Using the kaggle API to download the daatset

```
!pip install -q kaggle
```

```
!mkdir -p ~/.kaggle
```

```
!cp /content/kaggle.json ~/.kaggle/
```

```
!ls ~/.kaggle
```

```
kaggle.json
```

```
!chmod 600 /root/.kaggle/kaggle.json
```

```
from google.colab import files
files=files.upload()
```

```
Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving kaggle.json to kaggle.json
```

```
!kaggle competitions download -c ashrae-energy-prediction
```

```
Warning: Looks like you're using an outdated API Version, please consider updating (
Downloading weather_train.csv.zip to /content
  0% 0.00/1.27M [00:00<?, ?B/s]
100% 1.27M/1.27M [00:00<00:00, 84.6MB/s]
Downloading train.csv.zip to /content
 96% 115M/120M [00:04<00:00, 25.3MB/s]
100% 120M/120M [00:04<00:00, 30.8MB/s]
Downloading test.csv.zip to /content
 92% 153M/167M [00:06<00:00, 26.7MB/s]
100% 167M/167M [00:06<00:00, 27.8MB/s]
Downloading building_metadata.csv to /content
  0% 0.00/44.5k [00:00<?, ?B/s]
100% 44.5k/44.5k [00:00<00:00, 46.2MB/s]
Downloading weather_test.csv.zip to /content
  0% 0.00/2.53M [00:00<?, ?B/s]
100% 2.53M/2.53M [00:00<00:00, 83.4MB/s]
Downloading sample_submission.csv.zip to /content
 84% 74.0M/88.4M [00:02<00:00, 20.9MB/s]
100% 88.4M/88.4M [00:02<00:00, 38.2MB/s]
```

```
!unzip /content/weather_train.csv.zip
```

```
Archive:  /content/weather_train.csv.zip
  inflating: weather_train.csv
```

```
import warnings
```

```
warnings.filterwarnings('ignore')


import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


pd.options.display.float_format = '{:,.2f}'.format


df_train=pd.read_csv('train.csv')
```

The code which is mentioned below is taken from a github account which reduces the size of the dataset and helps in doing the EDA accurately.

```
def reduce_mem_usage(df, verbose=True):
    numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
    start_mem = df.memory_usage().sum() / 1024**2
    for col in df.columns:
        col_type = df[col].dtypes
        if col_type in numerics:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).ma›
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
    end_mem = df.memory_usage().sum() / 1024**2
    if verbose: print('Mem. usage decreased to {:5.2f} Mb ({:.1f}% reduction)'.format(end_
    return df



df_train_red=reduce_mem_usage(df_train,verbose=True)
```

```
 Mem. usage decreased to 289.19 Mb (53.1% reduction)
```

```
df_weather_train=pd.read_csv('weather_train.csv')


df_weather_train_red=reduce_mem_usage(df_weather_train,verbose=True)
```

```
Mem. usage decreased to  3.07 Mb (68.1% reduction)
```

```
df_building=pd.read_csv('building_metadata.csv')
```

```
df_building_red=reduce_mem_usage(df_building,verbose=True)
```

```
Mem. usage decreased to  0.03 Mb (60.3% reduction)
```

```
df_train_build=pd.merge(df_train_red,df_building_red,how='left',on=['building_id'])
```

```
df_train_merge=pd.merge(df_train_build,df_weather_train_red,how='left',on=['site_id','time
```

```
df_train_merge['timestamp']=pd.to_datetime(df_train_merge['timestamp'])
```

```
import seaborn as sns
```

Replacing the meter by their usage type

```
df_train_merge['meter'].replace([0,1,2,3],['electricity','chilledwater','steam','hotwater'
```

Starting the EDA for each and every site

```
df_train_site_0=df_train_merge.loc[df_train_merge['site_id']==0]
```

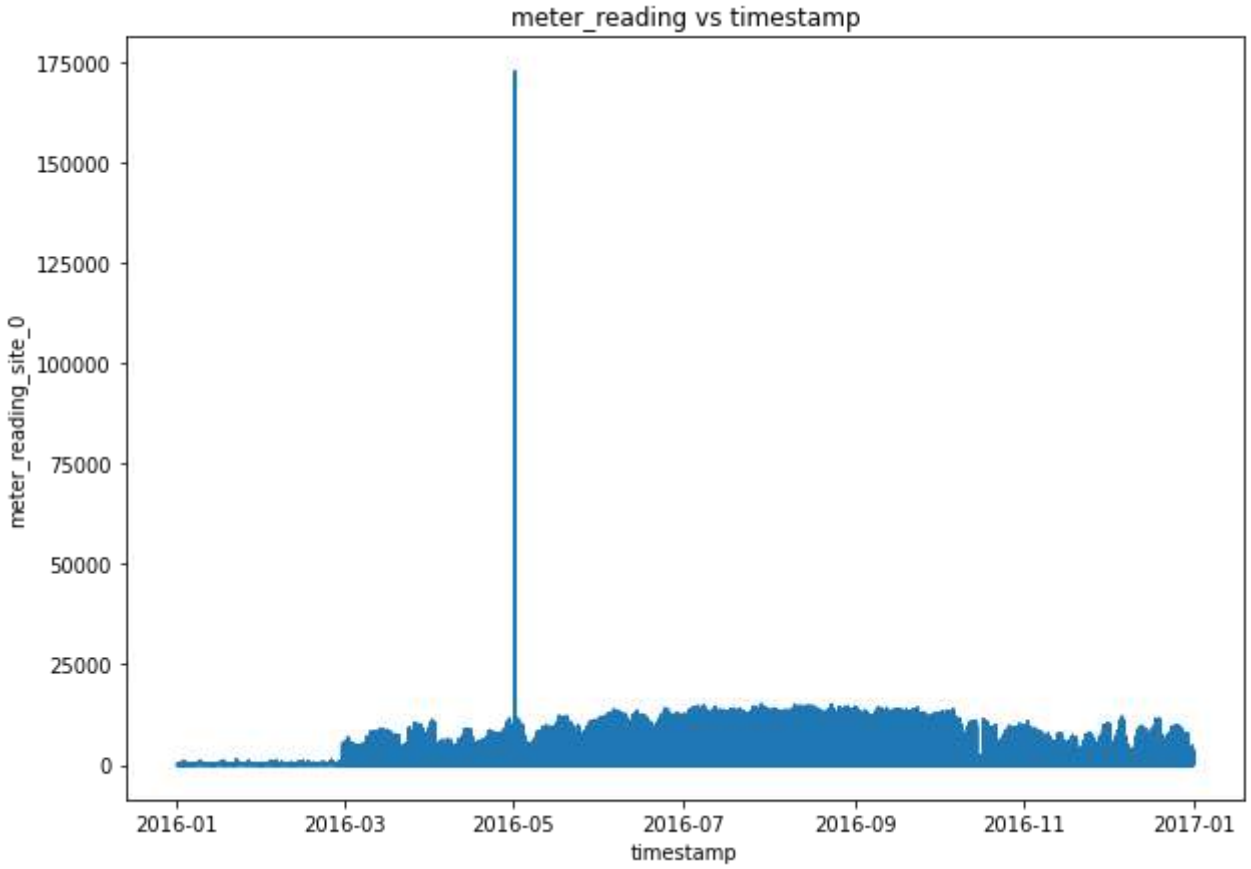
```
(df_train_site_0.isnull().sum()/df_train_site_0.shape[0])*100
```

```
building_id            0.00
meter                  0.00
timestamp              0.00
meter_reading          0.00
site_id                0.00
primary_use            0.00
square_feet            0.00
year_built             0.00
floor_count          100.00
air_temperature        0.03
cloud_coverage        43.41
dew_temperature        0.03
precip_depth_1_hr      0.01
sea_level_pressure     0.96
wind_direction         2.90
wind_speed             0.00
dtype: float64
```

Here we are checking the percentage of null values at site 0

1. First thing we can notice that floor count is 100% missing so for this site we can drop this feature.
2. Cloud Coverage is almost 44% missing and instead of dropping this we can impute it to fill the missing values.

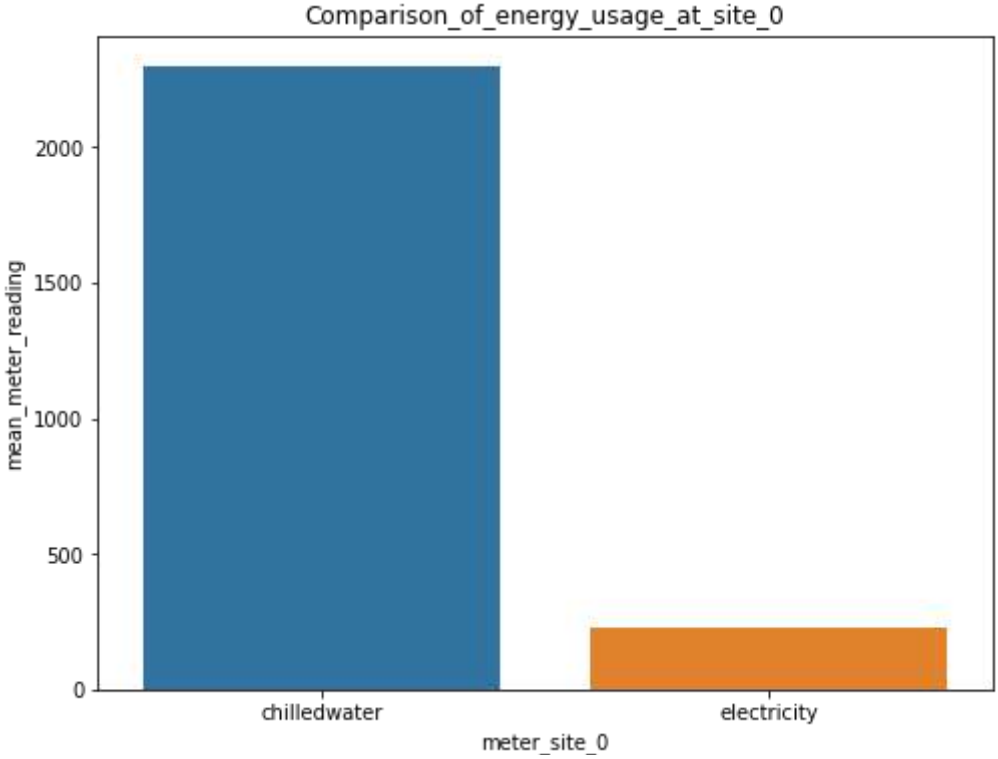
```
fig,ax=plt.subplots(figsize=(10,7))
ax.plot(df_train_site_0['timestamp'],df_train_site_0['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('meter_reading_site_0')
plt.title('meter_reading vs timestamp')
plt.show()
```



This is the plot of hourly meter reading at site 0 with the timestamp

1. One important observation we can see that the meter readings for the first few months are comparatively lower than the preceding months.
2. Now here in the 5 month a very high reading is observed which can be an anomaly.

```
z=df_train_site_0.groupby(['meter'])
fg,ax=plt.subplots(figsize=(8,6))
sns.barplot(data=z['meter_reading'].mean().reset_index(),x='meter',y='meter_reading',ax=ax
plt.title('Comparison_of_energy_usage_at_site_0')
plt.show()
```

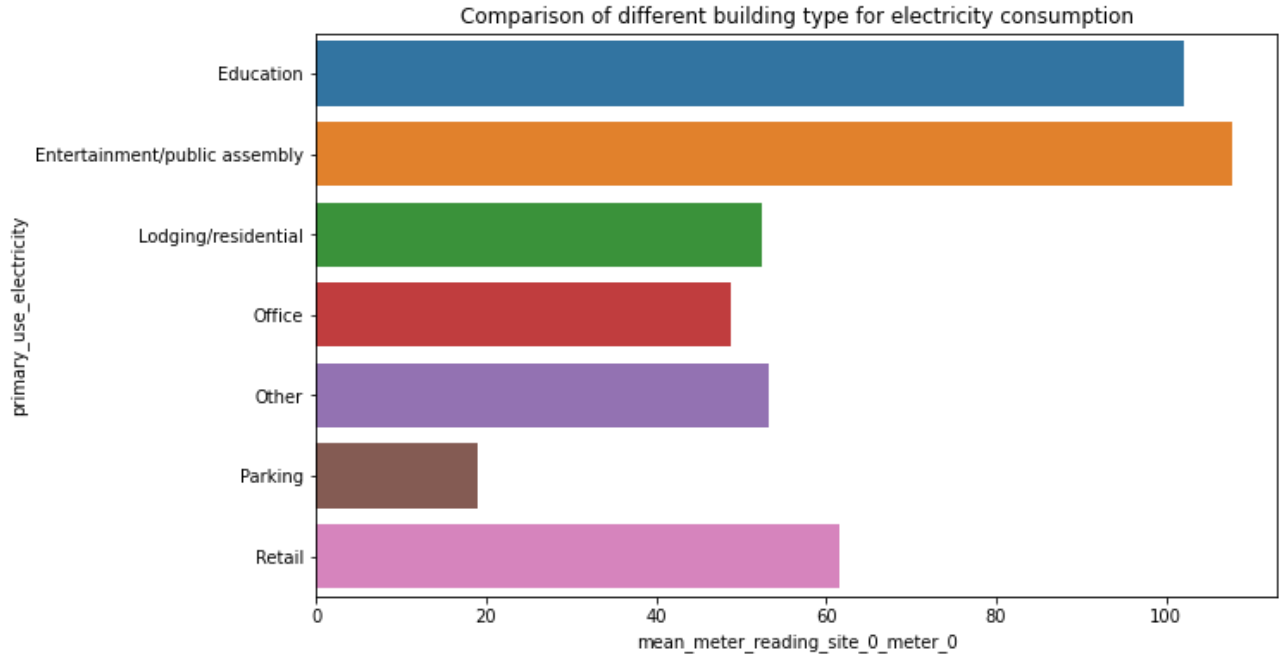


1. **This barplot brings an important observation to us that energy required for chilledwater usage is very high as compared to the electricity usage at site 0**

```
#Here I will be dividing the site by different energy usage types
df_train_site_0_meter_0=df_train_site_0.loc[df_train_site_0['meter']=='electricity']
df_train_site_0_meter_1=df_train_site_0.loc[df_train_site_0['meter']=='chilledwater']


#The electrical consumption at site 0 is in kBTU we need to convert it into kWh
df_train_site_0_meter_0['meter_reading']=df_train_site_0_meter_0['meter_reading']*0.2931


g=df_train_site_0_meter_0.groupby(['primary_use'])
sns.barplot(data=g['building_id'].nunique().reset_index(),x='building_id',y='primary_use')
plt.xlabel('building_count_site_0_meter_0')
plt.ylabel('primary_use_electrcity')
plt.title('building count vs primary usage')
plt.show()
```

This barplot represents the count of building at site 0 which is using electricity as its primary usage source


```
fig,ax=plt.subplots(figsize=(10,6))
k=df_train_site_0_meter_0.groupby(['primary_use'])
sns.barplot(ax=ax,data=g['meter_reading'].mean().reset_index(),x='meter_reading',y='primar
plt.xlabel('mean_meter_reading_site_0_meter_0')
plt.ylabel('primary_use_electricity')
plt.title('Comparison of different building type for electricity consumption')
plt.show()
```

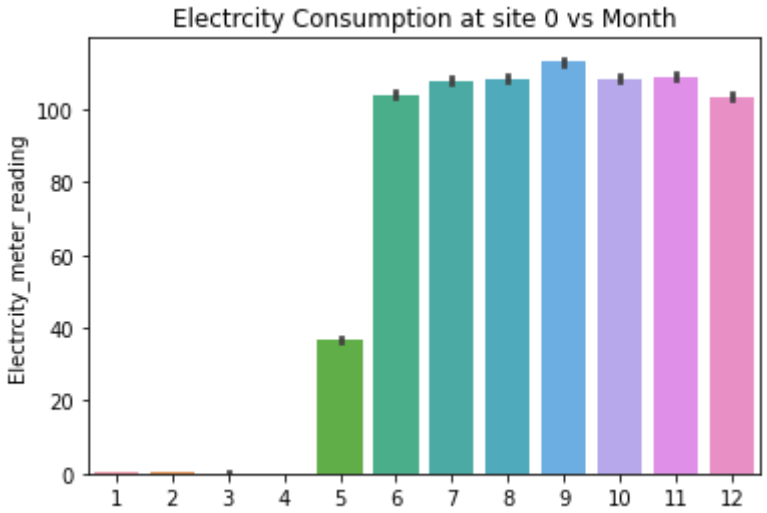


Important observation--> Although entertainment and public assembly buildings are less in number but are consuming electricity at a large scale.

Retail is also having high consumption but are less in numbers.

```
#Here we are adding month to check the seasonal variations of electricity consumption
df_train_site_0_meter_0['month']=df_train_site_0_meter_0['timestamp'].dt.month
```

```
z=df_train_site_0_meter_0
sns.barplot(data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('Electrcity_meter_reading')
plt.title('Electrcity Consumption at site 0 vs Month')
plt.show()
```

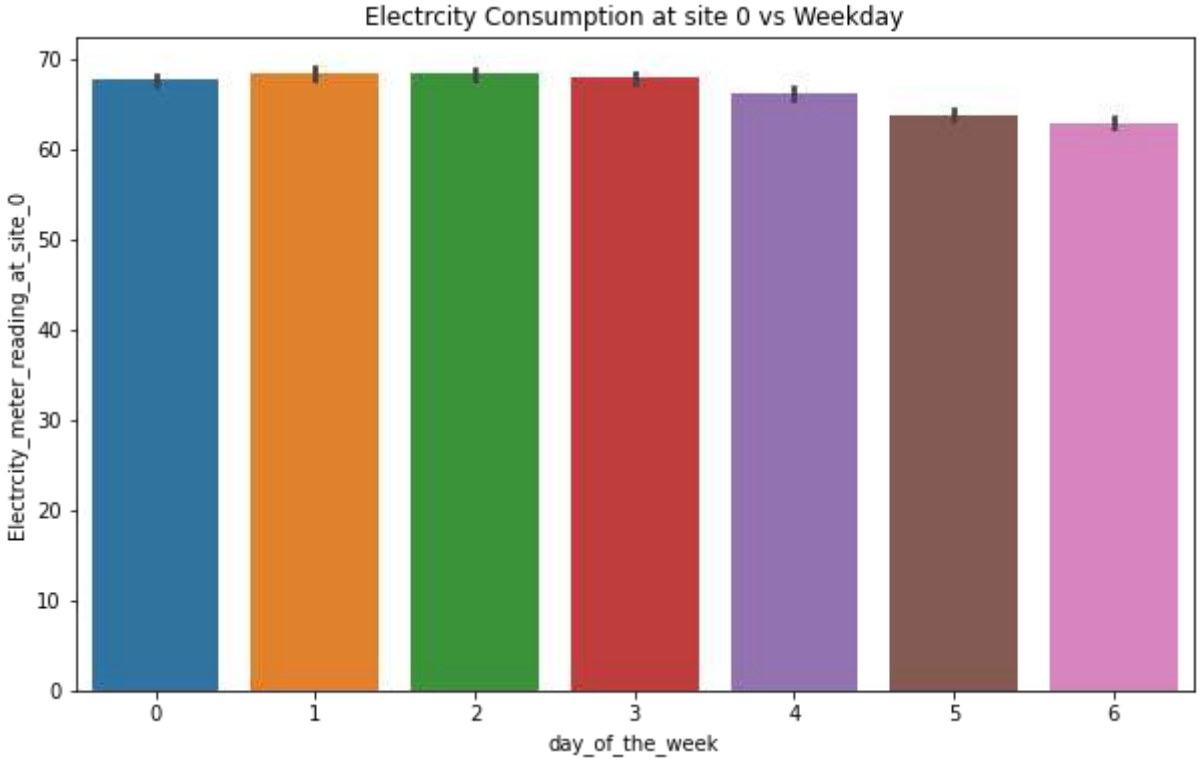


From this plot we can observe that the proper meter readings are observed after the 5 month

Now we can also see that the electricty consumption is slightly higher for the summer months as compared to the other months.

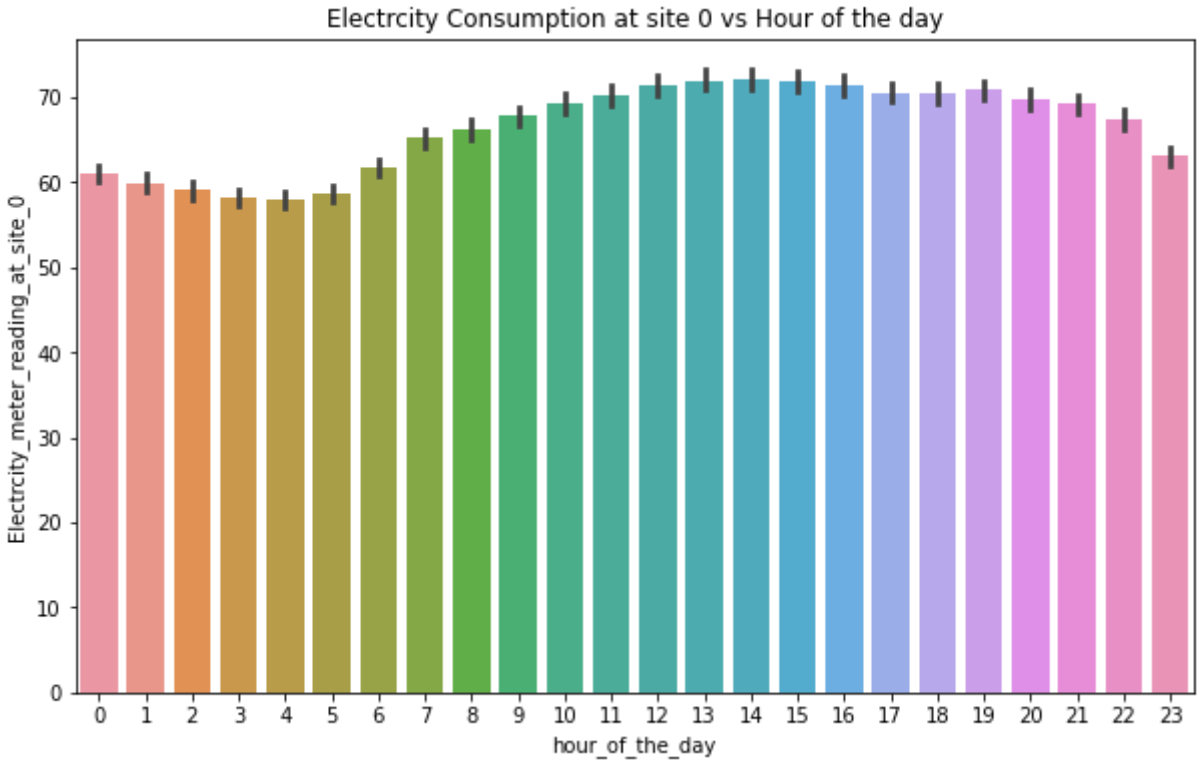
```
#Adding weekday and hour to check the variations during the week and the hour
df_train_site_0_meter_0['hour']=df_train_site_0_meter_0['timestamp'].dt.hour
df_train_site_0_meter_0['weekday']=df_train_site_0_meter_0['timestamp'].dt.weekday
```

```
fig,ax=plt.subplots(figsize=(10,6))
z=df_train_site_0_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('Electrcity_meter_reading_at_site_0')
plt.title('Electrcity Consumption at site 0 vs Weekday')
plt.show()
```



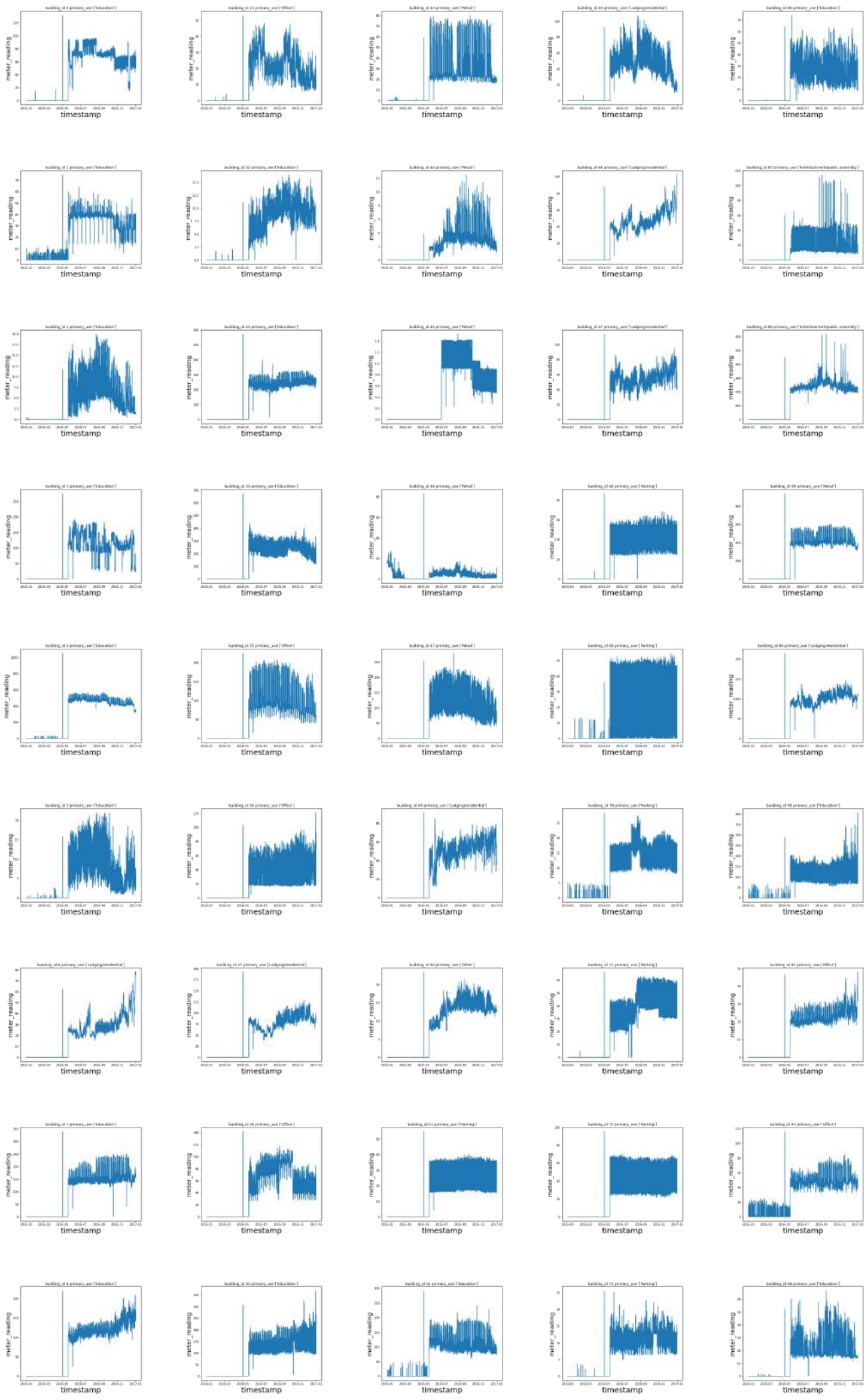
Electrical consumption over the weekend is less as compared to the weekdays

```
fig,ax=plt.subplots(figsize=(10,6))
z=df_train_site_0_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('Electrcity_meter_reading_at_site_0')
plt.title('Electrcity Consumption at site 0 vs Hour of the day')
plt.show()
```

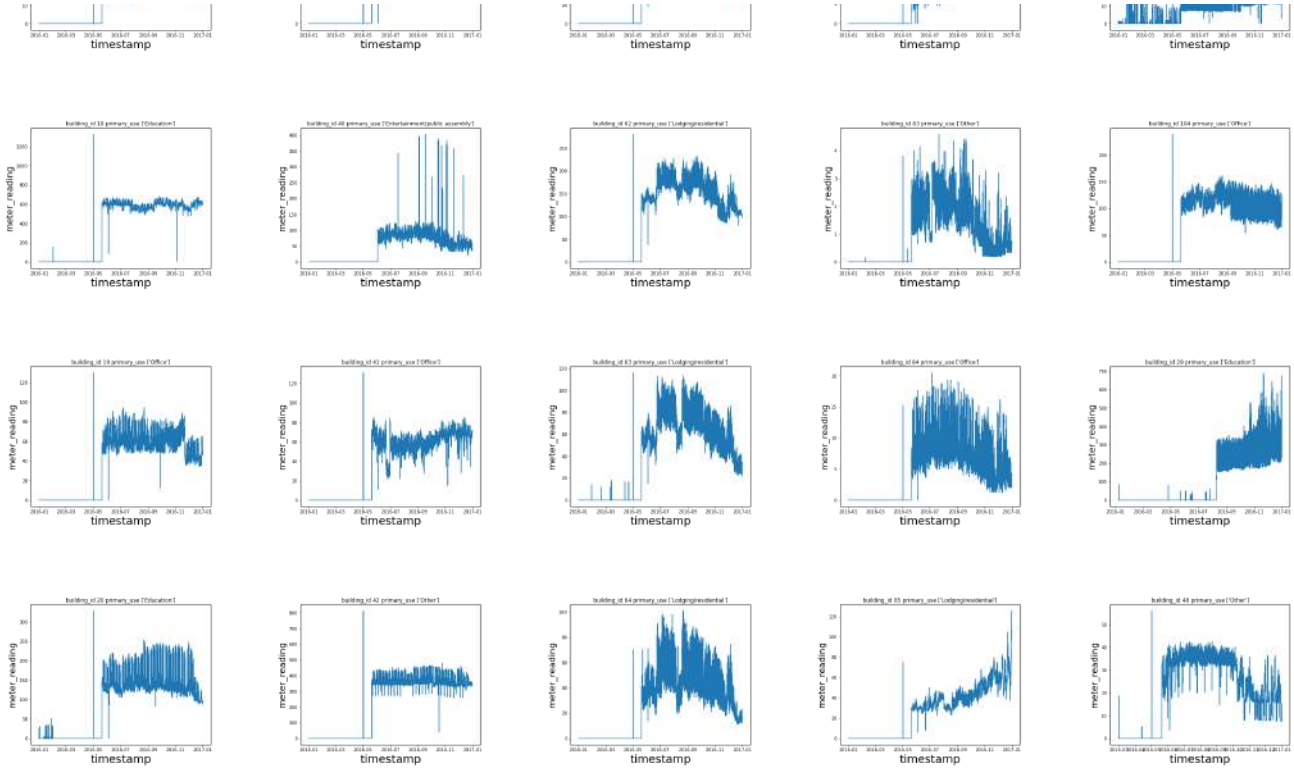


The Electrical consumption also shows variation over different hours of the day and it peaks around at 14:00 pm and then starts falling over time.

```
fig,axes=plt.subplots(nrows=21,ncols=5,figsize=(50,200))
for i in range(df_train_site_0_meter_0['building_id'].nunique()):
 g=df_train_site_0_meter_0['building_id'].unique()[i]
 z=df_train_site_0_meter_0[df_train_site_0_meter_0['building_id']==g]
 ax=axes[i%21][i//21]
 ax.plot(z['timestamp'],z['meter_reading'])
 ax.set_xlabel('timestamp',fontsize=25)
 ax.set_ylabel('meter_reading',fontsize=20)
 k=z[z['building_id']==g]
 ax.set_title('building_id {} primary_use {}'.format(g,k['primary_use'].unique()),fontsize
 plt.subplots_adjust(hspace=0.7,wspace=0.5)
```


meter_reading
timestamp


meter_reading
timestamp


timestamp
meter_reading

Here I have plotted the meter readings for all the buildings which consume electricty.

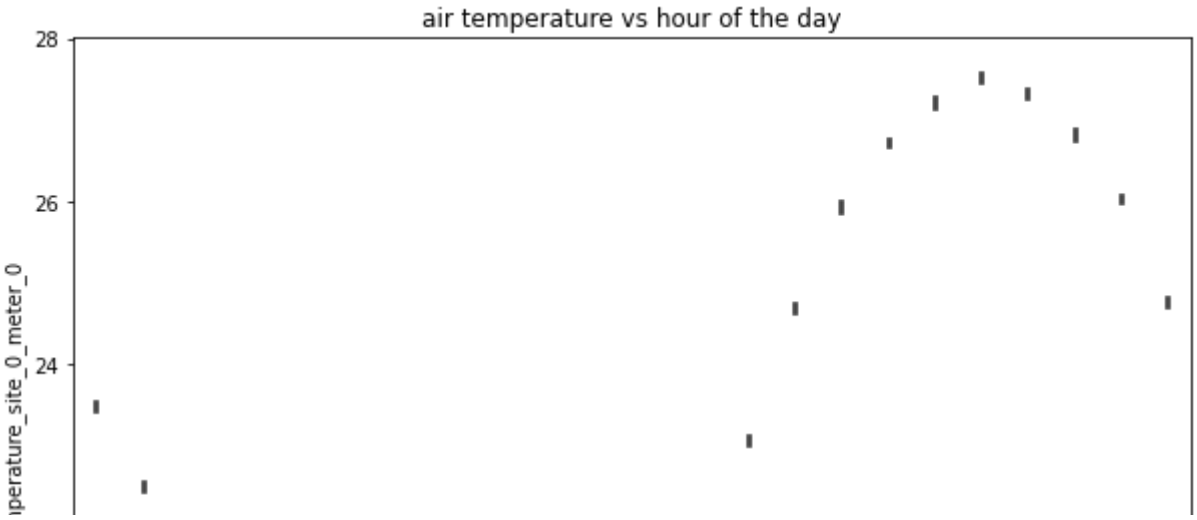
1) The meter readings are mostly zeros till May 20 as if proper readings are taken after that only therefore we need to remove all those readings as it will not be observed for the next year.

2) Building-->53 I will not use that building for training purpose as it shows high readings and zero for most part of the year which can definitely be an anomaly.

3) Building-> 45 and 53 are having zero meter readings till 7 month and 9 month which we have to take care of.

4) The reading which are extremely high in between normal readings we have to remove that also as it might be due to some fault.

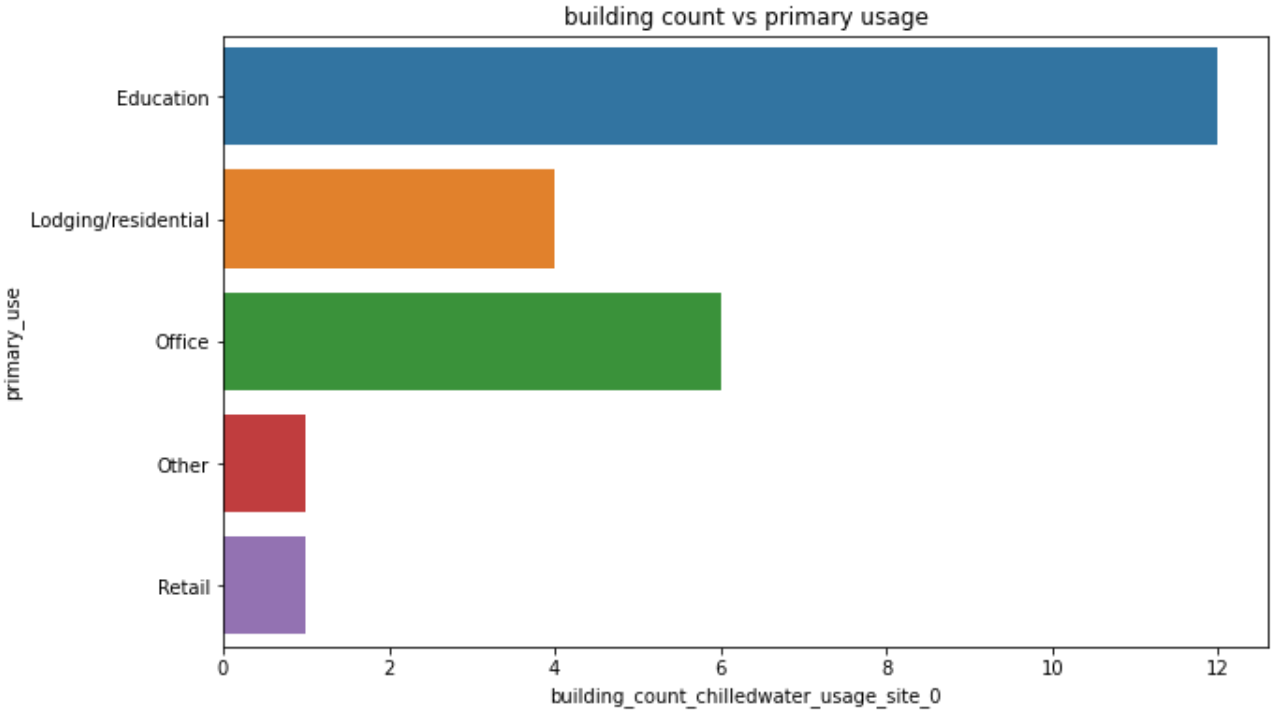
```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_0_meter_0,x='hour',y='air_temperature')
plt.ylabel('air_temperature_site_0_meter_0')
plt.xlabel('hour_of_the_day')
plt.title('air temperature vs hour of the day')
plt.show()
```



Weather Timestamp

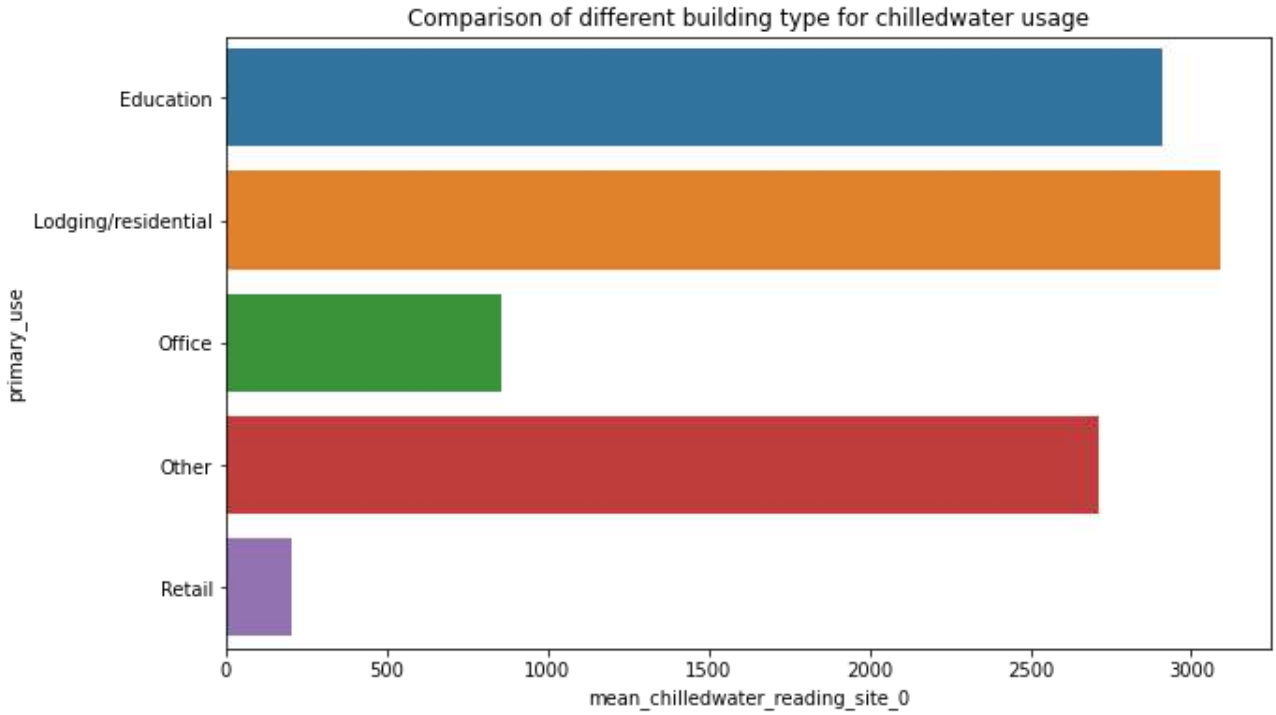
1. Here the temperature becomes maximun at 20:00 pm.
2. This implies that the weather timestamp is not in alignment with the local timestamp of the hourly meter reading for electricty consumption.

```
g=df_train_site_0_meter_1.groupby(['primary_use'])
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=g['building_id'].nunique().reset_index(),x='building_id',y='primary
plt.xlabel('building_count_chilledwater_usage_site_0')
plt.ylabel('primary_use')
plt.title('building count vs primary usage')
plt.show()
```



It shows the building count which uses chilledwater as primary usage

```
k=df_train_site_0_meter_1.groupby(['primary_use'])
k=k['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=k,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading_site_0')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for chilledwater usage')
plt.show()
```

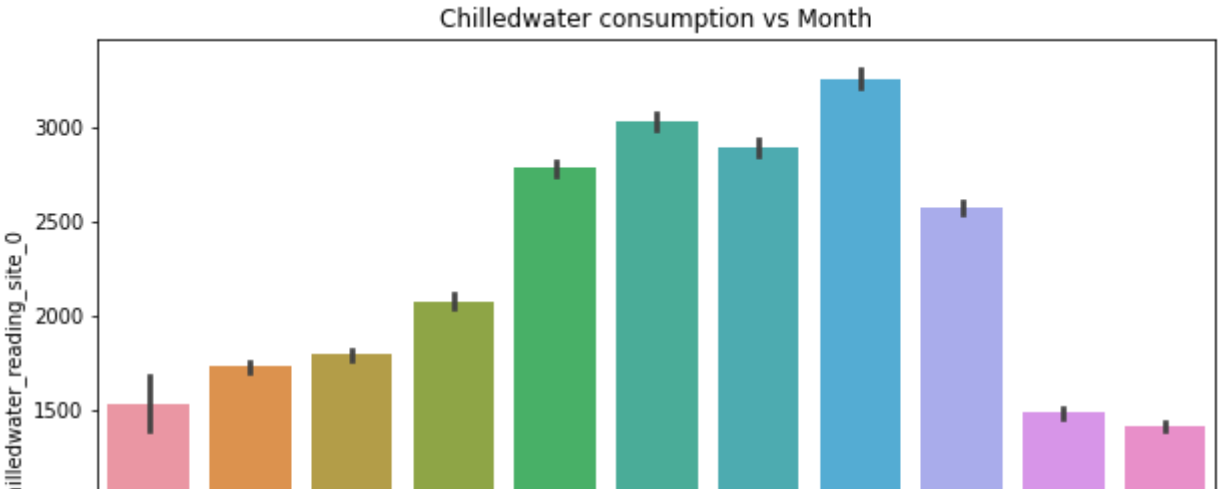


Lodging/Residential and Other are having higher chilledwater consumption

Higher consumption can be normal depending upon the building or it can be due to anomaly readings therefore we need to plot and see the variations in the consumption.

```
df_train_site_0_meter_1['month']=df_train_site_0_meter_1['timestamp'].dt.month
```

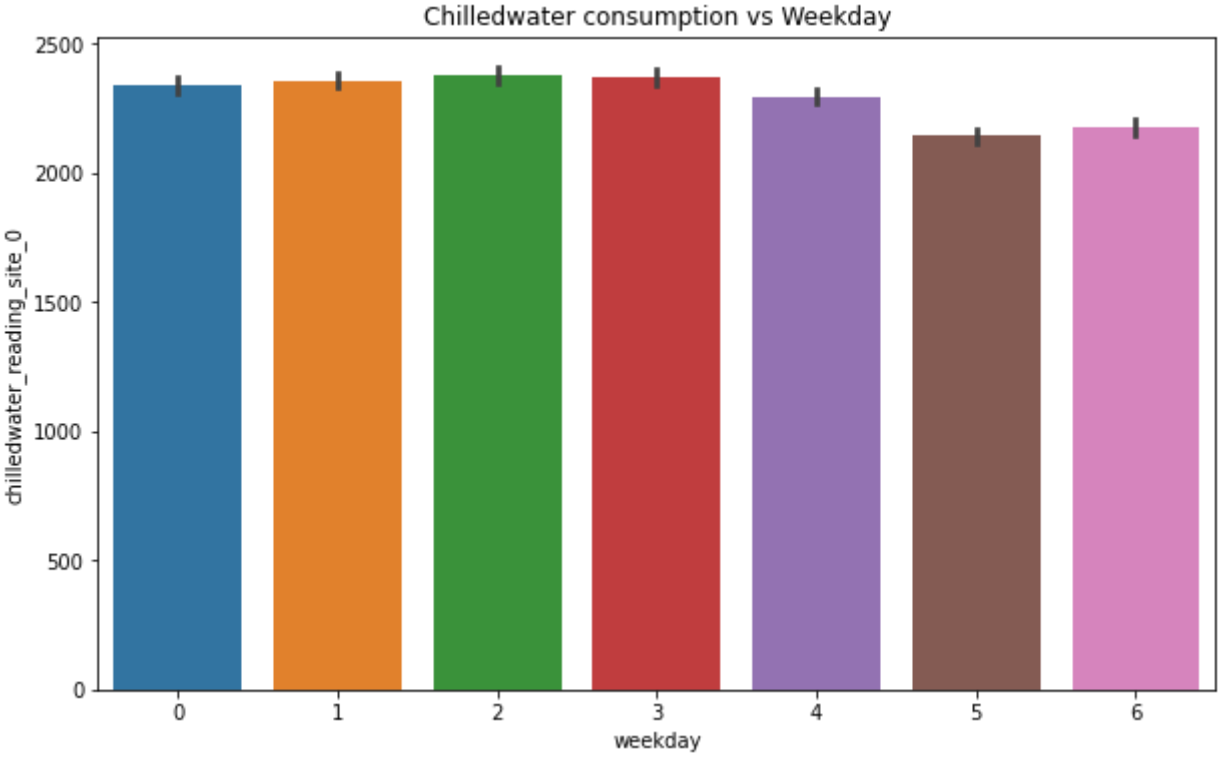
```
z=df_train_site_0_meter_1
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.ylabel('chilledwater_reading_site_0')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



Chilledwater Consumption shows a nice variation along with the seasons showing higher consumption in the summer months especially during the 9th month.

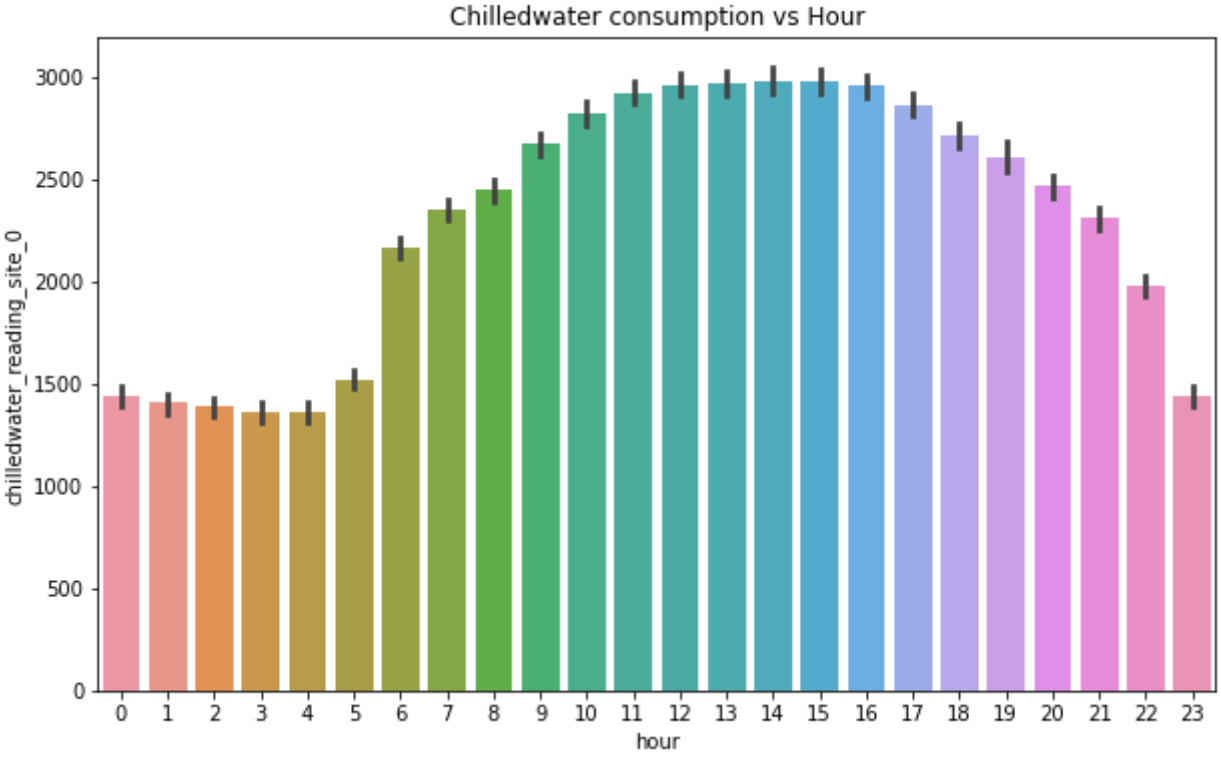
```
df_train_site_0_meter_1['hour']=df_train_site_0_meter_1['timestamp'].dt.hour
df_train_site_0_meter_1['weekday']=df_train_site_0_meter_1['timestamp'].dt.weekday
```

```
z=df_train_site_0_meter_1
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.ylabel('chilledwater_reading_site_0')
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



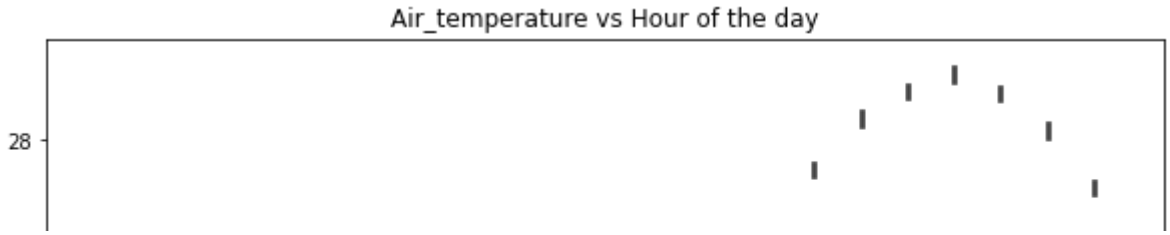
Chilledwater Consumption at site 0 shows lower trend during the weekend as compared to the weekdays

```
z=df_train_site_0_meter_1
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.ylabel('chilledwater_reading_site_0')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```



Chilledwater consumption shows a good hourly variation also and is peaking during the day time.

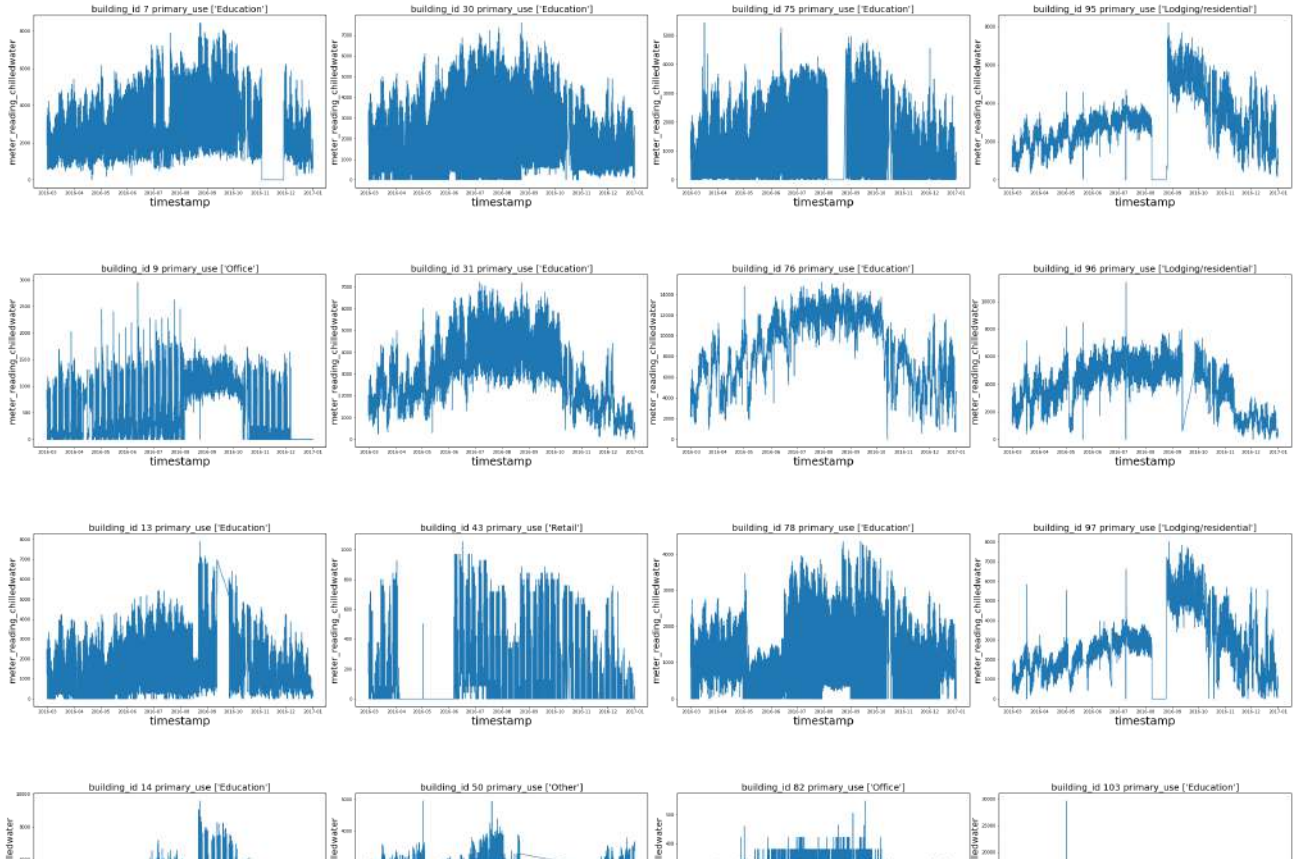
```
z=df_train_site_0_meter_1
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.ylabel('air_temperature_site_0')
plt.title('Air_temperature vs Hour of the day')
plt.show()
```



Weather Timestamp

Weather timestamp is not in alignment with the local timestamp of the hourly chilledwater readings as the temperature peaks around 19:00 pm.

