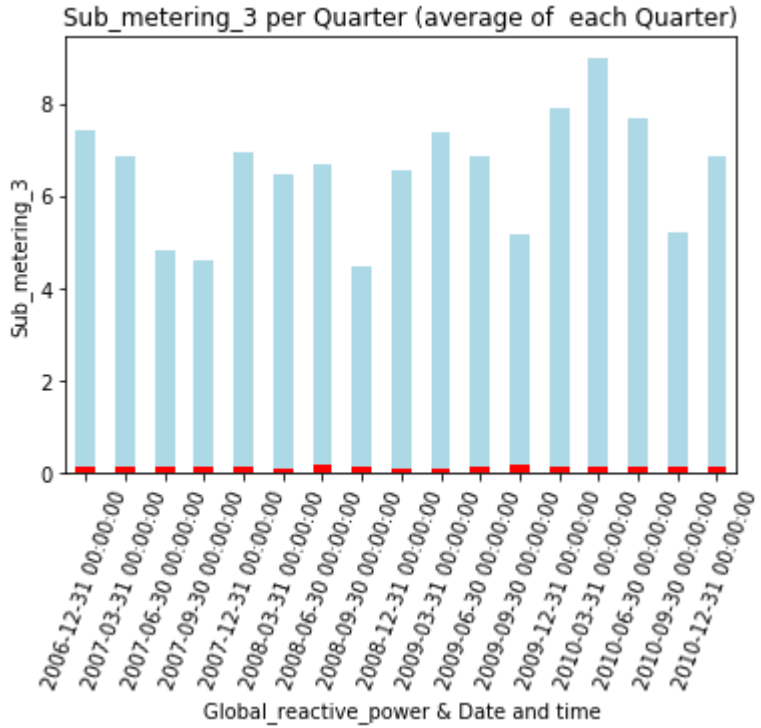
quarter i.e.,

#the wasteage of power and the power consumed by submetering_3

In [146]:
```
df['Sub_metering_3'].resample('M').mean().plot(kind='bar', color='lightblue')
df['Global_reactive_power'].resample('M').mean().plot(kind='bar', color='red')
plt.xticks(rotation= 70)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Sub_metering_3')
plt.title('Sub_metering_3 per Month (average of  each Month)')
plt.show()
```

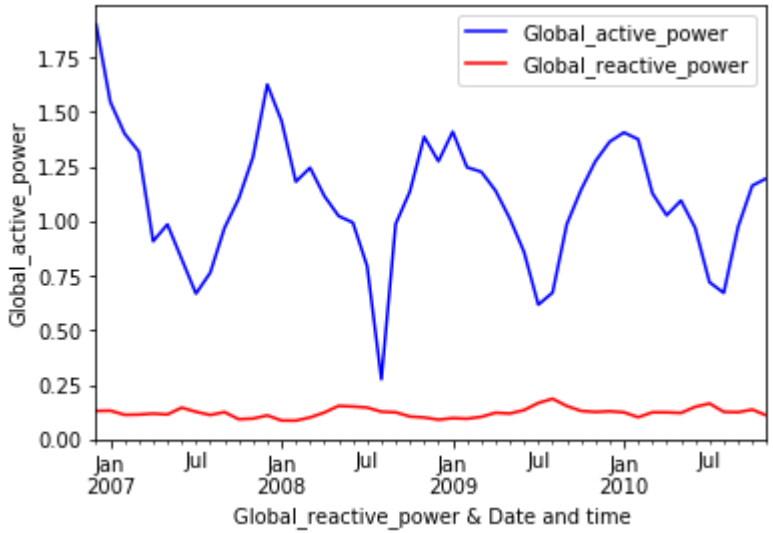


In [147]:
```
df['Sub_metering_3'].resample('Q').mean().plot(kind='bar', color='lightblue')
df['Global_reactive_power'].resample('Q').mean().plot(kind='bar', color='red')
plt.xticks(rotation=70)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Sub_metering_3')
plt.title('Sub_metering_3 per Quarter (average of  each Quarter)')
plt.show()
```

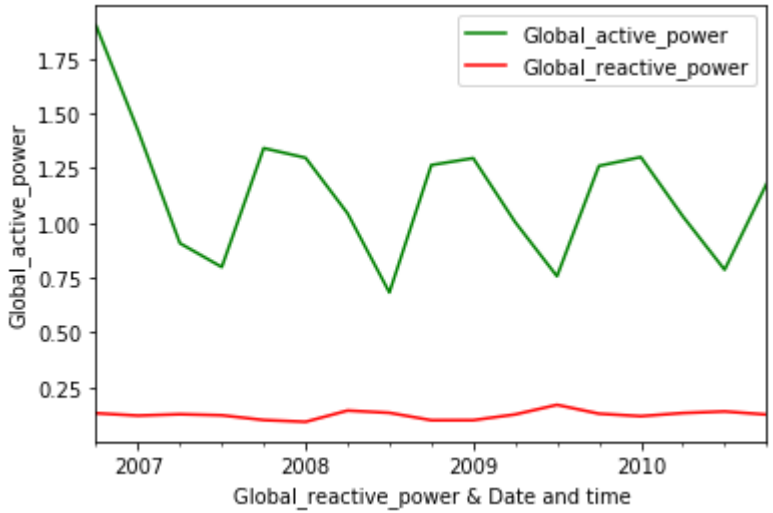


Analysing the Global_Active power vs Reactive power over Month , quarter and year