```
fig,axes=plt.subplots(nrows=6,ncols=4,figsize=(50,60))
for i in range(df_train_site_0_meter_1['building_id'].nunique()):
 g=df_train_site_0_meter_1['building_id'].unique()[i]
 z=df_train_site_0_meter_1[df_train_site_0_meter_1['building_id']==g]
 ax=axes[i%6][i//6]
 ax.plot(z['timestamp'],z['meter_reading'])
 ax.set_xlabel('timestamp',fontsize=25)
 ax.set_ylabel('meter_reading_chilledwater',fontsize=20)
 k=z[z['building_id']==g]
 ax.set_title('building_id {} primary_use {}'.format(g,k['primary_use'].unique()),fontsize
 plt.subplots_adjust(hspace=0.5,wspace=0.1)
```



Building 75,95,97,43,98,60 are showing constant zero meter readings which needs to be filtered out.

Building 7 is showing zero reading in the 11th month but it shows meter reading in the 12th month which implies that it is an anomaly observed in the meter reading..

Building 103,60,98 are showing very high consumption for the 5th and the 7th month which needs to be removed as it is an anomaly.


```
#From here on starting the analysis for the 1st site
```

```
df_train_site_1=df_train_merge.loc[df_train_merge['site_id']==1]
```

```
df_train_site_1.isnull().sum()/df_train_site_1.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           0.25
floor_count          0.00
air_temperature      0.00
cloud_coverage       0.81
dew_temperature      0.00
precip_depth_1_hr    1.00
sea_level_pressure   0.01
wind_direction       0.00
```

```
wind_speed           0.00
dtype: float64
```

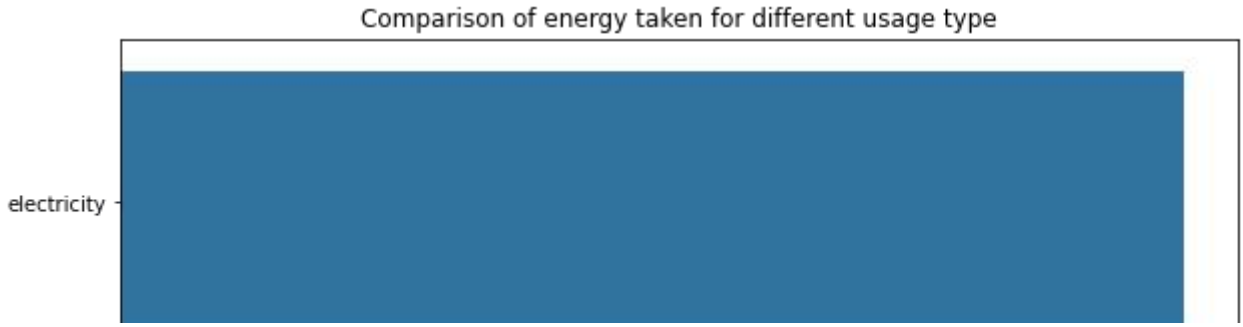
1. The null values for site 1 are not greater than 1% and we need to impute these values as they can be used in the training process.

```
df_corr_1=df_train_site_1.corr()
df_corr_1.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	-0.28	nan	-0.16	-0.16	-0.05	0.00
meter_reading	-0.28	1.00	nan	0.55	-0.09	0.33	-0.02
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	-0.16	0.55	nan	1.00	-0.22	0.57	-0.00
year_built	-0.16	-0.09	nan	-0.22	1.00	-0.05	0.00
floor_count	-0.05	0.33	nan	0.57	-0.05	1.00	-0.00
air_temperature	0.00	-0.02	nan	-0.00	0.00	-0.00	1.00
cloud_coverage	-0.00	-0.01	nan	0.00	-0.00	0.00	-0.09
dew_temperature	0.00	-0.05	nan	-0.00	0.00	-0.00	0.82
precip_depth_1_hr	nan	nan	nan	nan	nan	nan	nan
sea_level_pressure	-0.00	-0.01	nan	-0.00	-0.00	-0.00	0.01
wind_direction	0.00	0.00	nan	-0.00	0.00	-0.00	0.04
wind_speed	0.00	0.03	nan	0.00	-0.00	0.00	0.17

The meter readings from the correlation plot does not show high correlation with any of the features.

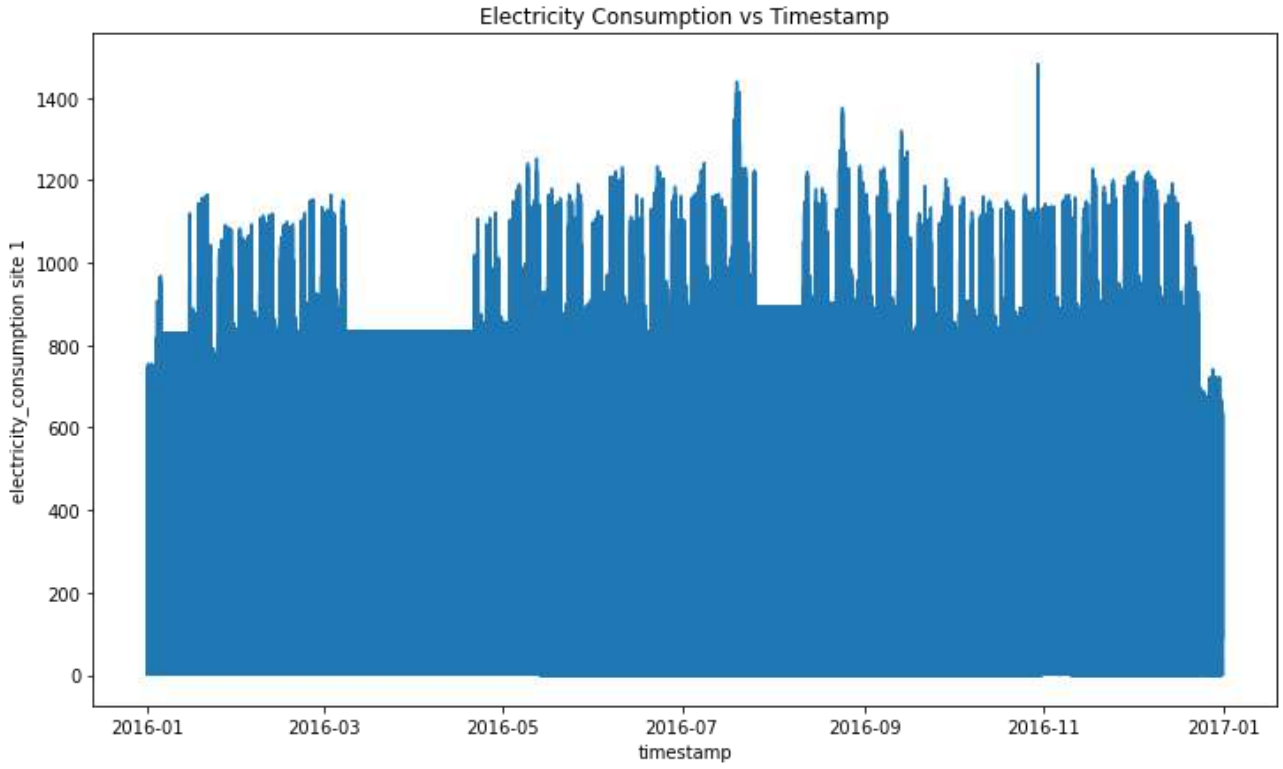
```
z=df_train_site_1.groupby(['meter'])
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z['meter_reading'].mean().reset_index(),x='meter_reading',y='meter'
plt.xlabel('mean_meter_reading_site_1')
plt.ylabel('meter')
plt.title('Comparison of energy taken for different usage type')
plt.show()
```



This site consumes energy for electricity and hotwater usage only with electricity consuming having higher energy requirements

```
df_train_site_1_meter_0=df_train_site_1.loc[df_train_site_1['meter']=='electricity']
df_train_site_1_meter_3=df_train_site_1.loc[df_train_site_1['meter']=='hotwater']
```

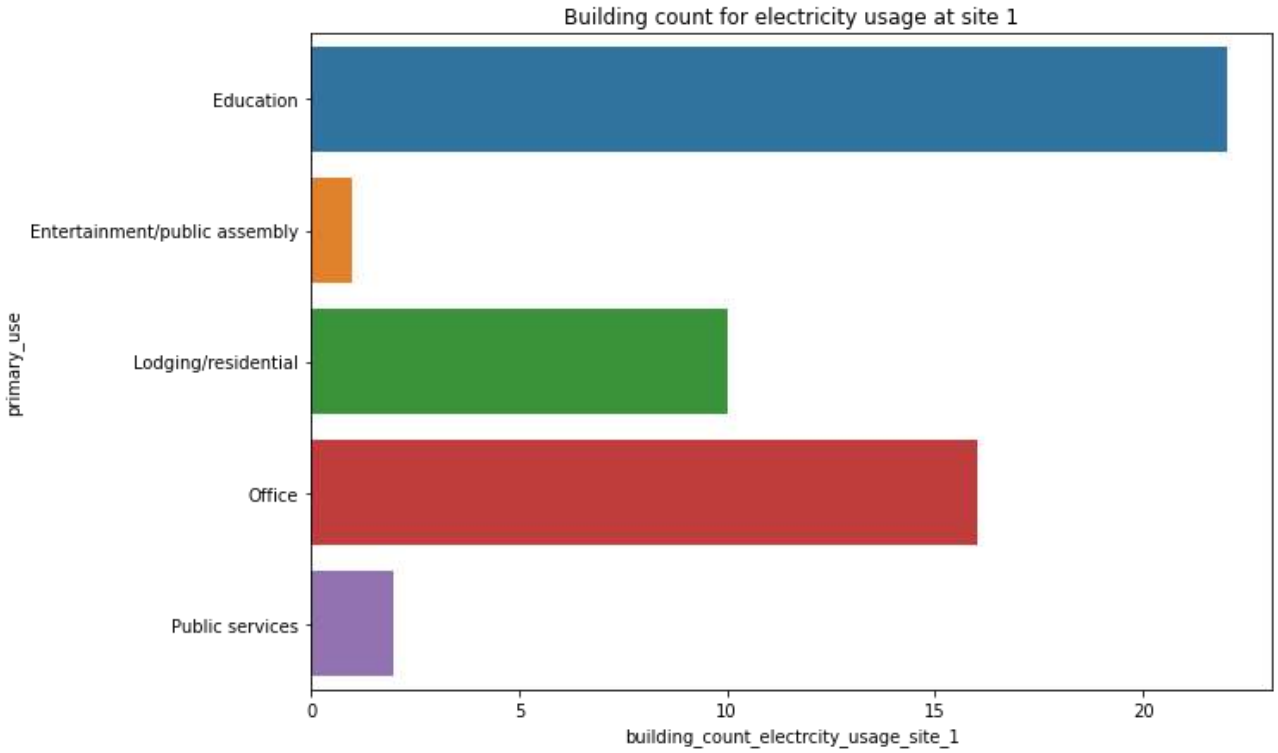
```
fig,ax=plt.subplots(figsize=(12,7))
plt.plot(df_train_site_1_meter_0['timestamp'],df_train_site_1_meter_0['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('electricity_consumption site 1')
plt.title('Electricity Consumption vs Timestamp')
plt.show()
```



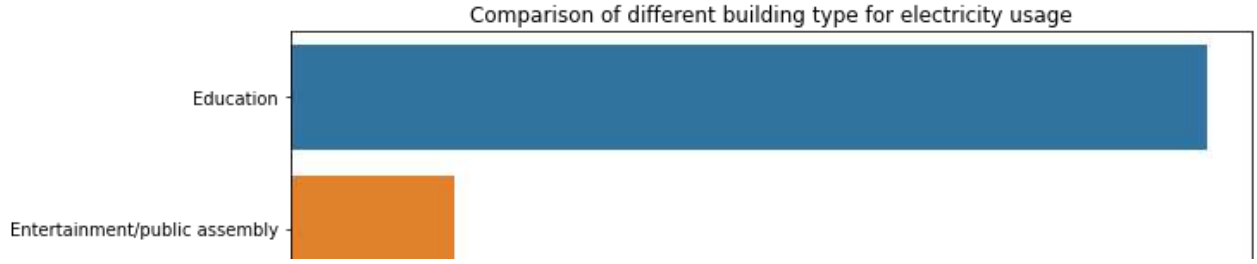
This is a rough plot shows electrical consumption variations over the timestamp.

```
fig,ax=plt.subplots(figsize=(10,7))
g=df_train_site_1_meter_0.groupby(['primary_use'])
sns.barplot(ax=ax,data=g['building_id'].nunique().reset_index(),x='building_id',y='primary
```

```
plt.xlabel('building_count_electrcity_usage_site_1')
plt.ylabel('primary_use')
plt.title('Building count for electricity usage at site 1')
plt.show()
```



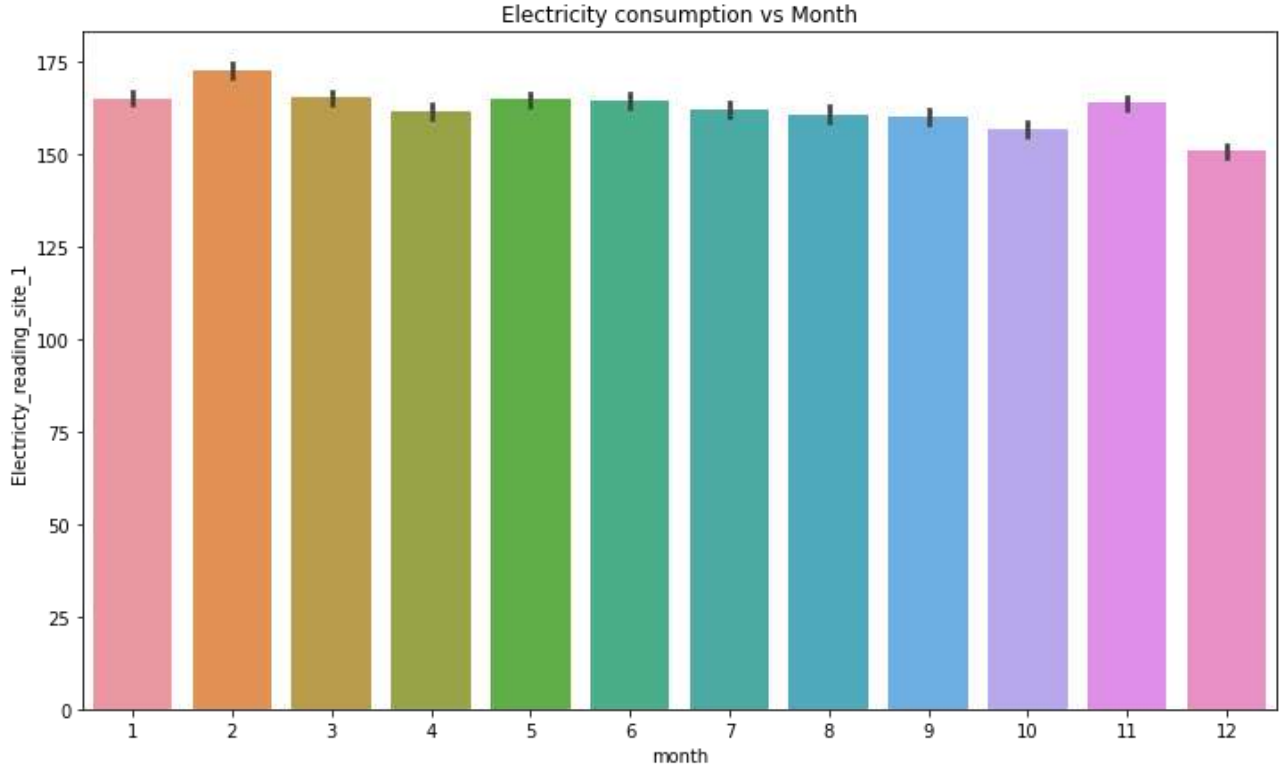

```
fig,ax=plt.subplots(figsize=(10,7))
g=df_train_site_1_meter_0.groupby(['primary_use'])
sns.barplot(ax=ax,data=g['meter_reading'].mean().reset_index(),x='meter_reading',y='primar
plt.xlabel('mean_electricity_reading_site_1')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for electricity usage')
plt.show()
```



Here Educational buildings are having the highest electricity consumption


```
df_train_site_1_meter_0['month']=df_train_site_1_meter_0['timestamp'].dt.month
```

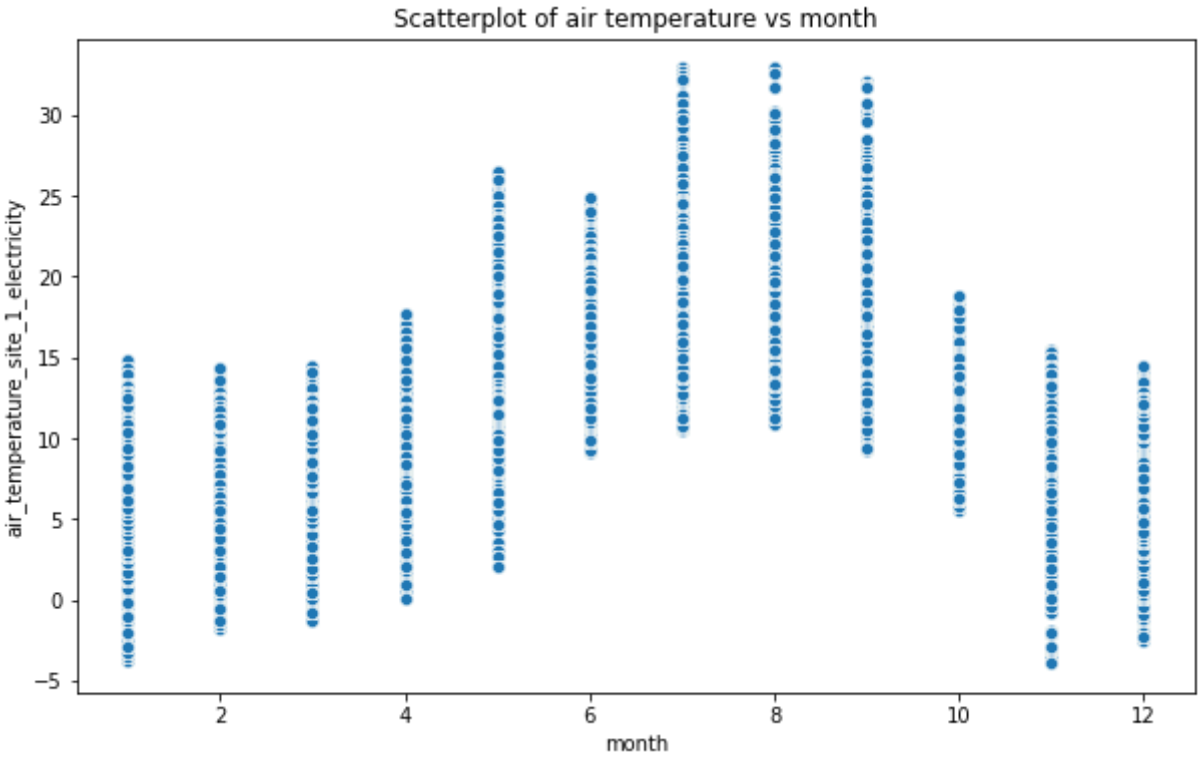
```
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=df_train_site_1_meter_0,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('Electricty_reading_site_1')
plt.title('Electricity consumption vs Month')
plt.show()
```



Here we can see that the consumption is relatively higher for the winter months as compared to the other.This can happen due to various reasons like the extreme temperature energy requirements type of the building.

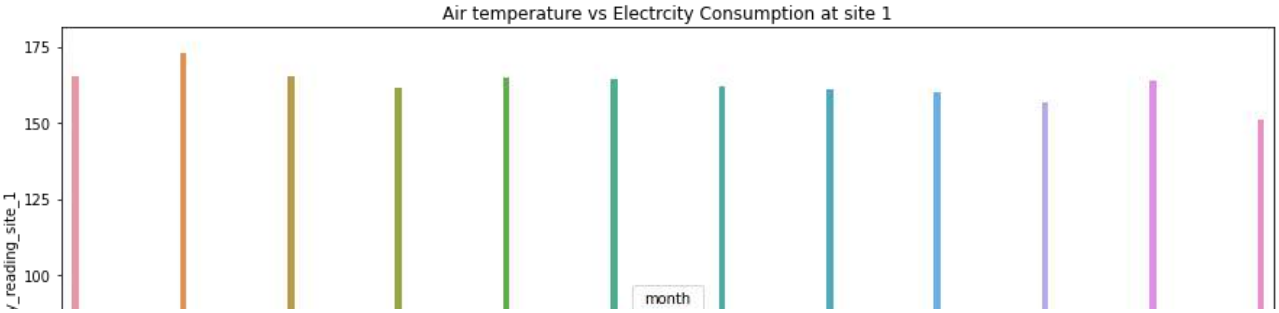
```
fig,ax=plt.subplots(figsize=(10,6))
sns.scatterplot(ax=ax,data=df_train_site_1_meter_0,x='month',y='air_temperature')
plt.ylabel('air_temperature_site_1_electricity')
plt.xlabel('month')
```

```
plt.title('Scatterplot of air temperature vs month')
plt.show()
```



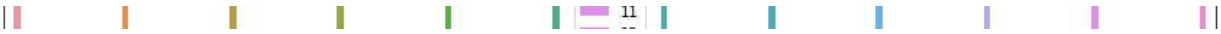
This scatter plot of air temperature vs month shows that the temperature during the winter months reaches to extreme as compared to the summer months which can be a reason for higher energy consumption during winter month.

```
z=df_train_site_1_meter_0.groupby(['month'])
z=z['air_temperature','meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(15,7))
sns.barplot(ax=ax,data=z,y='meter_reading',x='air_temperature',order=z['air_temperature'],
plt.xlabel('mean_air_temperature')
plt.ylabel('mean_electricity_reading_site_1')
plt.title('Air temperature vs Electrcity Consumption at site 1')
plt.show()
```

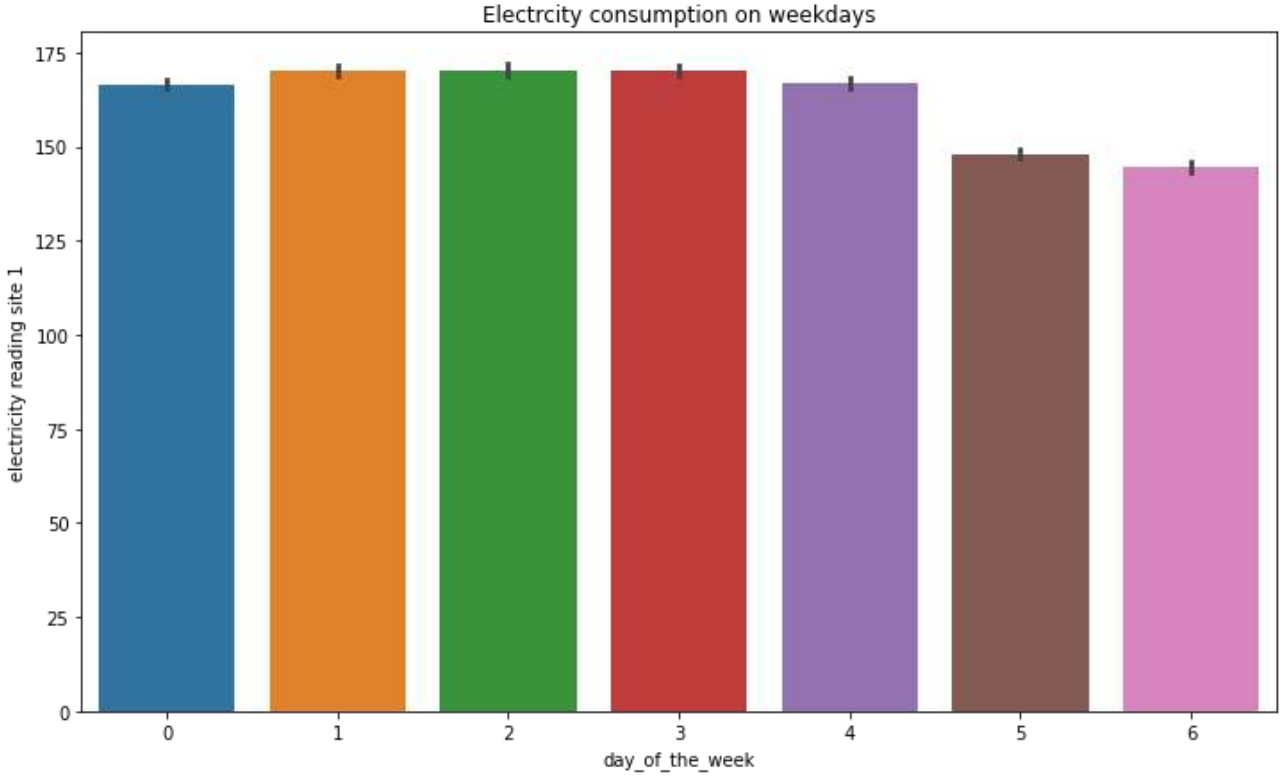


This is a plot for mean temperature vs mean electricity reading

1. For lower temperatures more energy is consumed for the heating purpose which can be a major reason for higher electricity consumption in the winter month.



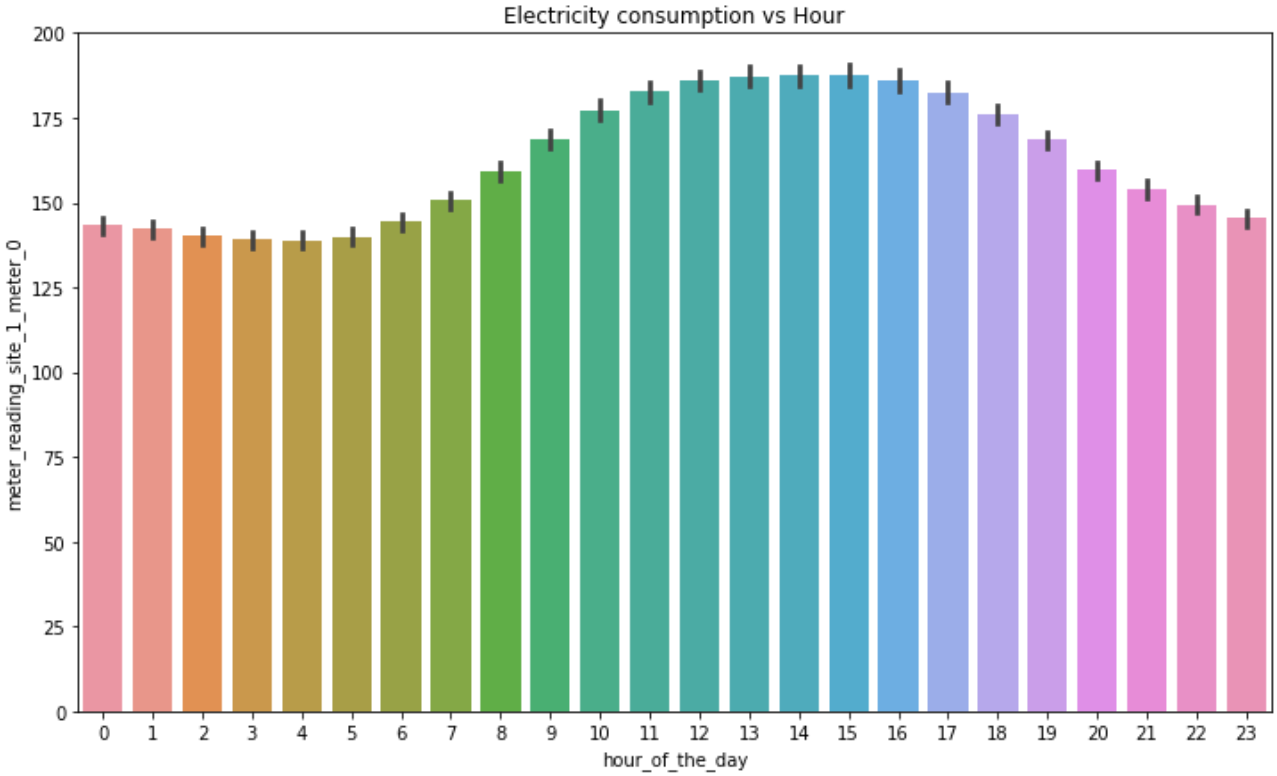

```
df_train_site_1_meter_0['weekday']=df_train_site_1_meter_0['timestamp'].dt.weekday
df_train_site_1_meter_0['hour']=df_train_site_1_meter_0['timestamp'].dt.hour


fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=df_train_site_1_meter_0,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity reading site 1')
plt.title('Electrcity consumption on weekdays')
plt.show()
```



Electricity Consumption is lower on the weekend as it is a holiday

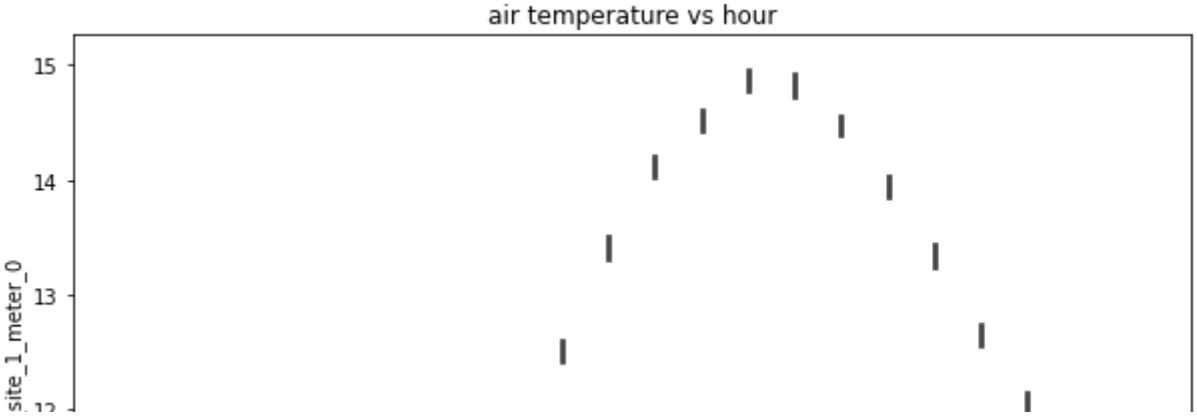
```
fig,ax=plt.subplots(figsize=(12,7))
```

```
sns.barplot(ax=ax,data=df_train_site_1_meter_0,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('meter_reading_site_1_meter_0')
plt.title('Electricity consumption vs Hour')
plt.show()
```



Electricity Consumption peaks during the day time as the occupancy is highest during that time which is proportional to the higher energy demands.

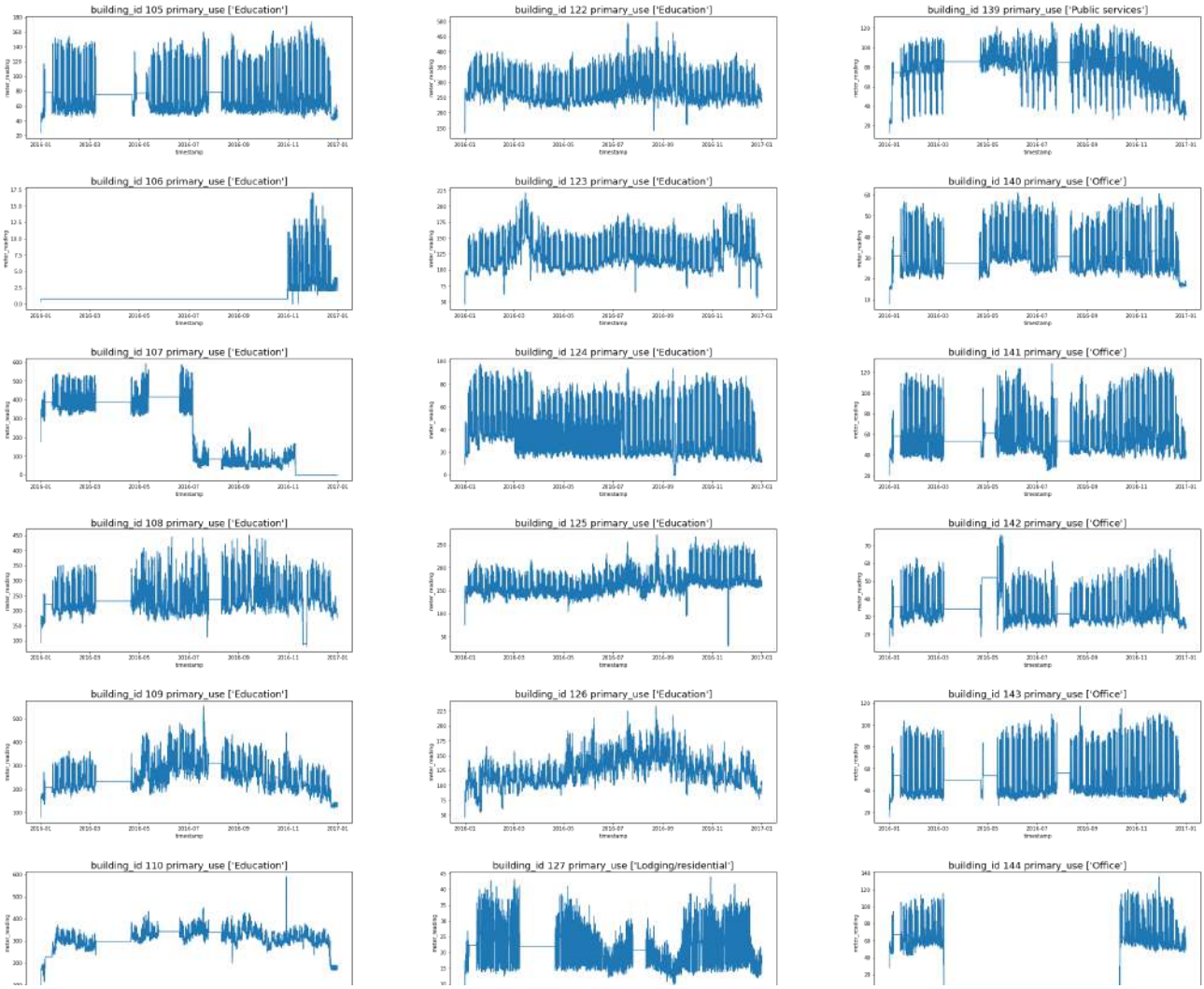
```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_1_meter_0,x='hour',y='air_temperature')
plt.ylabel('air_temperature_site_1_meter_0')
plt.xlabel('hour_of_the_day')
plt.title('air temperature vs hour')
plt.show()
```



Weather Timestamp

- This plot shows that the weather timestamp is in alignment with the local timestamp for hourly electrical readings.
- The temperature peaks during the afternoon.

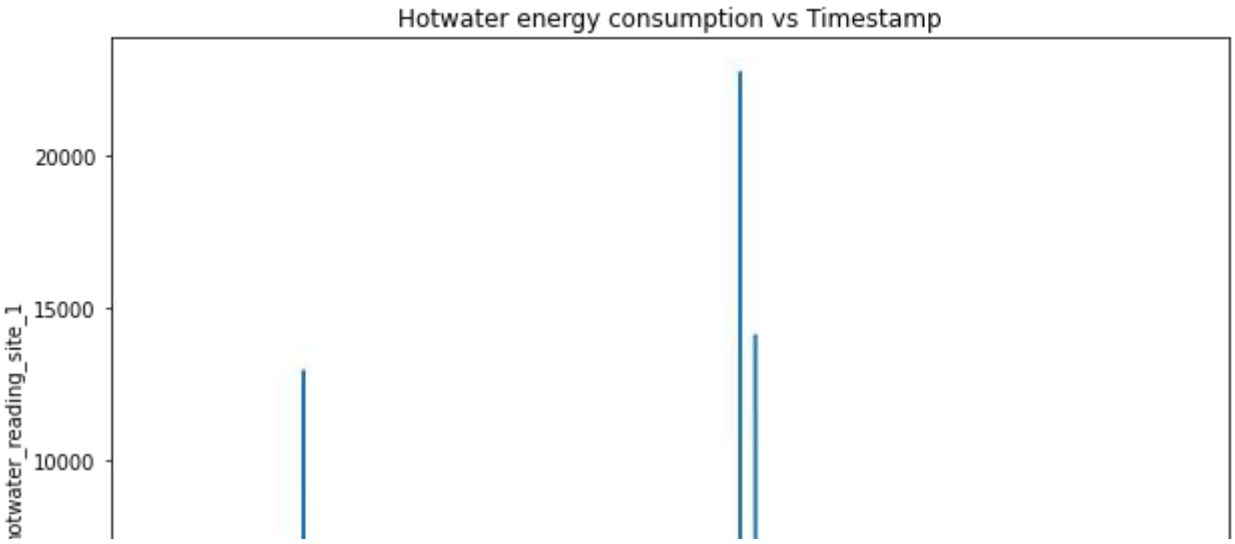
```
fig,ax=plt.subplots(figsize=(40,100),nrows=17,ncols=3)
for i in range(df_train_site_1_meter_0['building_id'].nunique()):
  g=df_train_site_1_meter_0['building_id'].unique()[i]
  axes=ax[i%17][i//17]
  z=df_train_site_1_meter_0.loc[df_train_site_1_meter_0['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('meter_reading')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```



Important observations

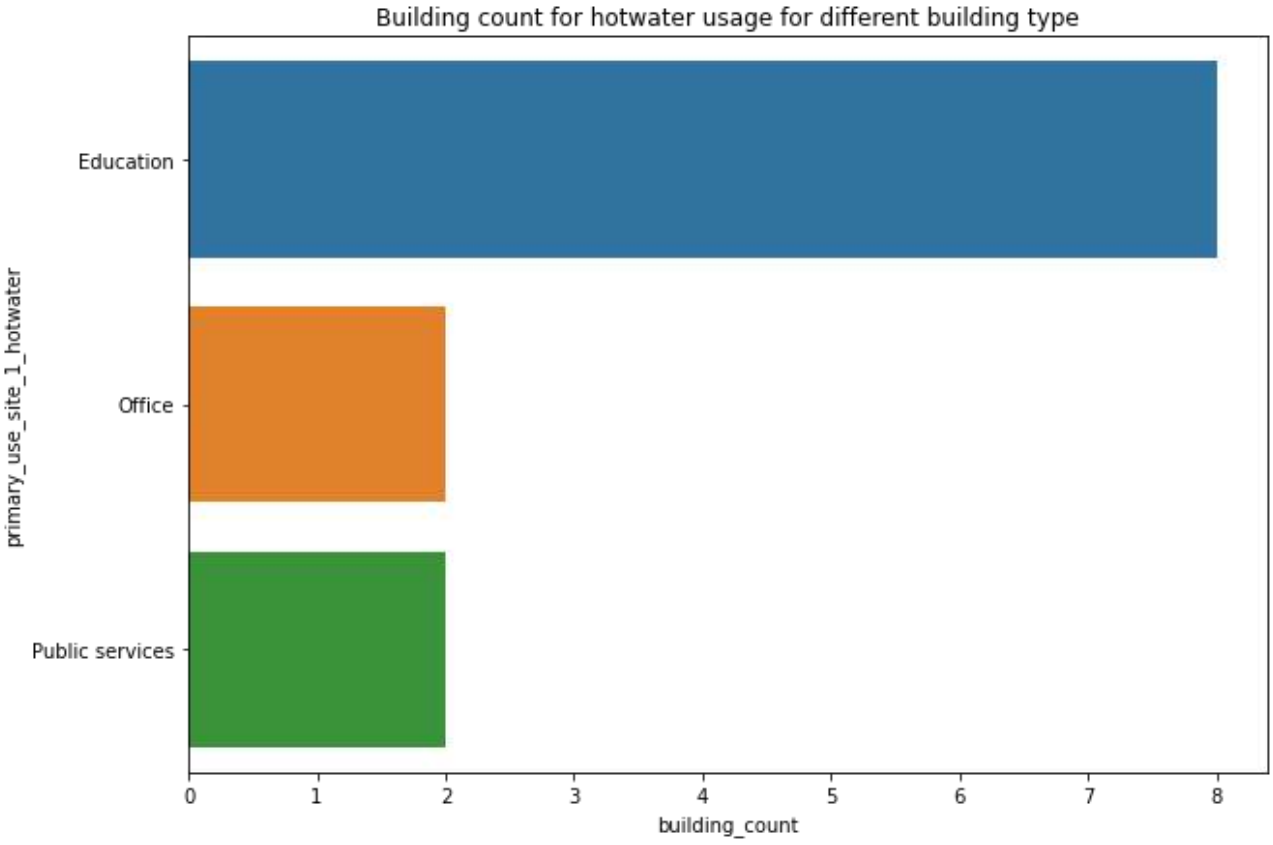
- Here we can observe that the buildings at site 1 which are consuming electricity are showing constant streaks of values for certain months which is definitely an anomaly and we need to remove those values.We would not want our model to fit on these constant values as these are faulty readings taken from the meter and it would make our model worse.

```
fig,ax=plt.subplots(figsize=(10,7))
ax.plot(df_train_site_1_meter_3['timestamp'],df_train_site_1_meter_3['meter_reading'])
plt.ylabel('hotwater_reading_site_1')
plt.xlabel('timestamp')
plt.title('Hotwater energy consumption vs Timestamp')
plt.show()
```



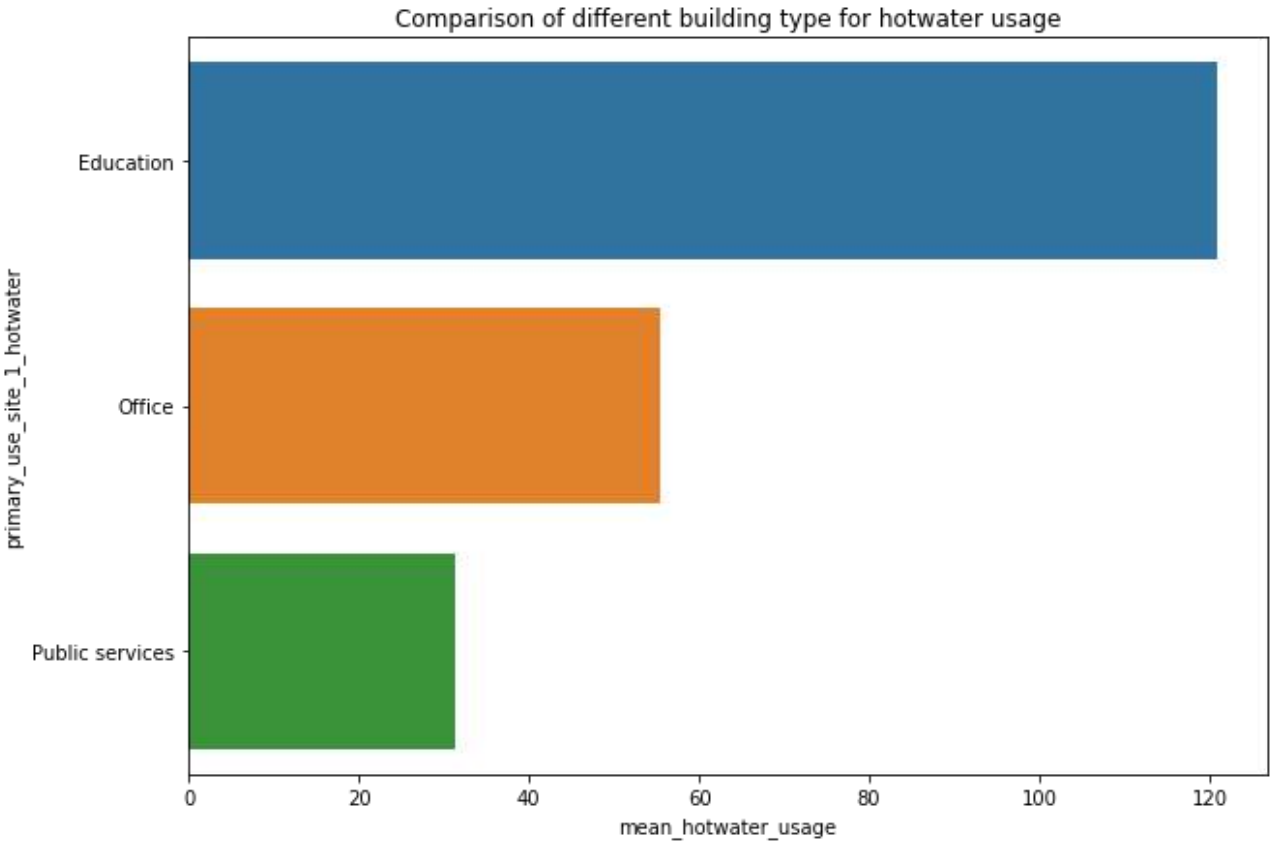
This rough plot for energy usage for hotwater shows 2-3 large peaked values which needs to be removed as it is an outlier.

```
fig,ax=plt.subplots(figsize=(10,7))
g=df_train_site_1_meter_3.groupby(['primary_use'])
sns.barplot(ax=ax,data=g['building_id'].nunique().reset_index(),x='building_id',y='primary
plt.xlabel('building_count')
plt.ylabel('primary_use_site_1_hotwater')
plt.title('Building count for hotwater usage for different building type')
plt.show()
```



This plot shows the building count which are using hotwater at site 1 for different building type.

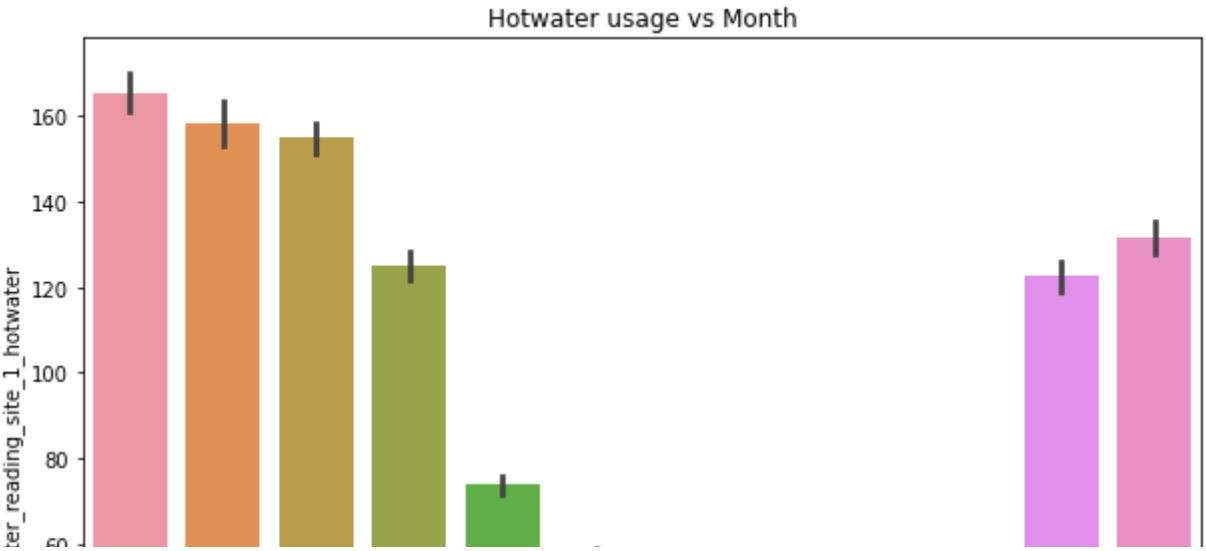
```
fig,ax=plt.subplots(figsize=(10,7))
g=df_train_site_1_meter_3.groupby(['primary_use'])
sns.barplot(ax=ax,data=g['meter_reading'].mean().reset_index(),x='meter_reading',y='primar
plt.xlabel('mean_hotwater_usage')
plt.ylabel('primary_use_site_1_hotwater')
plt.title('Comparison of different building type for hotwater usage')
plt.show()
```



Education shows highest energy required for hotwater usage

```
df_train_site_1_meter_3['month']=df_train_site_1_meter_3['timestamp'].dt.month
```

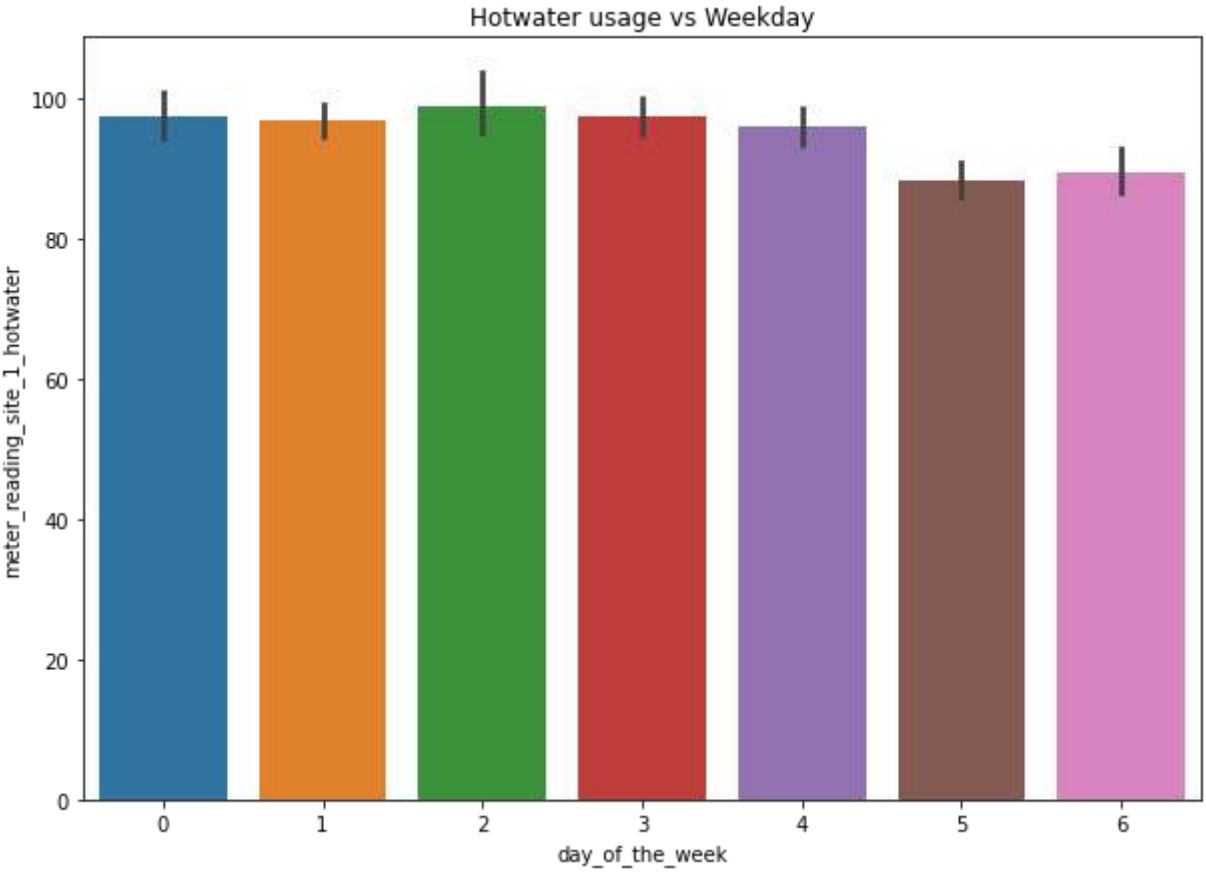
```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_1_meter_3,x='month',y='meter_reading')
plt.ylabel('meter_reading_site_1_hotwater')
plt.xlabel('month')
plt.title('Hotwater usage vs Month')
plt.show()
```



Hotwater usage shows a good variation over the seasons(month)

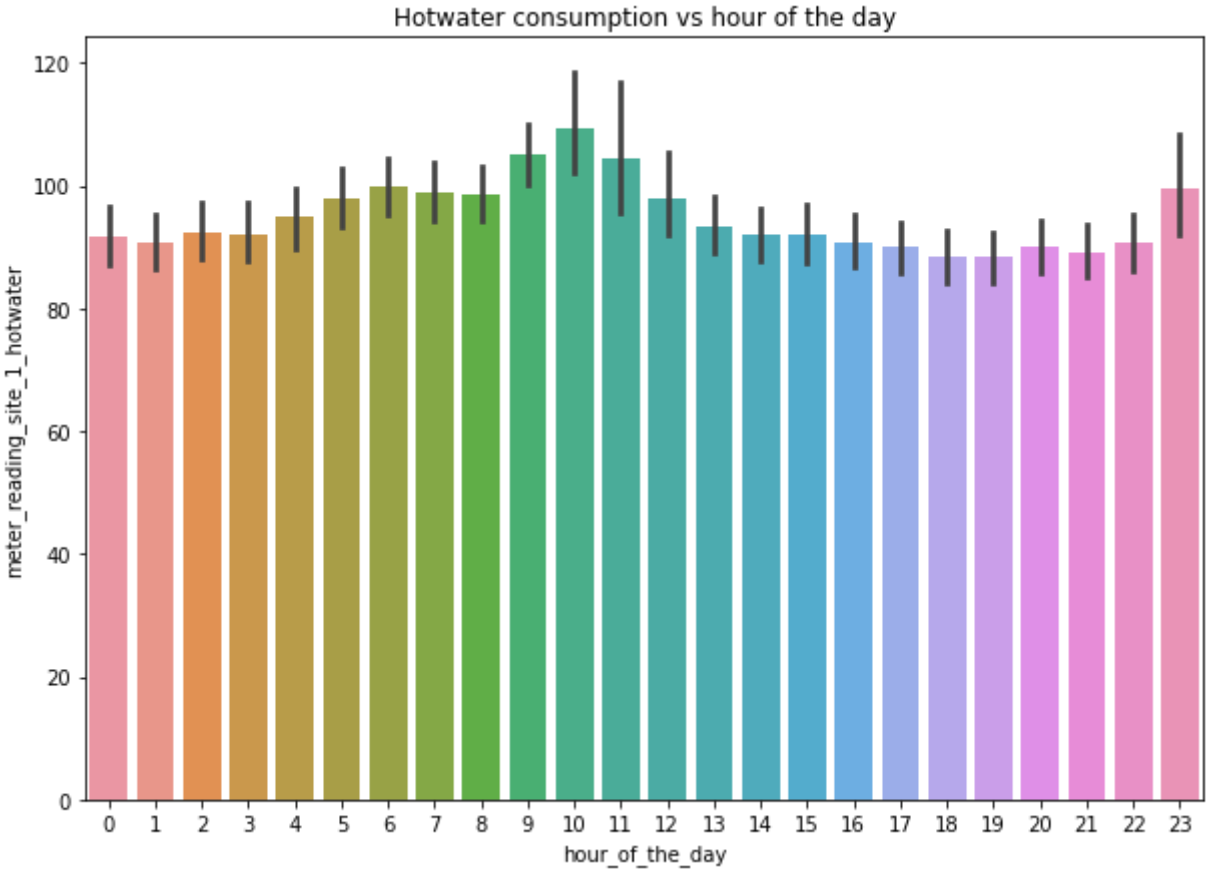
```
df_train_site_1_meter_3['weekday']=df_train_site_1_meter_3['timestamp'].dt.weekday
df_train_site_1_meter_3['hour']=df_train_site_1_meter_3['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_1_meter_3,x='weekday',y='meter_reading')
plt.ylabel('meter_reading_site_1_hotwater')
plt.xlabel('day_of_the_week')
plt.title('Hotwater usage vs Weekday')
plt.show()
```



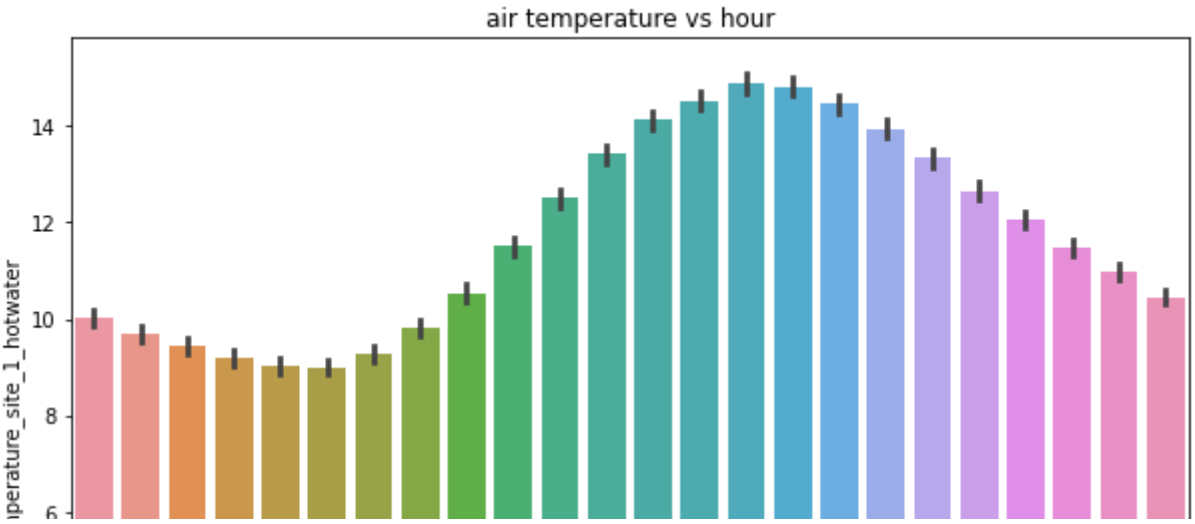
Hotwater usage is lesser for the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_1_meter_3,x='hour',y='meter_reading')
plt.ylabel('meter_reading_site_1_hotwater')
plt.xlabel('hour_of_the_day')
plt.title('Hotwater consumption vs hour of the day')
plt.show()
```



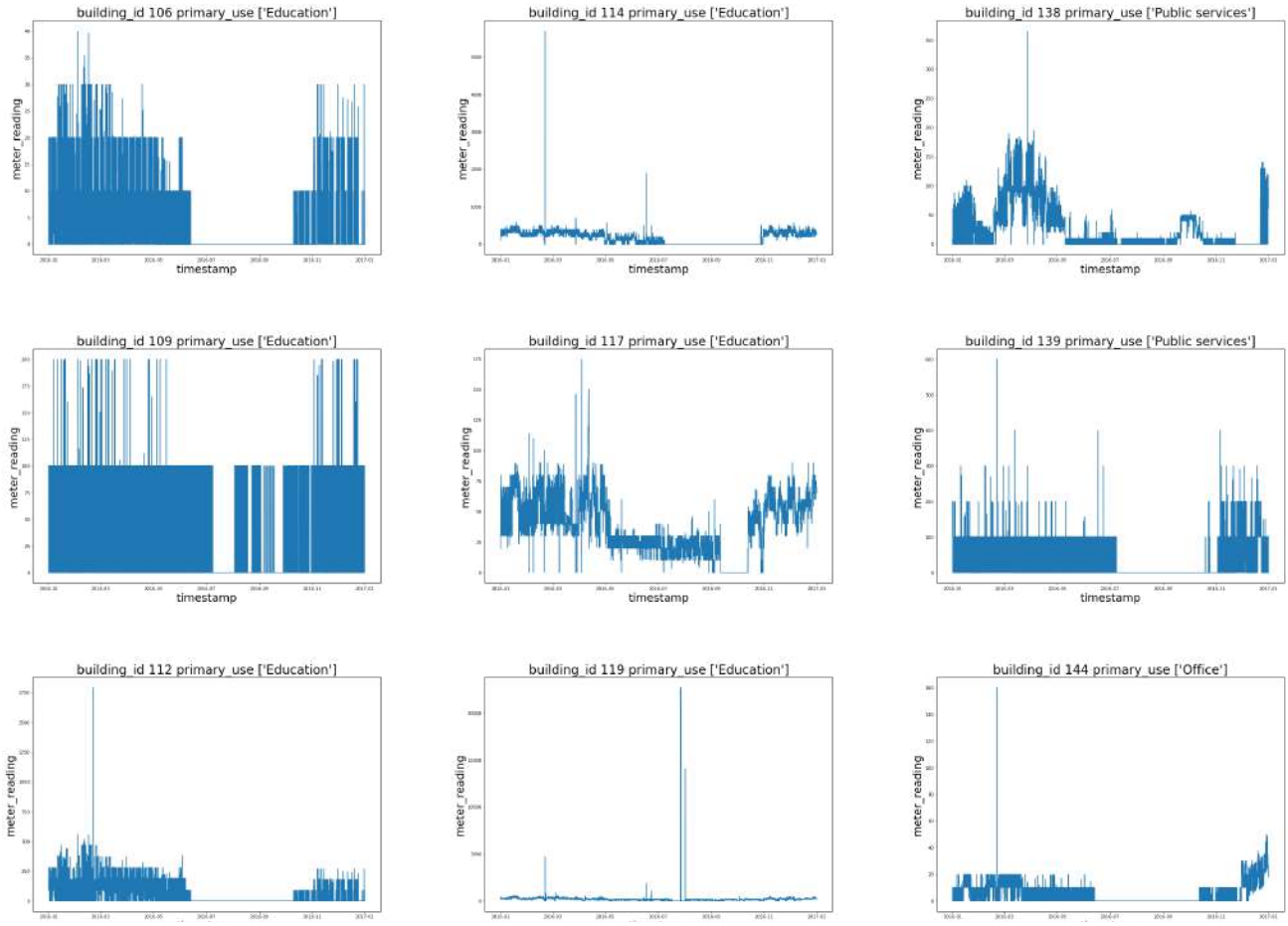
This plot shows us the hourly variations of hotwater usage which shows that it peaks around 11:00 am.

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_1_meter_3,x='hour',y='air_temperature')
plt.ylabel('air_temperature_site_1_hotwater')
plt.xlabel('hour_of_the_day')
plt.title('air temperature vs hour')
plt.show()
```



This plot shows us that the weather timestamp is in alignment with the local timestamp of the hourly hotwater readings.

```
fig,ax=plt.subplots(figsize=(50,50),nrows=4,ncols=3)
for i in range(df_train_site_1_meter_3['building_id'].nunique()):
  g=df_train_site_1_meter_3['building_id'].unique()[i]
  axes=ax[i%4][i//4]
  z=df_train_site_1_meter_3.loc[df_train_site_1_meter_3['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp',fontsize=25)
  axes.set_ylabel('meter_reading',fontsize=25)
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```



Most of the buildings are showing zero hotwater usage from roughly 6-10 month which is a valid reading as these are summer months and we do not need to remove them.

We can also see that many of the buildings are showing very high peaked values for the hotwater usage even during the summer month which is definitely an anomaly and we need to remove them cause we dont want our model to overfit.

```
#Starting the analysis for the 2nd site


df_train_site_2=df_train_merge.loc[df_train_merge['site_id']==2]


df_train_site_2.isnull().sum()/df_train_site_2.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           0.23
floor_count          1.00
air_temperature      0.00
cloud_coverage       0.27
dew_temperature      0.00
precip_depth_1_hr    0.01
sea_level_pressure   0.01
wind_direction       0.07
```

```
wind_speed           0.00
dtype: float64
```

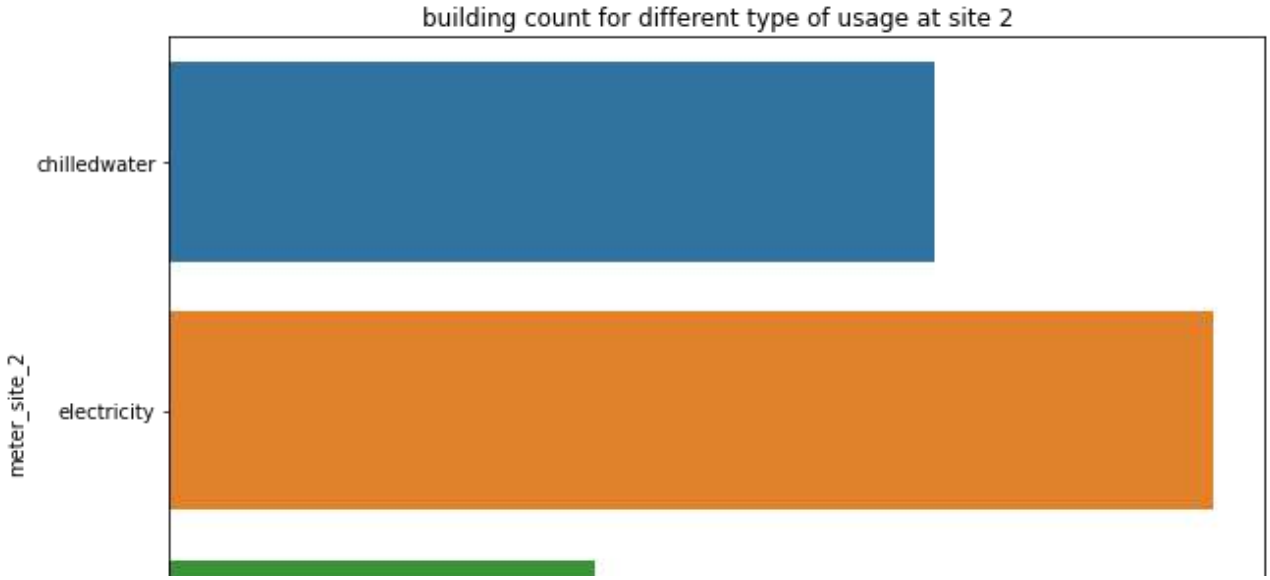
We need to impute the missing values so that it can be used in the training process.This will be handled very well in the preprocessing part

```
df_corr_2=df_train_site_3.corr()
df_corr_2.style.background_gradient(cmap='hot_r').set_precision(2)
```

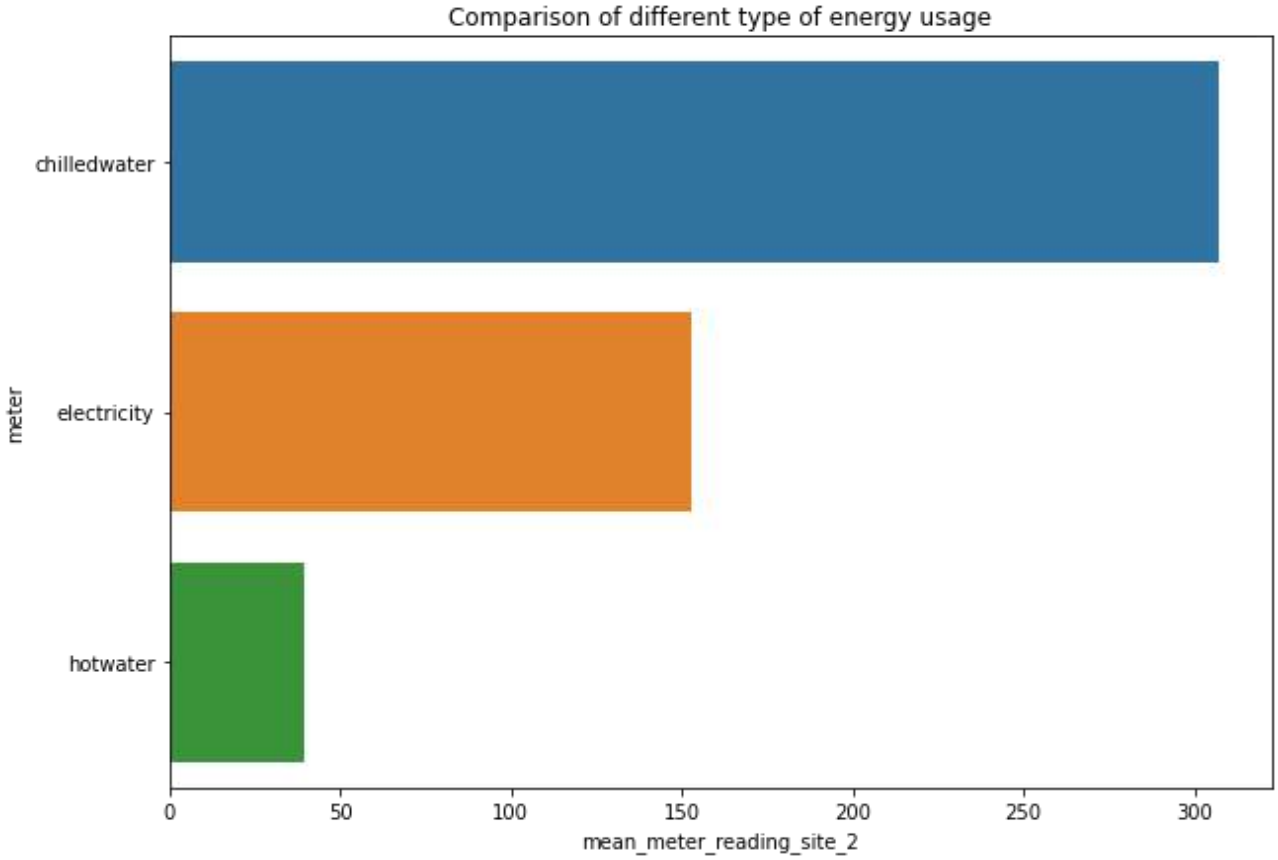
	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.06	nan	0.00	0.12	nan	0.00
meter_reading	0.06	1.00	nan	0.84	0.10	nan	0.03
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.00	0.84	nan	1.00	0.08	nan	0.00
year_built	0.12	0.10	nan	0.08	1.00	nan	-0.00
floor_count	nan	nan	nan	nan	nan	nan	nan
air_temperature	0.00	0.03	nan	0.00	-0.00	nan	1.00
cloud_coverage	-0.00	0.01	nan	0.00	0.00	nan	0.14
dew_temperature	0.00	0.04	nan	0.00	-0.00	nan	0.89
precip_depth_1_hr	-0.00	-0.00	nan	-0.00	-0.00	nan	-0.01
sea_level_pressure	0.00	-0.01	nan	-0.00	0.00	nan	-0.29
wind_direction	0.00	-0.00	nan	0.00	-0.00	nan	-0.10
wind_speed	-0.00	-0.01	nan	-0.00	0.00	nan	-0.06
month	0.00	-0.01	nan	0.00	-0.00	nan	0.27
weekday	-0.00	-0.03	nan	-0.00	-0.00	nan	-0.02
hour	0.00	0.01	nan	0.00	0.00	nan	0.14

Here meter reading shows a very high correlation with the square feet of the building which implies that as the area increases the energy usage also increases linearly.

```
z=df_train_site_2.groupby(['meter'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=z,x='building_id',y='meter')
plt.xlabel('building_count')
plt.ylabel('meter_site_2')
plt.title('building count for different type of usage at site 2')
plt.show()
```



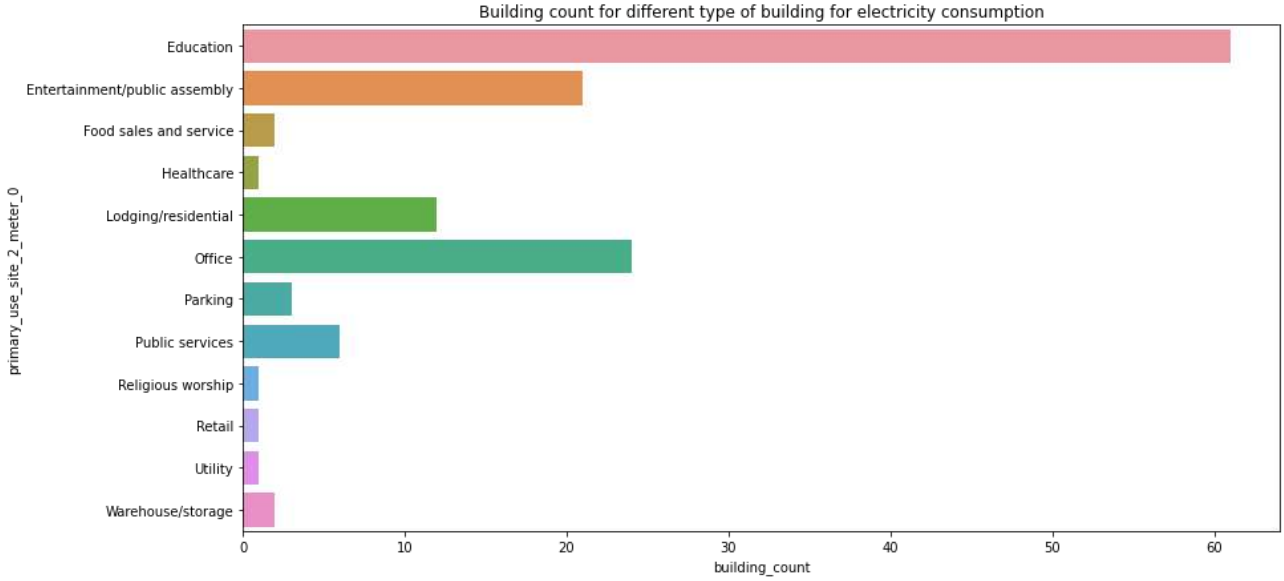

```
z=df_train_site_2.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading_site_2')
plt.ylabel('meter')
plt.title('Comparison of different type of energy usage')
plt.show()
```



Chilledwater is the highest consumption of energy usage at site 2

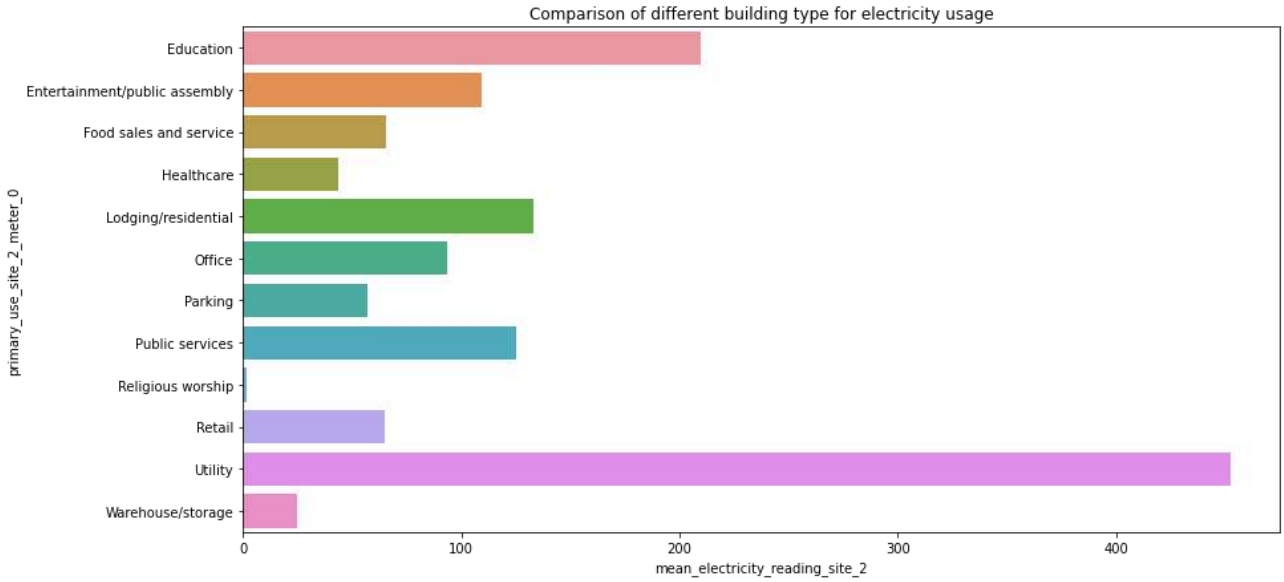
```
df_train_site_2_meter_0=df_train_site_2.loc[df_train_site_2['meter']=='electricity']
df_train_site_2_meter_1=df_train_site_2.loc[df_train_site_2['meter']=='chilledwater']
```

```
df_train_site_2_meter_3=df_train_site_2.loc[df_train_site_2['meter']=='hotwater']


df_train_site_2_meter_0['month']=df_train_site_2_meter_0['timestamp'].dt.month
df_train_site_2_meter_0['weekday']=df_train_site_2_meter_0['timestamp'].dt.weekday
df_train_site_2_meter_0['hour']=df_train_site_2_meter_0['timestamp'].dt.hour


fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_2_meter_0.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_site_2_meter_0')
plt.title('Building count for different type of building for electricity consumption')
plt.show()
```



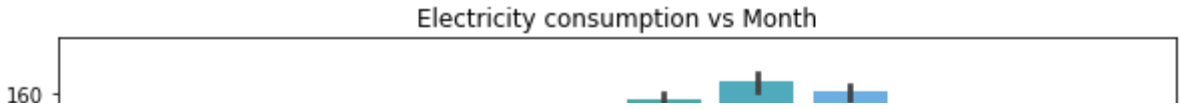
It shows the building count for diffrent type of buildings using electricity

```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_2_meter_0.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_reading_site_2')
plt.ylabel('primary_use_site_2_meter_0')
plt.title('Comparison of different building type for electricity usage')
plt.show()
```



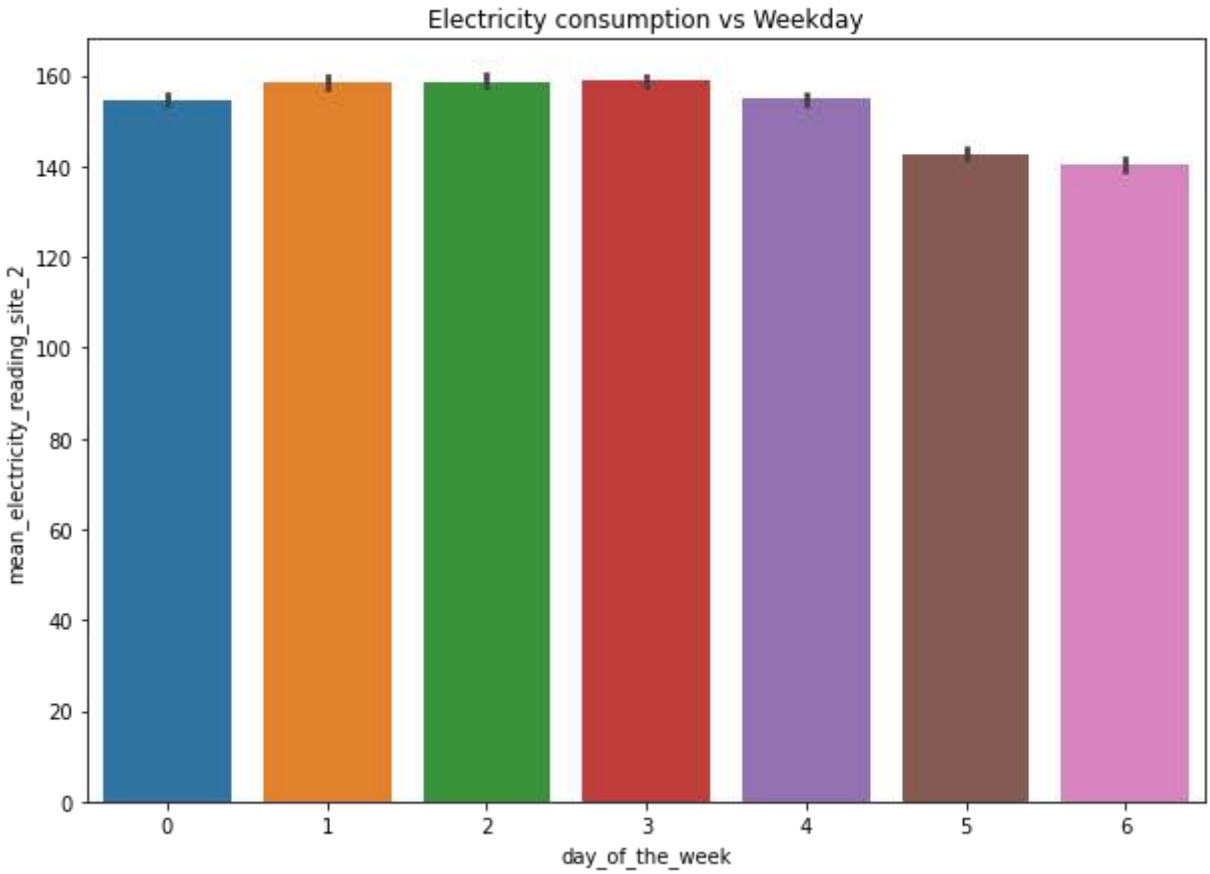
Although utility buildings are small in number but it is having the highest electricity consumption.We will investigate it further when we plot the energy consumption of all the buildings with the timestamp.

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_0,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('mean_electricity_reading_site_2')
plt.title('Electricity consumption vs Month')
plt.show()
```



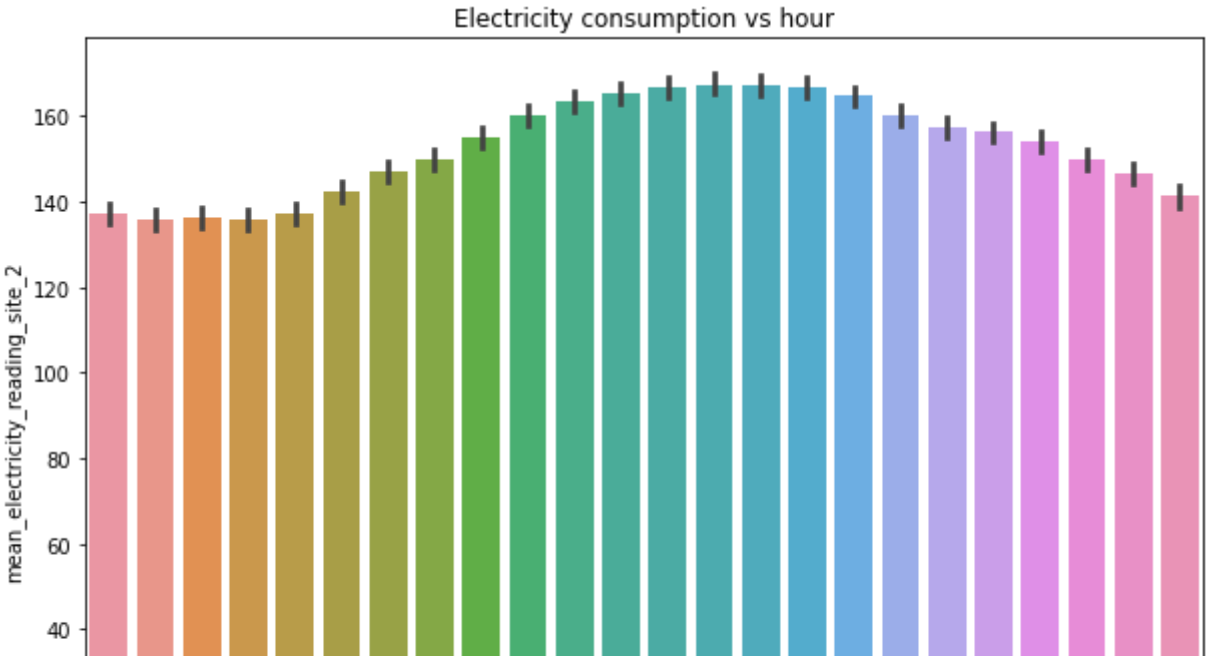
Seasonal variations of electricity consumption

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_0,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('mean_electricity_reading_site_2')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



Electricity Consumption is less over the weekend than the weekday.

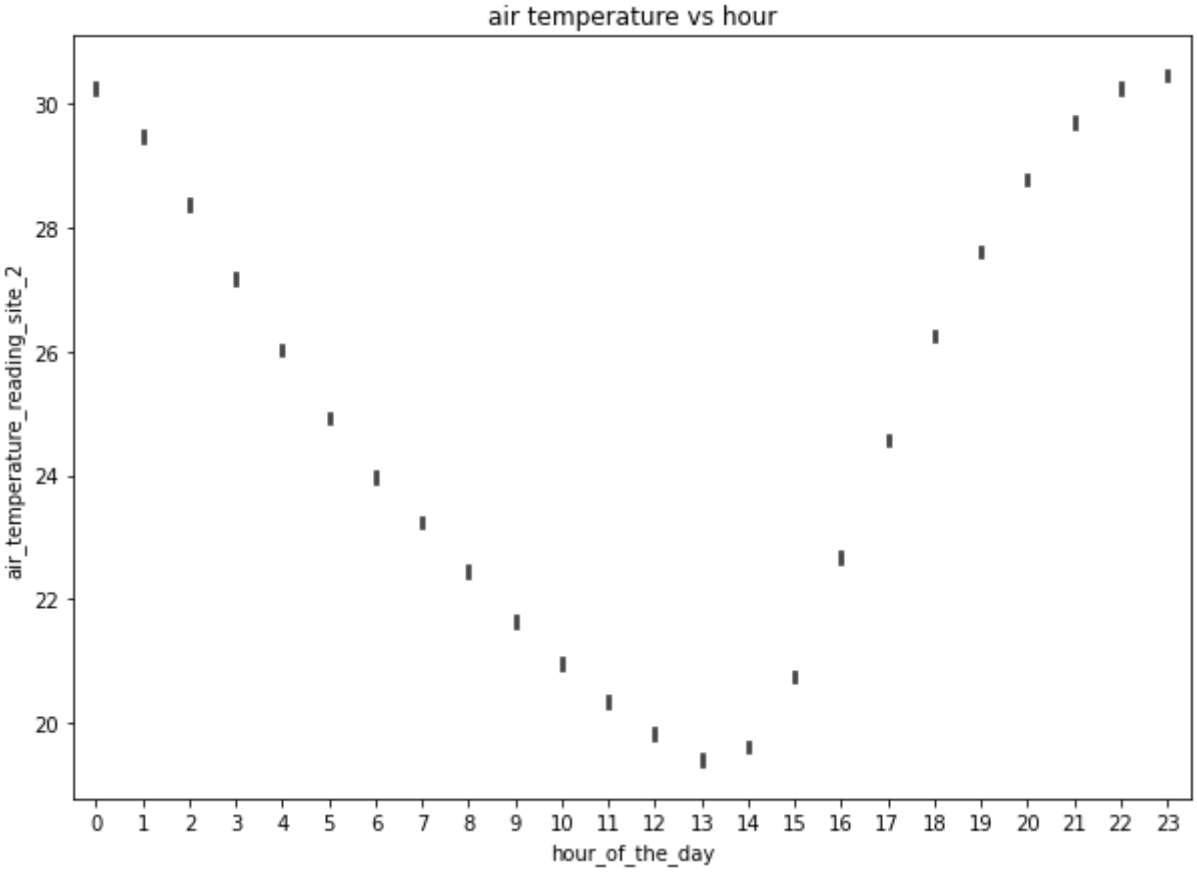
```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_0,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('mean_electricity_reading_site_2')
plt.title('Electricity consumption vs hour')
plt.show()
```



Electricity Consumption peaks over the afternoon hours and decreses gradually after that as the occupancy might be the highest during the afternoon hours.The weather conditions might also affect these energy consumption.

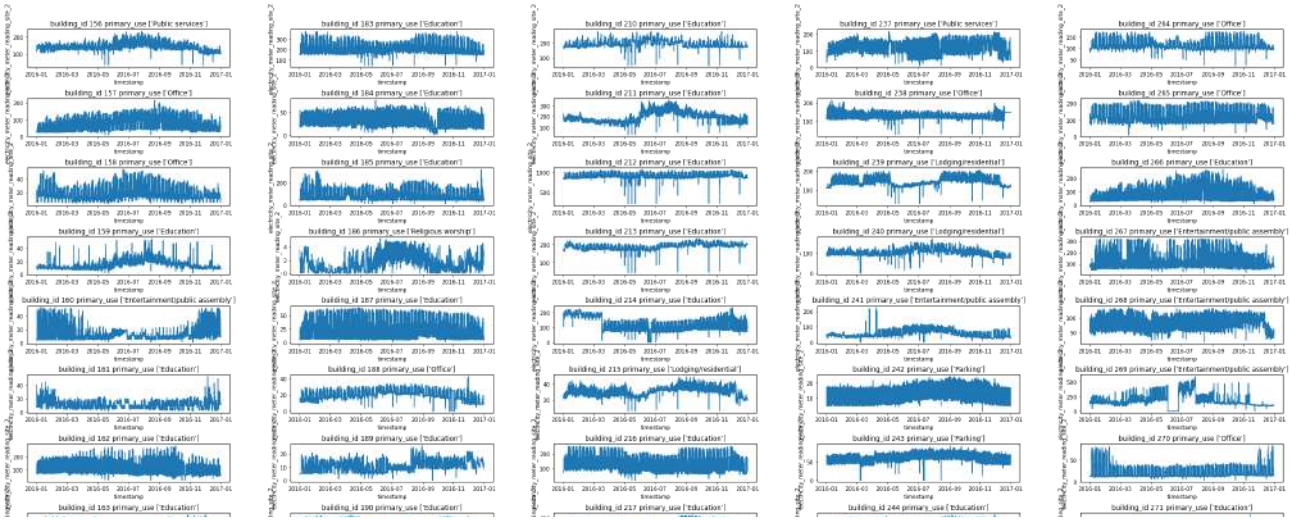
hour of the day

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_0,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_reading_site_2')
plt.title('air temperature vs hour')
plt.show()
```



This plot shows us that the weather timestamp is not in alignment with the local timestamp of hourly readings for site 2 as the air temperature peaks around 23:00 pm

```
fig,ax=plt.subplots(figsize=(40,60),nrows=27,ncols=5)
for i in range(df_train_site_2_meter_0['building_id'].nunique()):
  g=df_train_site_2_meter_0['building_id'].unique()[i]
  axes=ax[i%27][i//27]
  z=df_train_site_2_meter_0.loc[df_train_site_2_meter_0['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_meter_reading_site_2')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
  plt.subplots_adjust(hspace=0.8,wspace=0.3)
```

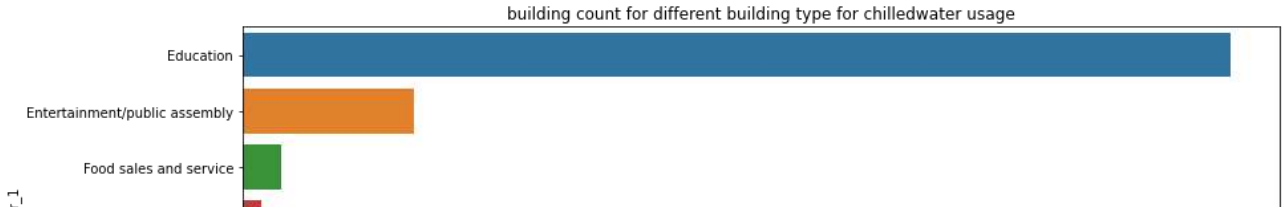


Important Observations

1. Building 218,180 are showing streaks of constant values and zeros which we need to remove as it is an anomaly.
2. Building 247,248,250,278,258,177,269 are also streak of constant zero values for different month which needs to be removed too as it might be an anomaly.

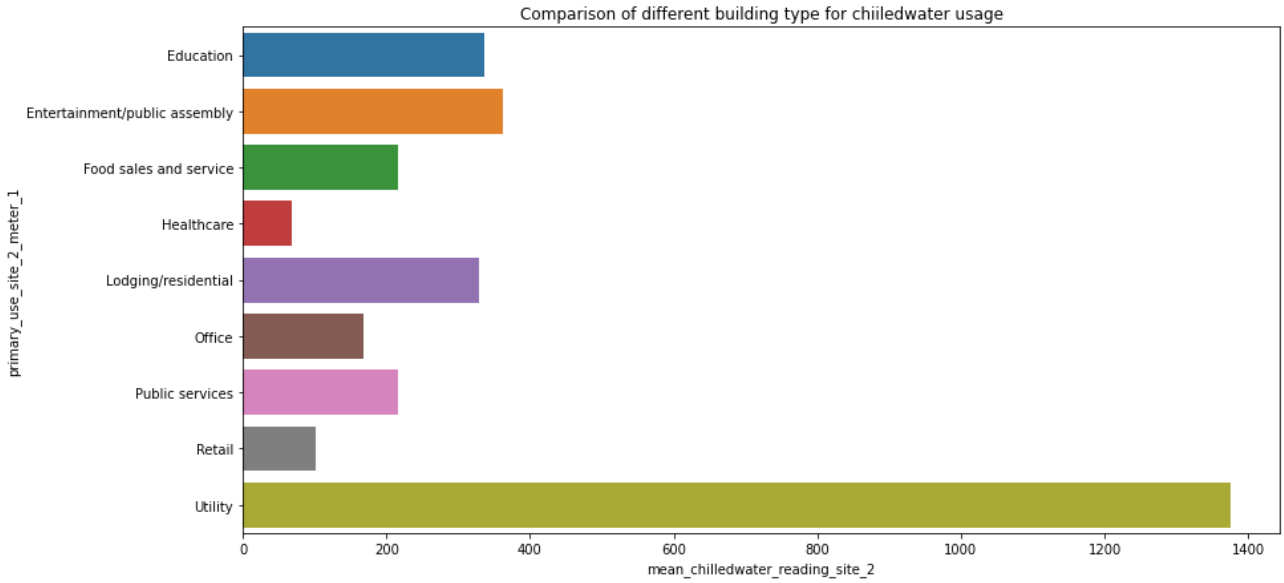
```
df_train_site_2_meter_1['month']=df_train_site_2_meter_1['timestamp'].dt.month
df_train_site_2_meter_1['weekday']=df_train_site_2_meter_1['timestamp'].dt.weekday
df_train_site_2_meter_1['hour']=df_train_site_2_meter_1['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_2_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_site_2_meter_1')
plt.title('building count for different building type for chilledwater usage')
plt.show()
```



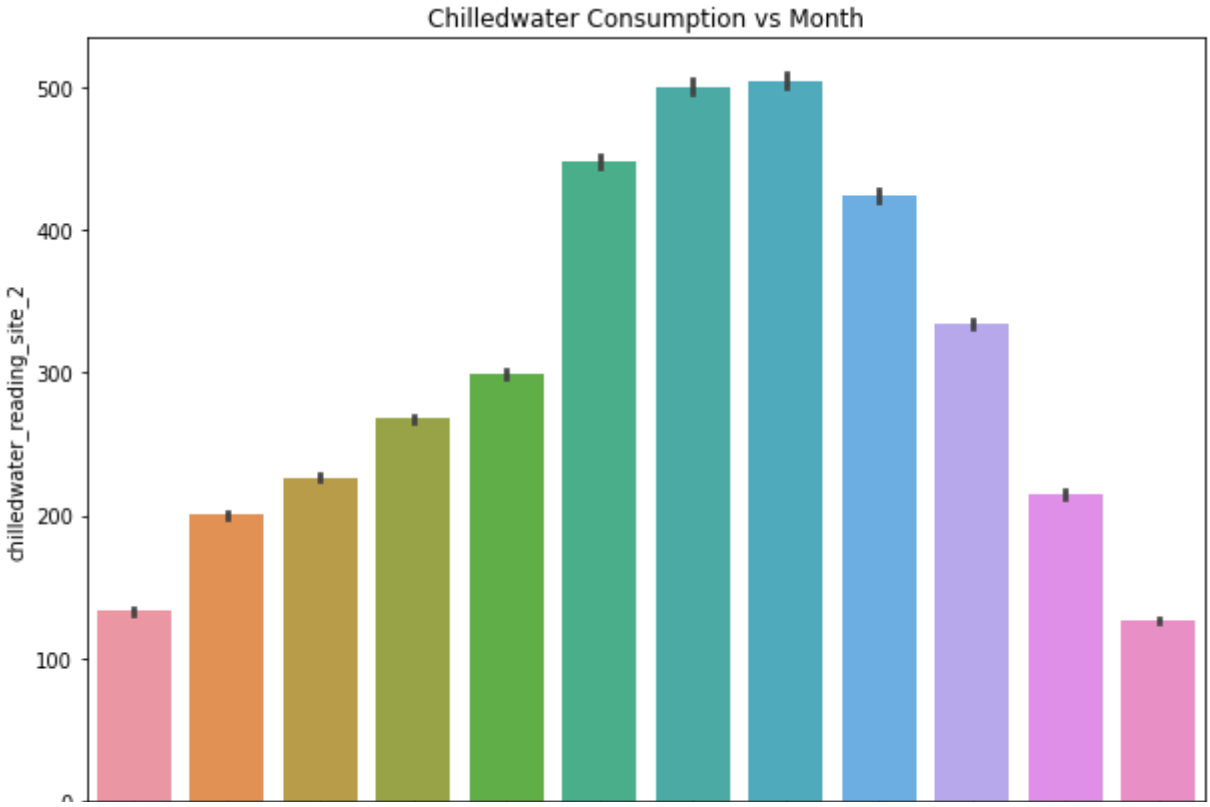
Building Count for different building type for chilledwater consumption

```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_2_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading_site_2')
plt.ylabel('primary_use_site_2_meter_1')
plt.title('Comparison of different building type for chiiledwater usage')
plt.show()
```



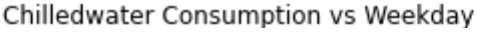
Utility is having the higher chilledwater usage

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_1,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading_site_2')
plt.title('Chilledwater Consumption vs Month')
plt.show()
```



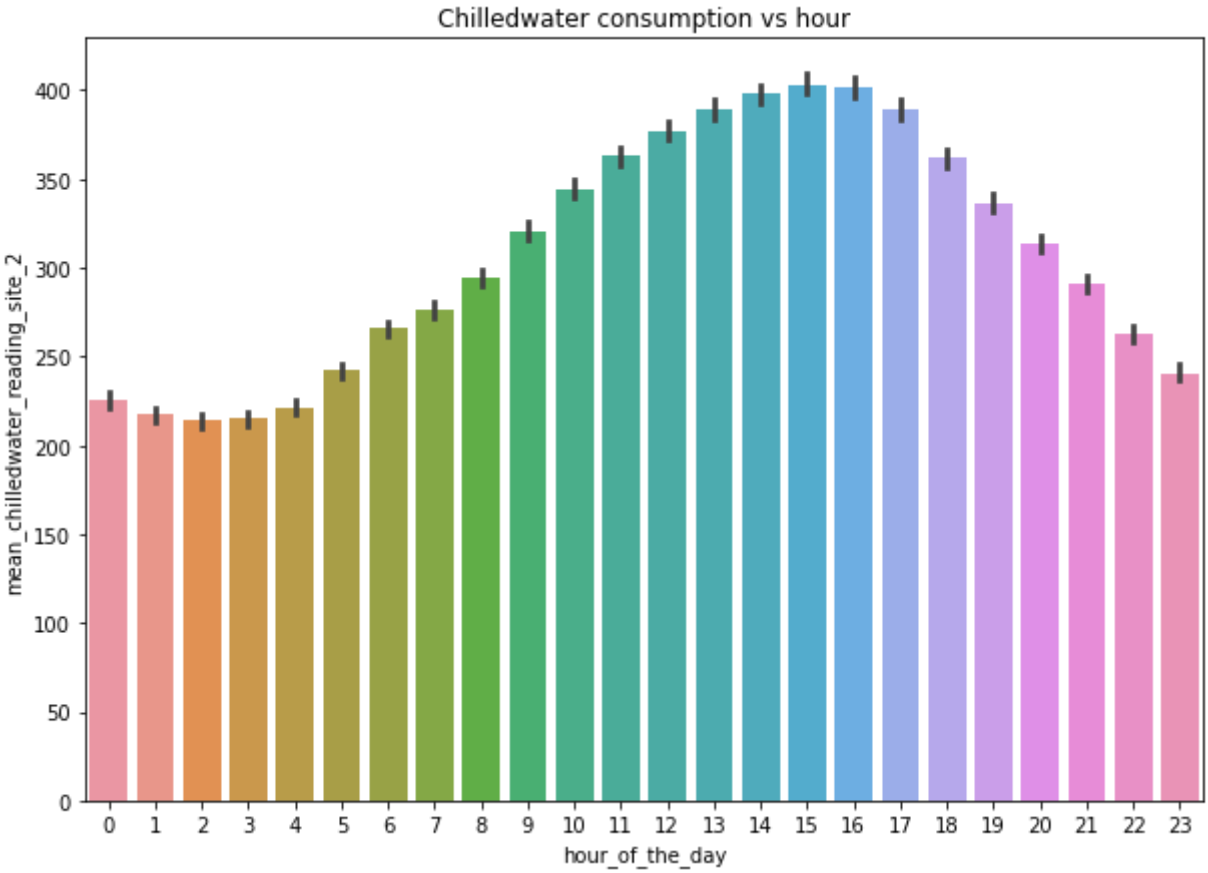
Chilledwater consumption is showing higher values for the summer month

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_1,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading_site_2')
plt.title('Chilledwater Consumption vs Weekday')
plt.show()
```



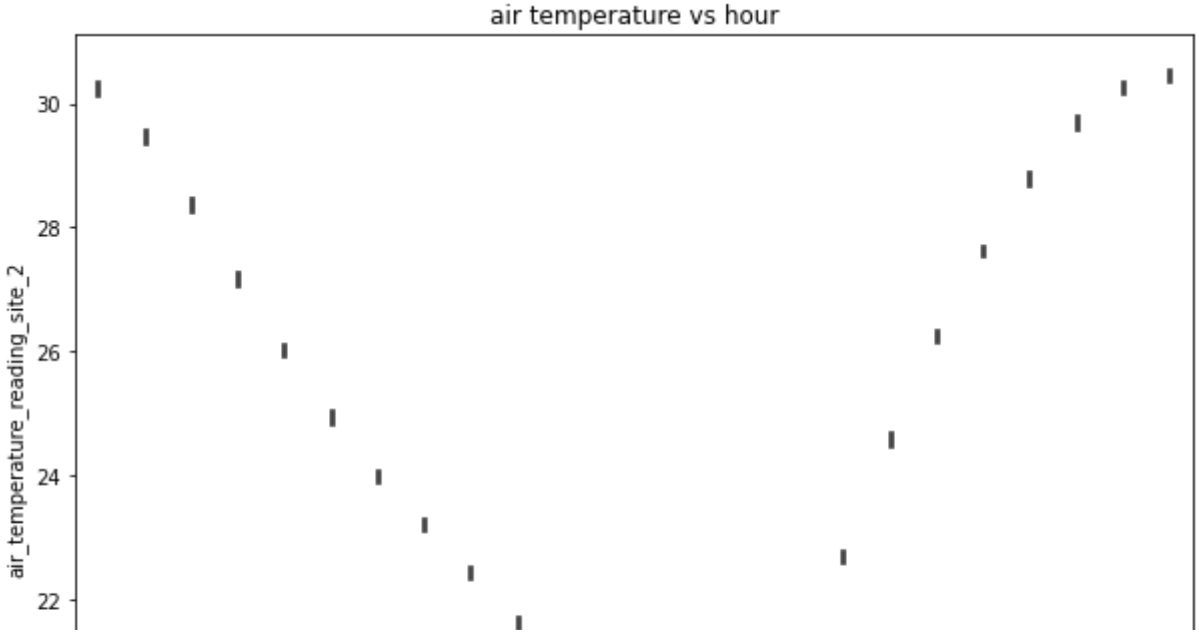
Chilledwater consumption is higher for weekdays as compared to the weekend

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_1,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('mean_chilledwater_reading_site_2')
plt.title('Chilledwater consumption vs hour')
plt.show()
```



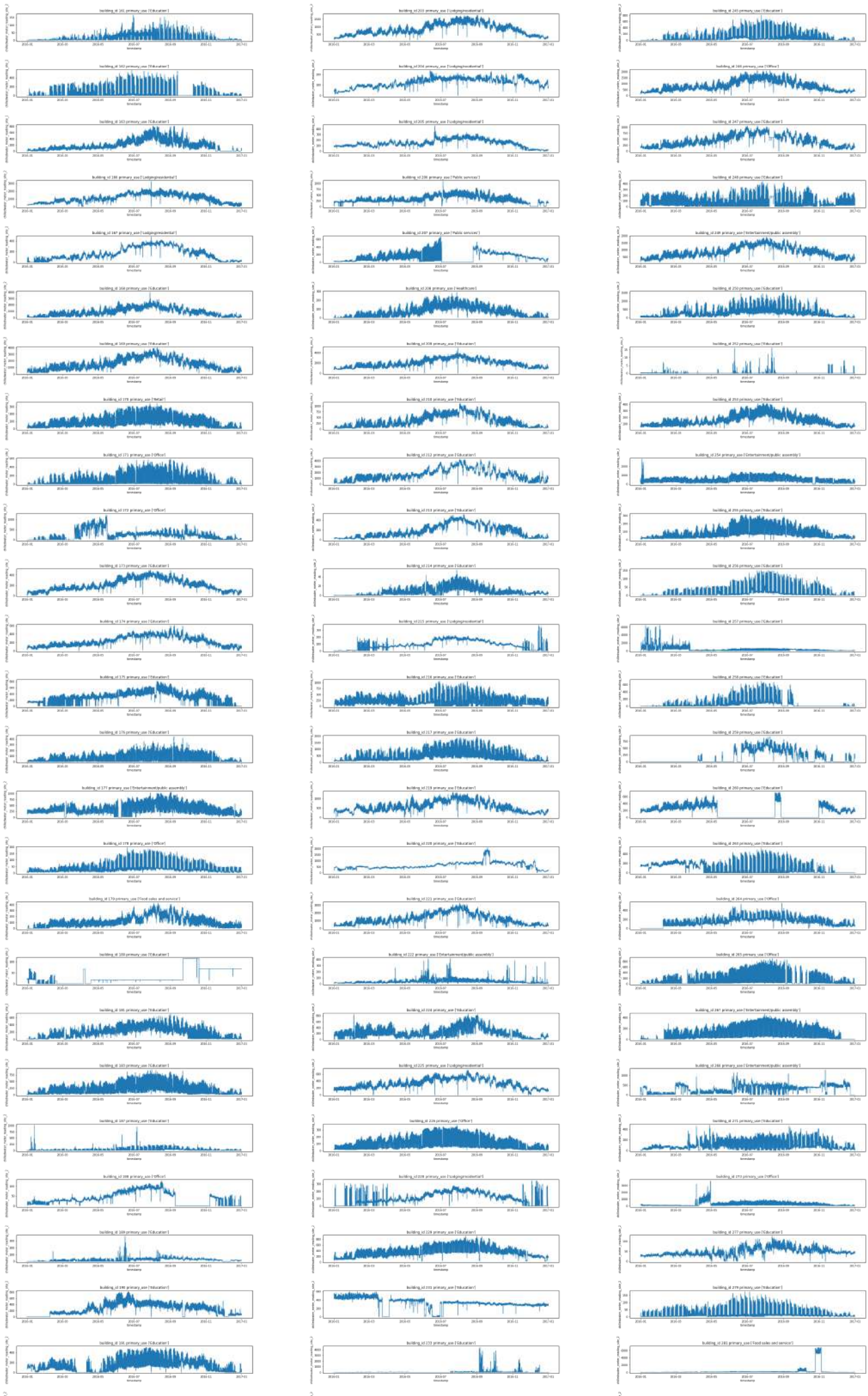
This plot shows us the hourly variations of chilledwater usage and it peaks during the afternoon hours.

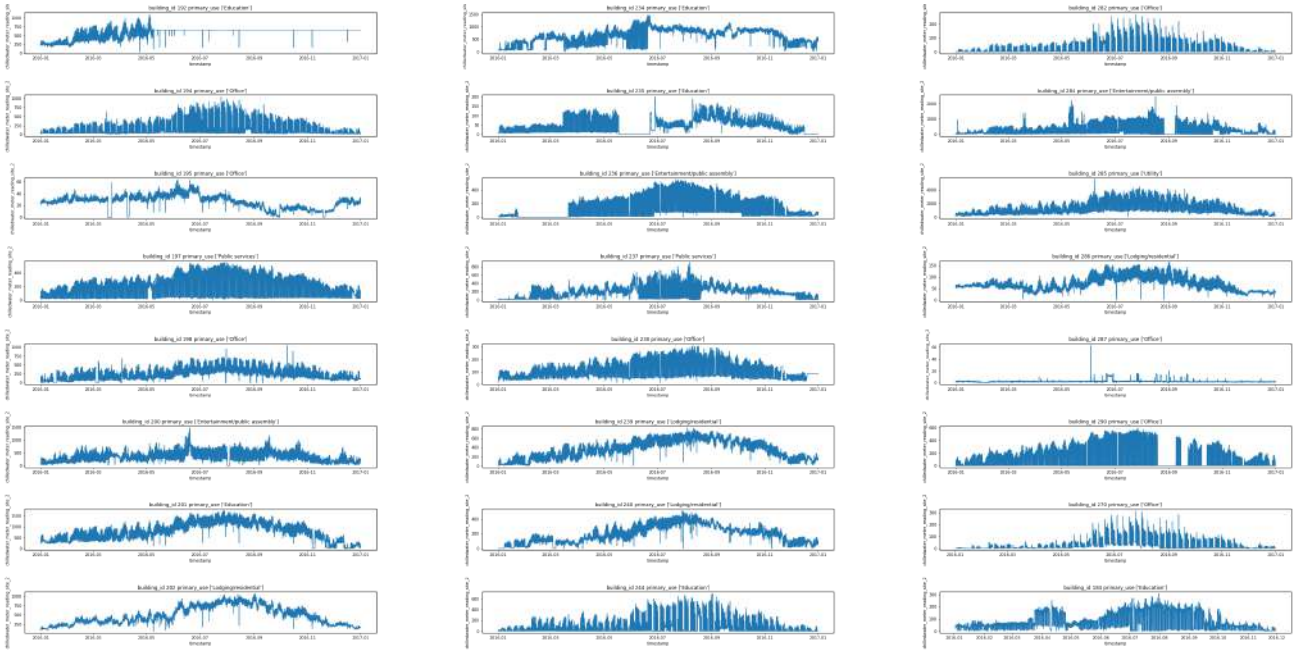
```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_1,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_reading_site_2')
plt.title('air temperature vs hour')
plt.show()
```



This plot shows that weather timestamp of the air temperature is not in alignment with the local timestamp of the hourly meter readings.