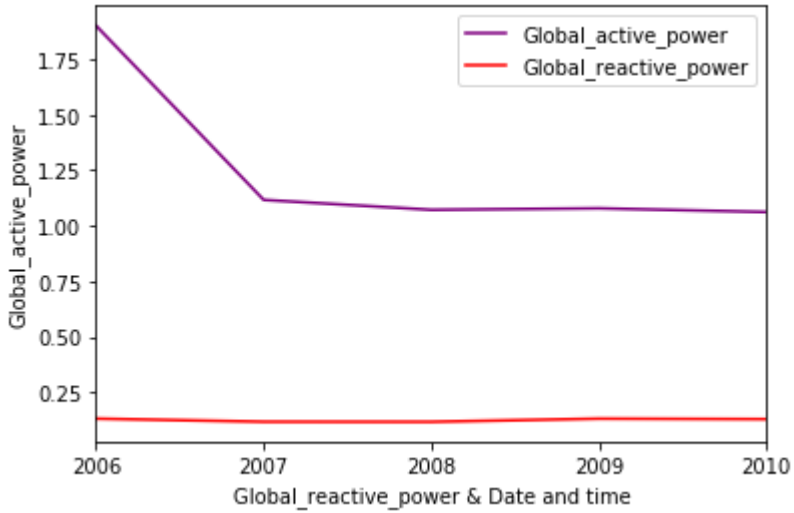
In [148]:
```
df.Global_active_power.resample('M').mean().plot(color='blue', legend=True)
df.Global_reactive_power.resample('M').mean().plot(color='red', legend=True)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Global_active_power')
plt.show()
```



In [149]:
```
df.Global_active_power.resample('Q').mean().plot(color='green', legend=True)
df.Global_reactive_power.resample('Q').mean().plot(color='red', legend=True)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Global_active_power')
plt.show()
```



In [150]:
```
df.Global_active_power.resample('y').mean().plot(color='purple', legend=True)
df.Global_reactive_power.resample('y').mean().plot(color='red', legend=True)
plt.xlabel('Global_reactive_power & Date and time')
plt.ylabel('Global_active_power')
plt.show()
```



MODEL SELECTION

In [186]:
```
df.isnull().any()
```

Out[186]:
```
Global_active_power      False
Global_reactive_power    False
Voltage                  False
Global_intensity         False
Sub_metering_1           False
Sub_metering_2           False
Sub_metering_3           False
dtype: bool
```

In [187]:
```
df.head(1)
```

Out[187]:

dt	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2006-12-16 17:24:00	4.216	0.418	234.84	18.4	0.0	1.0	17.0

In [189]:
```
x = df.iloc[:,1:7].values
y = df.iloc[:,0:1].values
```

In [190]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2, random_state = 0)
```

In [191]:
```
x_train.shape
```

Out[191]: (1660207, 6)

In [192]:
```
y_train.shape
```

Out[192]: (1660207, 1)

In [193]:
```
x_test.shape
```

Out[193]: (415052, 6)

In [194]:
```
y_test.shape
```

Out[194]: (415052, 1)

RANDOM FOREST AND DECISION TREE REGRESSION MODEL

In [203]:
```
from sklearn.tree import  DecisionTreeRegressor
dtr = DecisionTreeRegressor(random_state = 0)
dtr.fit(x_train,y_train)
```

Out[203]:
```
DecisionTreeRegressor(random_state=0)
```

In [205]:
```
ydtr = dtr.predict(x_test)
```

In [206]:
```
accuratdtr = r2_score(y_test,ydtr)
```

In [207]:
```
accuratdtr
```

Out[207]:
```
0.9984153921918739
```

In [211]:
```
dtr.predict([[1.0,250,30,1.3,2.5,31.0]])
```

Out[211]:
```
array([7.272])
```

MULTILINEAR REGRESSION MODEL

In [195]:
```
from sklearn.linear_model import LinearRegression
mlr  = LinearRegression()
mlr.fit(x_train,y_train)
```

Out[195]:
```
LinearRegression()
```

In [201]:
```
y_test
```

Out[201]:
```
array([[3.112],
       [2.21 ],
       [0.666],
       ...,
       [0.31 ],
       [2.934],
       [2.412]])
```

In [202]:
```
y_train
```

Out[202]:
```
array([[1.734],
       [0.426],
       [2.306],
       ...,
       [0.218],
       [0.302],
       [0.41 ]])
```

In [196]:
```
y_pred = mlr.predict(x_test)
```

In [197]:
```
from sklearn.metrics import r2_score
accuracy = r2_score(y_test,y_pred)
```

In [198]:
```
accuracy
```

Out[198]:
```
0.9985267118834567
```

In [199]:
```
df.head(1)
```

Out[199]:

dt	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2006-12-16 17:24:00	4.216	0.418	234.84	18.4	0.0	1.0	17.0

In [200]:
```
mlr.predict([[1.0,250,30,1.3,2.5,31.0]])
```

Out[200]:
```
array([[7.07108389]])
```

In [ ]:
```

```