```
fig,ax=plt.subplots(figsize=(55,120),nrows=33,ncols=3)
for i in range(df_train_site_2_meter_1['building_id'].nunique()):
  g=df_train_site_2_meter_1['building_id'].unique()[i]
  axes=ax[i%33][i//33]
  z=df_train_site_2_meter_1.loc[df_train_site_2_meter_1['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('chilledwater_meter_reading_site_2')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
  plt.subplots_adjust(hspace=1,wspace=0.3)
```





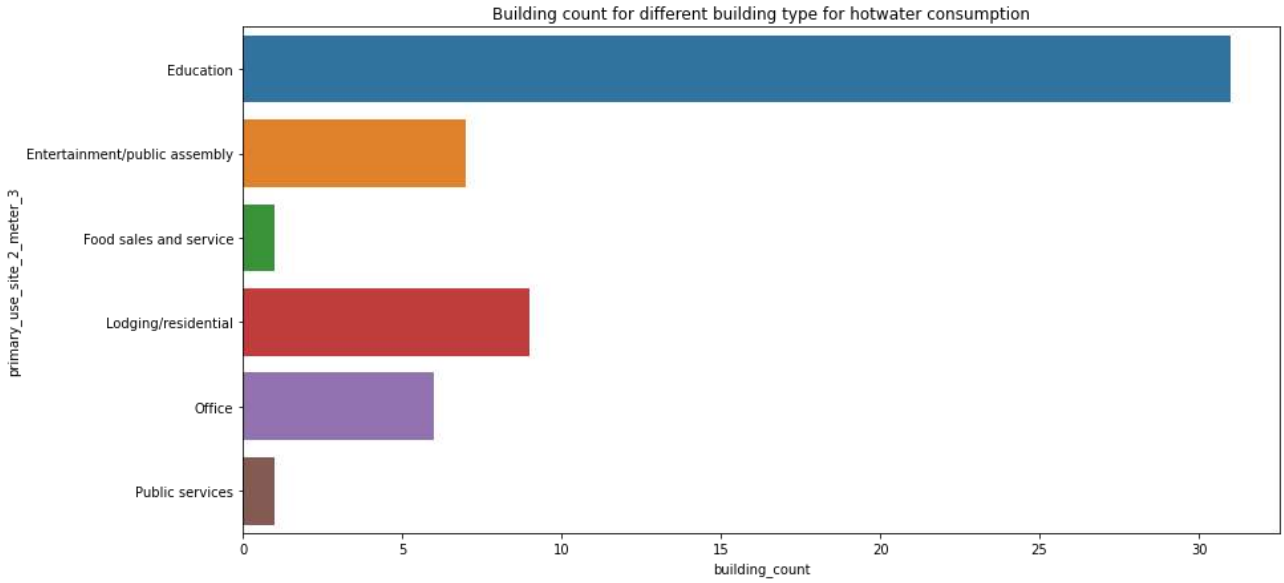
Important Observations

- The above plot for chilledwater readings shows anomaly in many of the buildings like in building 180 it starts streaks of constant zero and meter values from the mid of the 2nd month and it continues upto the last month of the year.Building 192 also shows constant values from the 5th month and continues upto the last month.These definitely might be due to some faults and we need to remove these values.
- Building 252 shows zero readings for most part of the year and shows little spikes in between the months so I will check whether to consider this building whether to consider this building or not in the training process.
- Now there are many zero readings observed between months here I am gonna remove zero readings only from summer months as this plot represents chilledwater energy consumption.
- For the zero readings in winter months I will keep them so that my model could learn seasonal variations.

```
df_train_site_2_meter_3['month']=df_train_site_2_meter_3['timestamp'].dt.month
df_train_site_2_meter_3['weekday']=df_train_site_2_meter_3['timestamp'].dt.weekday
```

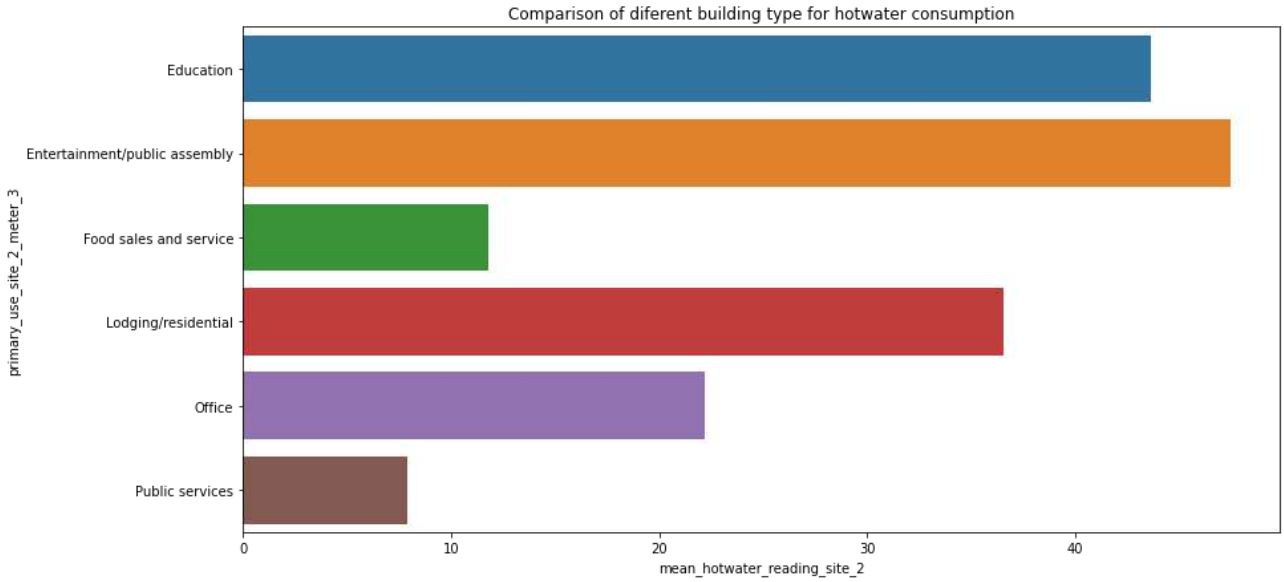
```
df_train_site_2_meter_3['hour']=df_train_site_2_meter_3['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_2_meter_3.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_site_2_meter_3')
plt.title('Building count for different building type for hotwater consumption')
plt.show()
```



This plot shows the building count for different types for hotwater consumption.

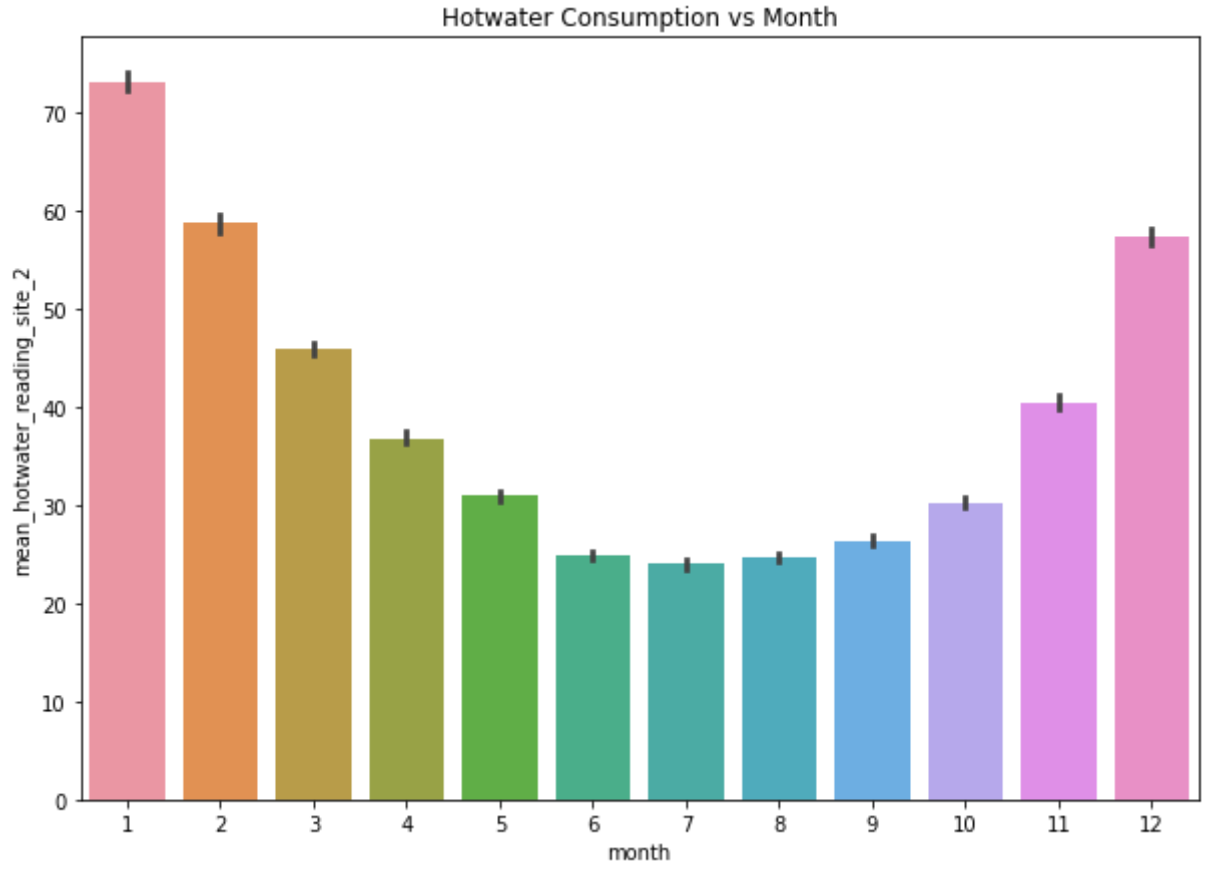
```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_2_meter_3.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_hotwater_reading_site_2')
plt.ylabel('primary_use_site_2_meter_3')
plt.title('Comparison of diferent building type for hotwater consumption')
plt.show()
```



Here we can see that the hotwater consumption is very high for the entertainment buildings althought heya re less in number.

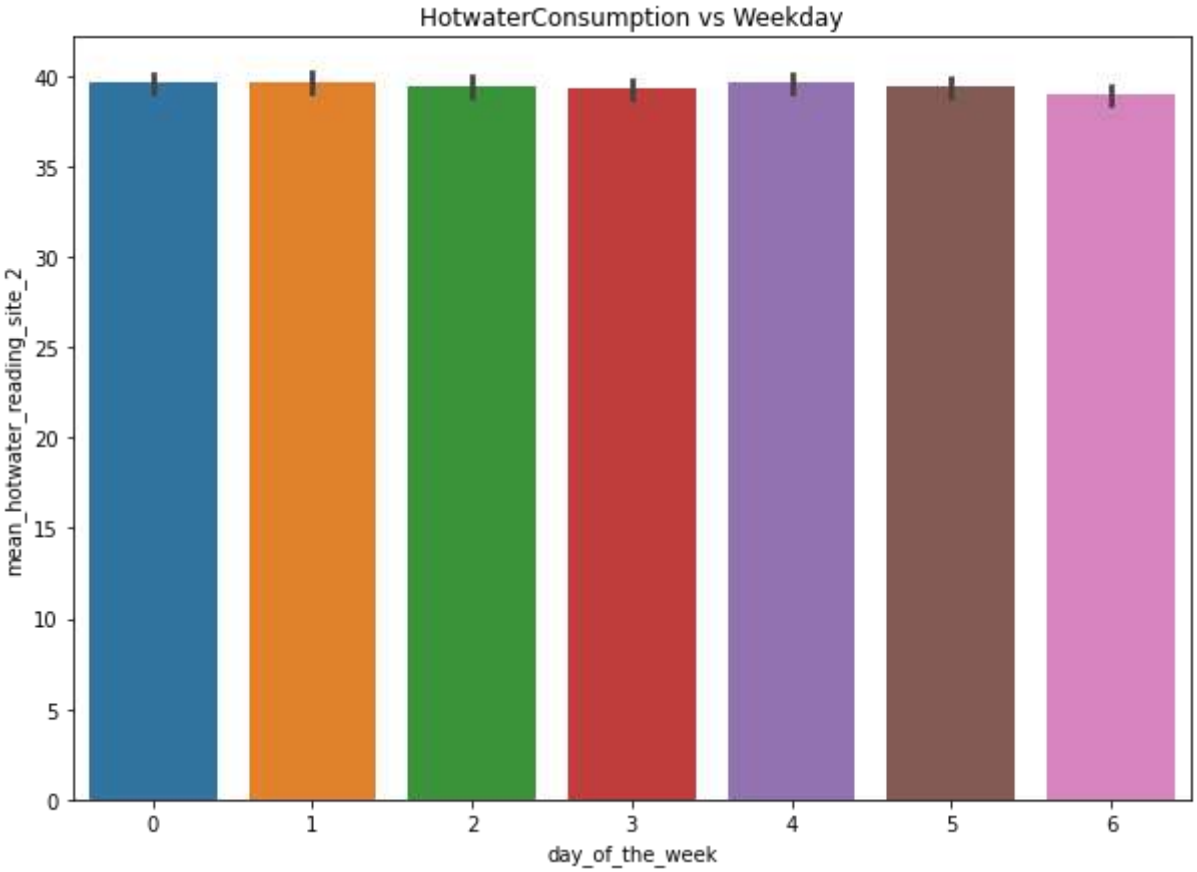
Residential buildings are also higher hotwater consumption but they are also less in number.

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_3,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('mean_hotwater_reading_site_2')
plt.title('Hotwater Consumption vs Month')
plt.show()
```



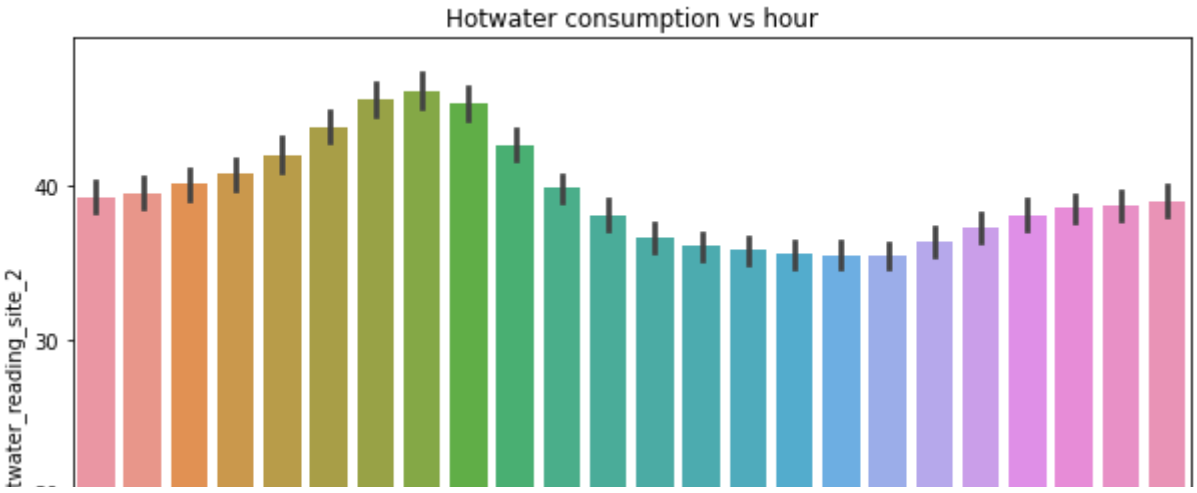
Hotwater consumption varies logically according to the seasonal transitions.

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_3,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('mean_hotwater_reading_site_2')
plt.title('HotwaterConsumption vs Weekday')
plt.show()
```



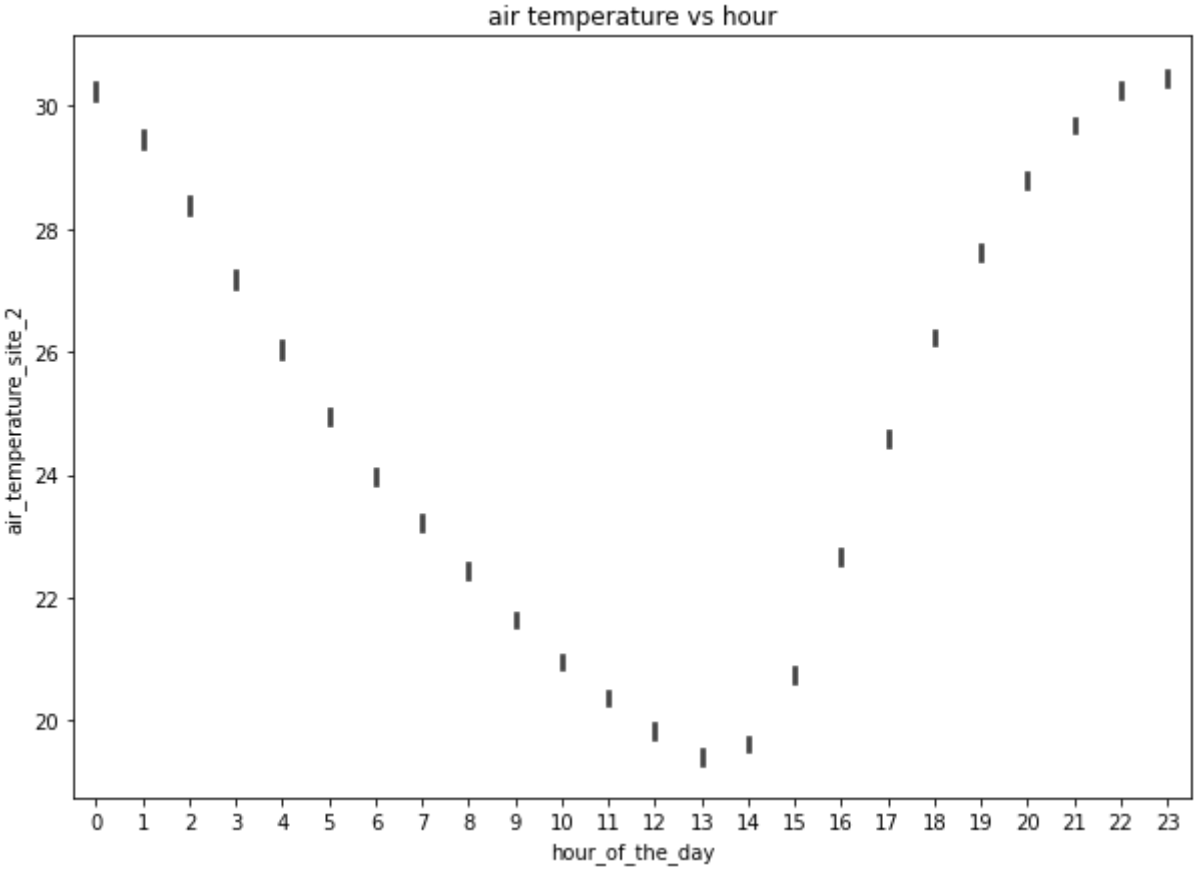
Here Hotwater consumption is showing equal usage on any day of the week

```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_3,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('mean_hotwater_reading_site_2')
plt.title('Hotwater consumption vs hour')
plt.show()
```



Hotwater Consumption shows highest consumption during the morning hours and then decreases rapidly during the daytime and then increases during the night time.

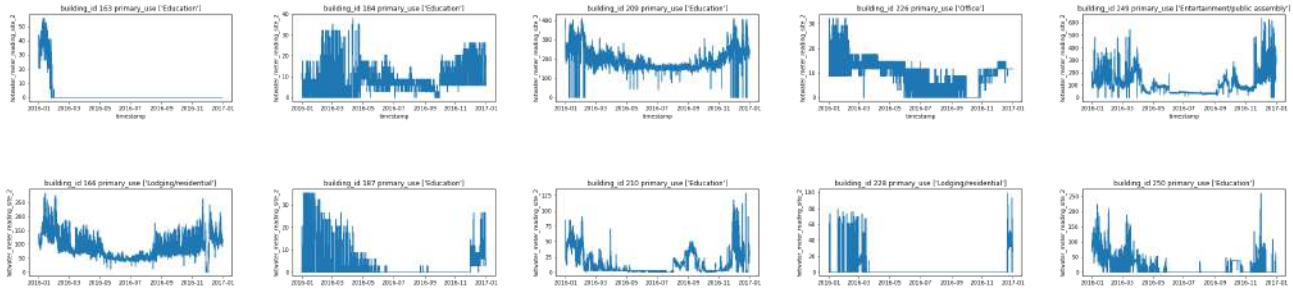
```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_2_meter_3,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_2')
plt.title('air temperature vs hour')
plt.show()
```



This plot shows that the weather timestamp is not in alignment with the local timestamp for the hourly readings of the hotwater consumption.

```
fig,ax=plt.subplots(figsize=(40,60),nrows=11,ncols=5)
```

```
  for i in range(df_train_site_2_meter_3['building_id'].nunique()):
    g=df_train_site_2_meter_3['building_id'].unique()[i]
    axes=ax[i%11][i//11]
    z=df_train_site_2_meter_3.loc[df_train_site_2_meter_3['building_id']==g]
    axes.plot(z['timestamp'],z['meter_reading'])
    axes.set_xlabel('timestamp')
    axes.set_ylabel('hotwater_meter_reading_site_2')
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
    plt.subplots_adjust(hspace=1,wspace=0.3)
```



Important Observations

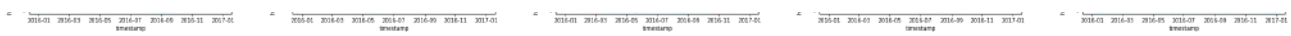
- Building 279 and 287 are having zero meter readings for the whole year just show a spike therefore we cannot use this in the training process.
- Building 263 is also having zer0 reading for most part of the year and shows some spike for winter months and somereading for the 4th and 5th month therfore it might not be good for considering it into the training process.

```
#Starting analysis for site 3
```

```
df_train_site_3=df_train_merge.loc[df_train_merge['site_id']==3]


df_train_site_3.isnull().sum()/df_train_site_3.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           0.52
floor_count          1.00
air_temperature      0.00
cloud_coverage       0.42
dew_temperature      0.00
precip_depth_1_hr    0.00
sea_level_pressure   0.02
wind_direction       0.02
wind_speed           0.00
dtype: float64
```


We need to impute the missing values so that it can be used in the training process.

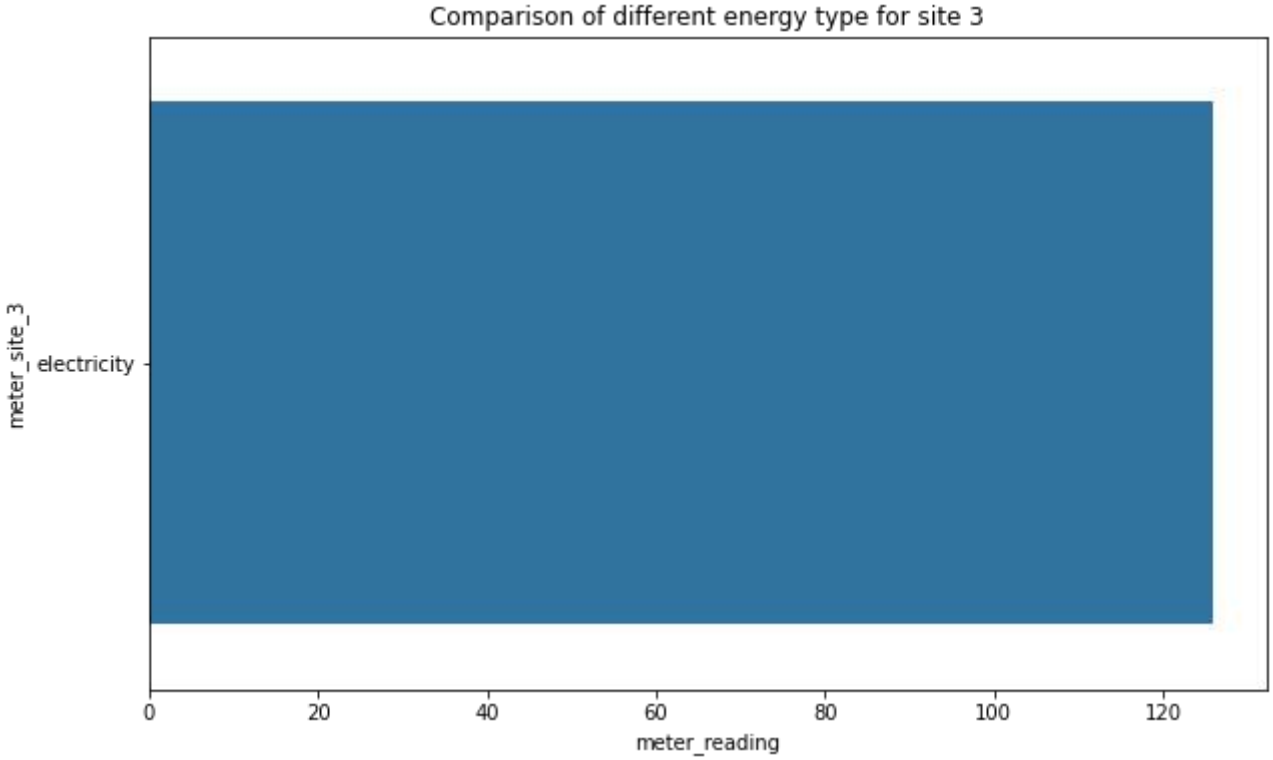


```
df_corr_3=df_train_site_3.corr()
df_corr_3.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.06	nan	0.00	0.12	nan	0.00
meter_reading	0.06	1.00	nan	0.84	0.10	nan	0.03
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.00	0.84	nan	1.00	0.08	nan	0.00
year_built	0.12	0.10	nan	0.08	1.00	nan	-0.00
floor_count	nan	nan	nan	nan	nan	nan	nan
air_temperature	0.00	0.03	nan	0.00	-0.00	nan	1.00
cloud_coverage	-0.00	0.01	nan	0.00	0.00	nan	0.14
dew_temperature	0.00	0.04	nan	0.00	-0.00	nan	0.89

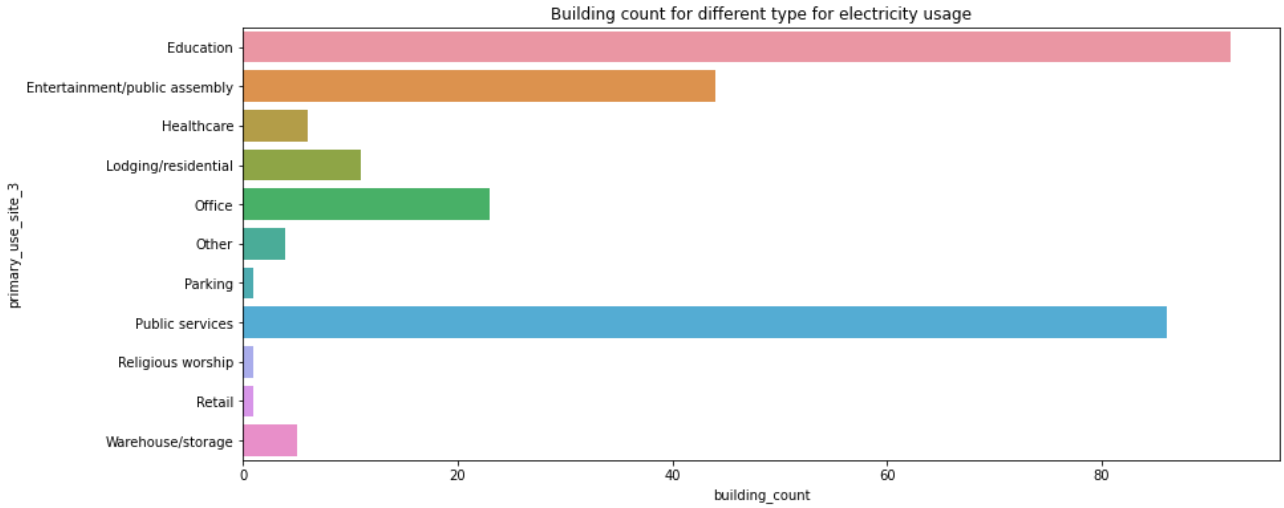
For site 3 meter reading shows a high correlation with the square feet of the building.

```
z=df_train_site_3.groupby(['meter'])
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z['meter_reading'].mean().reset_index(),x='meter_reading',y='meter'
plt.xlabel('meter_reading')
plt.ylabel('meter_site_3')
plt.title('Comparison of different energy type for site 3')
plt.show()
```



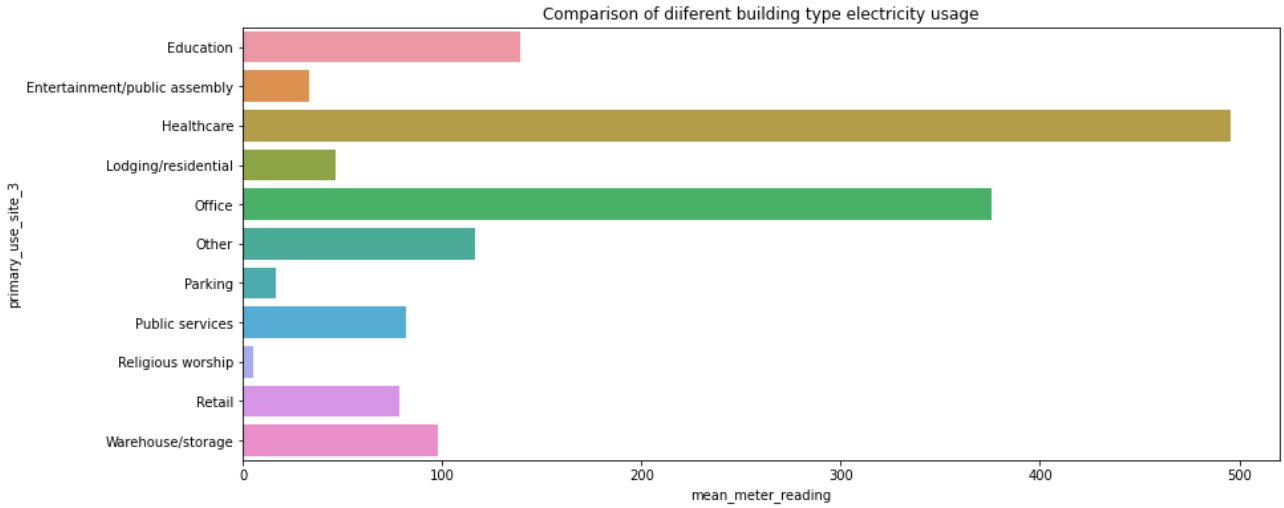
Site 3 only has electrical energy consumption

```
z=df_train_site_3.groupby(['primary_use'])
fig,ax=plt.subplots(figsize=(14,6))
sns.barplot(ax=ax,data=z['building_id'].nunique().reset_index(),x='building_id',y='primary
plt.xlabel('building_count')
plt.ylabel('primary_use_site_3')
plt.title('Building count for different type for electricity usage')
plt.show()
```



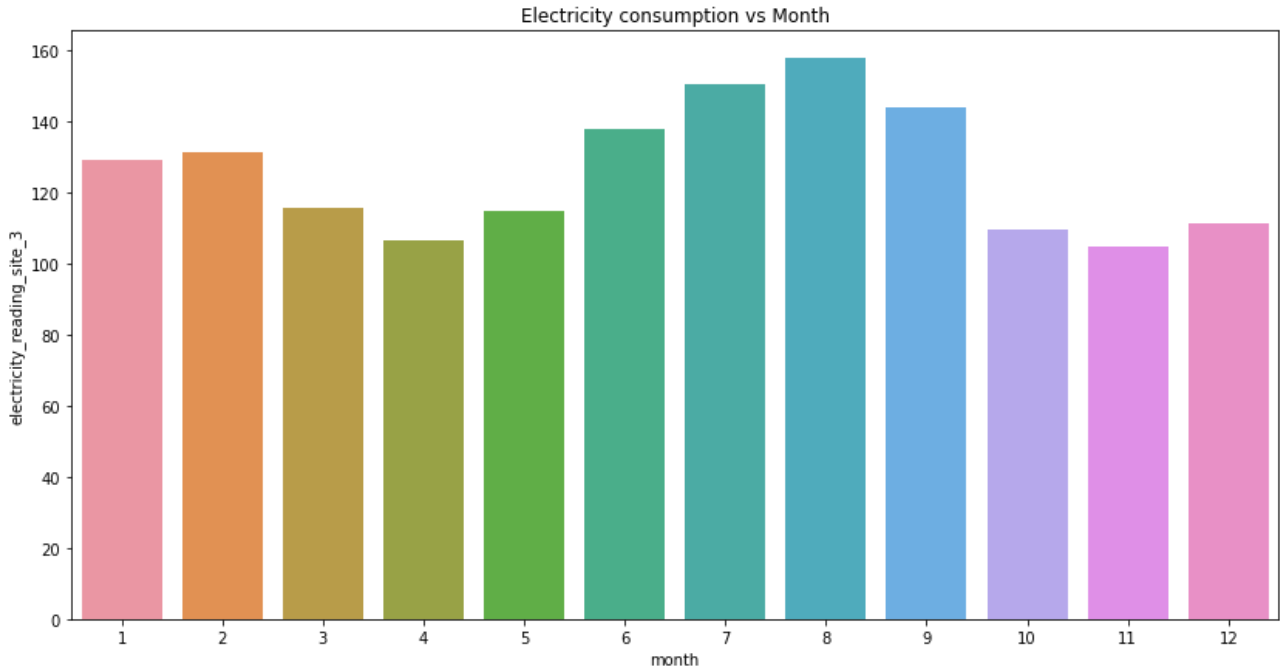
This plot shows the building count for different building type for electricity consumption.

```
z=df_train_site_3.groupby(['primary_use'])
fig,ax=plt.subplots(figsize=(14,6))
sns.barplot(ax=ax,data=z['meter_reading'].mean().reset_index(),x='meter_reading',y='primar
plt.xlabel('mean_meter_reading')
plt.ylabel('primary_use_site_3')
plt.title('Comparison of diiferent building type electricity usage')
plt.show()
```



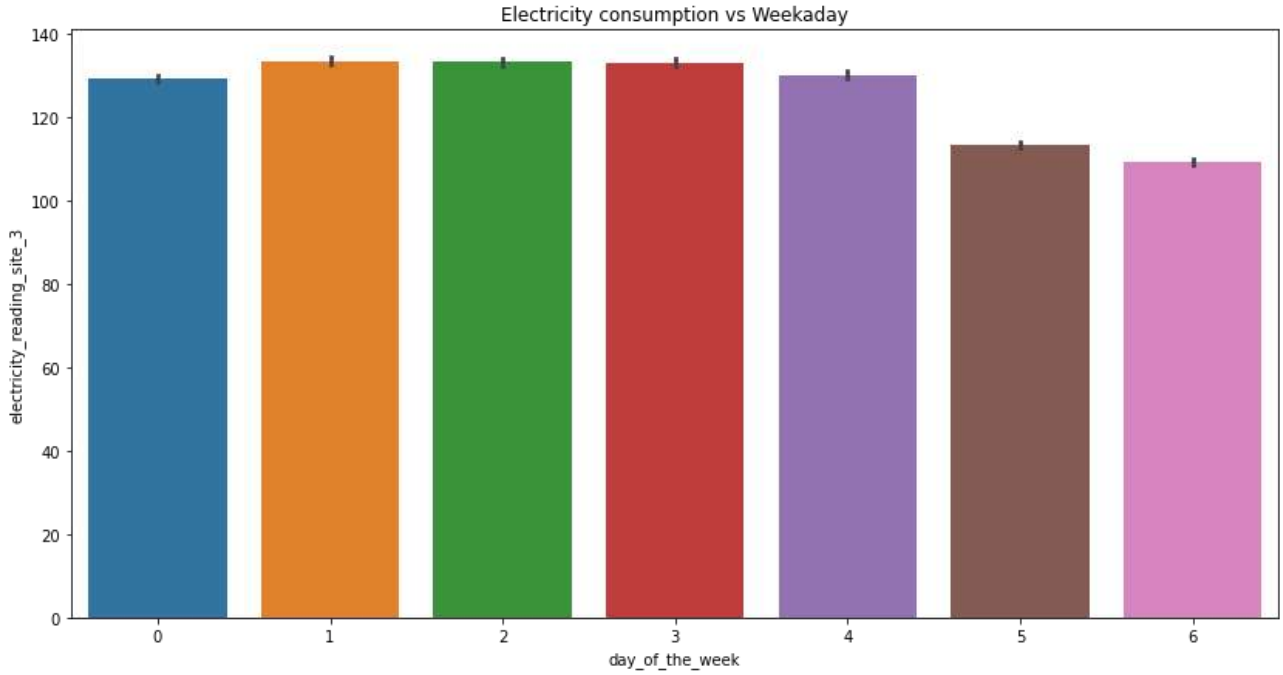
This plot shows that Healthcare industry has the highest electrical energy consumption.

```
df_train_site_3['month']=df_train_site_3['timestamp'].dt.month


fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_3.groupby(['month'])
sns.barplot(ax=ax,data=z['meter_reading'].mean().reset_index(),x='month',y='meter_reading'
plt.xlabel('month')
plt.ylabel('electricity_reading_site_3')
plt.title('Electricity consumption vs Month')
plt.show()
```



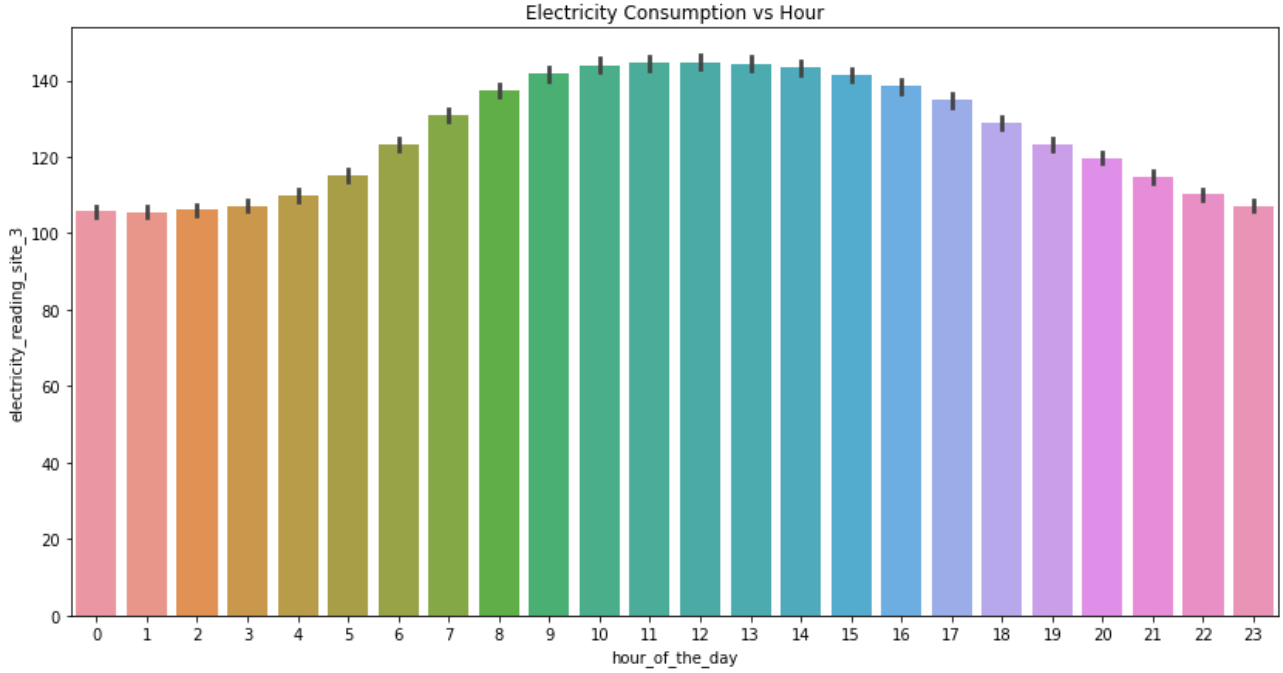
This plot shows the variation of electrical energy consumption over different month

```
df_train_site_3['weekday']=df_train_site_3['timestamp'].dt.weekday
df_train_site_3['hour']=df_train_site_3['timestamp'].dt.hour


fig,ax=plt.subplots(figsize=(14,7))
sns.barplot(ax=ax,data=df_train_site_3,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_3')
plt.title('Electricity consumption vs Weekaday')
plt.show()
```



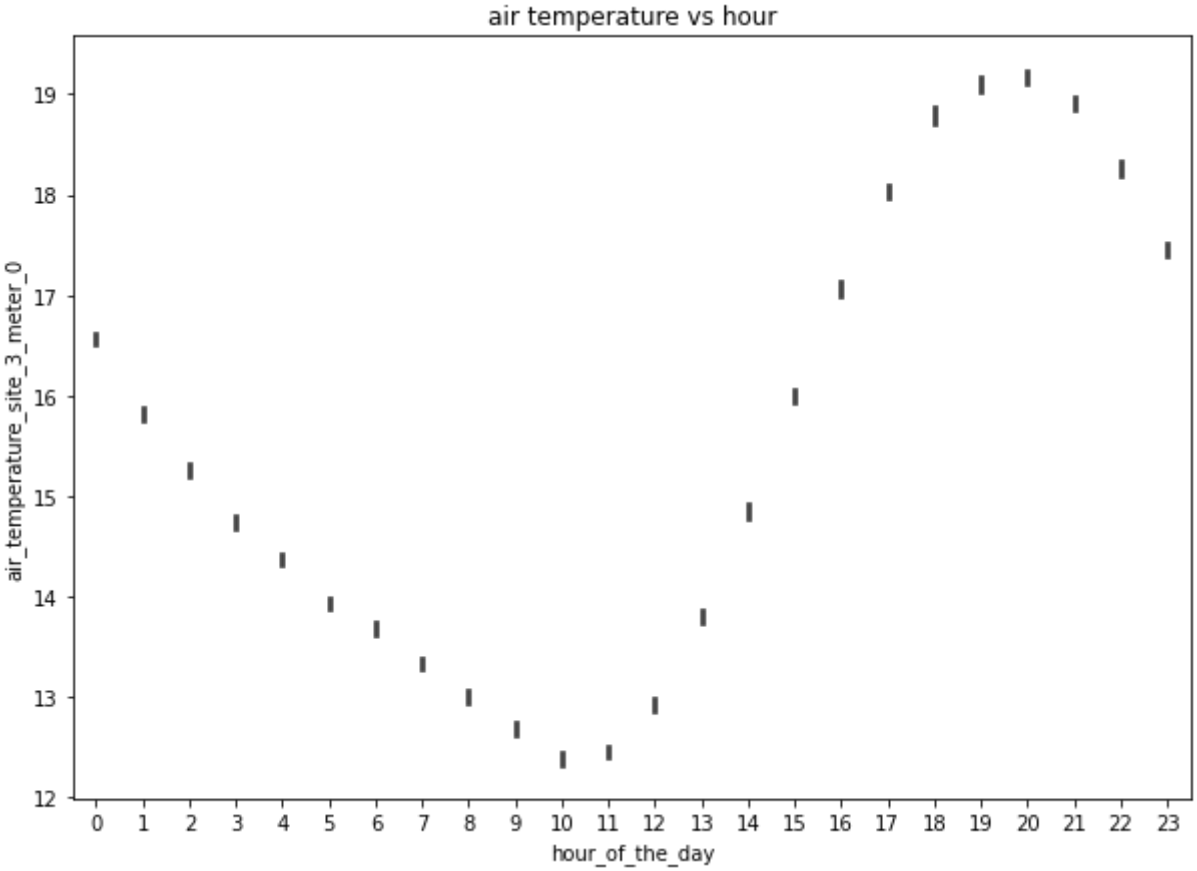
Electricity consumption is lesser on the wekend as compared to the weekdays

```
fig,ax=plt.subplots(figsize=(14,7))
sns.barplot(ax=ax,data=df_train_site_3,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_3')
plt.title('Electricity Consumption vs Hour')
plt.show()
```



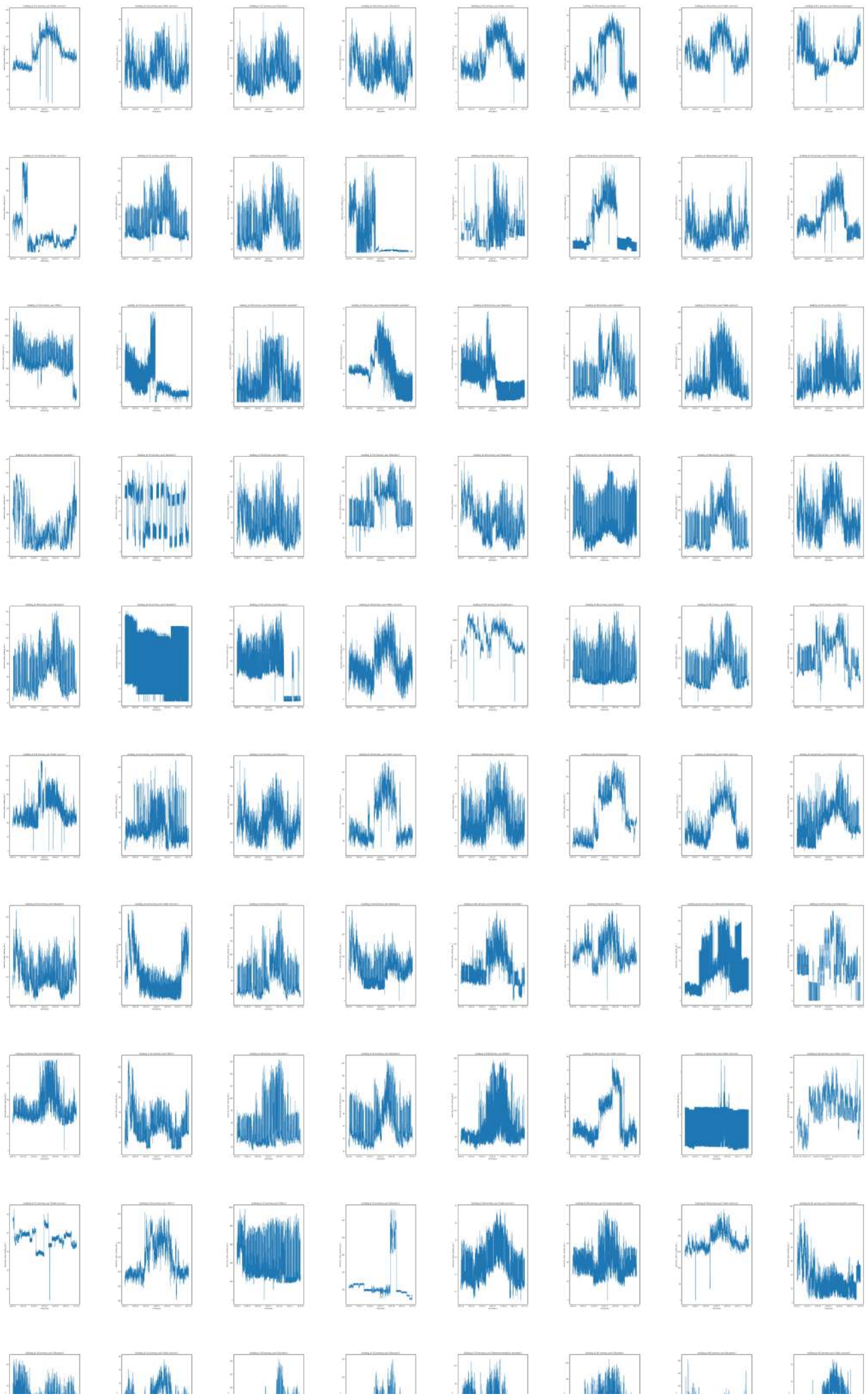
Electricity consumption over the hour of the day as we can see that it peaks during the daytime and starts to decrease gradually during the evening hours.

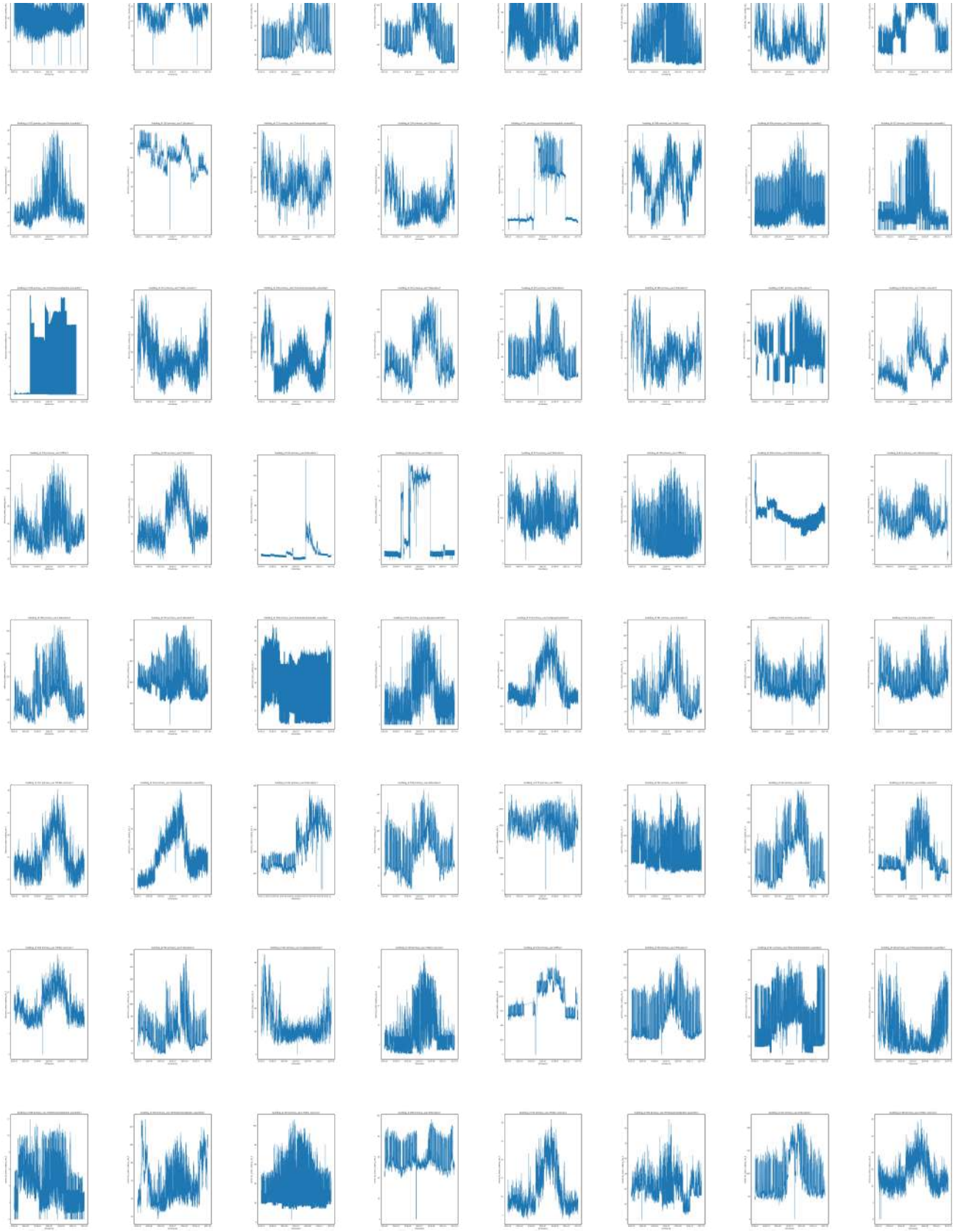
```
fig,ax=plt.subplots(figsize=(10,7))
sns.barplot(ax=ax,data=df_train_site_3,x='hour',y='air_temperature')
plt.ylabel('air_temperature_site_3_meter_0')
plt.xlabel('hour_of_the_day')
plt.title('air temperature vs hour')
plt.show()
```



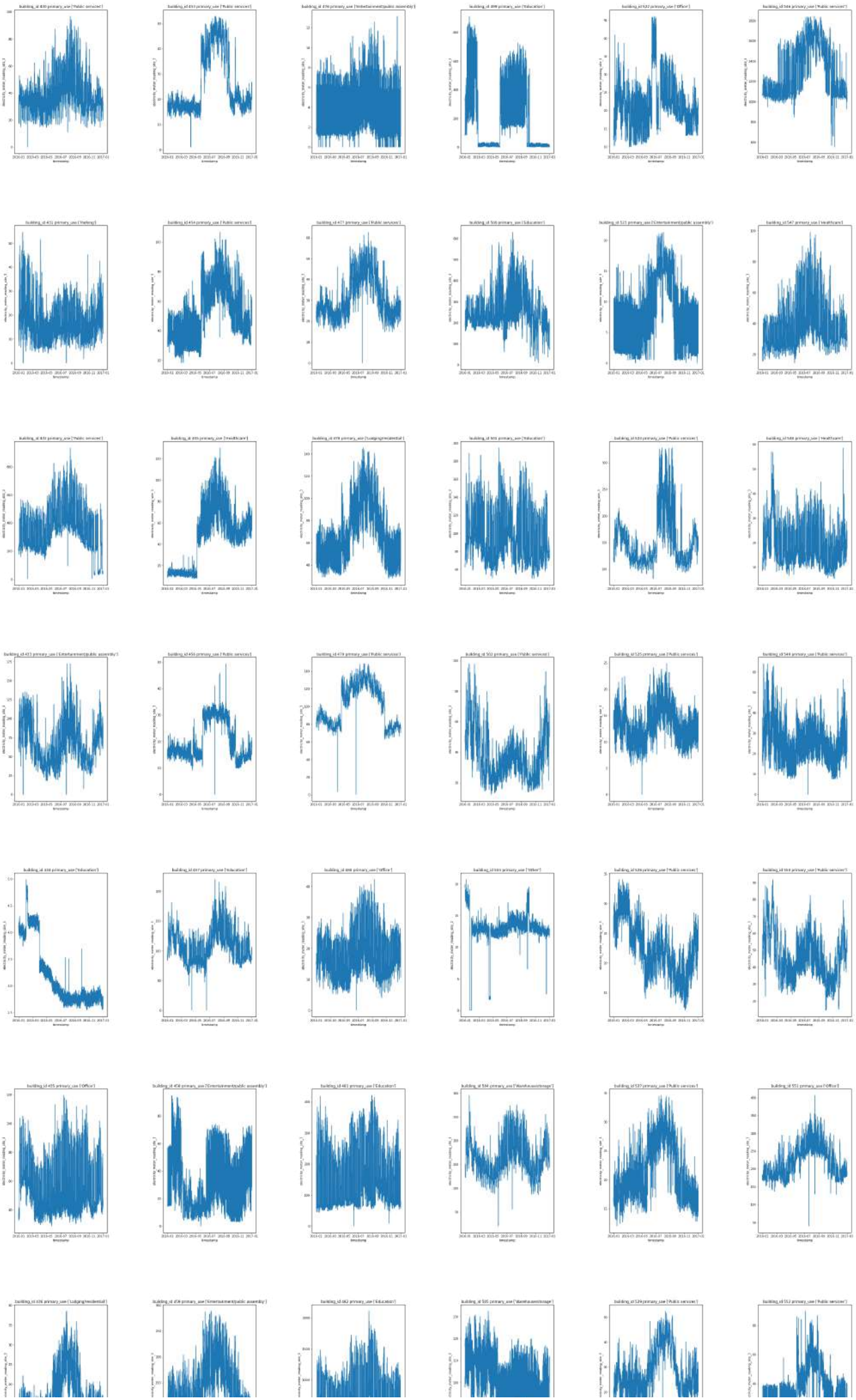
The weather timestamp is not aligned with the local timestamp of the hourly meter reading as the air temperature peaks around 20:00 pm

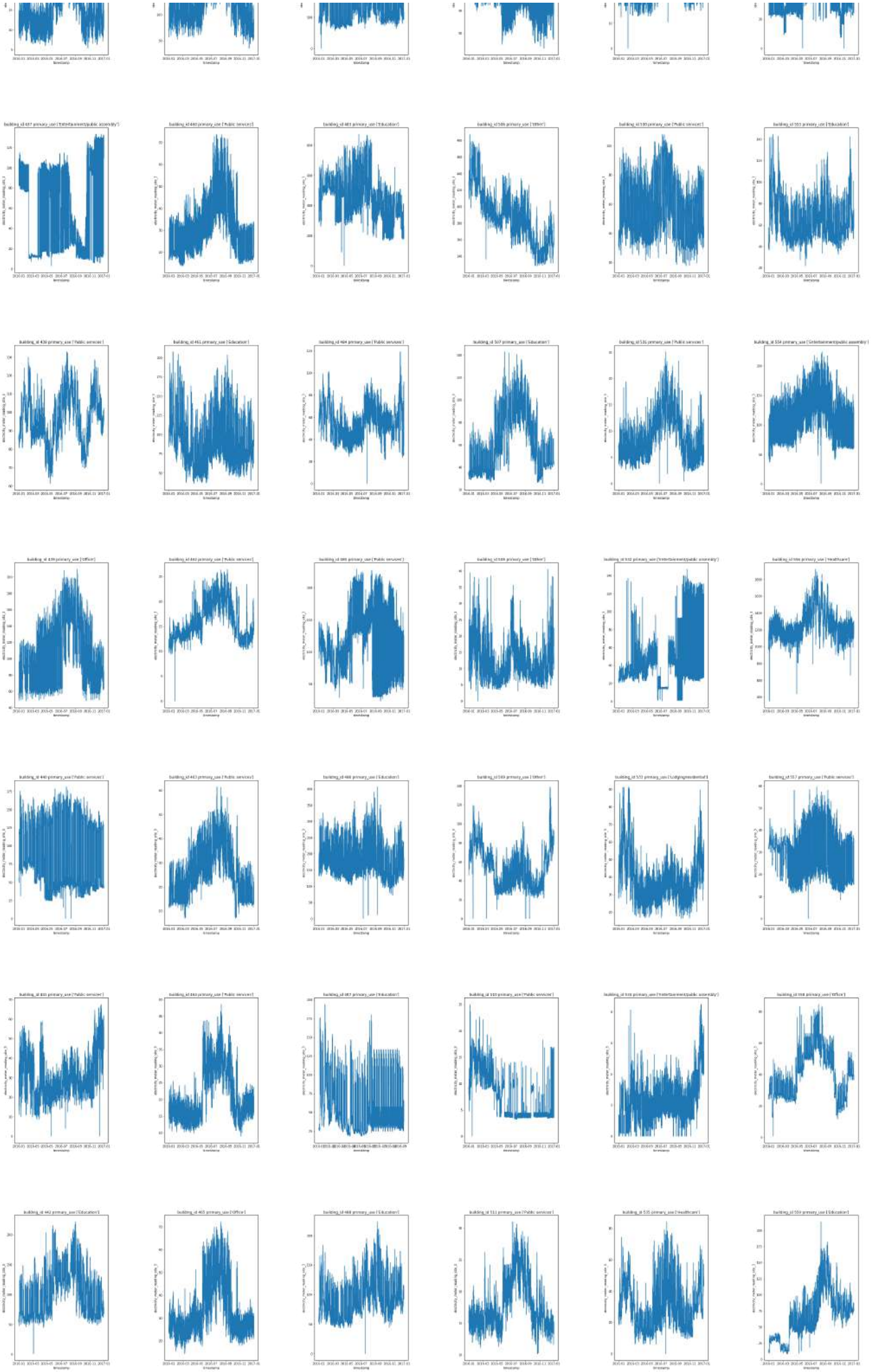
```
fig,ax=plt.subplots(figsize=(100,300),nrows=17,ncols=8)
for i in range(df_train_site_3['building_id'].nunique()-138):
  g=df_train_site_3['building_id'].unique()[i]
  axes=ax[i%17][i//17]
  z=df_train_site_3.loc[df_train_site_3['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_meter_reading_site_3')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
  plt.subplots_adjust(hspace=0.5,wspace=0.6)
```

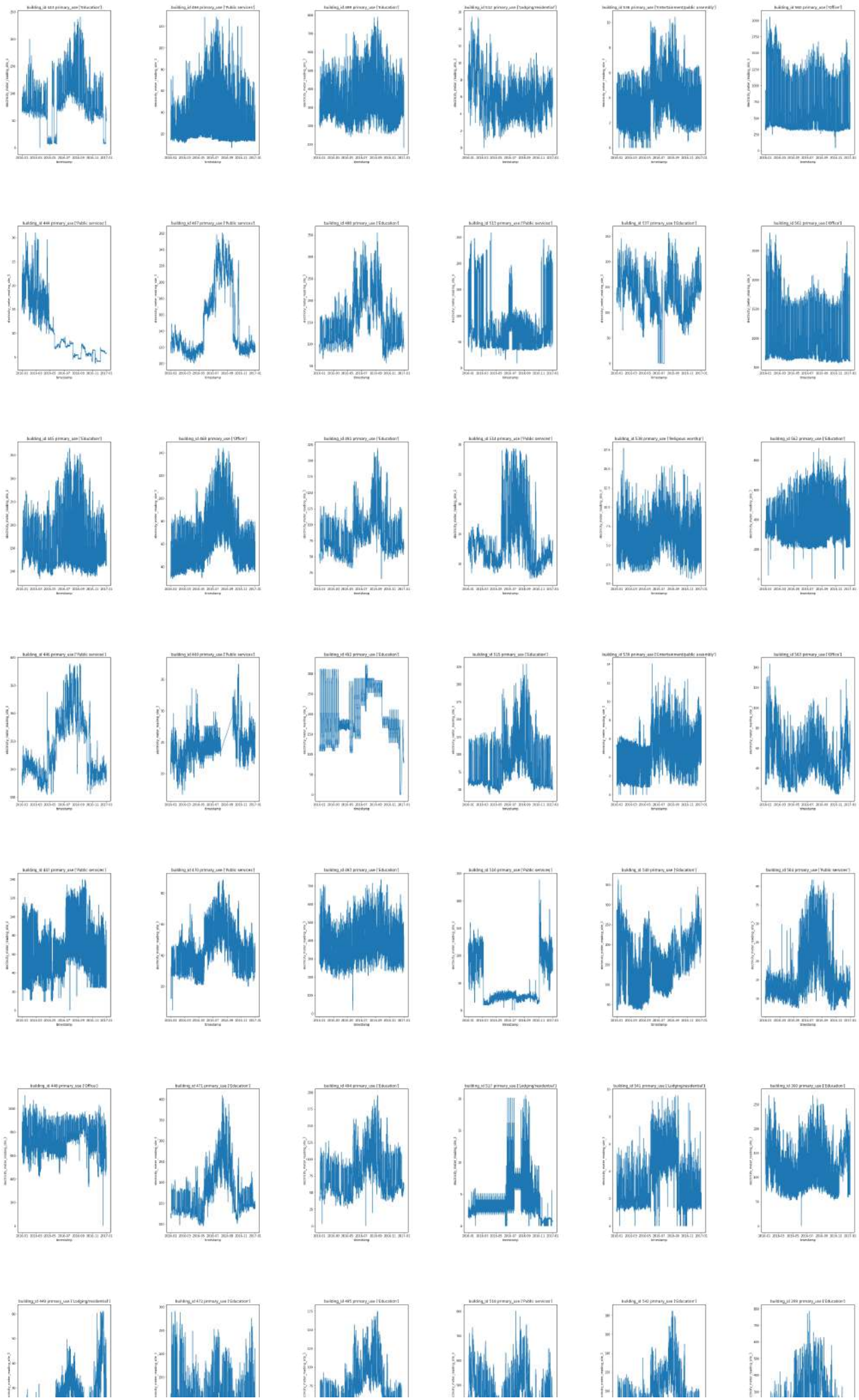


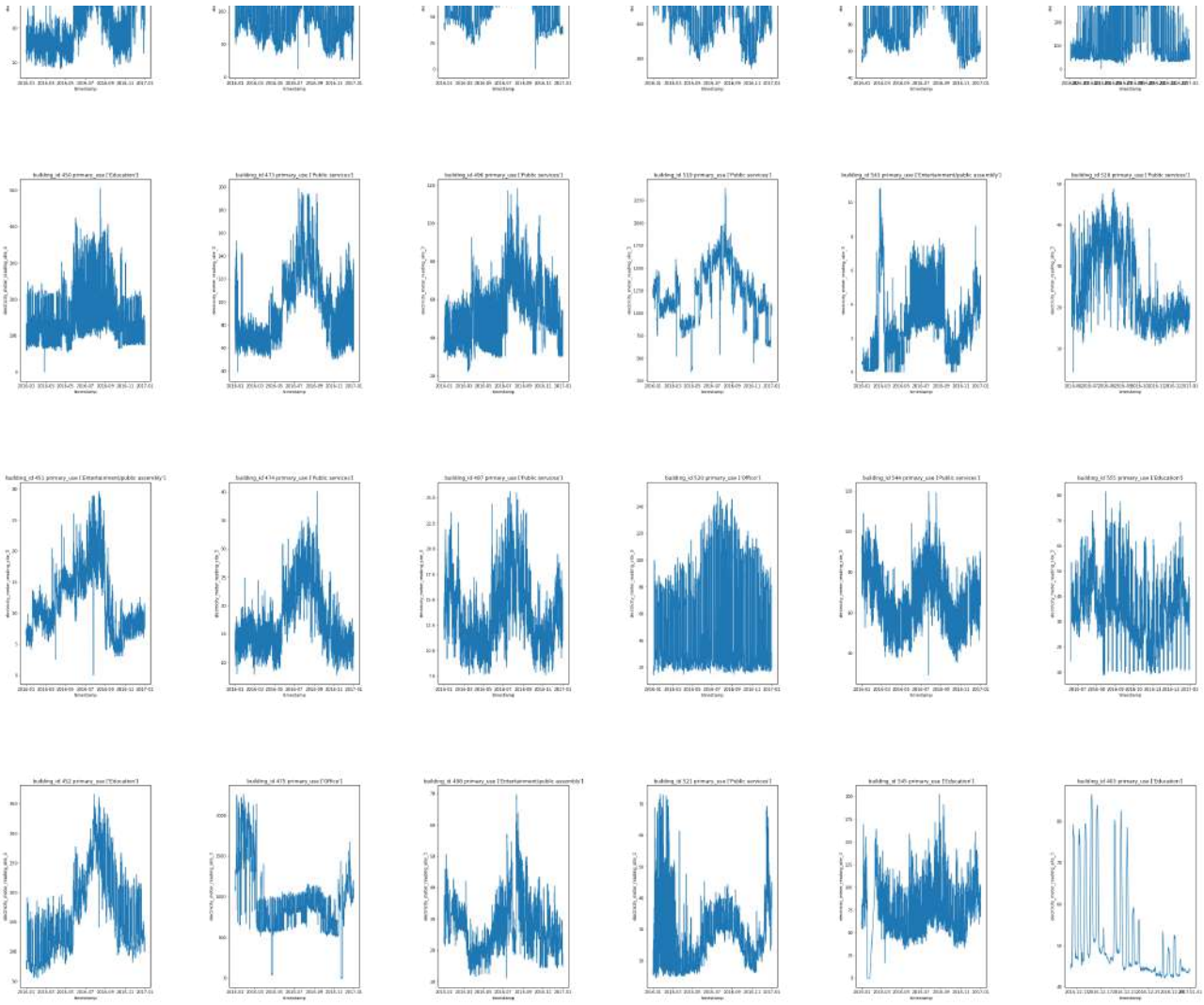


```
fig,ax=plt.subplots(figsize=(50,300),nrows=23,ncols=6)
for i in range(df_train_site_3['building_id'].nunique()-136):
  g=df_train_site_3['building_id'].unique()[136:274][i]
  axes=ax[i%23][i//23]
  z=df_train_site_3.loc[df_train_site_3['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_meter_reading_site_3')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
  plt.subplots_adjust(hspace=0.5,wspace=0.6)
```









There are a total of 274 buildings in total at site 3 which consumes electricity therefore to better represent it I didvided it into 136 and 138 buildings.

From the above plots what wecan see that we need to find the meter readings which are showing a huge spike and we need to remove them.Here we are not observing streaks of constant zeros for the buildings.

```
#Starting the analysis for site 4
```

```
df_train_site_4=df_train_merge.loc[df_train_merge['site_id']==4]
```

```
df_train_site_4.isnull().sum()/df_train_site_4.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           0.01
floor_count          0.00
air_temperature      0.00
cloud_coverage       0.48
dew_temperature      0.00
precip_depth_1_hr    0.15
sea_level_pressure   0.01
wind_direction       0.01
wind_speed           0.00
dtype: float64
```

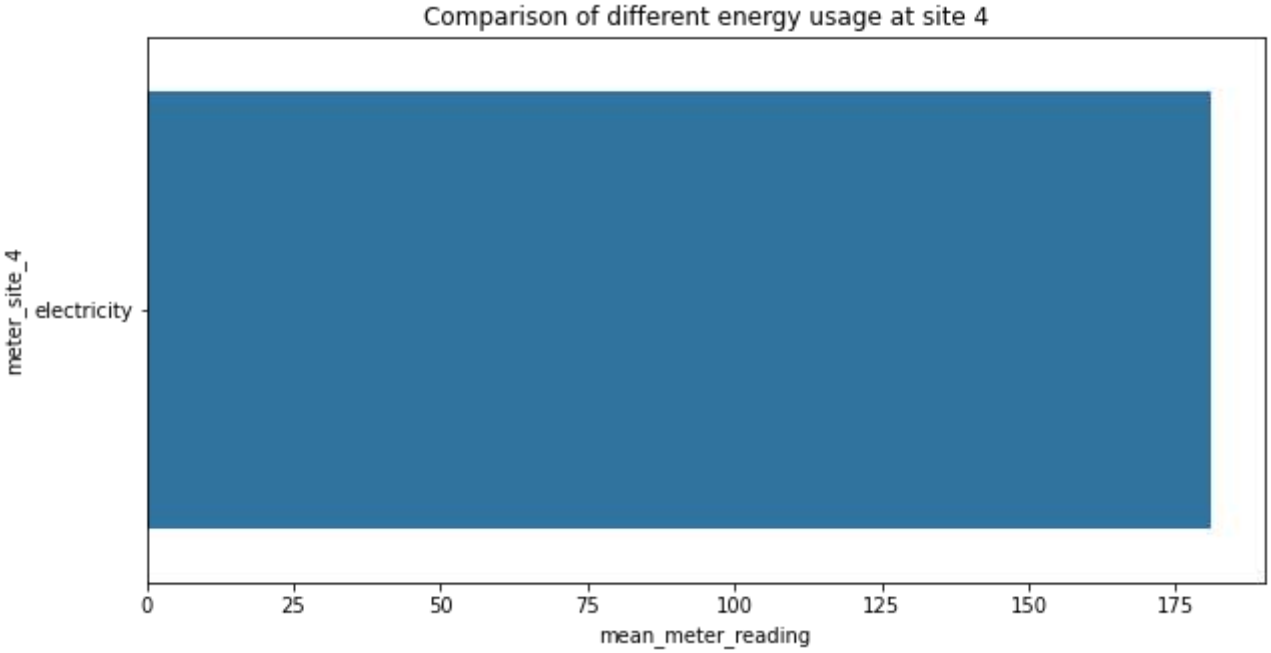
We need to fill up the missing values for the training process.

```
df_corr_4=df_train_site_4.corr()
df_corr_4.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.20	nan	0.18	0.01	0.18	-0.00
meter_reading	0.20	1.00	nan	0.62	0.38	0.63	-0.01
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.18	0.62	nan	1.00	0.27	0.64	-0.01
year_built	0.01	0.38	nan	0.27	1.00	0.30	0.01
floor_count	0.18	0.63	nan	0.64	0.30	1.00	-0.00
air_temperature	-0.00	-0.01	nan	-0.01	0.01	-0.00	1.00
cloud_coverage	0.00	-0.01	nan	0.00	-0.00	0.01	0.10
dew_temperature	-0.00	0.01	nan	-0.01	0.01	-0.01	0.58
precip_depth_1_hr	-0.00	-0.00	nan	-0.00	0.00	-0.00	-0.05
sea_level_pressure	0.00	0.00	nan	0.01	-0.01	0.00	-0.36
wind_direction	0.00	-0.02	nan	-0.00	0.00	0.00	0.43
wind_speed	0.00	-0.03	nan	-0.00	0.00	0.00	0.37

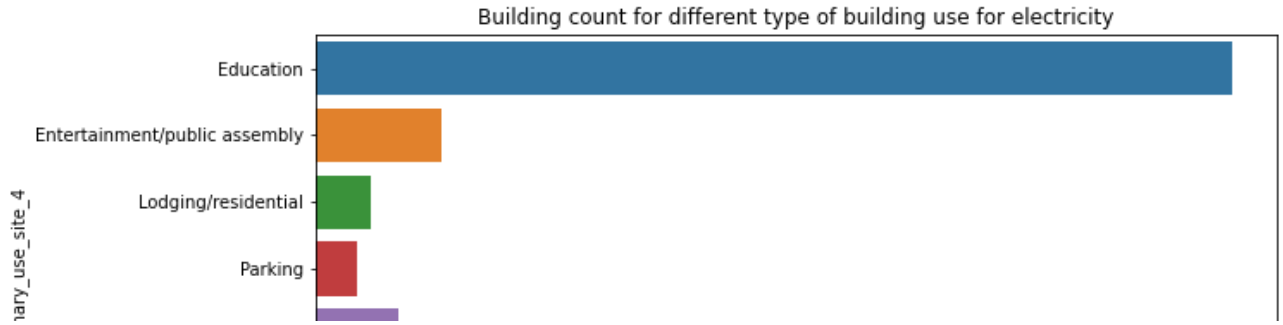
For site 4 we can see that the meter reading is having a normal correlation with the square feet and the floor count.

```
z=df_train_site_4.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(10,5))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading')
plt.ylabel('meter_site_4')
plt.title('Comparison of different energy usage at site 4')
plt.show()
```



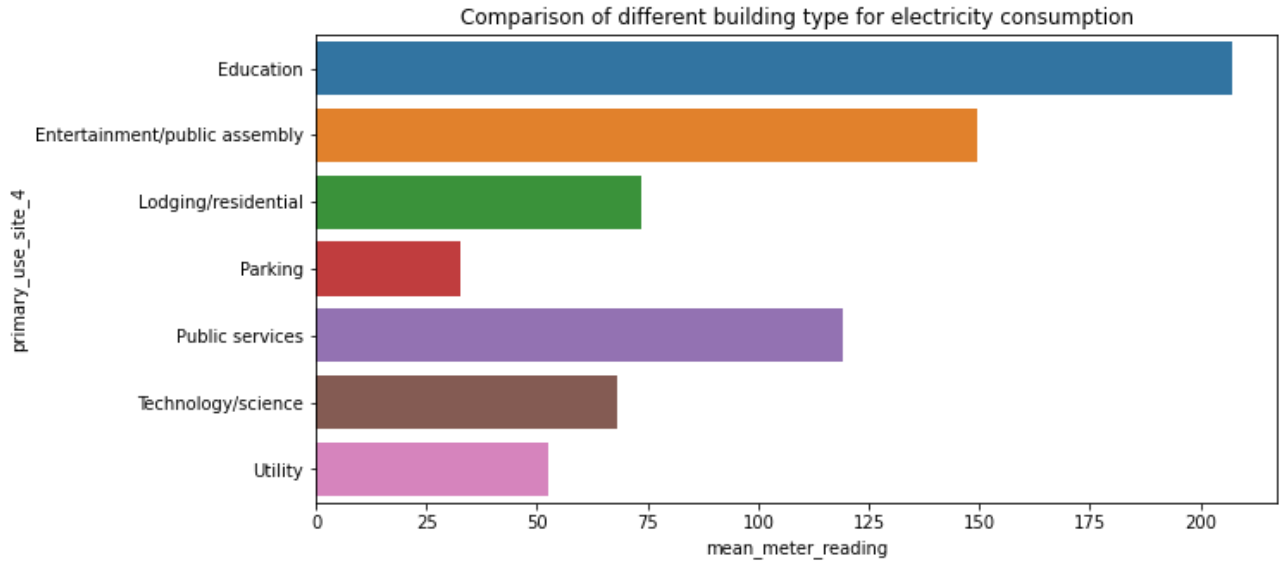
This plot shows that at site 4 the only source of energy consumption is electricity

```
z=df_train_site_4.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(10,5))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_site_4')
plt.title('Building count for different type of building use for electricity')
plt.show()
```



Building count for different building type for electricity usage

```
z=df_train_site_4.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(10,5))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_meter_reading')
plt.ylabel('primary_use_site_4')
plt.title('Comparison of different building type for electricity consumption')
plt.show()
```



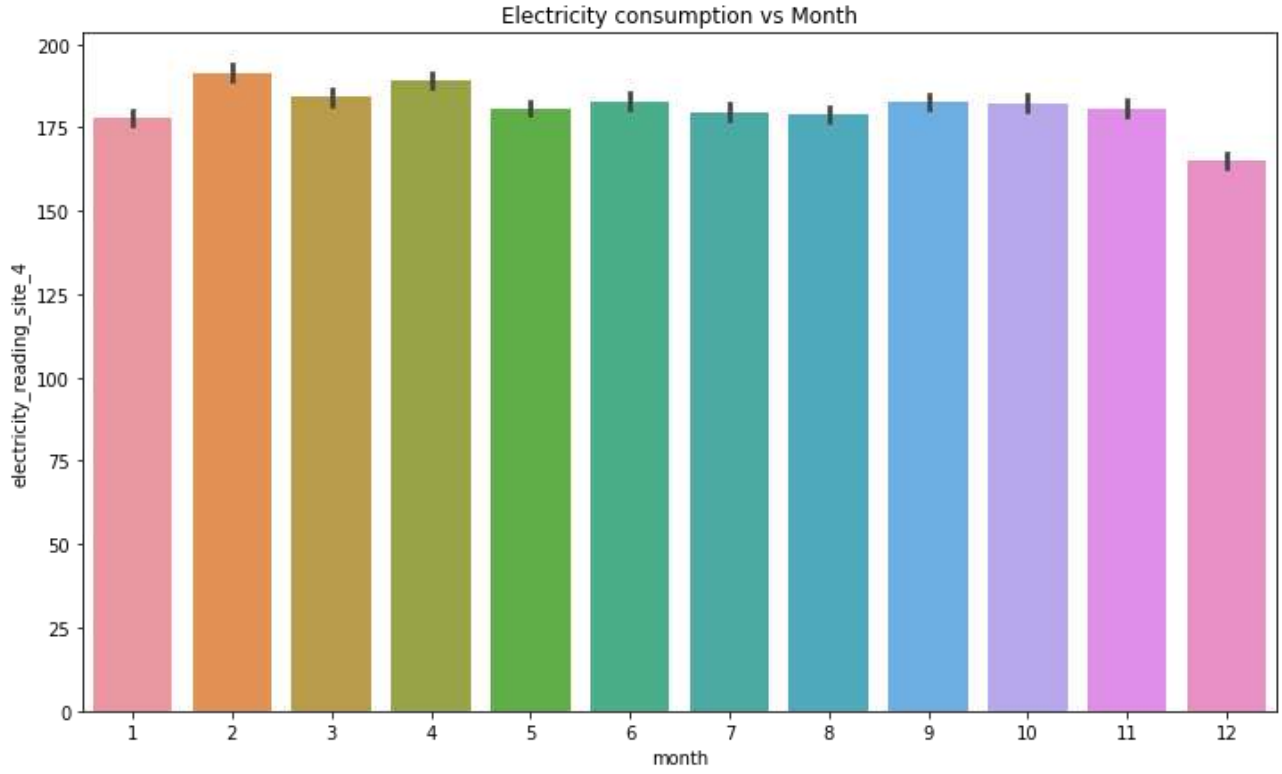
Here we can see that Education has the highest electricty consumption.

Entertainment buildings consume high electricity although they are less in number.

```
df_train_site_4['month']=df_train_site_4['timestamp'].dt.month
df_train_site_4['weekday']=df_train_site_4['timestamp'].dt.weekday
df_train_site_4['hour']=df_train_site_4['timestamp'].dt.hour
```

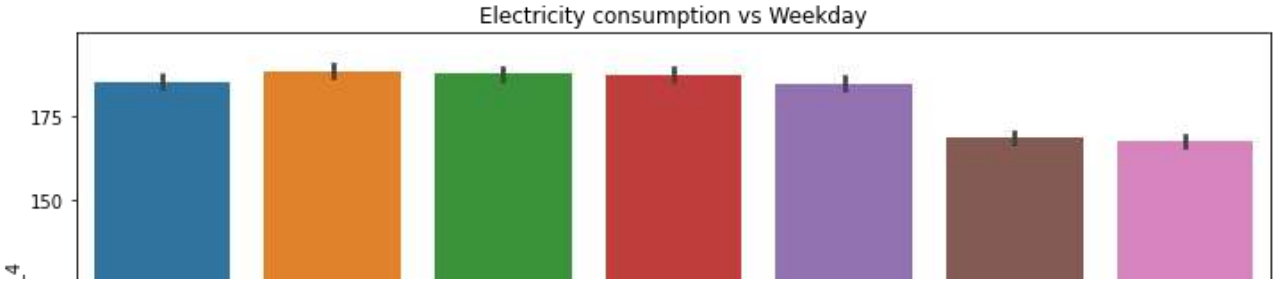
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_4
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_reading_site_4')
```

```
plt.title('Electricity consumption vs Month')
plt.show()
```



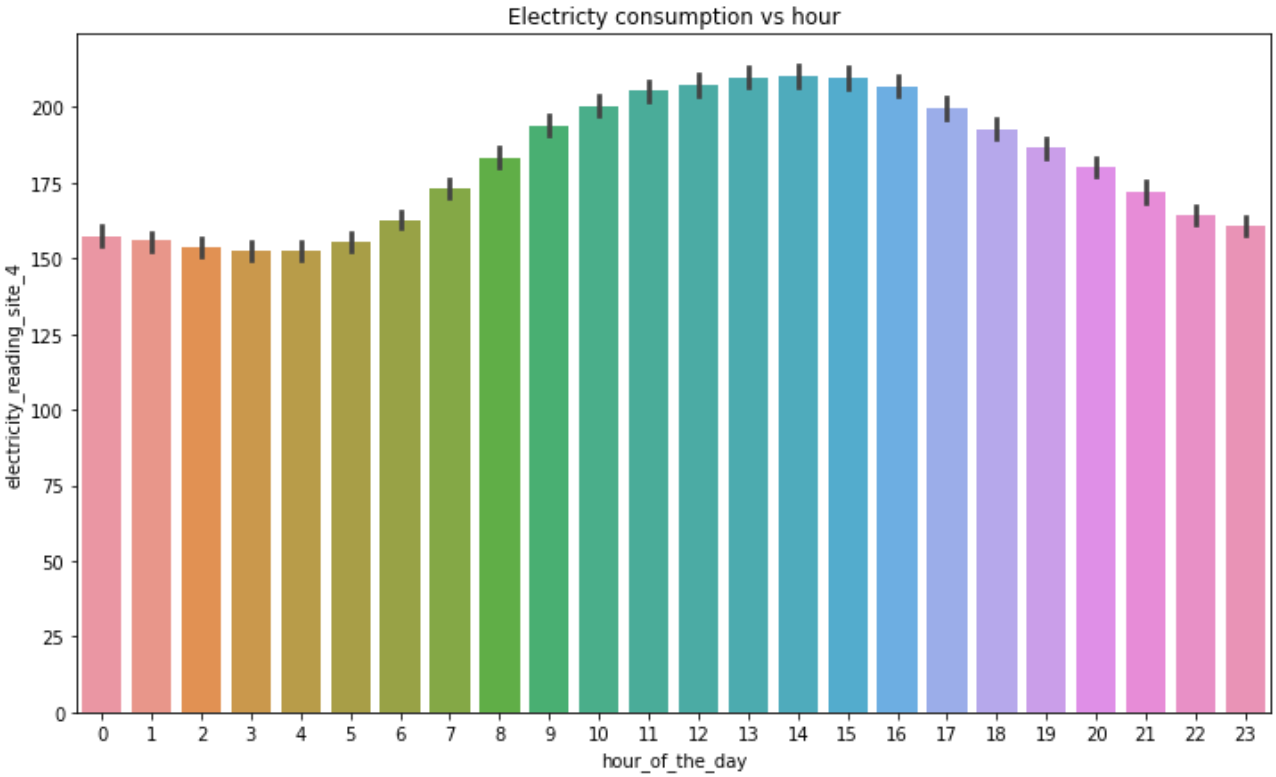
Electricity consumption shows variation over the month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_4
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_4')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



Electricity consumption is less for the weekend as compared to the weekday

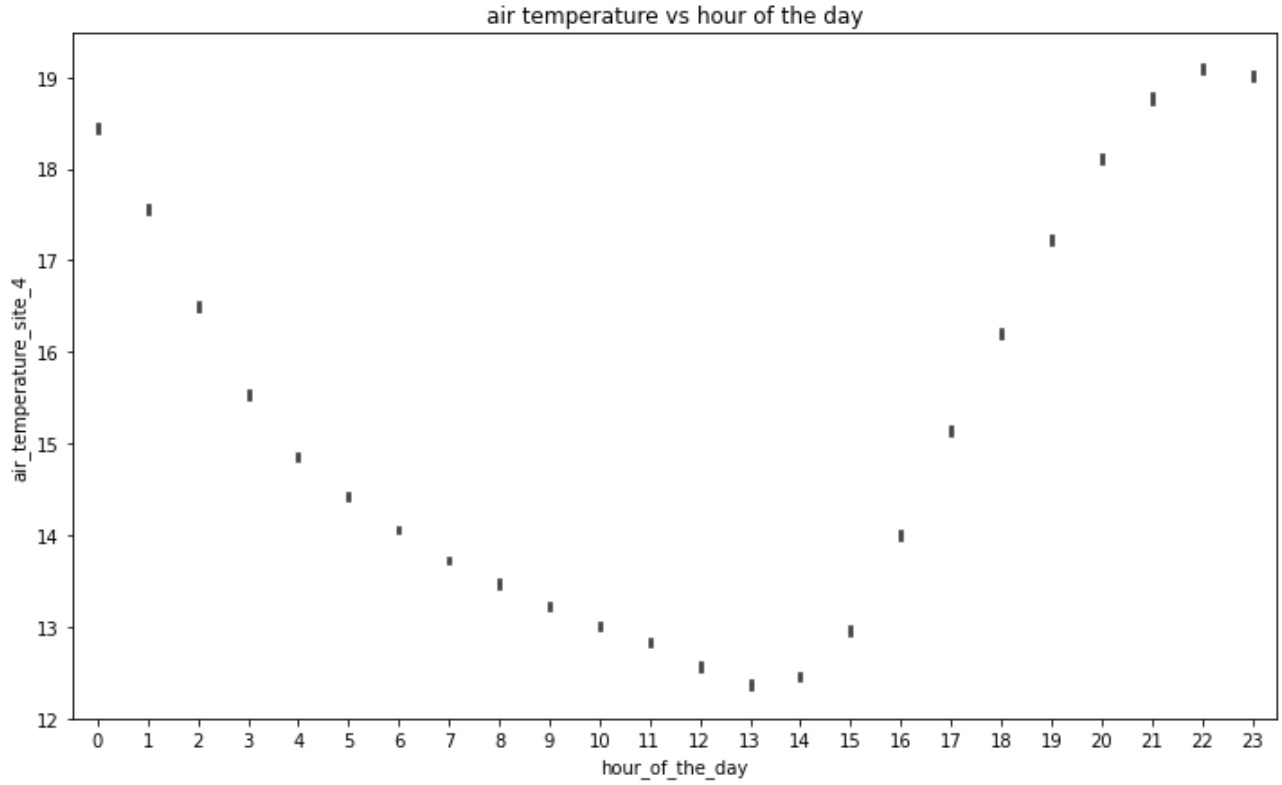

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_4
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_4')
plt.title('Electricty consumption vs hour')
plt.show()
```



Electricity consumption varies over hour of the day and it peaks during the day time and decreases gradually from the evening.

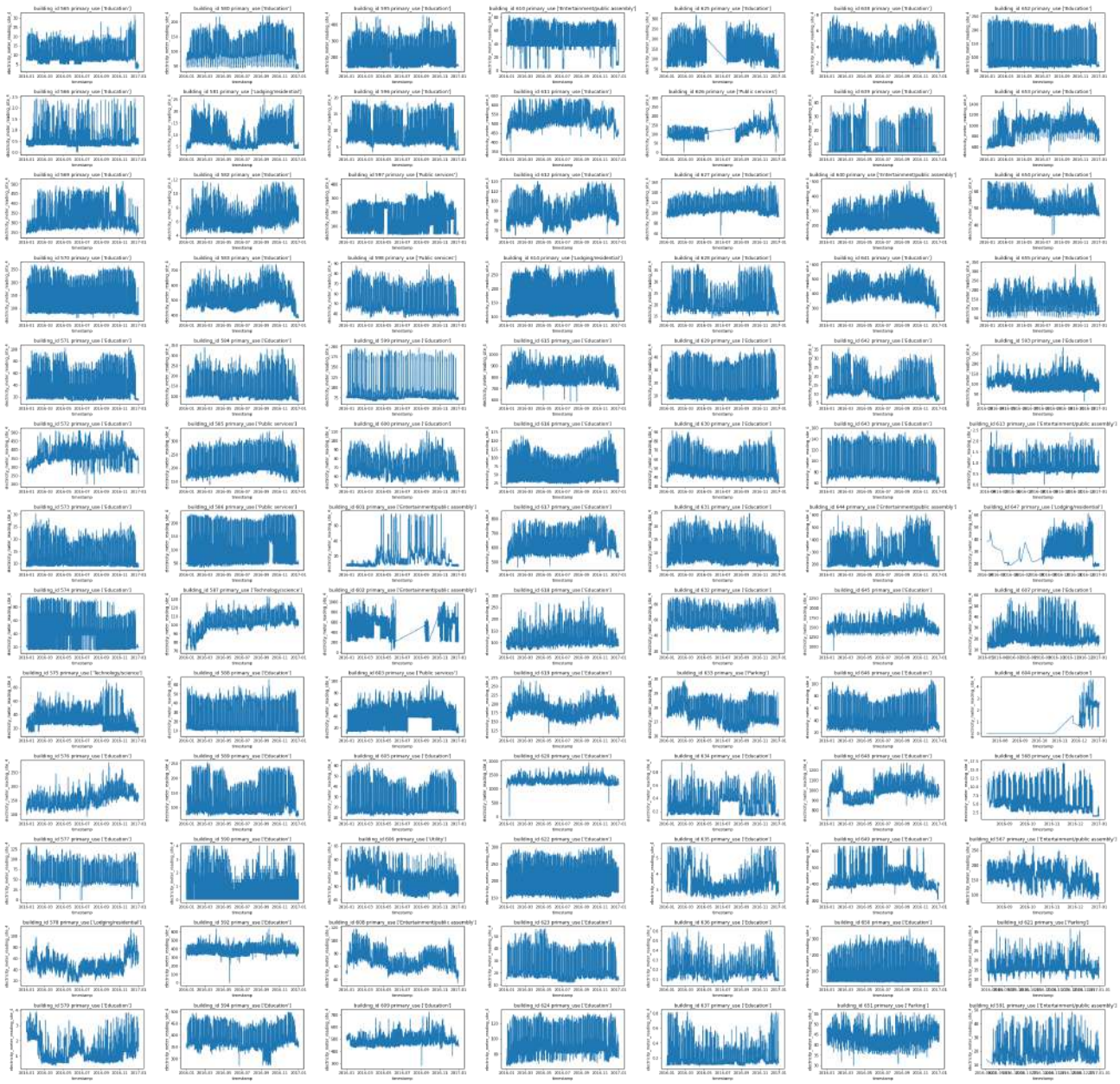
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_4
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_4')
plt.title('air temperature vs hour of the day')
plt.show()
```

```
plt.show()
```



Here we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings as the temperature peaks around 22:00 pm.

```
fig,ax=plt.subplots(figsize=(50,50),nrows=13,ncols=7)
for i in range(df_train_site_4['building_id'].nunique()):
  g=df_train_site_4['building_id'].unique()[i]
  axes=ax[i%13][i//13]
  z=df_train_site_4.loc[df_train_site_4['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_meter_reading_site_4')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```



Important observations

- Building 104 shows constant zero meter reading for the initial part of the plot which can definitely be an anomaly.
- Building 602,625,626 shows a constant slope for the meter reading during a certain period which also stands out from the other meter readings.

```
#Starting analysis for site 5
```

```
df_train_site_5=df_train_merge.loc[df_train_merge['site_id']==5]
```

```
df_train_site_5.isnull().sum()/df_train_site_5.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
```

```
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           0.01
floor_count          0.00
air_temperature      0.00
cloud_coverage       0.69
dew_temperature      0.00
precip_depth_1_hr    1.00
sea_level_pressure   1.00
wind_direction       0.04
wind_speed           0.00
dtype: float64
```

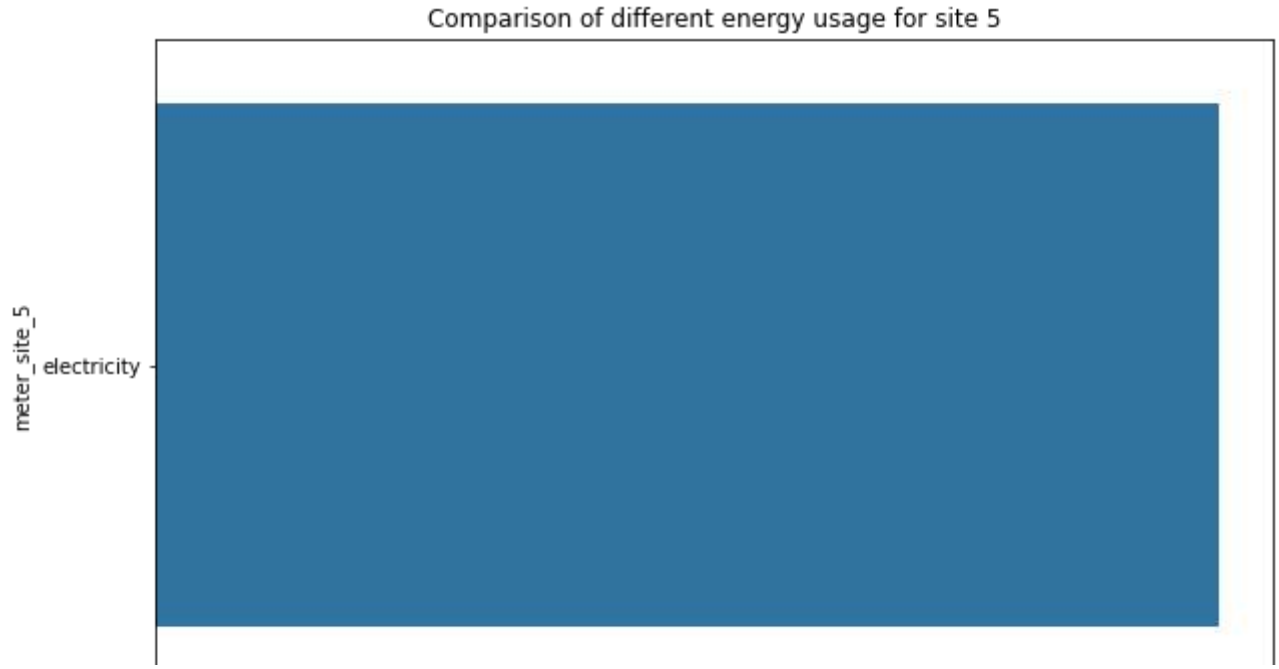
We need to fill the missing values for the training process.

```
df_corr_5=df_train_site_5.corr()
df_corr_5.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	-0.00	nan	0.19	-0.19	-0.01	-0.00
meter_reading	-0.00	1.00	nan	0.60	0.12	0.34	-0.00
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.19	0.60	nan	1.00	0.10	0.58	-0.00
year_built	-0.19	0.12	nan	0.10	1.00	-0.09	-0.00
floor_count	-0.01	0.34	nan	0.58	-0.09	1.00	0.00
air_temperature	-0.00	-0.00	nan	-0.00	-0.00	0.00	1.00
cloud_coverage	0.00	0.01	nan	-0.00	-0.00	-0.00	-0.05
dew_temperature	-0.00	-0.02	nan	0.00	-0.00	-0.00	0.90
precip_depth_1_hr	nan	nan	nan	nan	nan	nan	nan
sea_level_pressure	nan	nan	nan	nan	nan	nan	nan
wind_direction	-0.00	-0.02	nan	0.00	0.00	0.00	0.10
wind_speed	0.00	0.04	nan	0.00	-0.00	-0.00	0.05

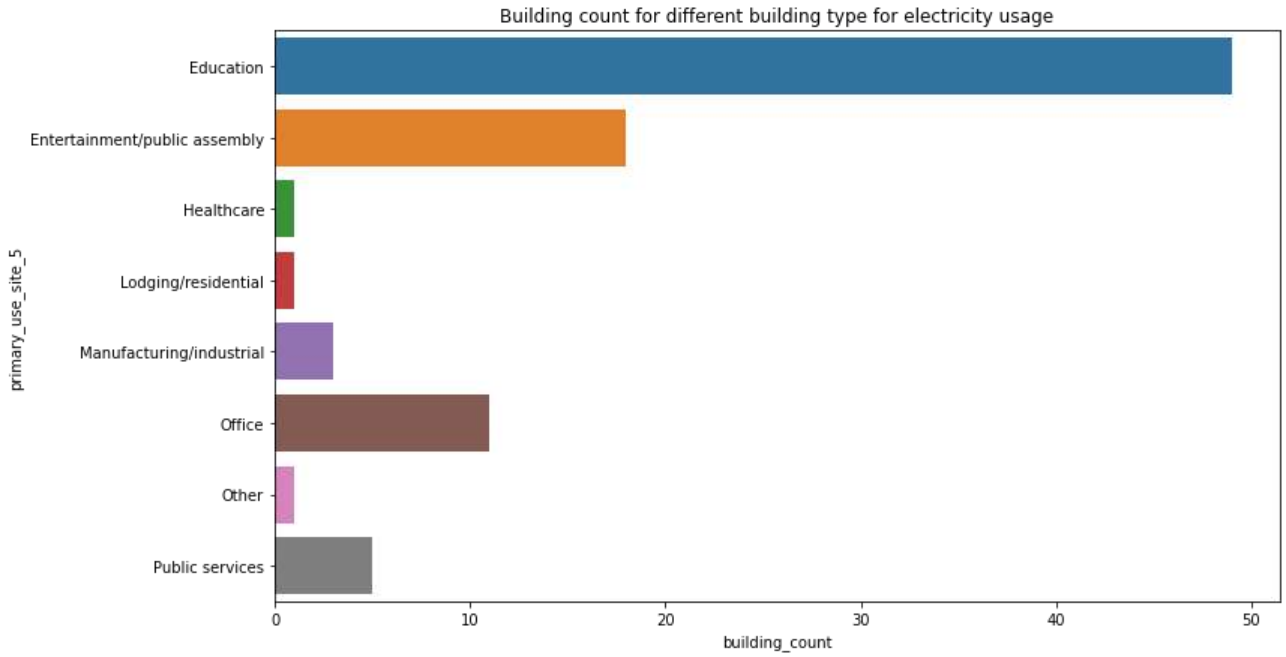
The correlation plot shows that the meter reading shows normal correlation with the square feet and is weakly correlated with the floor count.

```
z=df_train_site_5.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading')
plt.ylabel('meter_site_5')
plt.title('Comparison of different energy usage for site 5')
plt.show()
```



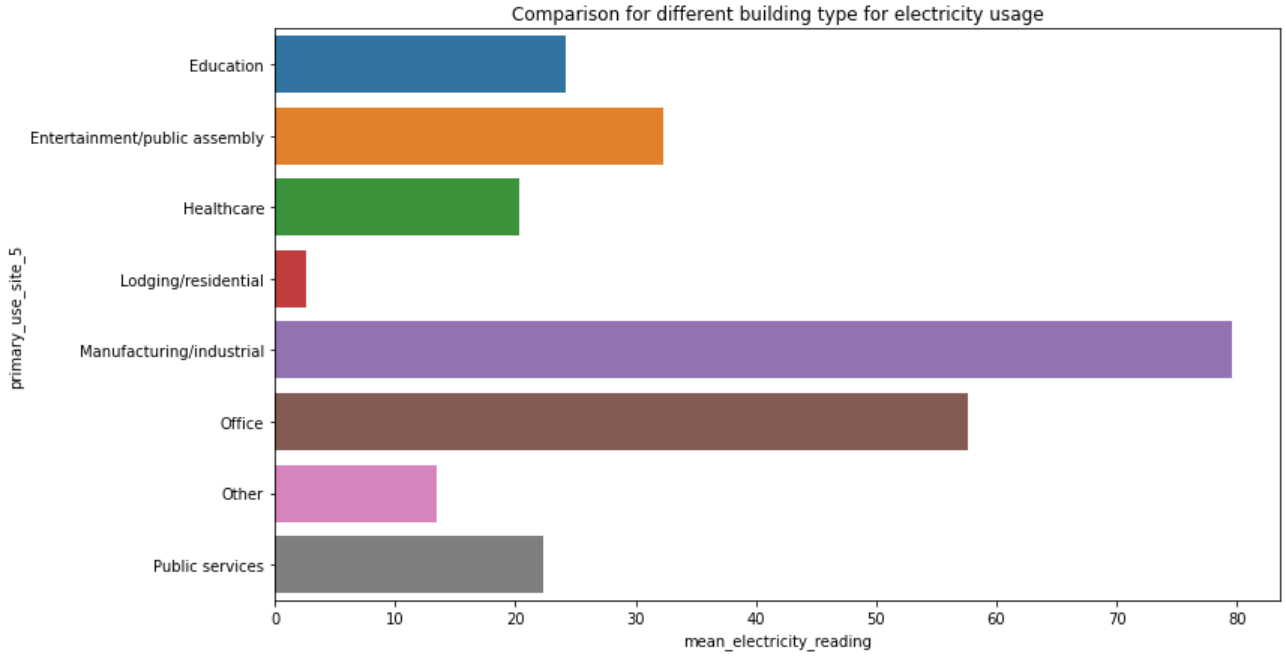
This plot shows that site 5 only consumes electricity

```
z=df_train_site_5.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_site_5')
plt.title('Building count for different building type for electricity usage')
plt.show()
```



The building count for different building type for electricity usage

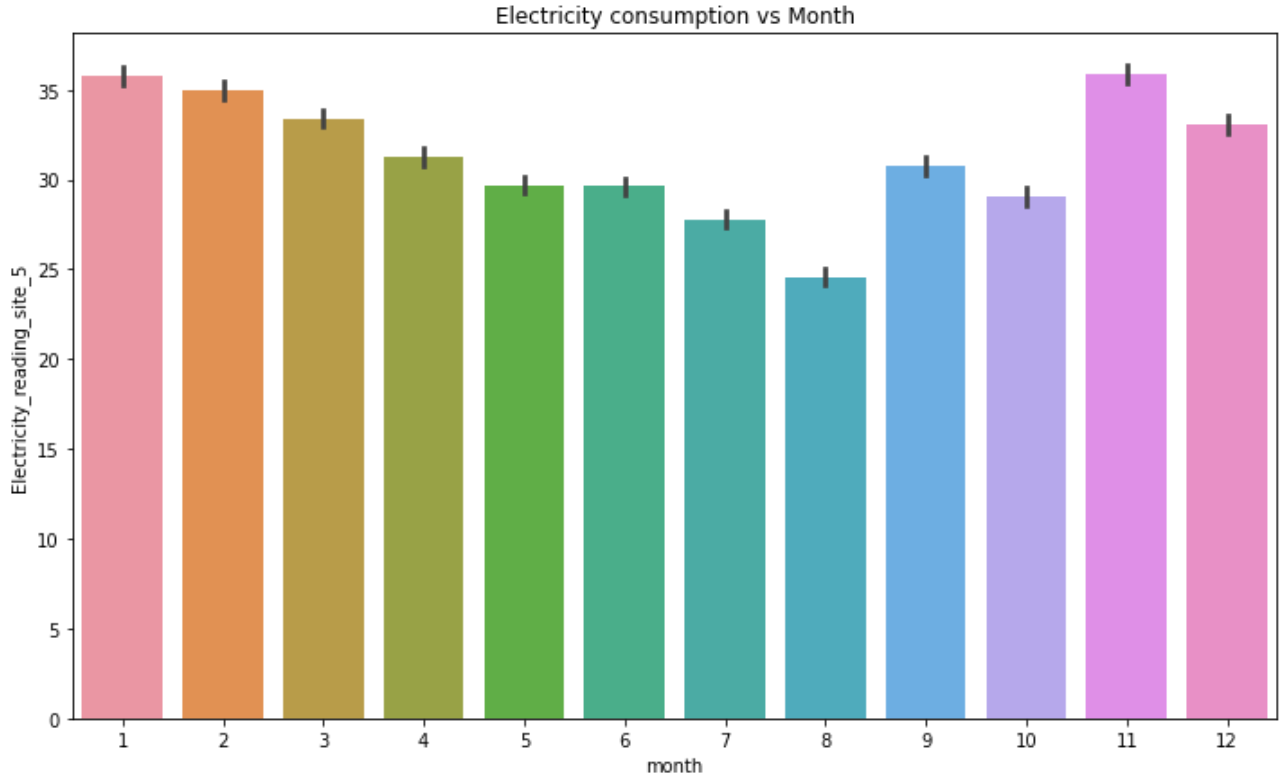
```
z=df_train_site_5.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_reading')
plt.ylabel('primary_use_site_5')
plt.title('Comparison for different building type for electricity usage')
plt.show()
```



This plot shows that manufacturing industries consumes higher electrical consumption

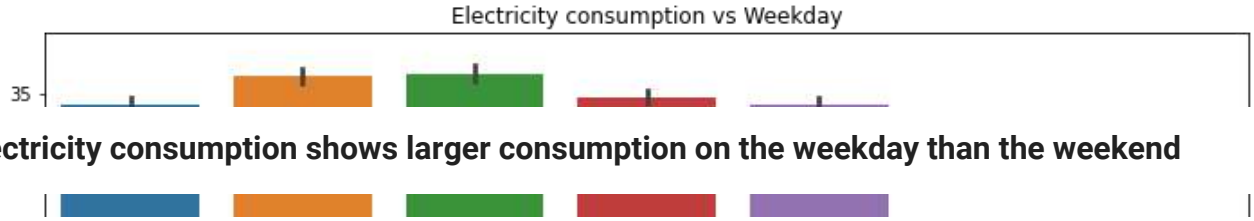
```
df_train_site_5['month']=df_train_site_5['timestamp'].dt.month
df_train_site_5['weekday']=df_train_site_5['timestamp'].dt.weekday
df_train_site_5['hour']=df_train_site_5['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_5
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('Electricity_reading_site_5')
plt.title('Electricity consumption vs Month')
plt.show()
```



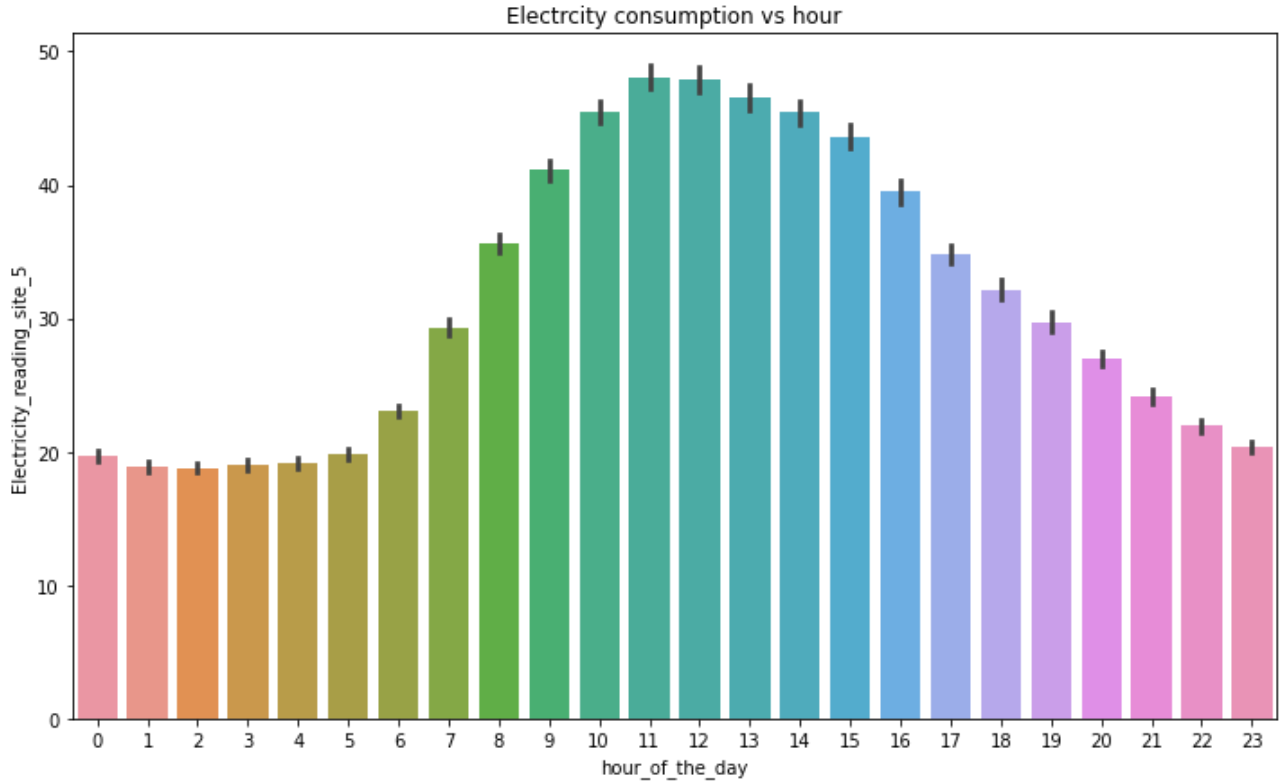
This plot shows electrical energy consumption over different month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_5
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('Electricity_reading_site_5')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



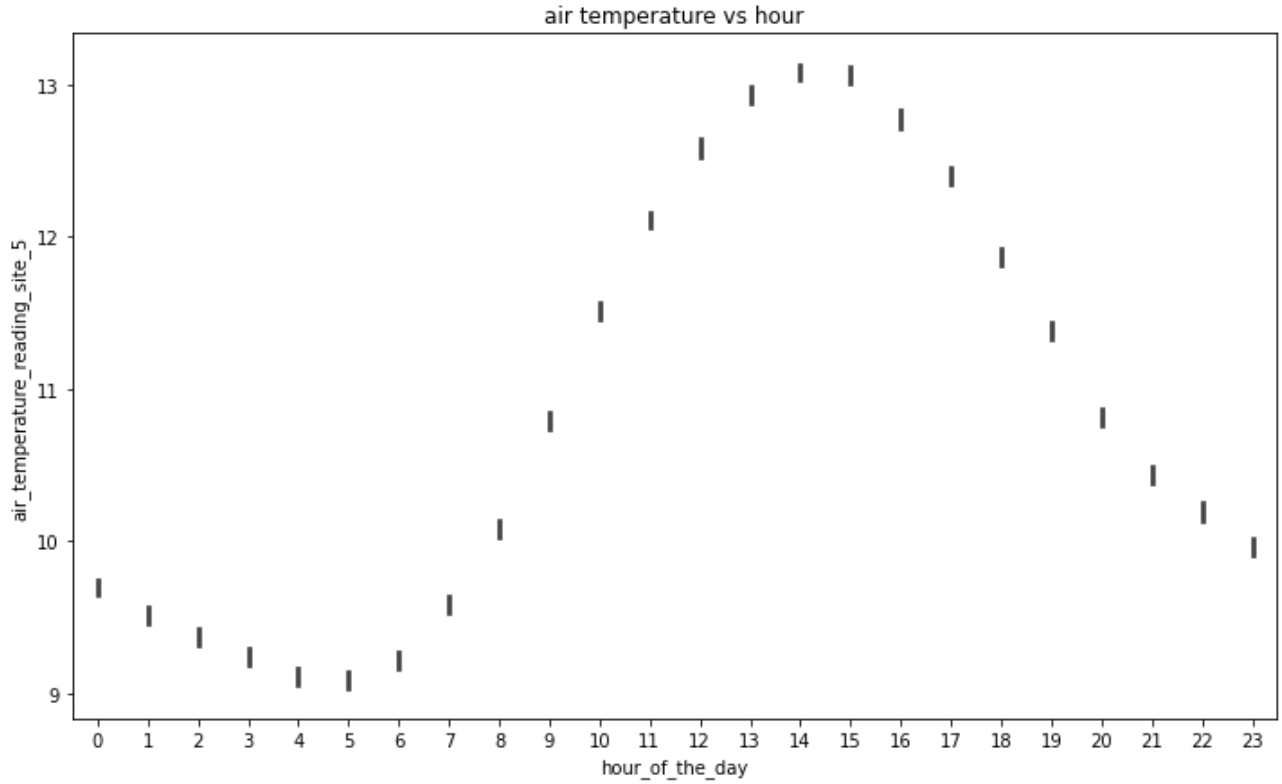
The Electricity consumption shows larger consumption on the weekday than the weekend

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_5
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('Electricity_reading_site_5')
plt.title('Electrcity consumption vs hour')
plt.show()
```



The Electrical consumption peaks during the daytime and decreases gradually as evening approaches

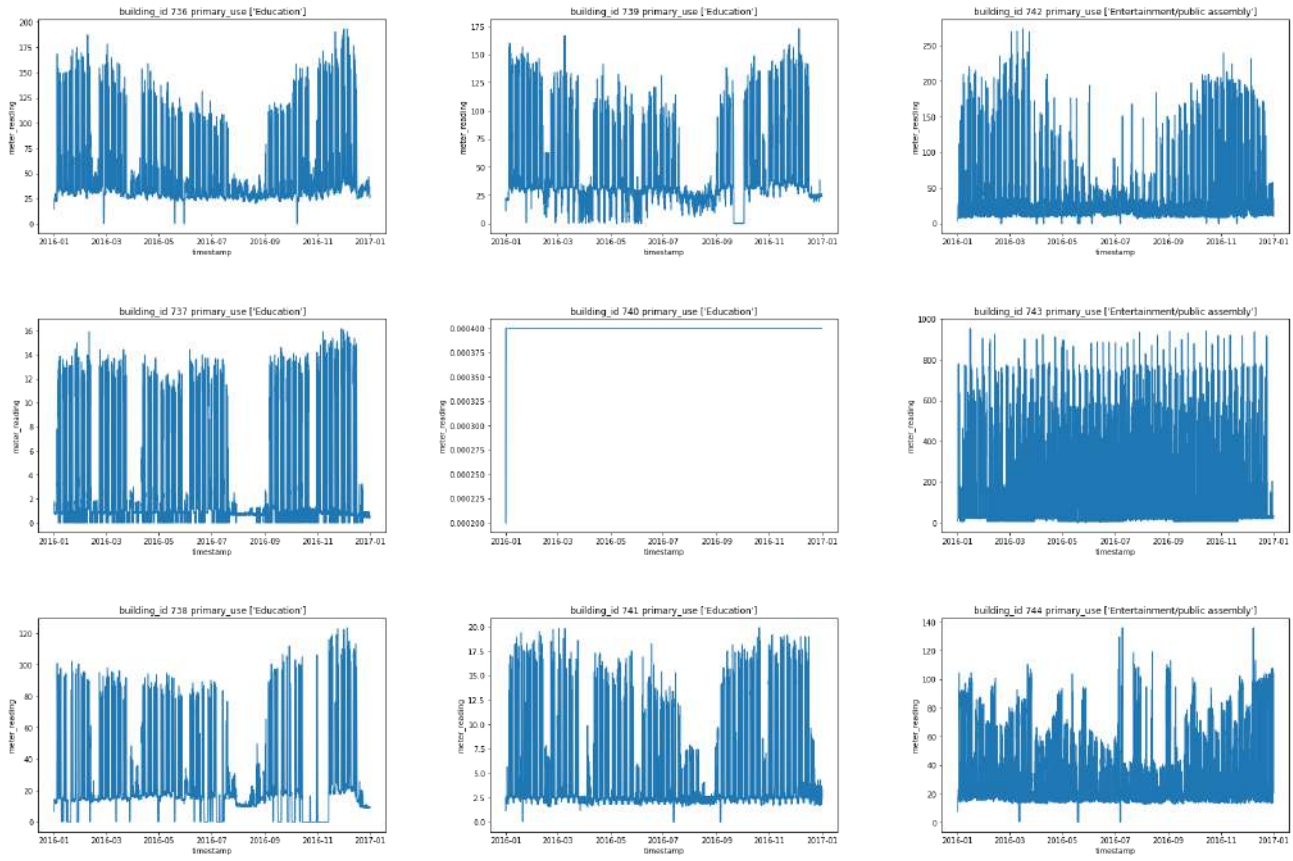
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_5
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_reading_site_5')
plt.title('air temperature vs hour')
plt.show()
```



The weather timestamp is in alignment with the local timestamp as the air temperature peaks around 14:00 pm

```
fig,ax=plt.subplots(figsize=(40,50),nrows=16,ncols=5)
for i in range(80):
  g=df_train_site_5['building_id'].unique()[i]
  axes=ax[i%16][i//16]
  z=df_train_site_5.loc[df_train_site_5['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_meter_reading_site_5')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```

```
fig,ax=plt.subplots(figsize=(30,20),nrows=3,ncols=3)
for i in range(9):
  g=df_train_site_5['building_id'].unique()[80:90][i]
  axes=ax[i%3][i//3]
  z=df_train_site_5.loc[df_train_site_5['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('meter_reading')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```



The total number of building at site 5 is 89 and I am dividing it into 80 and 9 so that I can represent it clearly in the plot.

Building 723 693 681 733 739 738 shows streaks of constant zero values during certain months and that might be an anomaly and we need to remove that.

Building 740 shows a constant reading over the whole year which can definitely be an anomalous reading and we might have to drop that building during training as we would not want to predict constant reading for that next year.

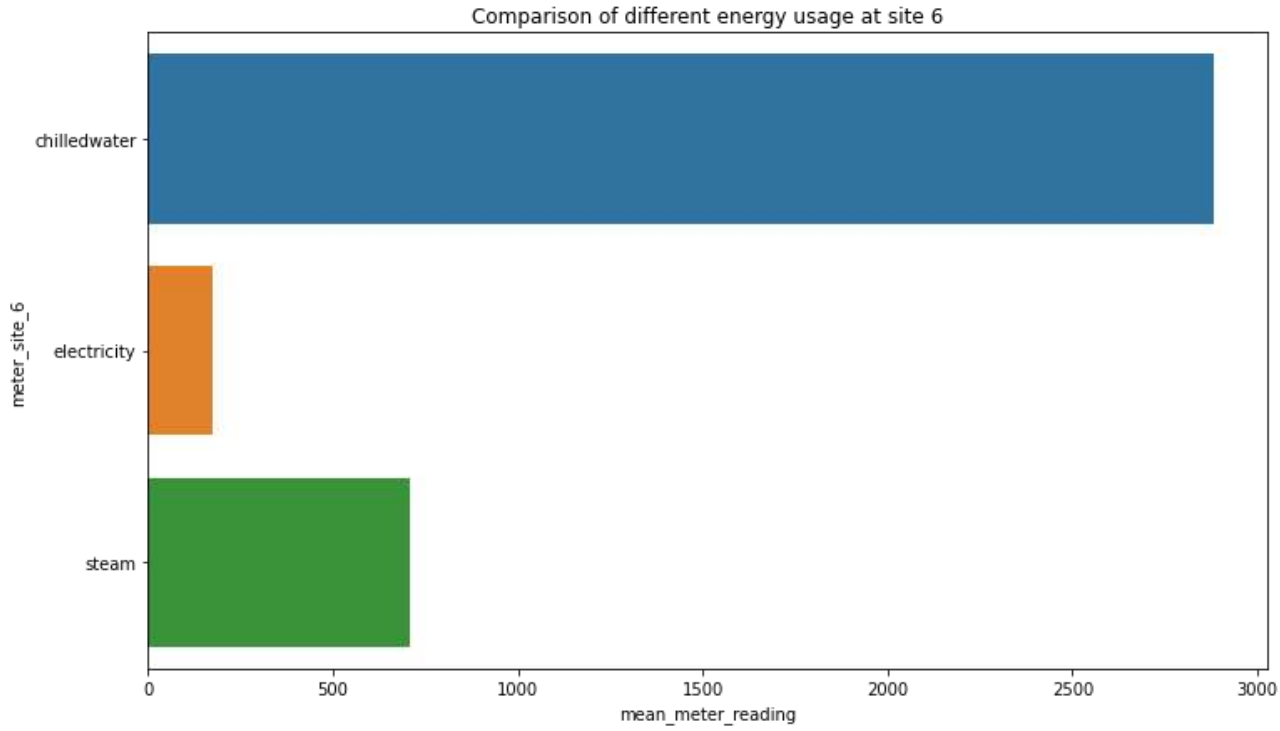
```
#Starting analysis for site 6


df_train_site_6=df_train_merge.loc[df_train_merge['site_id']==6]


df_corr_6=df_train_site_6.corr()
df_corr_6.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.04	nan	0.82	nan	nan	-0.01
meter_reading	0.04	1.00	nan	0.02	nan	nan	0.02
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.82	0.02	nan	1.00	nan	nan	-0.01
year_built	nan	nan	nan	nan	nan	nan	nan
floor_count	nan	nan	nan	nan	nan	nan	nan

Here the meter reading has not got strong correlation with any of the features.

dew_temperature	-0.01	0.02	nan	-0.01	nan	nan	0.86

```
z=df_train_site_6.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading')
plt.ylabel('meter_site_6')
plt.title('Comparison of different energy usage at site 6')
plt.show()
```

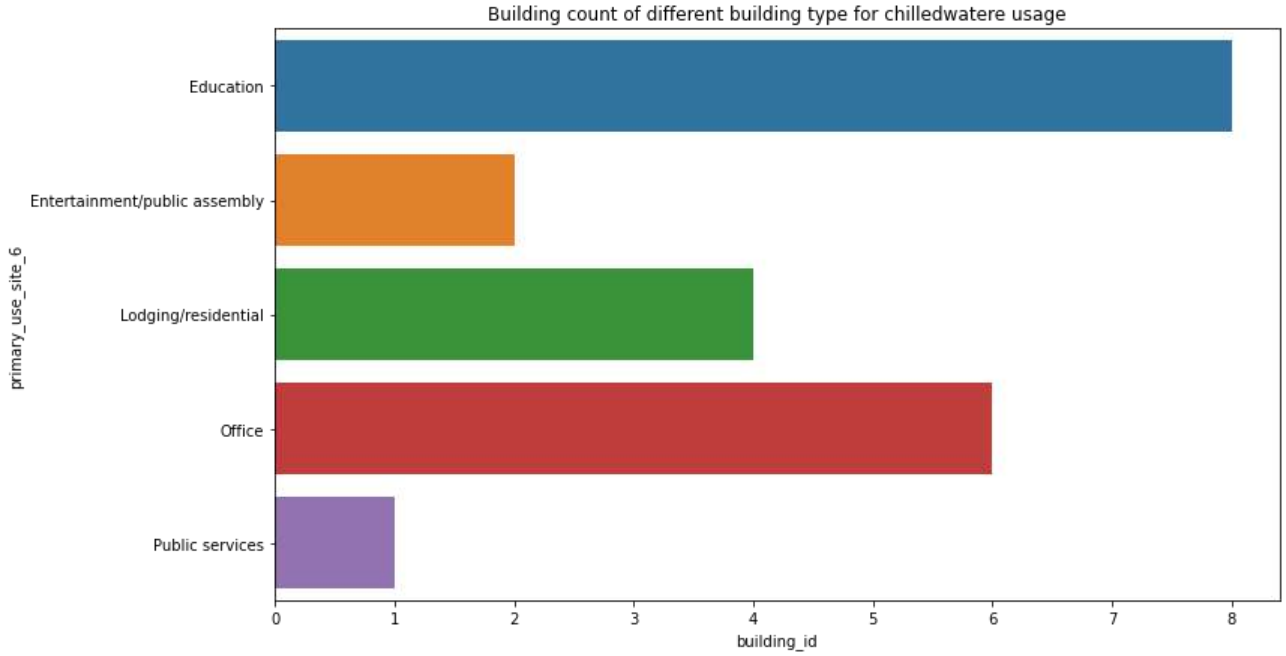


This plot shows that chilledwater consumption is highest at site 6

```
df_train_site_6_meter_1=df_train_site_6.loc[df_train_site_6['meter']=='chilledwater']
```

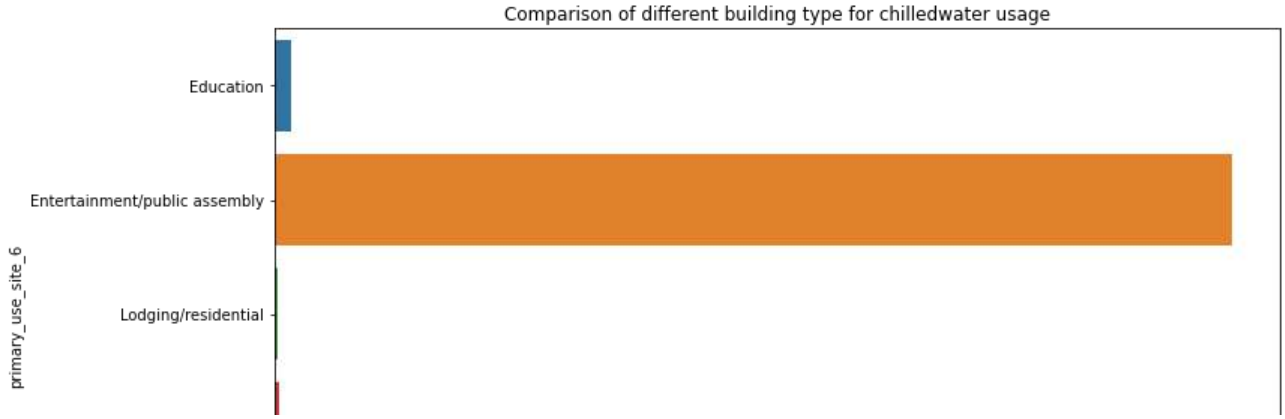
```
z=df_train_site_6_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
```

```
plt.xlabel('building_id')
plt.ylabel('primary_use_site_6')
plt.title('Building count of different building type for chilledwatere usage')
plt.show()
```



The above plot represents the count of different building type for chilledwater usage.

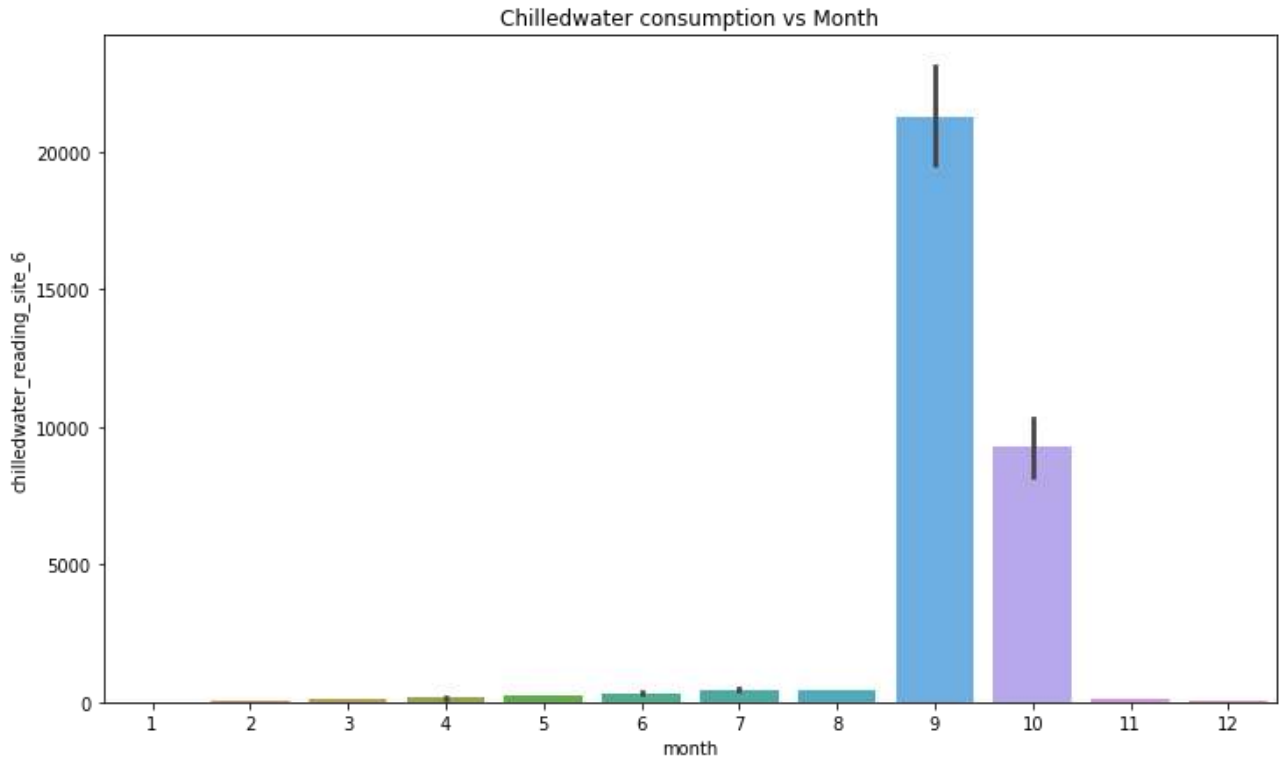
```
z=df_train_site_6_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading')
plt.ylabel('primary_use_site_6')
plt.title('Comparison of different building type for chilledwater usage')
plt.show()
```



At site 6 we can see that enteratinment building type has highest consumption of chilledwater.All other type of buildings are having very less chilledwater consumption.

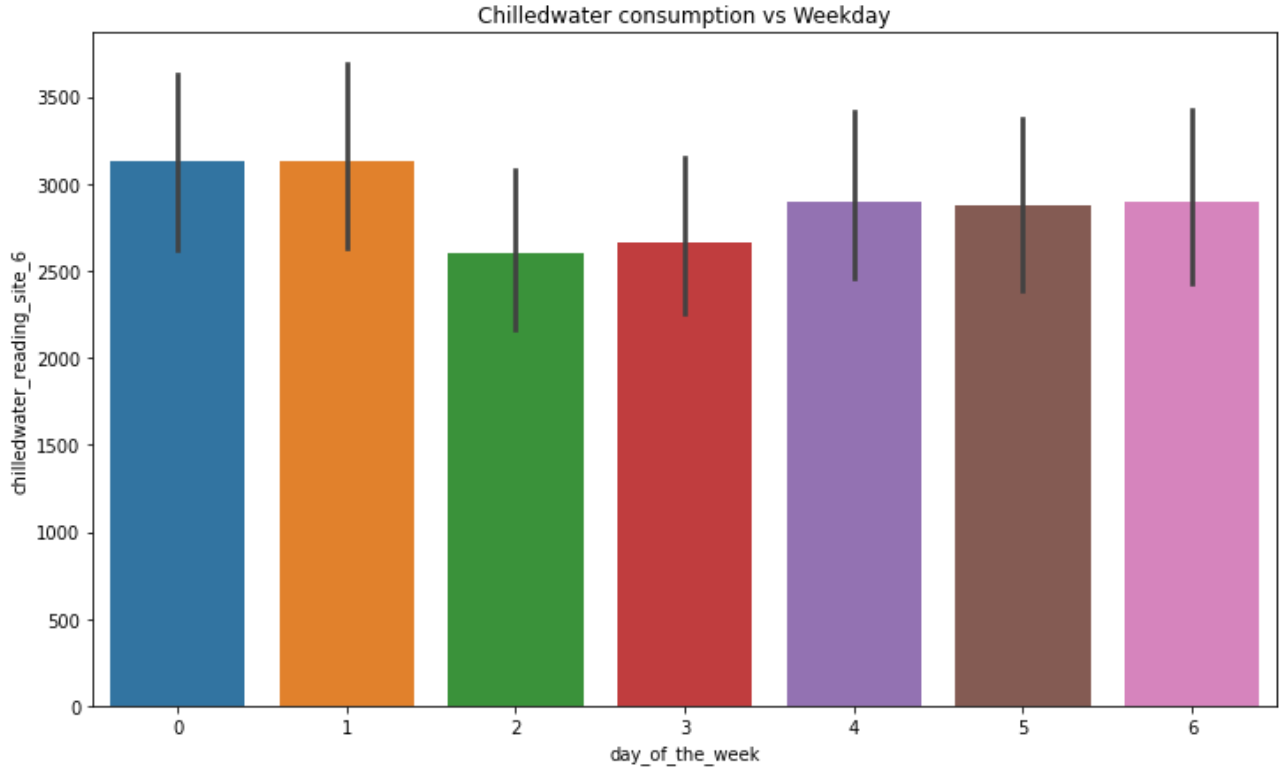
```
df_train_site_6_meter_1['month']=df_train_site_6_meter_1['timestamp'].dt.month
df_train_site_6_meter_1['weekday']=df_train_site_6_meter_1['timestamp'].dt.weekday
df_train_site_6_meter_1['hour']=df_train_site_6_meter_1['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=df_train_site_6_meter_1,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading_site_6')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



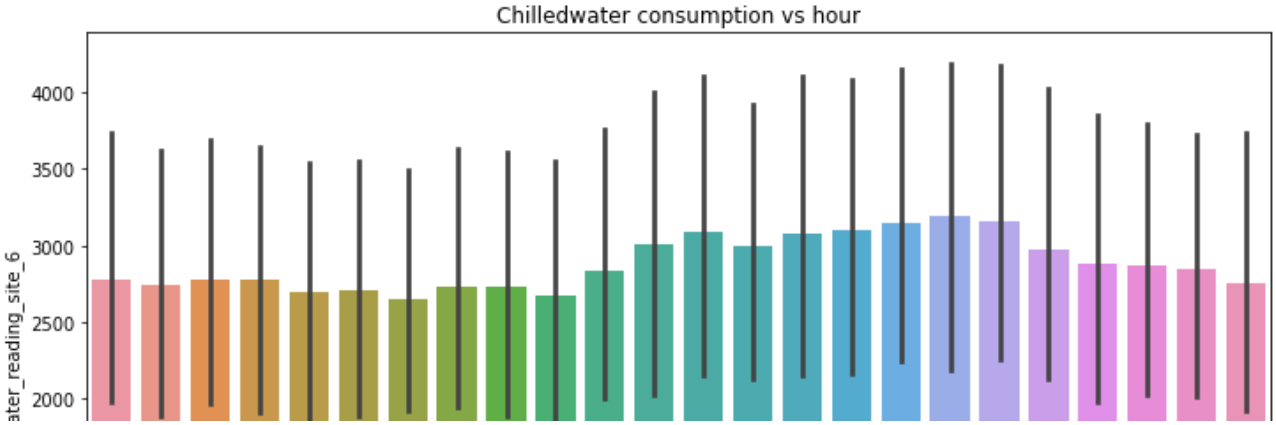
The above plot shows that the consumption of chilled water is significant for 9-10 month.

```
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=df_train_site_6_meter_1,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading_site_6')
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



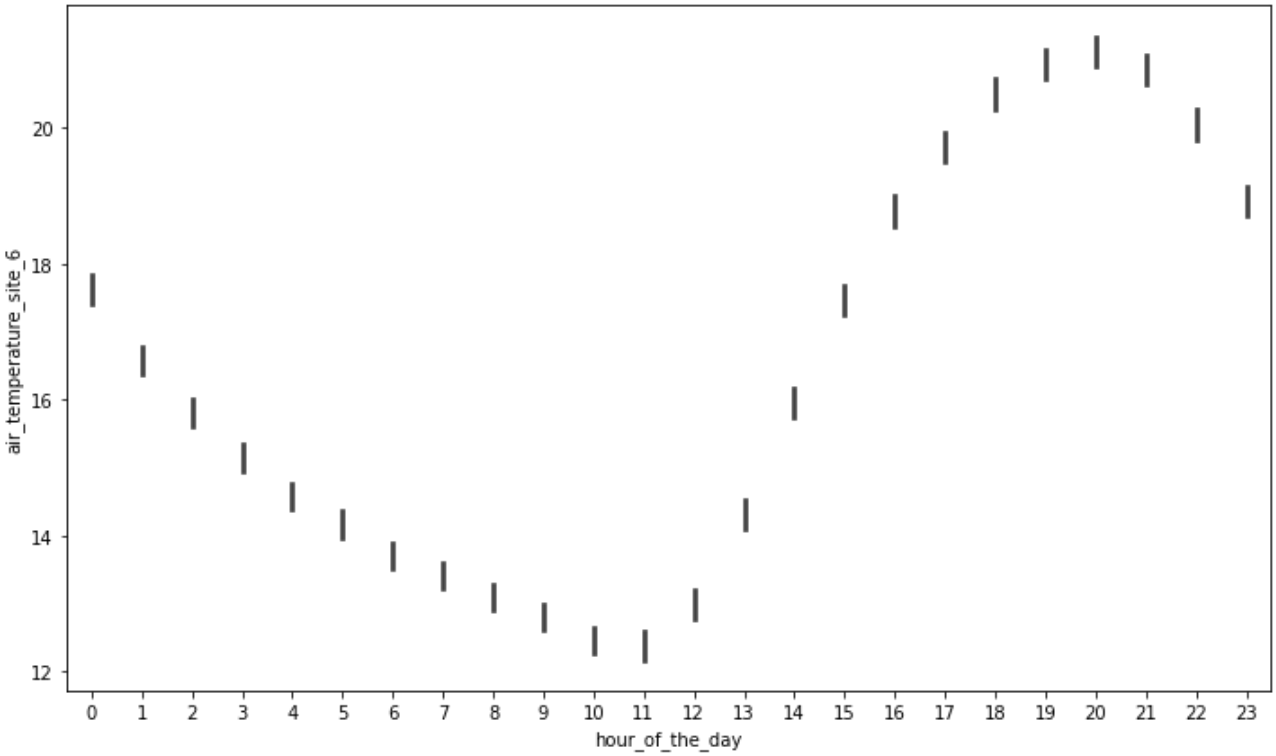
Chilledwater consumption consumption is varying over the week

```
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=df_train_site_6_meter_1,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilledwater_reading_site_6')
plt.title('Chilledwater consumption vs hour')
plt.show()
```



Chilledwater consumption is the highest between 10:00 am and 19:00 pm

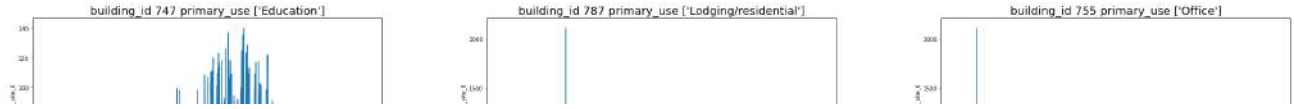
```
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=df_train_site_6_meter_1,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_6')
plt.show()
```



Weather timestamp is not in alignment with the local timestamp of the hourly meter readings as the air temperature peaks around 20:00 pm.

```
fig,ax=plt.subplots(figsize=(40,70),nrows=7,ncols=3)
for i in range(df_train_site_6_meter_1['building_id'].nunique()):
  g=df_train_site_6_meter_1['building_id'].unique()[i]
  axes=ax[i%7][i//7]
  z=df_train_site_6_meter_1.loc[df_train_site_6_meter_1['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
```

```
    axes.set_xlabel('timestamp')
    axes.set_ylabel('chilledwater_meter_reading_site_6')
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
    plt.subplots_adjust(hspace=0.4,wspace=0.3)
```



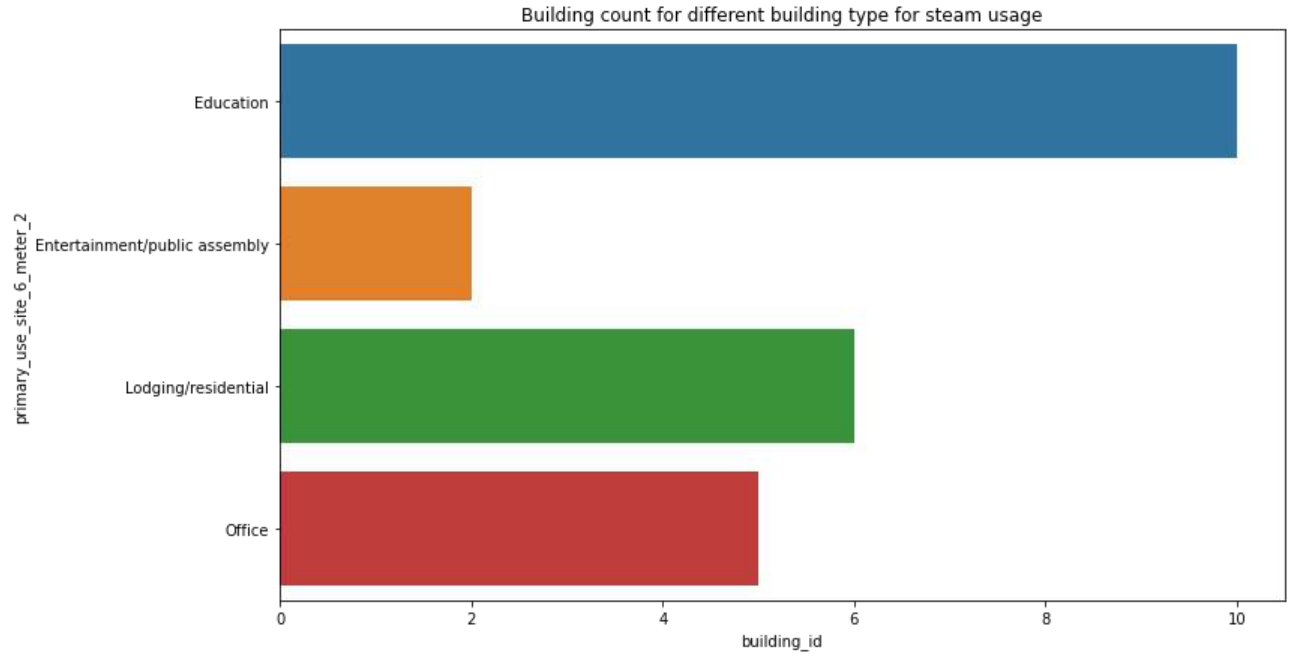
Important Observations

- Almost in all the buildings above we can see a single large spike at which needs to be removed as it might be due to faulty reading.

```
df_train_site_6_meter_2=df_train_site_6.loc[df_train_site_6['meter']=='steam']
```



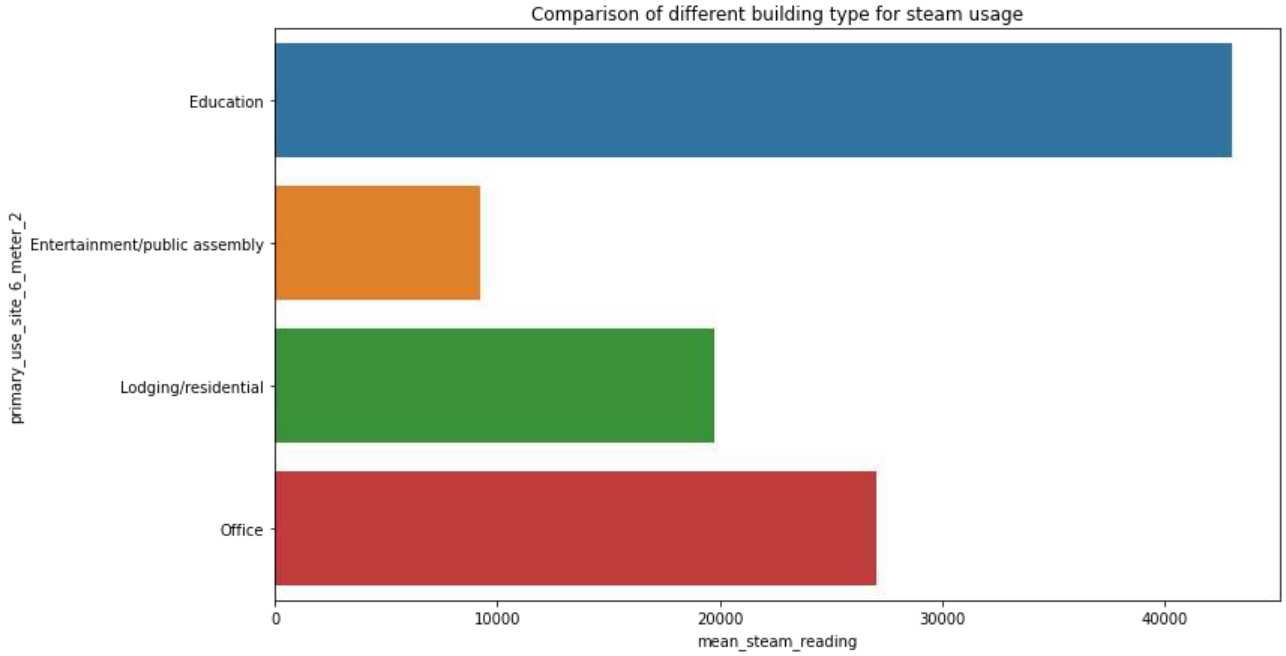
```
z=df_train_site_6_meter_2.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_id')
plt.ylabel('primary_use_site_6_meter_2')
plt.title('Building count for different building type for steam usage')
plt.show()
```



This plot shows the building count for different building type for steam usage.


```
z=df_train_site_6_meter_2.groupby(['primary_use'])
z=z['meter_reading'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
```

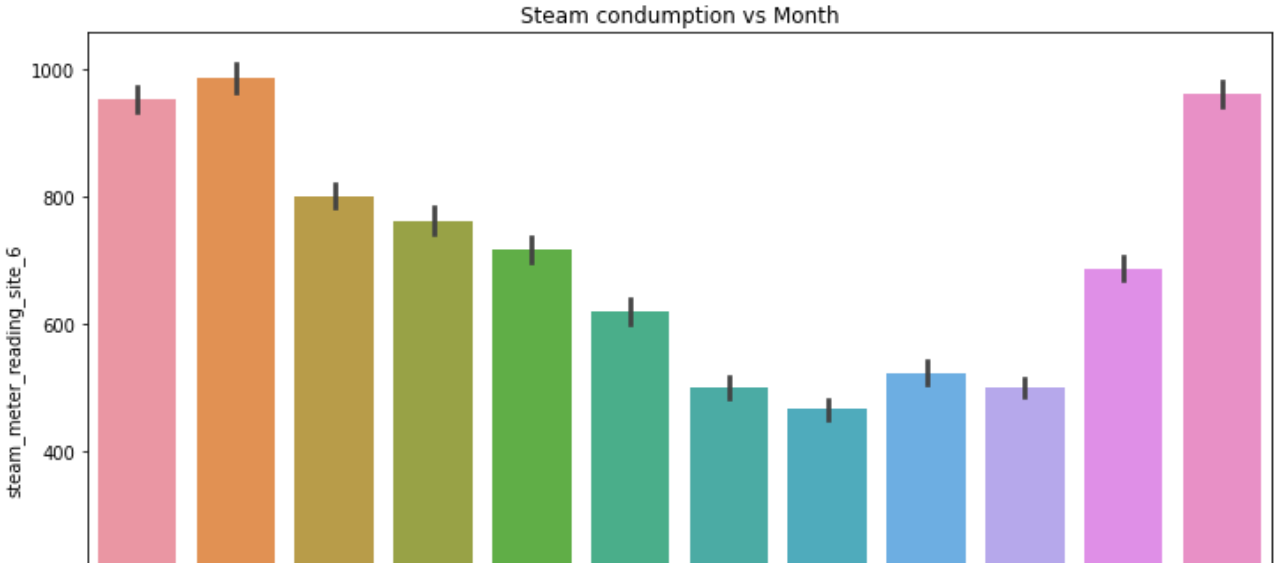
```
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_steam_reading')
plt.ylabel('primary_use_site_6_meter_2')
plt.title('Comparison of different building type for steam usage')
plt.show()
```



The above plot shows that educational building shows the highest steam consumption.

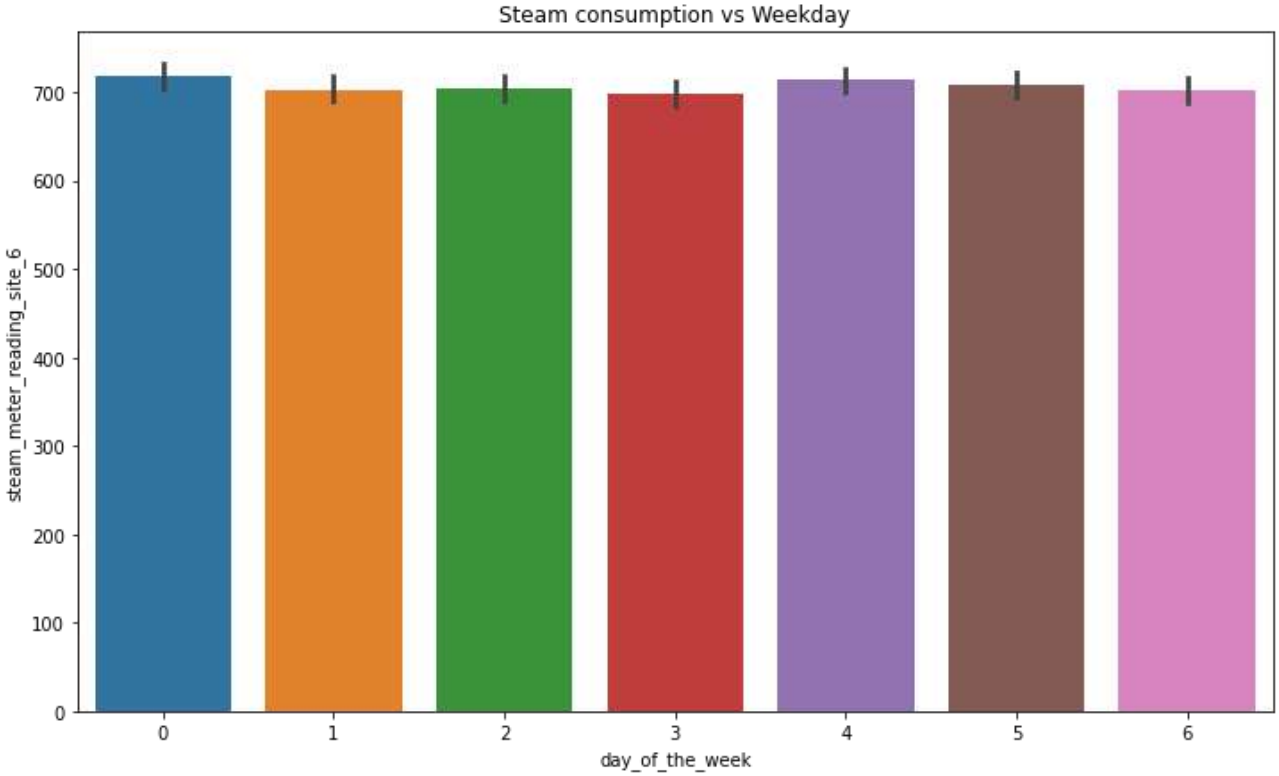
```
df_train_site_6_meter_2['month']=df_train_site_6_meter_2['timestamp'].dt.month
df_train_site_6_meter_2['weekday']=df_train_site_6_meter_2['timestamp'].dt.weekday
df_train_site_6_meter_2['hour']=df_train_site_6_meter_2['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_2
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('steam_meter_reading_site_6')
plt.title('Steam condumption vs Month')
plt.show()
```



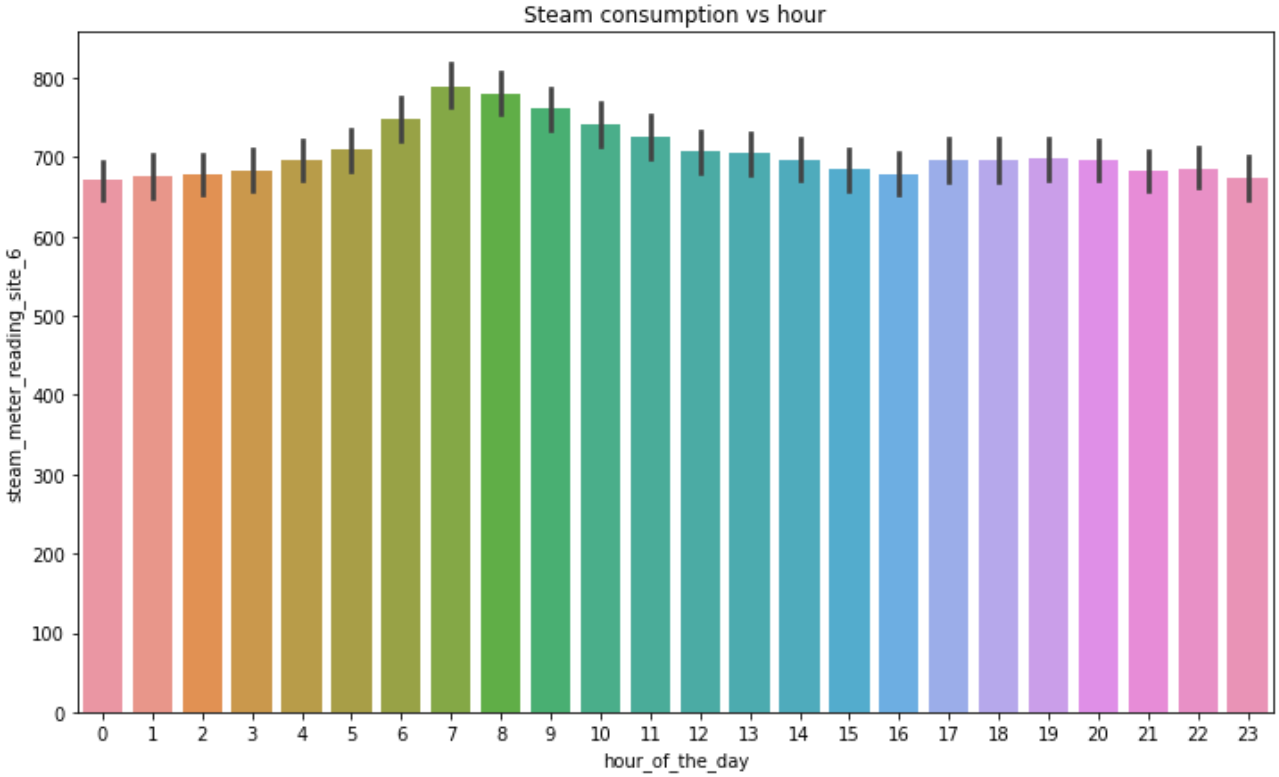
As we can see that the steam consumption is higher for the winter months and gradually decreases in the winter.


```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_2
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('steam_meter_reading_site_6')
plt.title('Steam consumption vs Weekday')
plt.show()
```



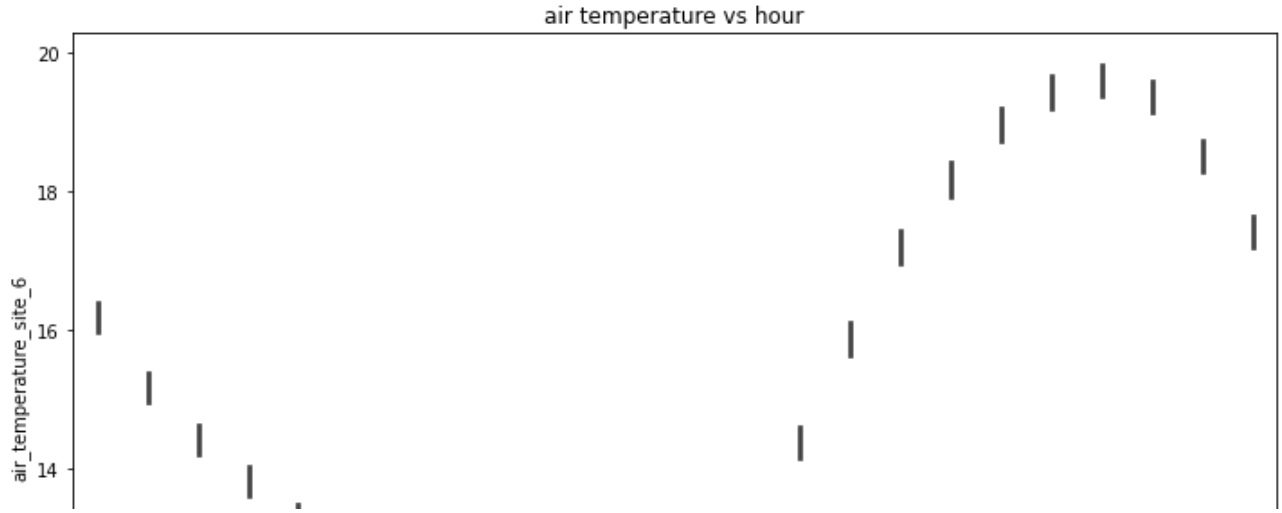
The steam consumption doed not show much variation over the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('steam_meter_reading_site_6')
plt.title('Steam consumption vs hour')
plt.show()
```



Steam consumption shows higher consumption during the morning time

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_6')
plt.title('air temperature vs hour')
plt.show()
```

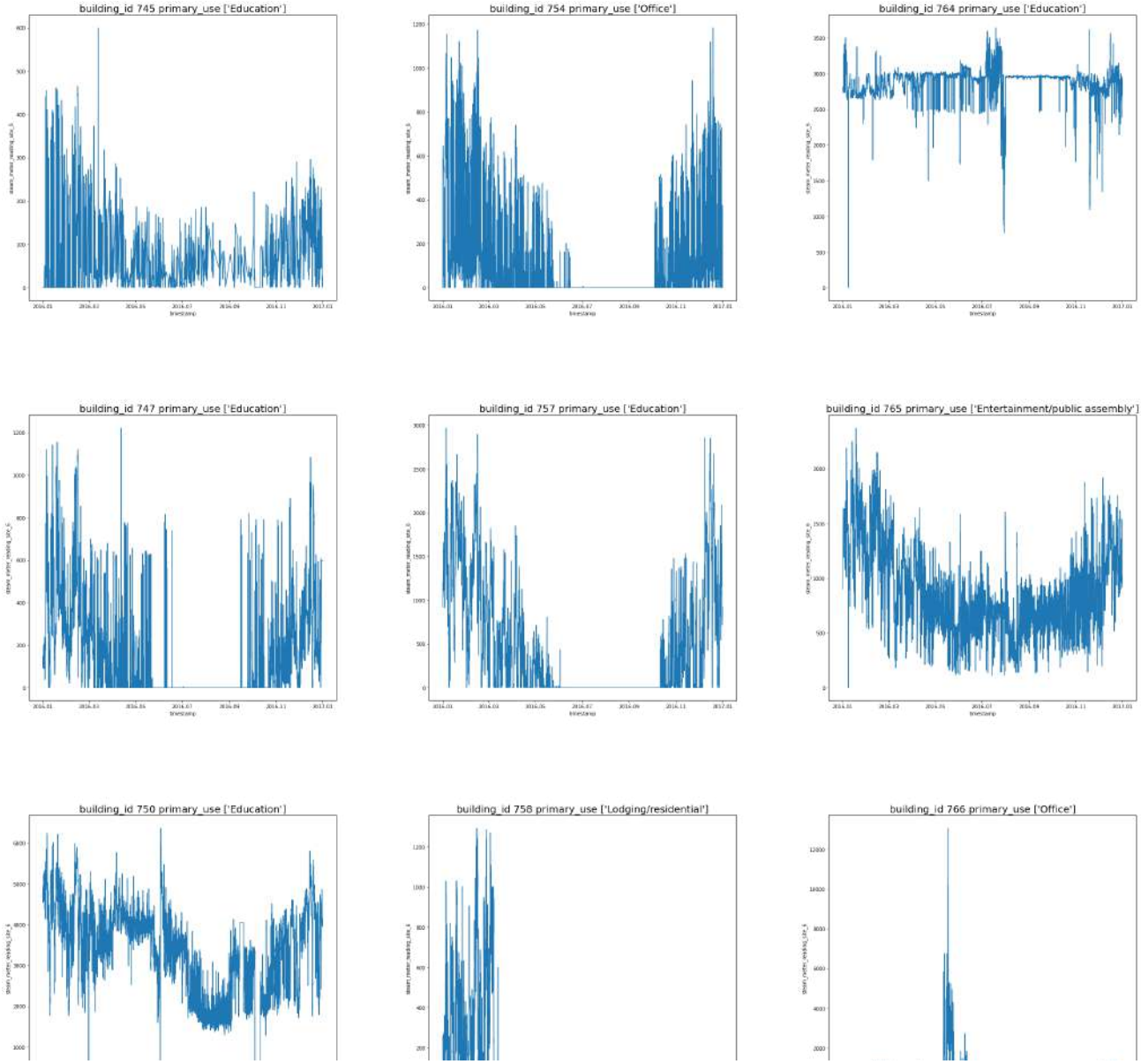


Weather timestamp is not in alignment with the local timestamp with the hourly meter readings as the air temperature peaks around 20:00 pm.

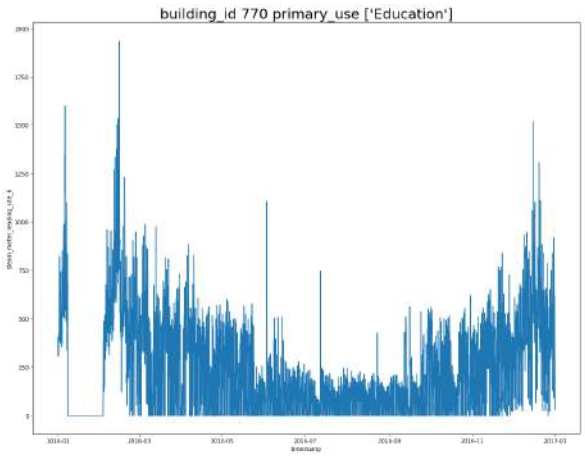
```
df_train_site_6_meter_2['building_id'].nunique()
```

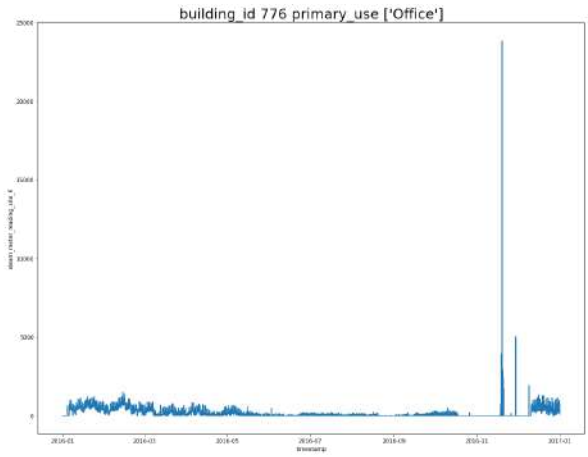
```
23
```

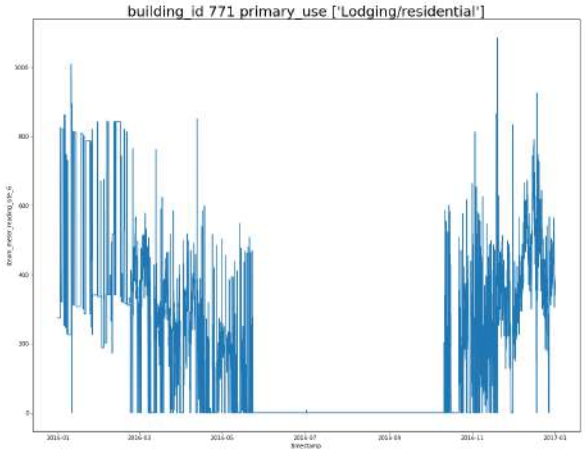
```
fig,ax=plt.subplots(figsize=(40,70),nrows=5,ncols=3)
for i in range(15):
  g=df_train_site_6_meter_2['building_id'].unique()[i]
  axes=ax[i%5][i//5]
  z=df_train_site_6_meter_2.loc[df_train_site_6_meter_2['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('steam_meter_reading_site_6')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```

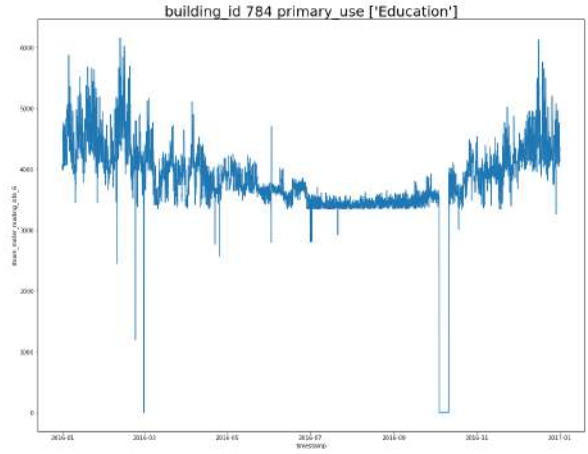


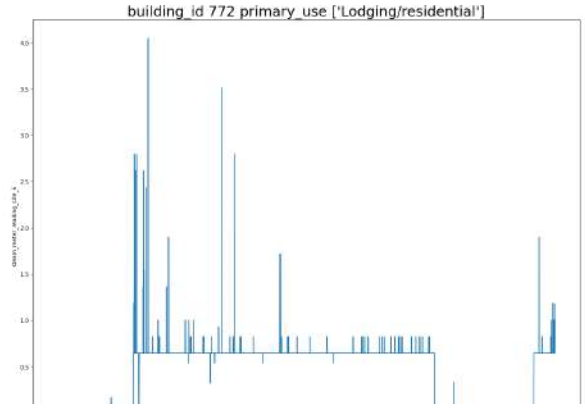

```
fig,ax=plt.subplots(figsize=(40,70),nrows=4,ncols=2)
for i in range(8):
  g=df_train_site_6_meter_2['building_id'].unique()[15:24][i]
  axes=ax[i%4][i//4]
  z=df_train_site_6_meter_2.loc[df_train_site_6_meter_2['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('steam_meter_reading_site_6')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```

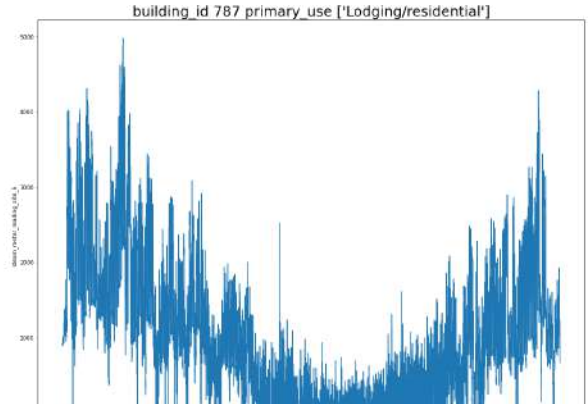












The building at site 6 meter 2 has 23 buildings so I divided that into 15 and 8 buildings for better representation.

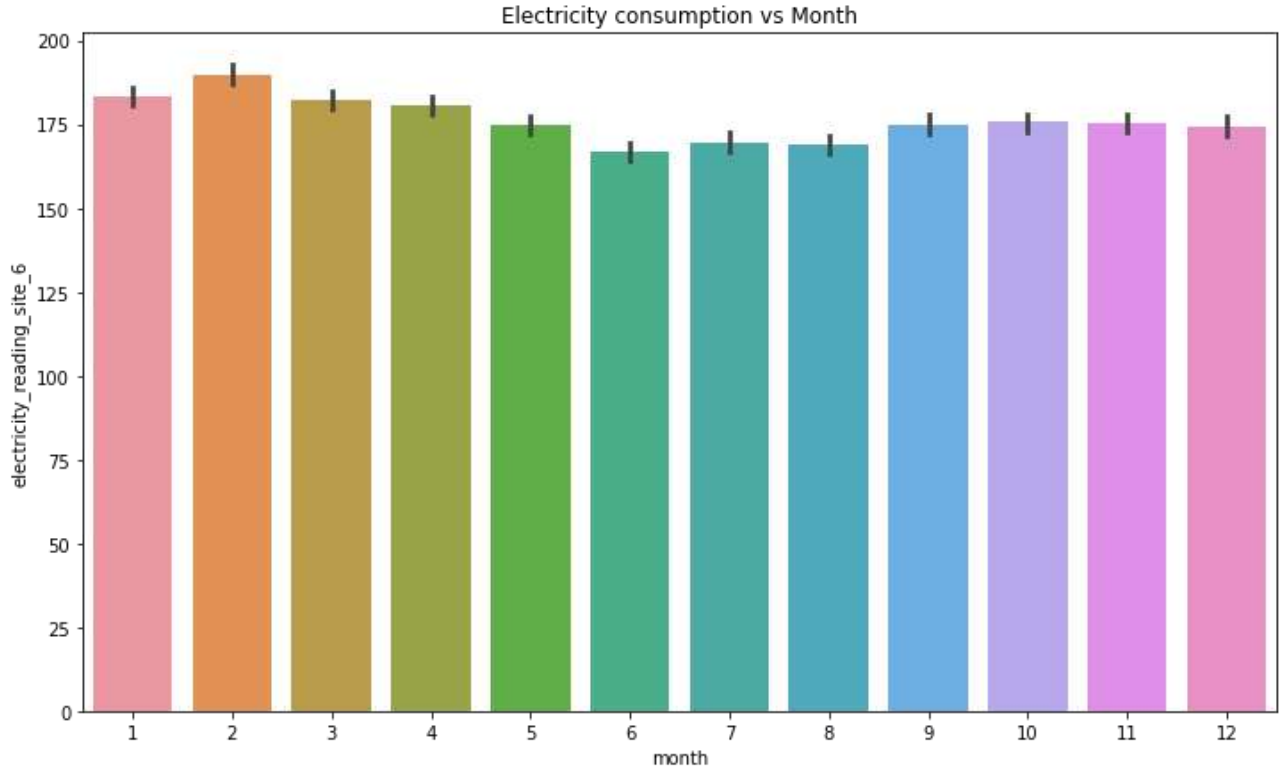
Important Observations

- We can observe from the above plot that many of the buildings are showing unusual spikes and we need to remove them.

```
df_train_site_6_meter_0=df_train_site_6.loc[df_train_site_6['meter']=='electricity']
```

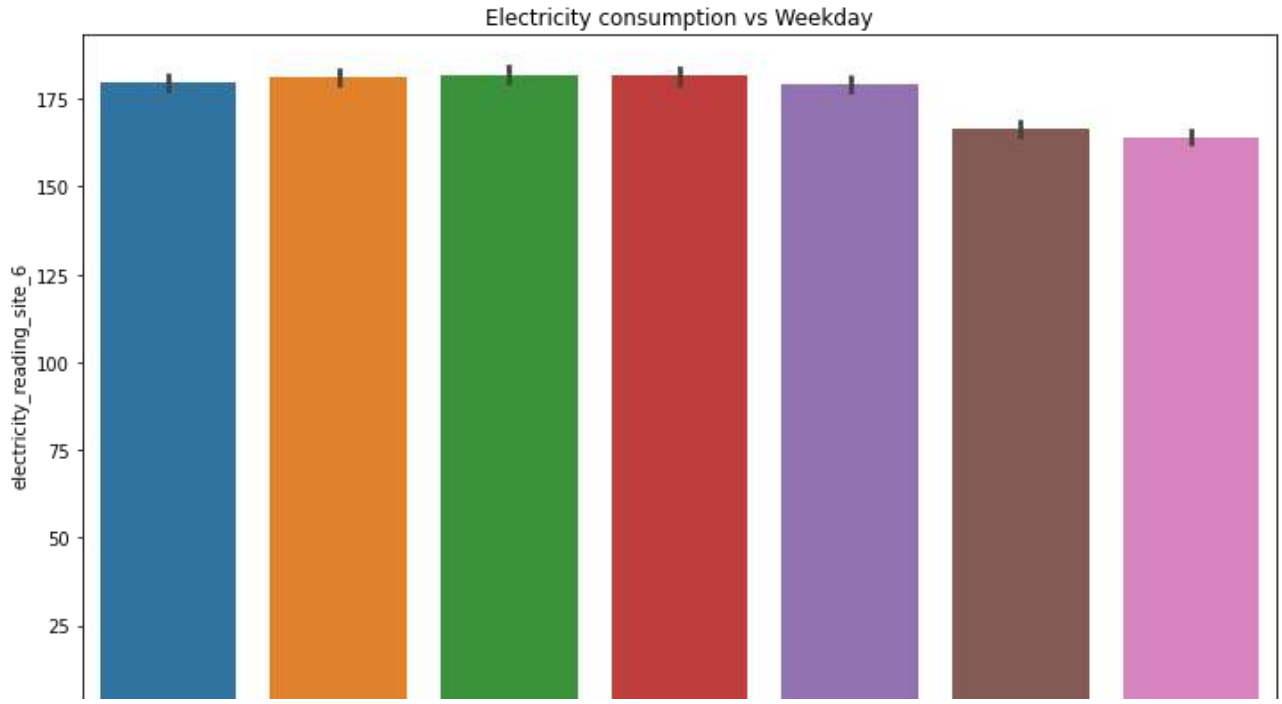
```
df_train_site_6_meter_0['month']=df_train_site_6_meter_0['timestamp'].dt.month
df_train_site_6_meter_0['weekday']=df_train_site_6_meter_0['timestamp'].dt.weekday
df_train_site_6_meter_0['hour']=df_train_site_6_meter_0['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_reading_site_6')
plt.title('Electricity consumption vs Month')
plt.show()
```



The above plot shows that the electricity consumption are varying according to the month.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_6')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



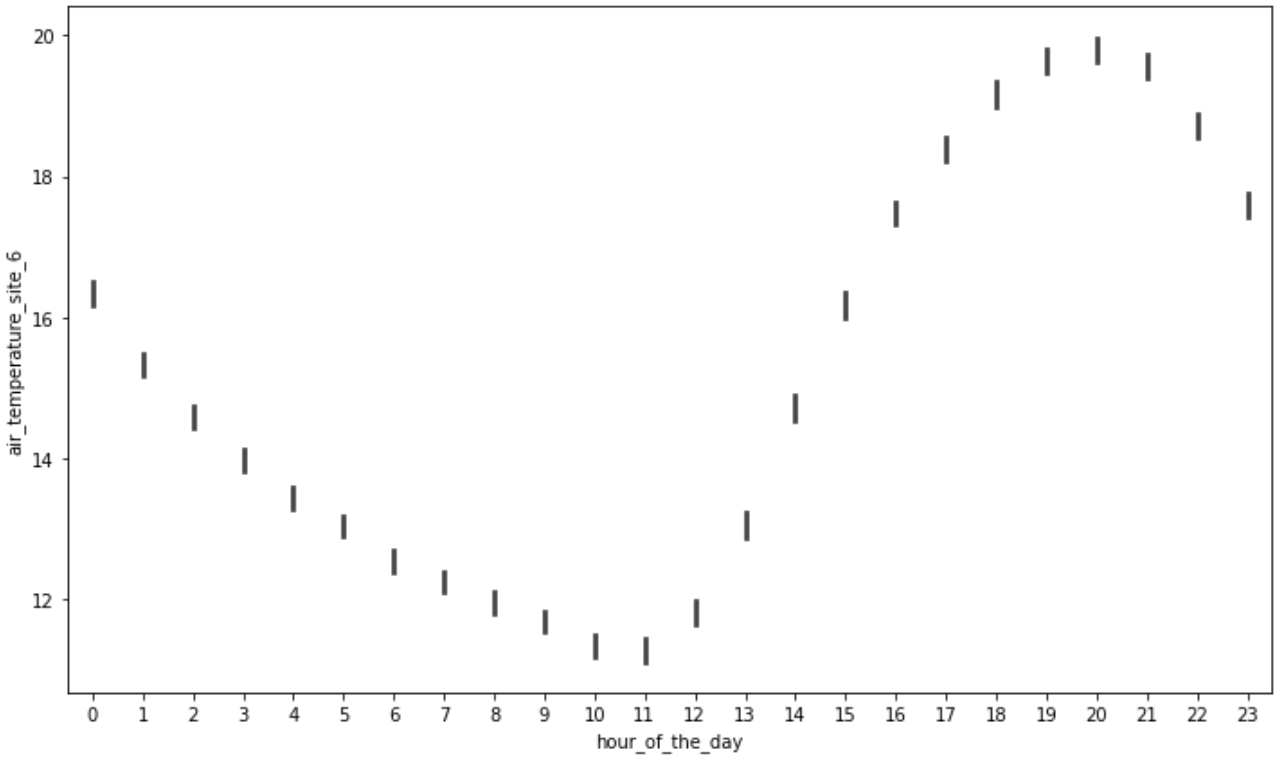
The above plot shoes that electricity consumption is lesser over the weekend as compared to the weekday.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_6')
plt.title('Electricity consumption vs hour')
plt.show()
```

Electricity consumption vs hour

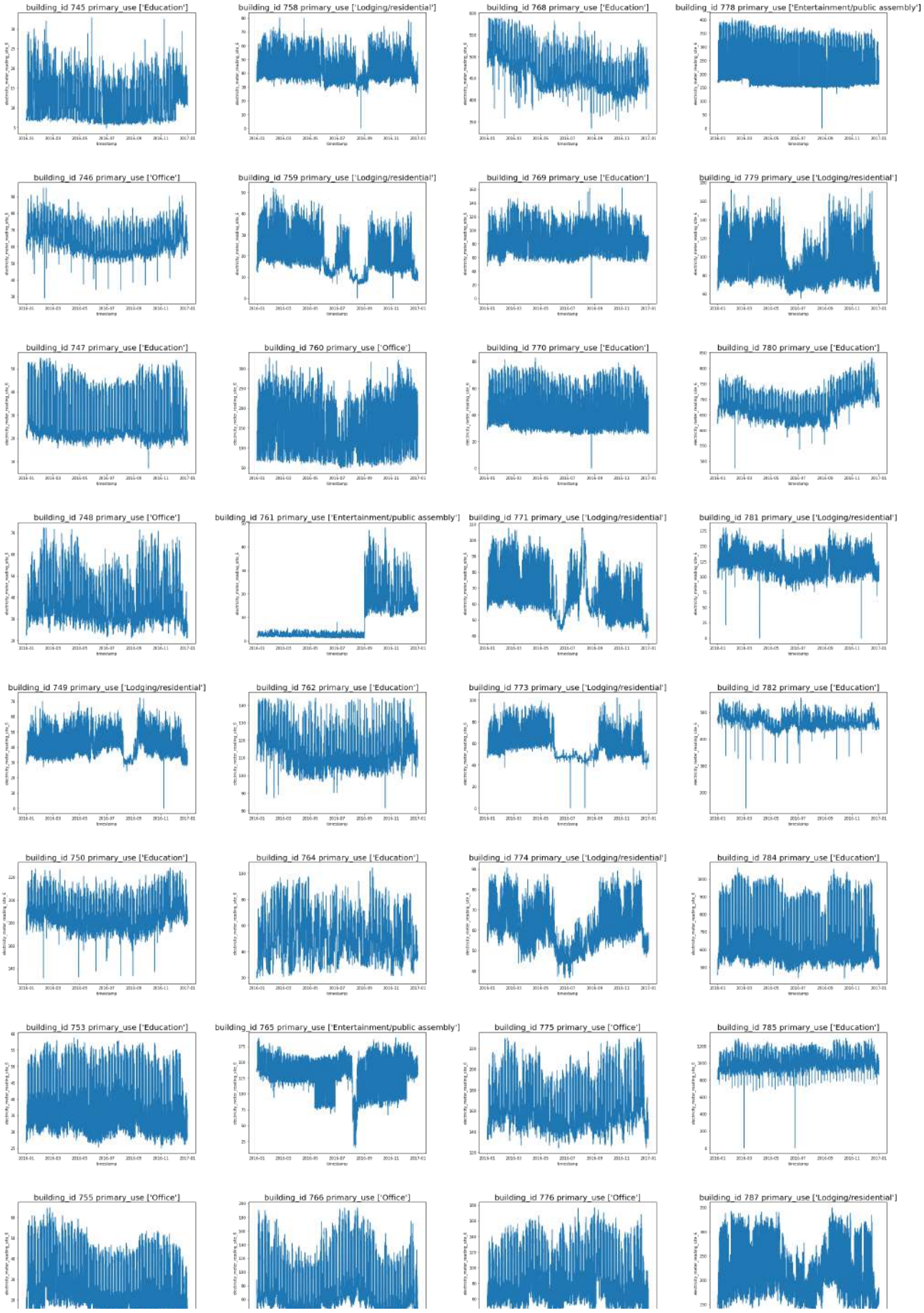
The above plot shows that the electrical consumption is peaking during the afternoon hours.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_6_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_6')
plt.show()
```



The weather timeatmp is not aligned with the local timestamp of the hourly meter readings as the temperature peaks around 20:00 pm.

```
fig,ax=plt.subplots(figsize=(40,70),nrows=9,ncols=4)
for i in range(df_train_site_6_meter_0['building_id'].nunique()):
  g=df_train_site_6_meter_0['building_id'].unique()[i]
  axes=ax[i%9][i//9]
  z=df_train_site_6_meter_0.loc[df_train_site_6_meter_0['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_meter_reading_site_6')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```



Important Observations

- We need to remove anomalous reading from building 749,758,770,773,778,781,785.

```
#Starting analysis for site 7
```

```
df_train_site_7=df_train_merge.loc[df_train_merge['site_id']==7]
```

```
df_train_site_7.isnull().sum()/df_train_site_7.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           0.07
floor_count          0.00
air_temperature      0.02
cloud_coverage       1.00
dew_temperature      0.02
precip_depth_1_hr    0.92
sea_level_pressure   0.02
wind_direction       0.02
wind_speed           0.02
dtype: float64
```

From this we are getting the precentage of null values this information is super necessary as we need the fill the missing values as null values cannot be fed to the model directly.

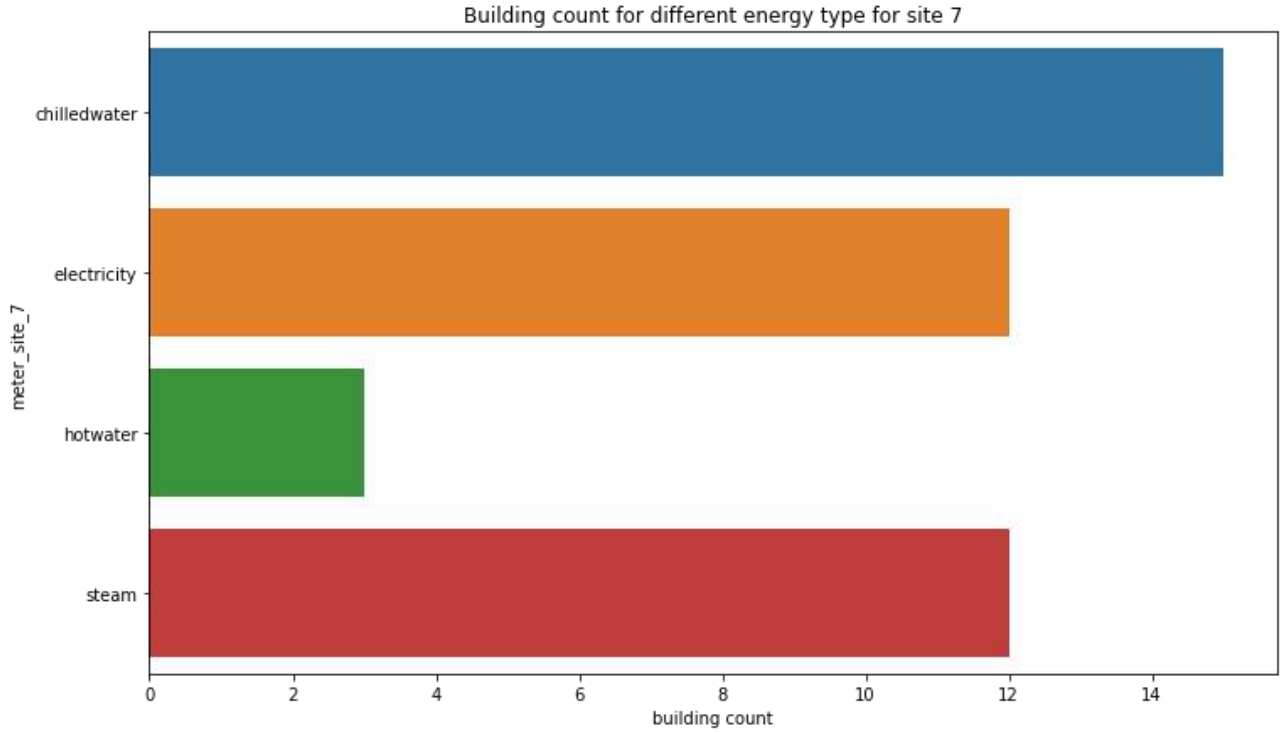
```
df_corr_7=df_train_site_7.corr()
df_corr_7.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.12	nan	0.18	0.62	0.30	-0.01
meter_reading	0.12	1.00	nan	0.36	0.10	0.09	-0.10
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.18	0.36	nan	1.00	0.48	0.51	-0.00
year_built	0.62	0.10	nan	0.48	1.00	0.59	-0.01
floor_count	0.30	0.09	nan	0.51	0.59	1.00	-0.00
air_temperature	-0.01	-0.10	nan	-0.00	-0.01	-0.00	1.00
cloud_coverage	nan	nan	nan	nan	nan	nan	nan
dew_temperature	-0.01	-0.07	nan	-0.00	-0.01	-0.00	0.91
precip_depth_1_hr	-0.00	-0.05	nan	0.00	-0.00	0.00	0.31
sea_level_pressure	0.00	0.01	nan	0.00	0.00	0.00	-0.20
wind_direction	0.00	-0.00	nan	0.00	0.00	-0.00	0.03
wind_speed	-0.00	-0.01	nan	-0.00	-0.00	-0.00	-0.05

From the correlation plot we can see that meter reading is not highly correlated with any of the features.

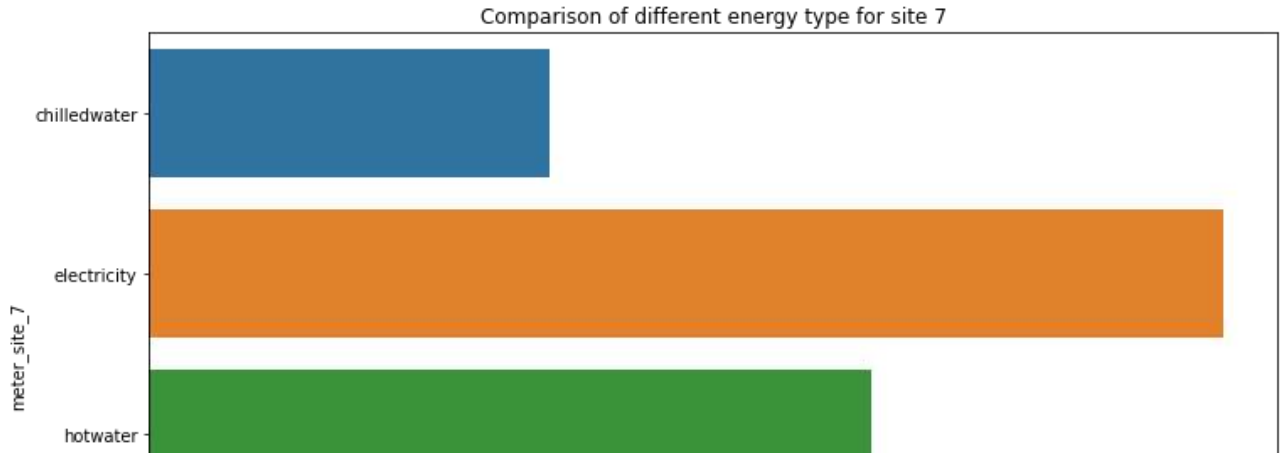
```
z=df_train_site_7.groupby(['meter'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
```

```
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='meter')
plt.xlabel('building count')
plt.ylabel('meter_site_7')
plt.title('Building count for different energy type for site 7')
plt.show()
```



The above plot shows the building count for different energy usage type.

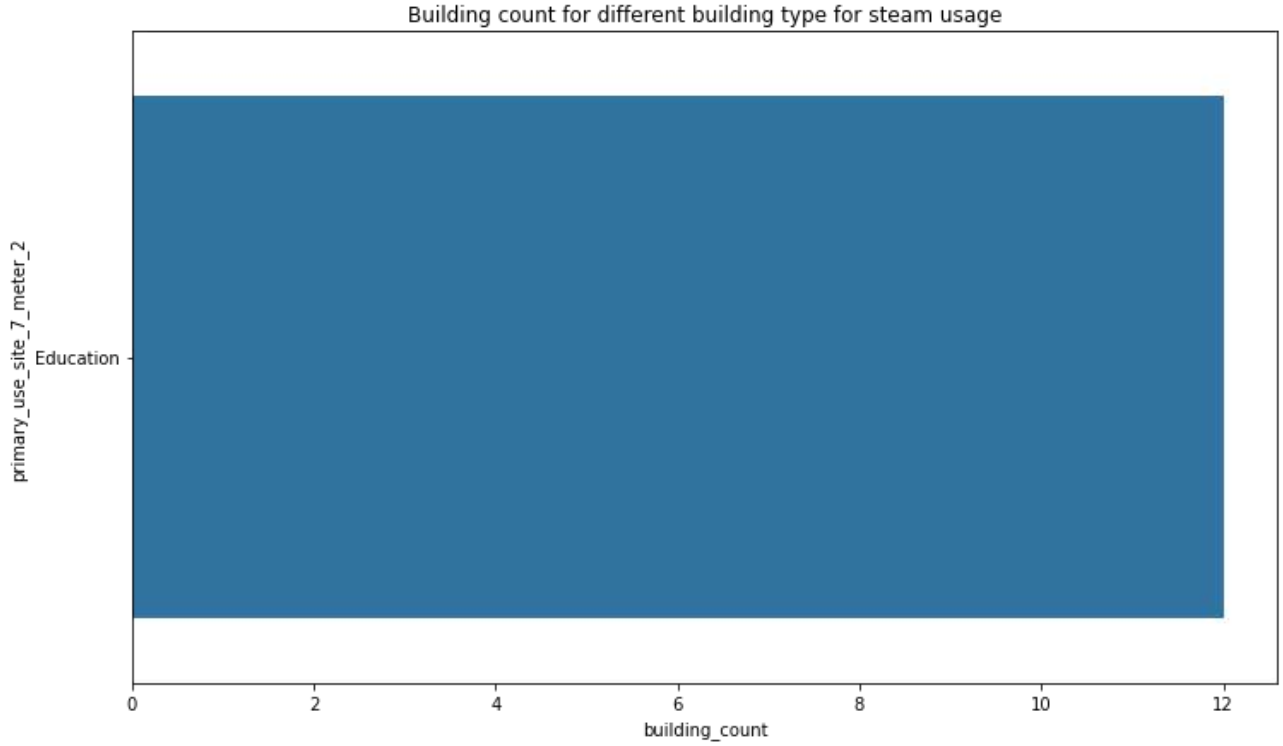
```
z=df_train_site_7.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading')
plt.ylabel('meter_site_7')
plt.title('Comparison of different energy type for site 7')
plt.show()
```



The above plot shows that at site 7 electricity is having the highest energy consumption.

```
df_train_site_7_meter_2=df_train_site_7.loc[df_train_site_7['meter']=='steam']
```

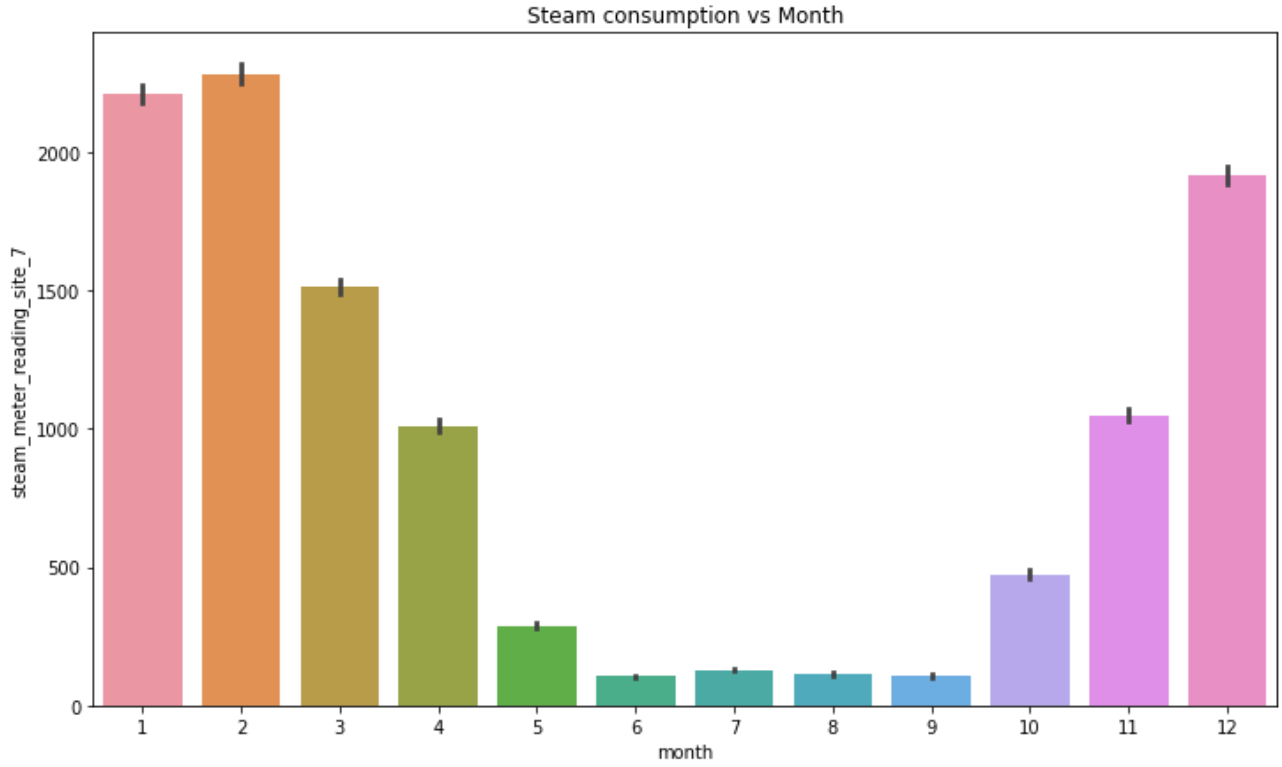
```
z=df_train_site_7_meter_2.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_site_7_meter_2')
plt.title('Building count for different building type for steam usage')
plt.show()
```



The above plot shows that at site 7 educational buildings is the only consumption of steam

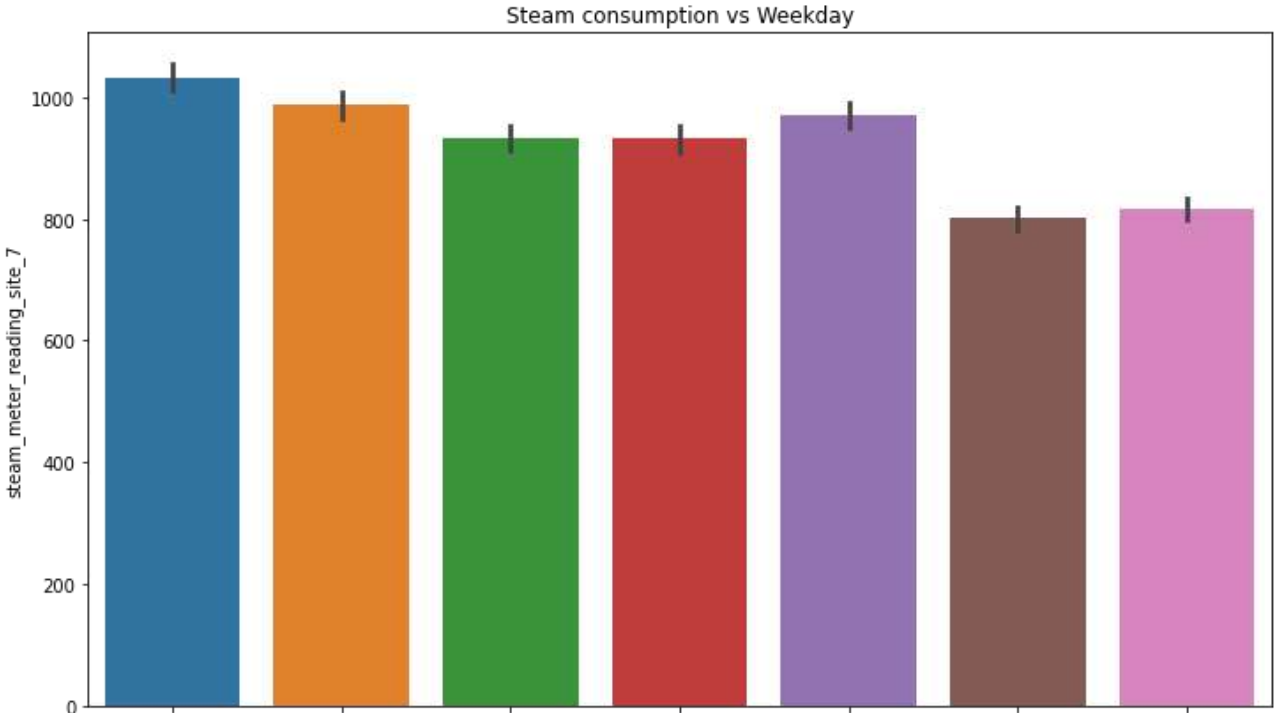
```
df_train_site_7_meter_2['month']=df_train_site_7_meter_2['timestamp'].dt.month
df_train_site_7_meter_2['weekday']=df_train_site_7_meter_2['timestamp'].dt.weekday
df_train_site_7_meter_2['hour']=df_train_site_7_meter_2['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_2
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('steam_meter_reading_site_7')
plt.title('Steam consumption vs Month')
plt.show()
```



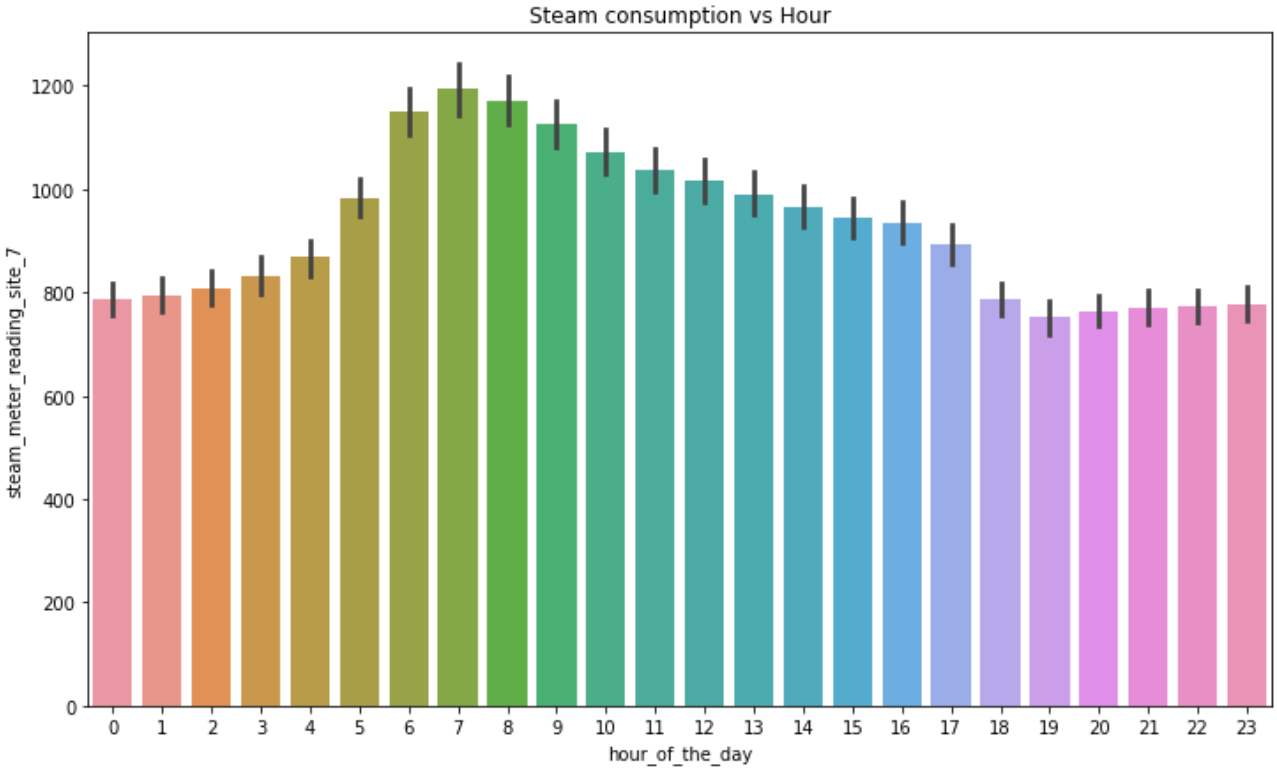
The above plot shows steam consumption is varying over month and it shows higher consumption during the winter month and very low consumption during the summer month.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_2
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('steam_meter_reading_site_7')
plt.title('Steam consumption vs Weekday')
plt.show()
```



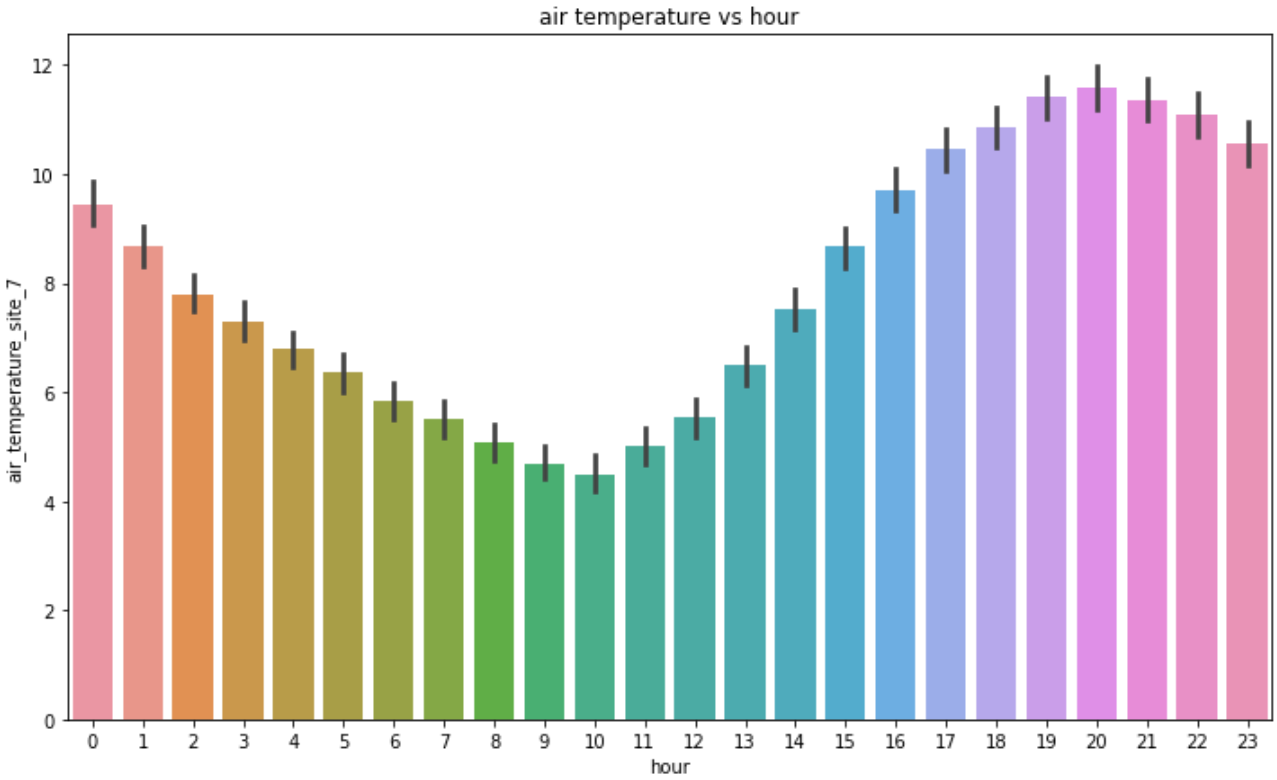
Steam consumption is higher over the weekdays as compared to the weekend.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('steam_meter_reading_site_7')
plt.title('Steam consumption vs Hour')
plt.show()
```



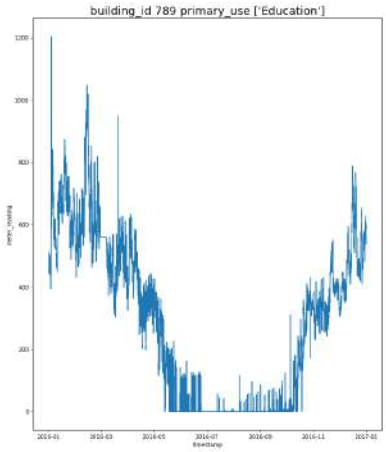
The above plot shows that steam consumption is higher over the morning time as decreases gradually over the time of the day.

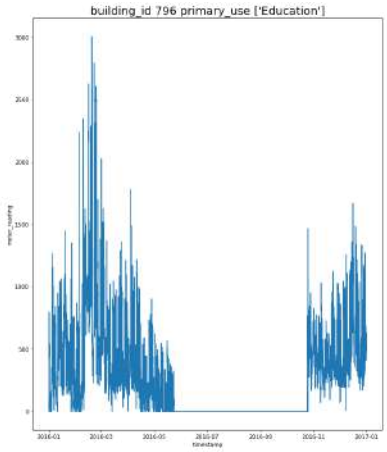
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour')
plt.ylabel('air_temperature_site_7')
plt.title('air temperature vs hour')
plt.show()
```

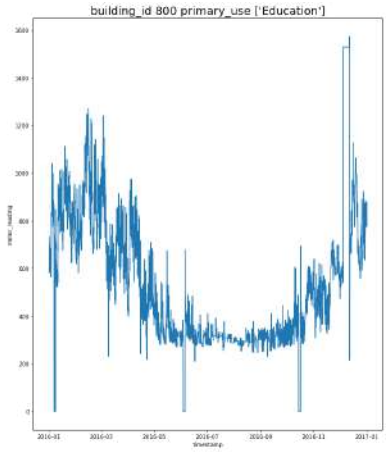


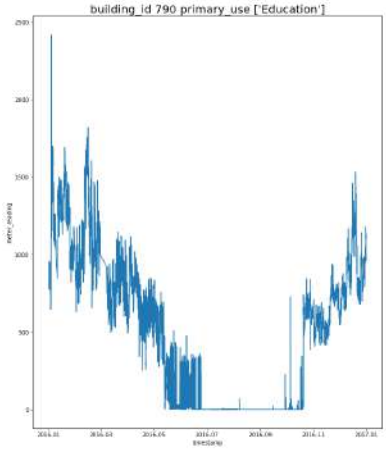
The above plot shows that the weather timesatmp is not in alignment with the local timestamp of the hourly meter reading as the temperature peaks around 20:00 pm.

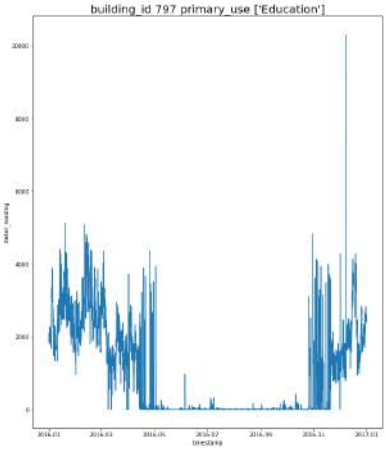
```
fig,ax=plt.subplots(figsize=(40,70),nrows=4,ncols=3)
for i in range(df_train_site_7_meter_2['building_id'].nunique()):
  g=df_train_site_7_meter_2['building_id'].unique()[i]
  axes=ax[i%4][i//4]
  z=df_train_site_7_meter_2.loc[df_train_site_7_meter_2['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('meter_reading')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4,wspace=0.3)
```

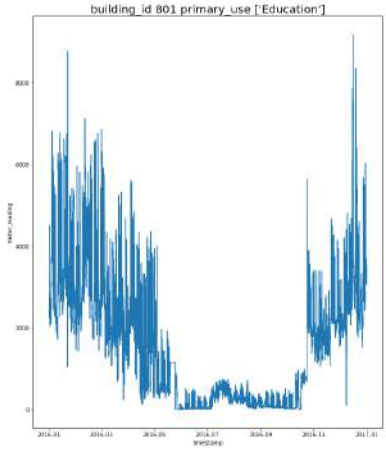


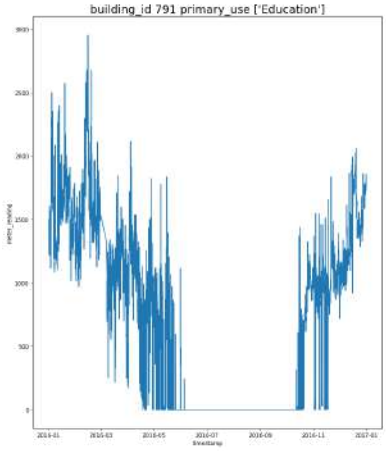


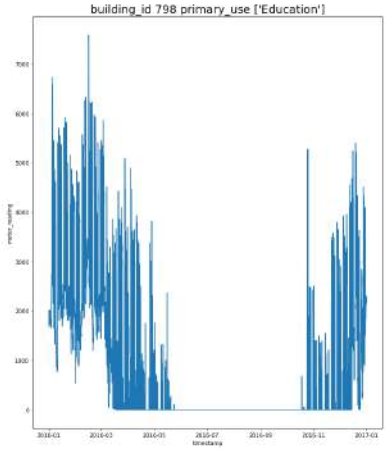


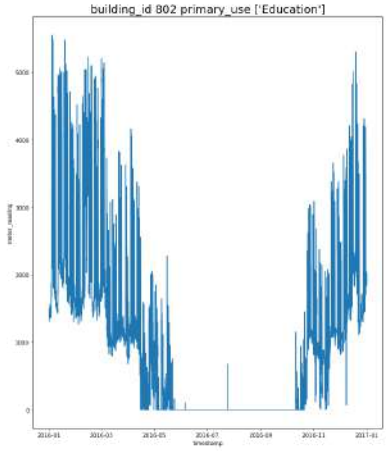


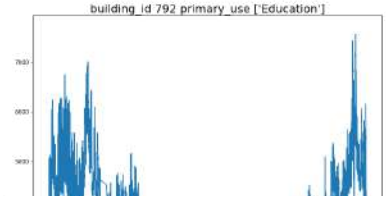


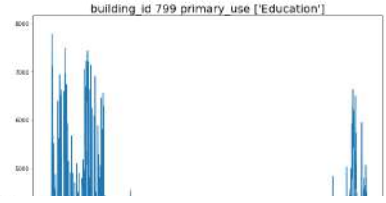


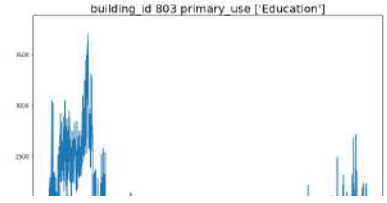








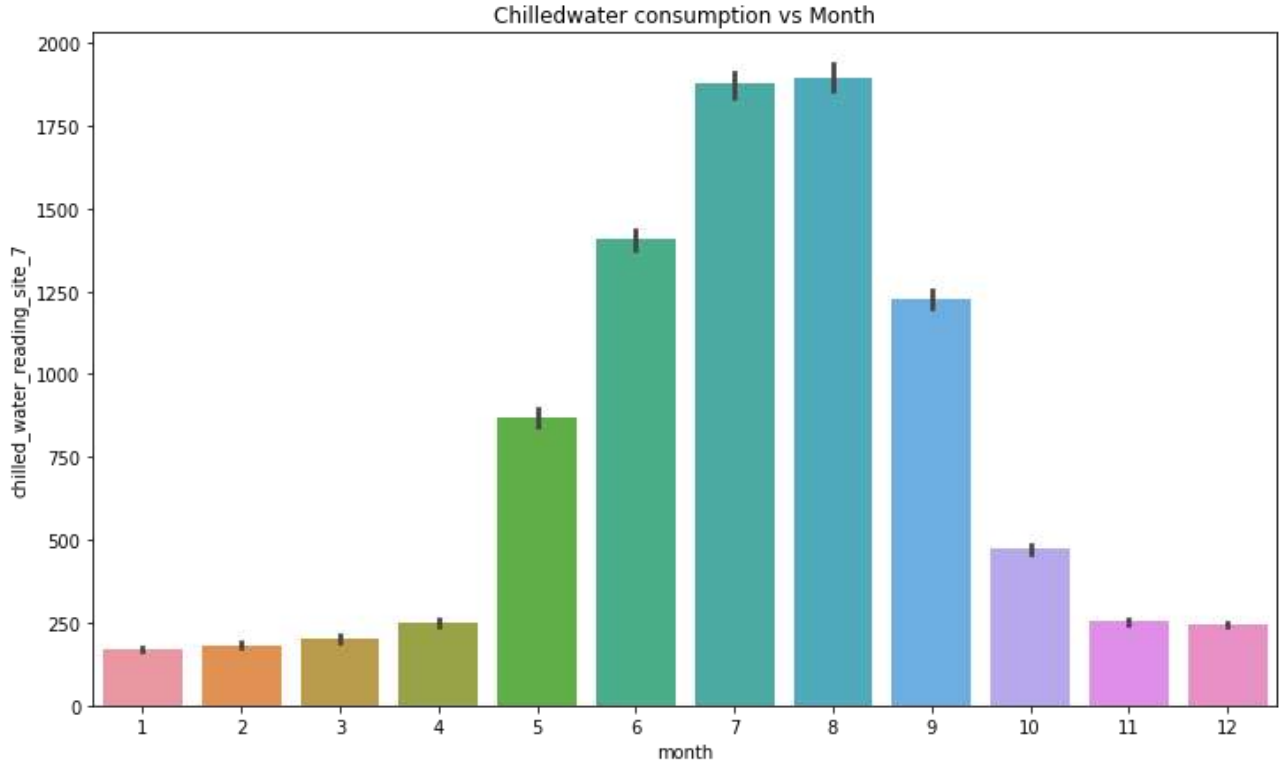




Building 797 shows avery high spike and we need to remove that reading.

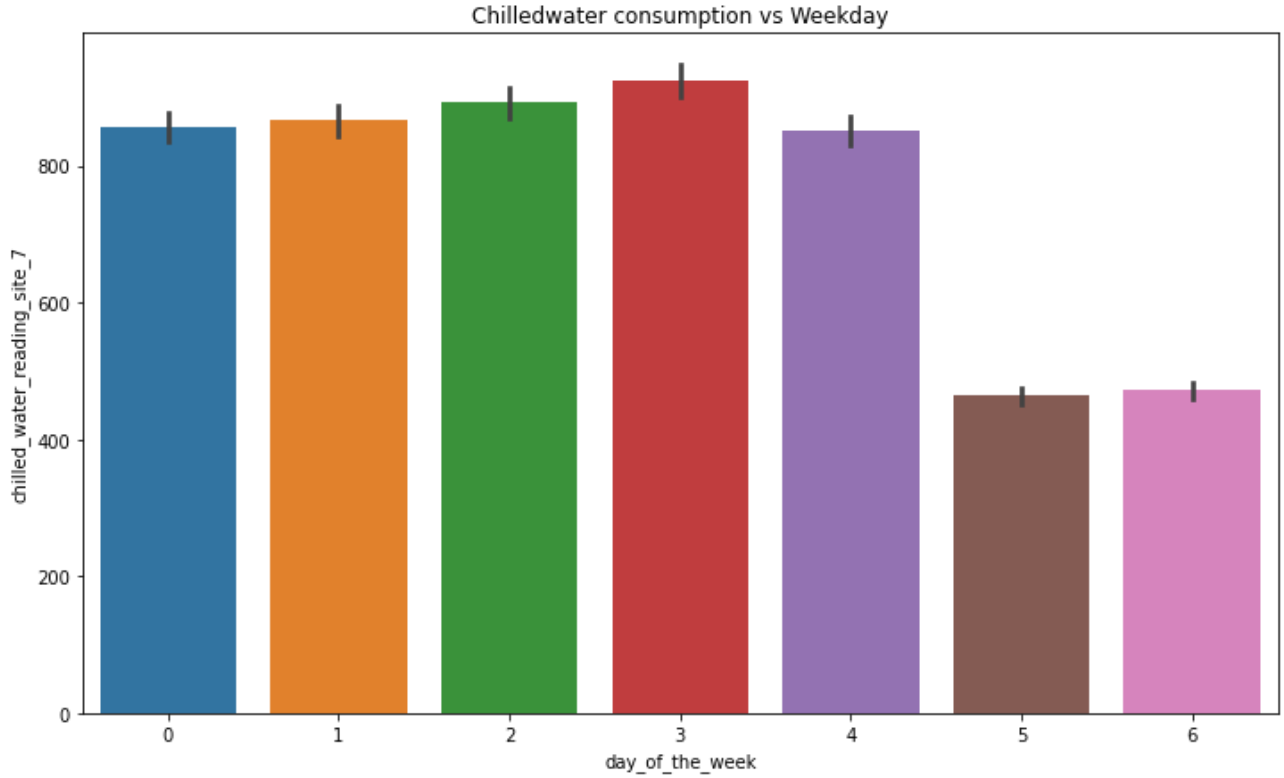
Building 800 803 792 we can remove the zero meter reading which can be observed in the

```
df_train_site_7_meter_1=df_train_site_7.loc[df_train_site_7['meter']=='chilledwater']


df_train_site_7_meter_1['month']=df_train_site_7_meter_1['timestamp'].dt.month
df_train_site_7_meter_1['weekday']=df_train_site_7_meter_1['timestamp'].dt.weekday
df_train_site_7_meter_1['hour']=df_train_site_7_meter_1['timestamp'].dt.hour


fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_1
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilled_water_reading_site_7')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



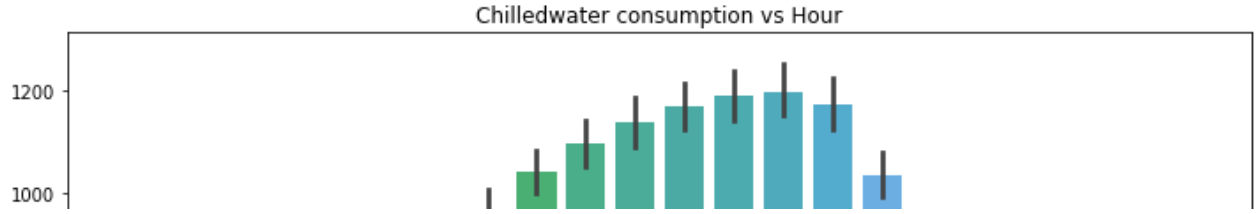
The above plot shows Chilledwater consumption over the month and we are seeing higher consumption during the summer month.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_1
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilled_water_reading_site_7')
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



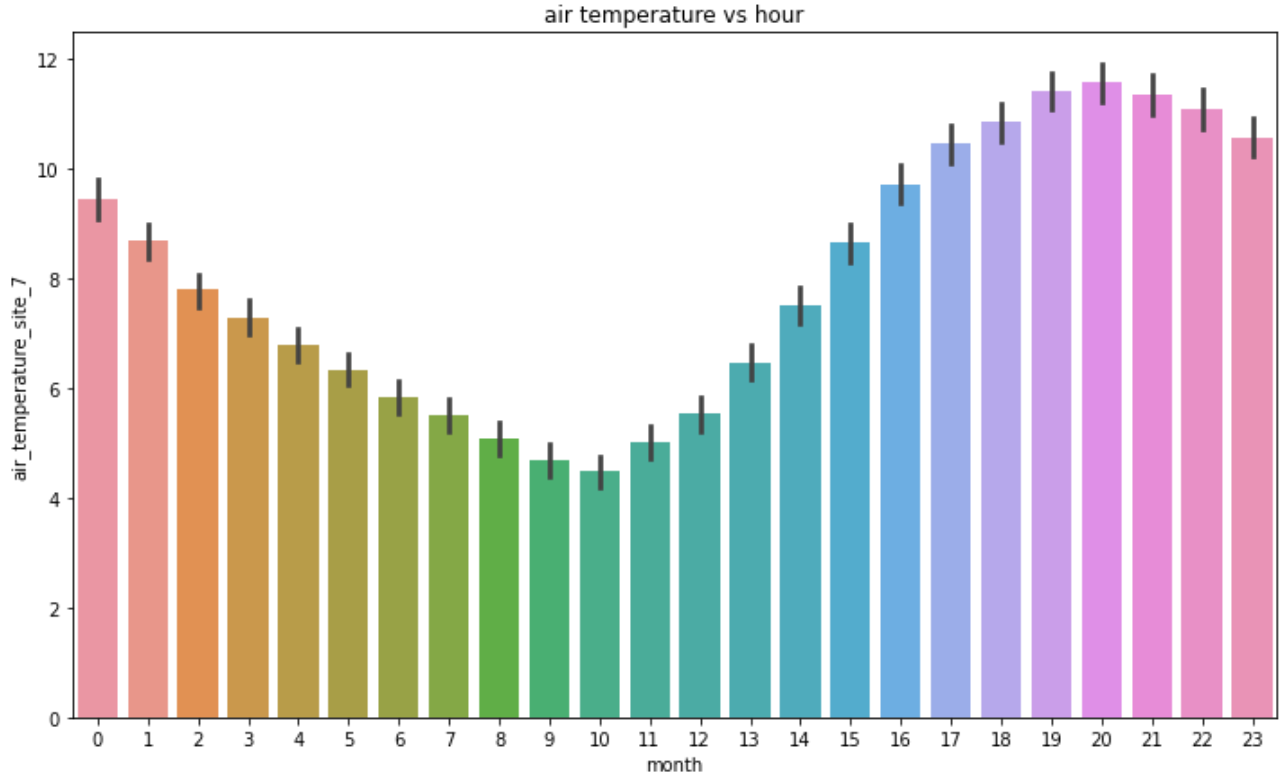
Chilledwater consumption shows a larger consumption over the weekday as compared to the weekend

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilled_water_reading_site_7')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```



Chilledwater consumption starts increasing from the morning time and peaks around 14:00 pm and starts gradually decreasing after that

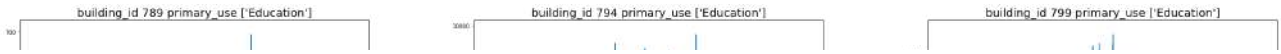
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('month')
plt.ylabel('air_temperature_site_7')
plt.title('air temperature vs hour')
plt.show()
```



The above plot shows that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings as it peaks around 20:00 pm

```
fig,ax=plt.subplots(figsize=(40,70),nrows=5,ncols=3)
for i in range(df_train_site_7_meter_1['building_id'].nunique()):
  g=df_train_site_7_meter_1['building_id'].unique()[i]
  axes=ax[i%5][i//5]
  z=df_train_site_7_meter_1.loc[df_train_site_7_meter_1['building_id']==g]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
```

```
    axes.set_ylabel('chilledwater_meter_reading_site_7')
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
    plt.subplots_adjust(hspace=0.4,wspace=0.3)
```



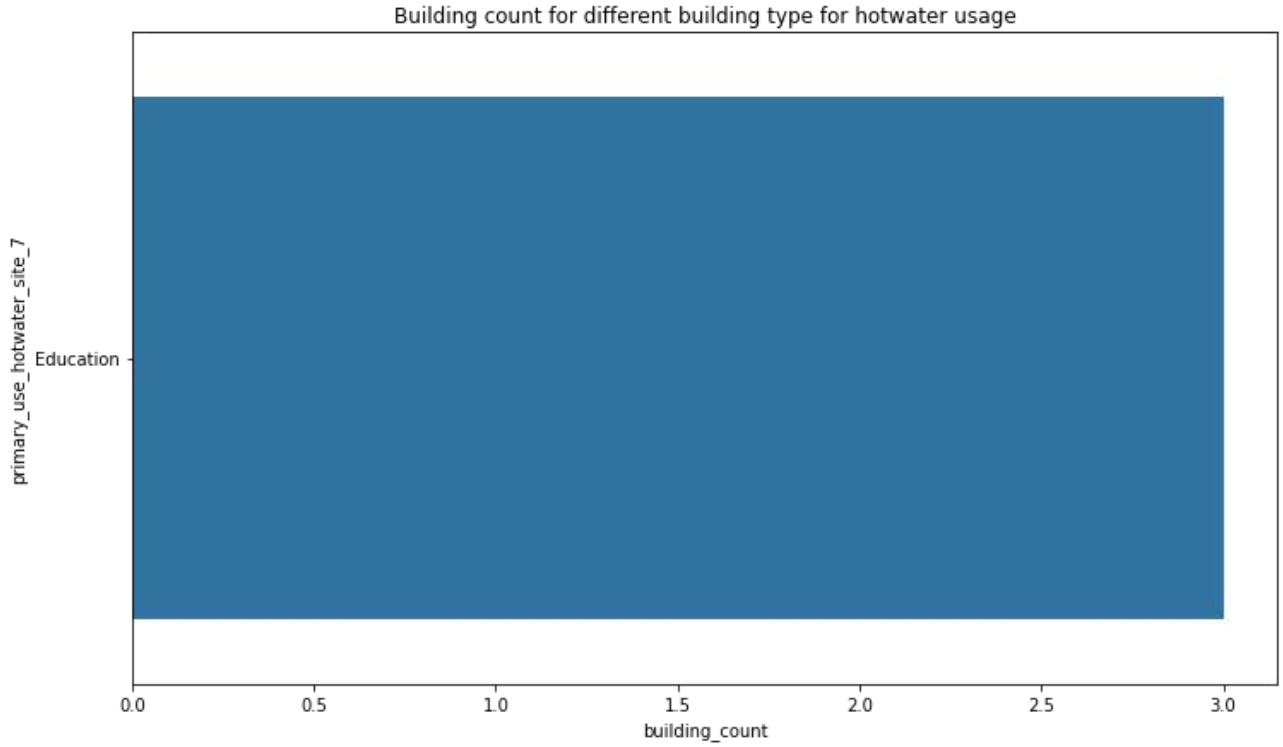
Important Observations

- Building 799 and 800 are having a peaked chilledwater consumption after the 11th month which we need to remove them.
- Building 789 790 792 we can remove the constant zero reading which starts coming in between the months for sometime.


```
df_train_site_7_meter_3=df_train_site_7.loc[df_train_site_7['meter']=='hotwater']
```



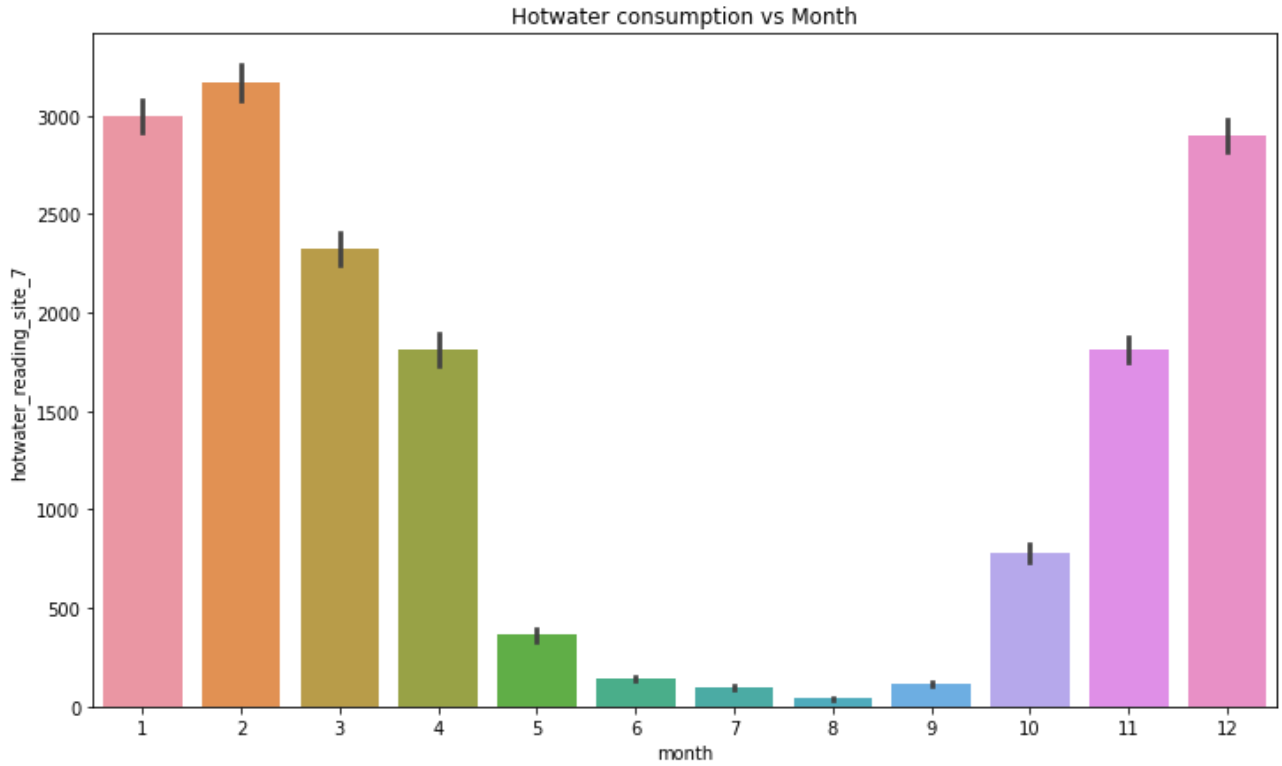
```
z=df_train_site_7_meter_3.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_hotwater_site_7')
plt.title('Building count for different building type for hotwater usage')
plt.show()
```



For hotwater usage at site 7 we can see that only educational building is present.

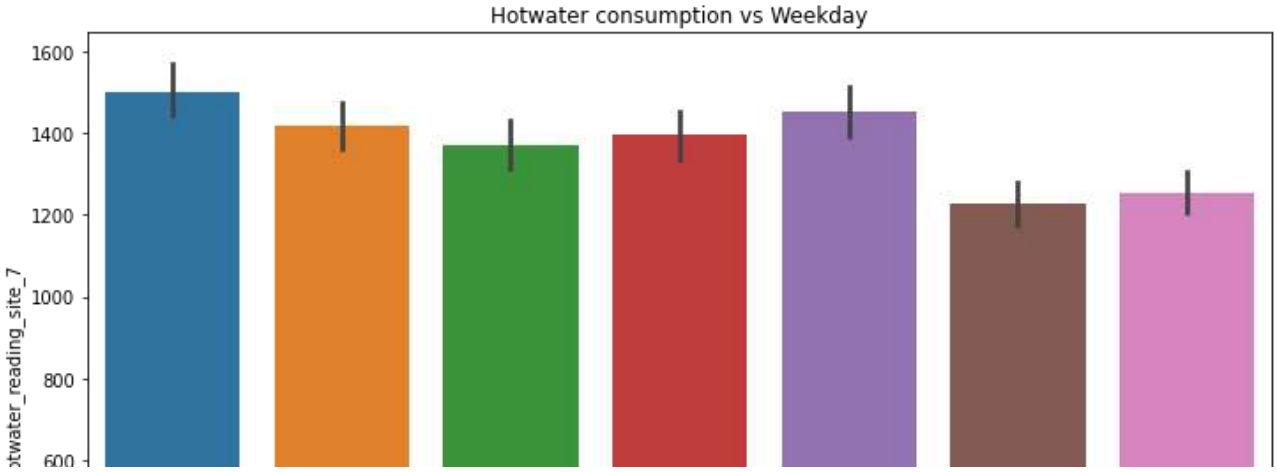
```
df_train_site_7_meter_3['month']=df_train_site_7_meter_3['timestamp'].dt.month
df_train_site_7_meter_3['weekday']=df_train_site_7_meter_3['timestamp'].dt.weekday
df_train_site_7_meter_3['hour']=df_train_site_7_meter_3['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_3
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('hotwater_reading_site_7')
plt.title('Hotwater consumption vs Month')
plt.show()
```



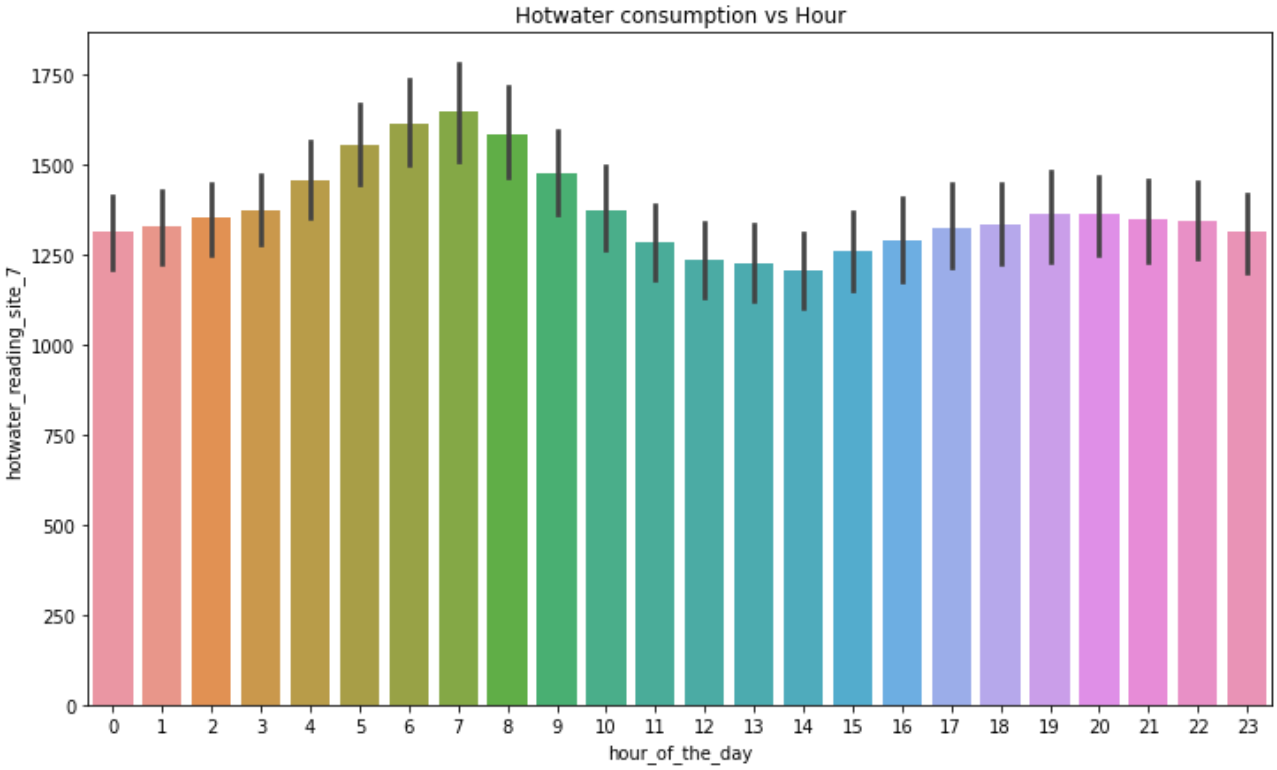
This above plot shows the usage of hotwater over the month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_3
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('hotwater_reading_site_7')
plt.title('Hotwater consumption vs Weekday')
plt.show()
```



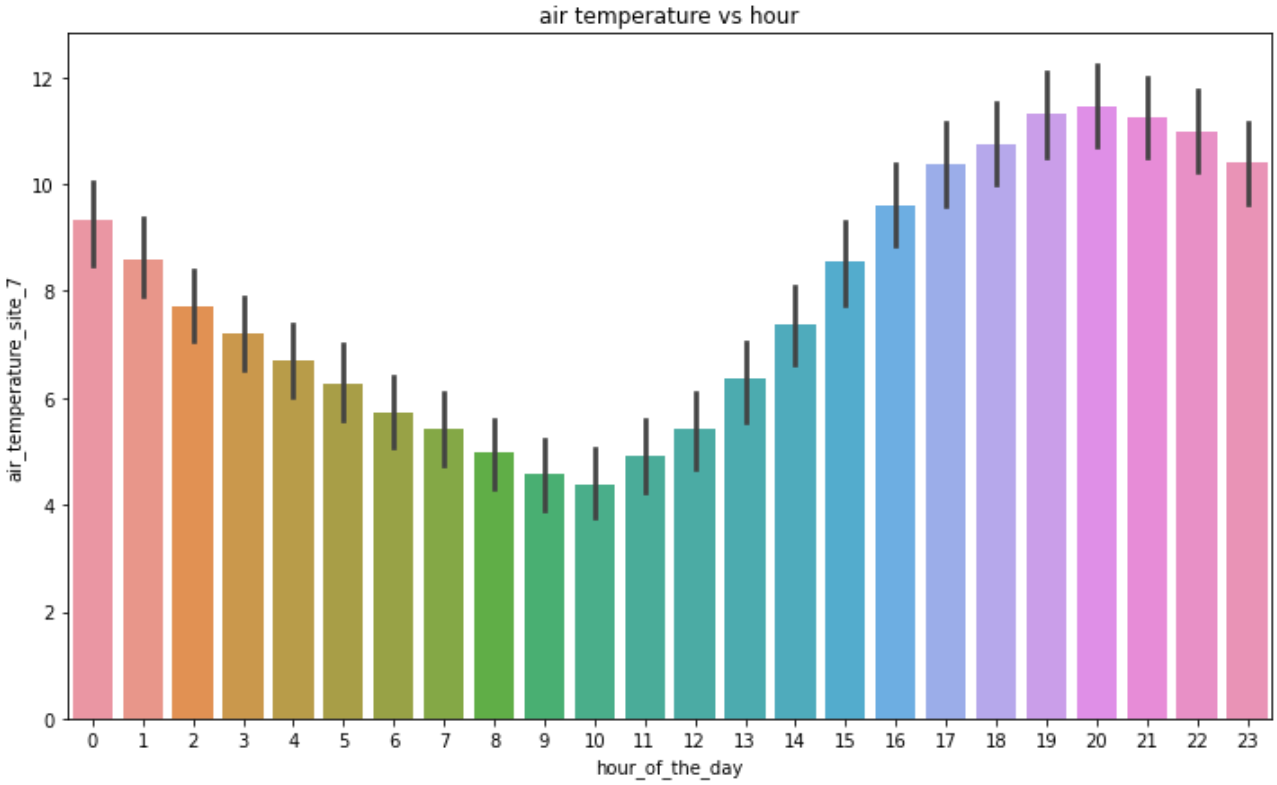
The above plot shows the consumption of hotwater over the weekdays.We can see that consumption is lesser over the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('hotwater_reading_site_7')
plt.title('Hotwater consumption vs Hour')
plt.show()
```



The above plot shows that the consumption of hotwater is higher over the morning hours decreases gradually and then again increases during the night time

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_7')
plt.title('air temperature vs hour')
plt.show()
```

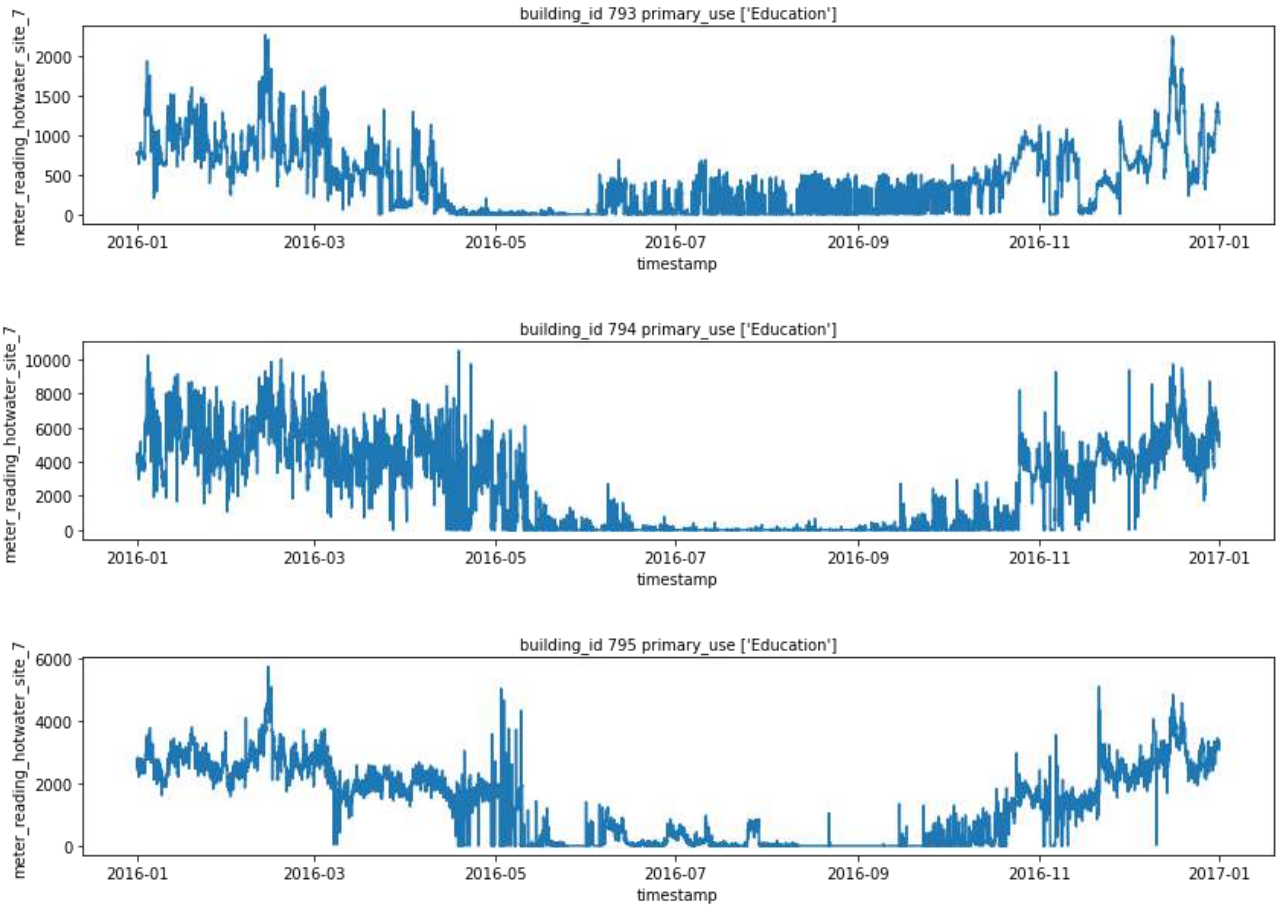


The above plot shows that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings as the temperature peaks around 20:00 pm

```
df_train_site_7_meter_3['building_id'].unique()
```

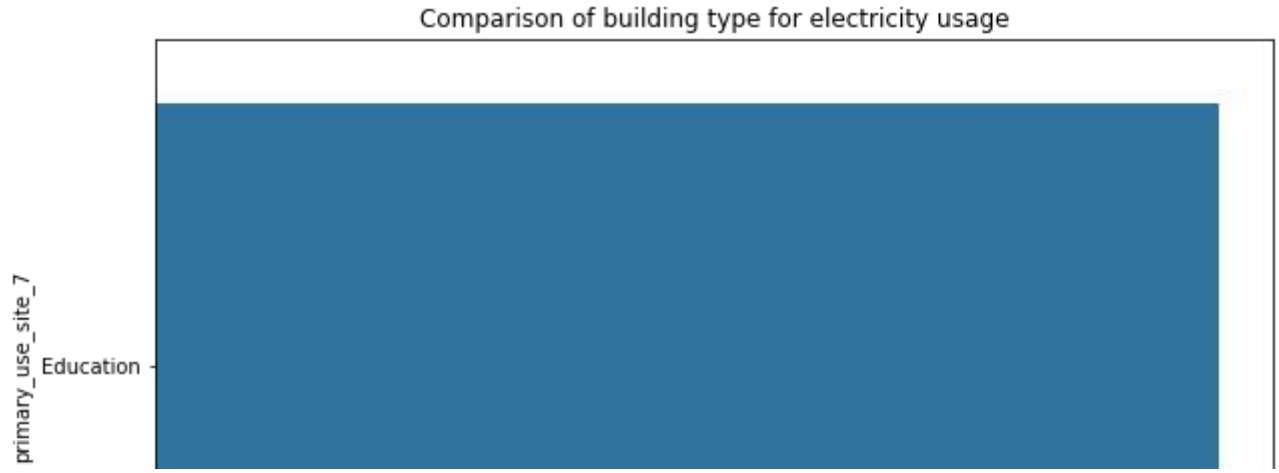
```
array([793, 794, 795], dtype=int16)
```

```
fig,axs=plt.subplots(figsize=(14,10),nrows=3,ncols=1,squeeze=False)
for i in range(df_train_site_7_meter_3['building_id'].nunique()):
   g=df_train_site_7_meter_3['building_id'].unique()[i]
   axes=axs[i%3][i//3]
   z=df_train_site_7_meter_3.loc[df_train_site_7_meter_3['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('meter_reading_hotwater_site_7')
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.6,wspace=0.3)
```



From the above plot for hotwater usage at site 7 we can see that none of the 3 buildings have are showing abnormal usage.

```
df_train_site_7_meter_0=df_train_site_7.loc[df_train_site_7['meter']=='electricity']


z=df_train_site_7_meter_0.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_reading')
plt.ylabel('primary_use_site_7')
plt.title('Comparison of building type for electricity usage')
plt.show()
```

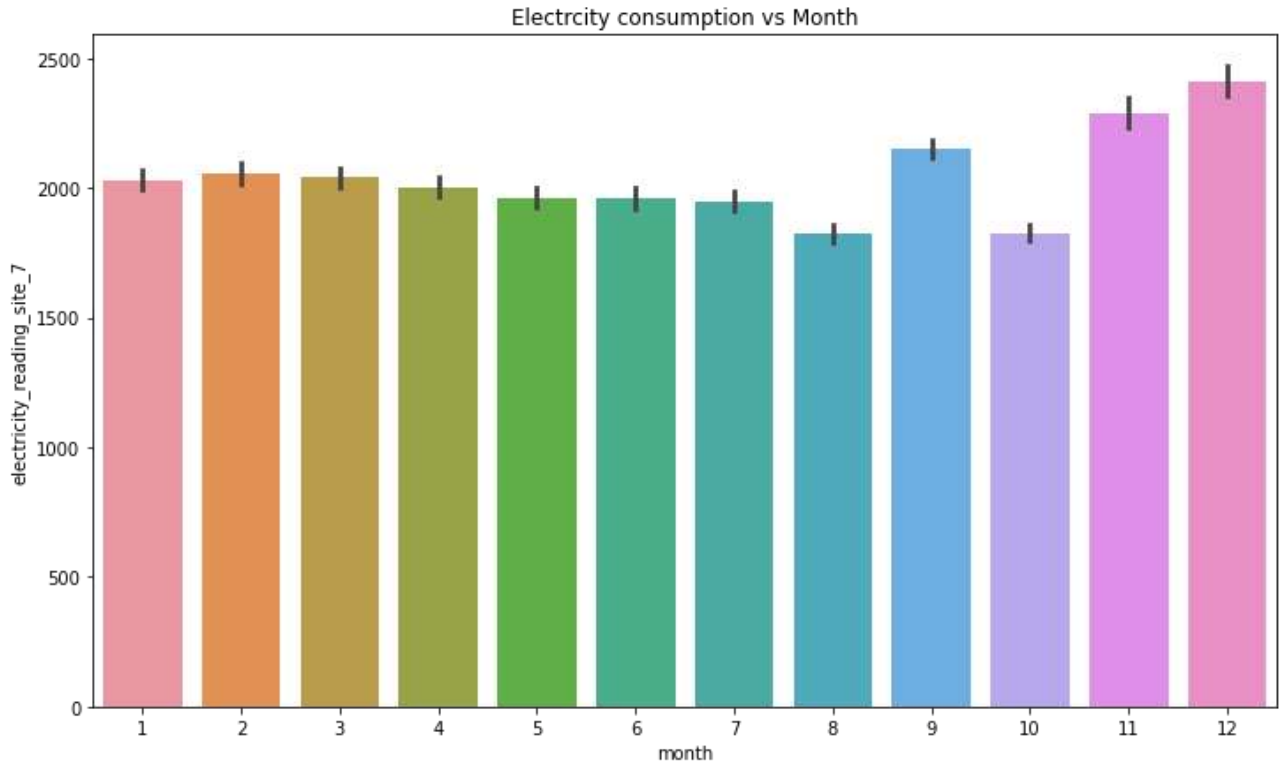


From the above plot we can see that at site 7 only educational building is using electricity

```
df_train_site_7_meter_0['month']=df_train_site_7_meter_0['timestamp'].dt.month
df_train_site_7_meter_0['weekday']=df_train_site_7_meter_0['timestamp'].dt.weekday
df_train_site_7_meter_0['hour']=df_train_site_7_meter_0['timestamp'].dt.hour
```

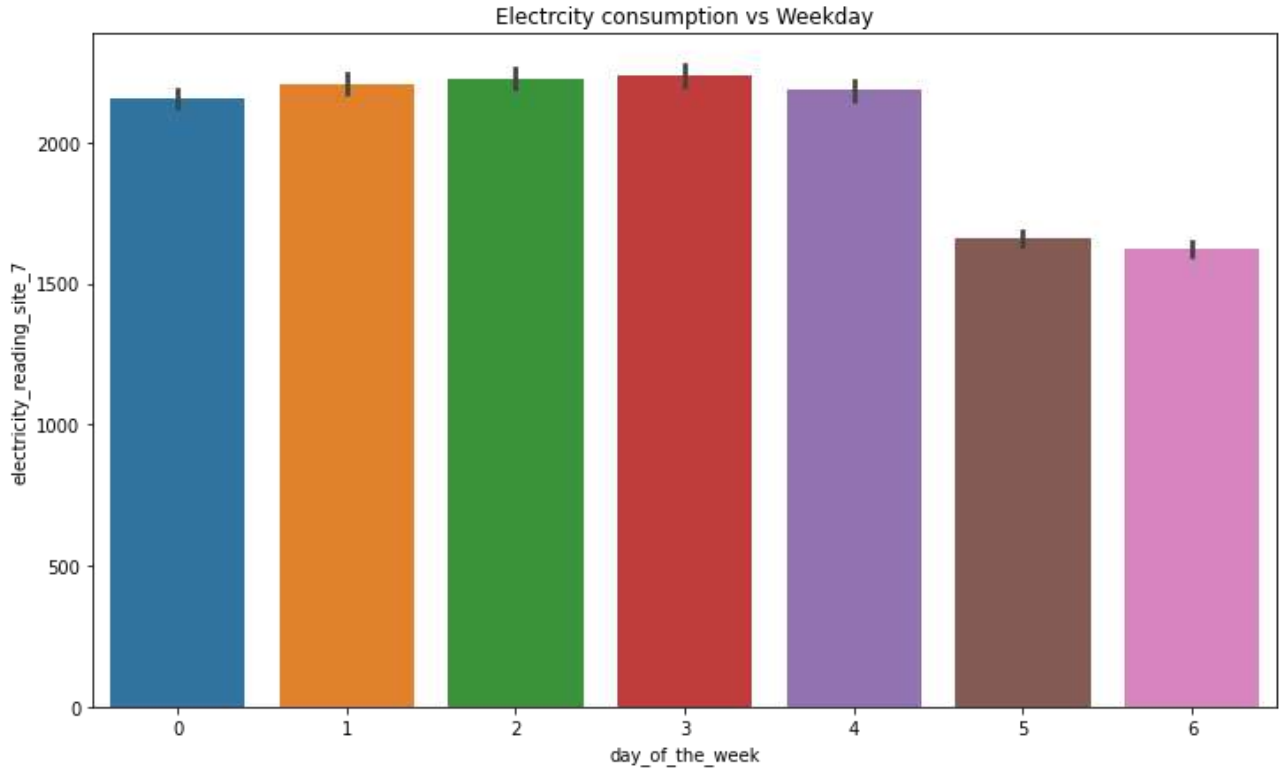
mean_electricity_reading

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_reading_site_7')
plt.title('Electrcity consumption vs Month')
plt.show()
```



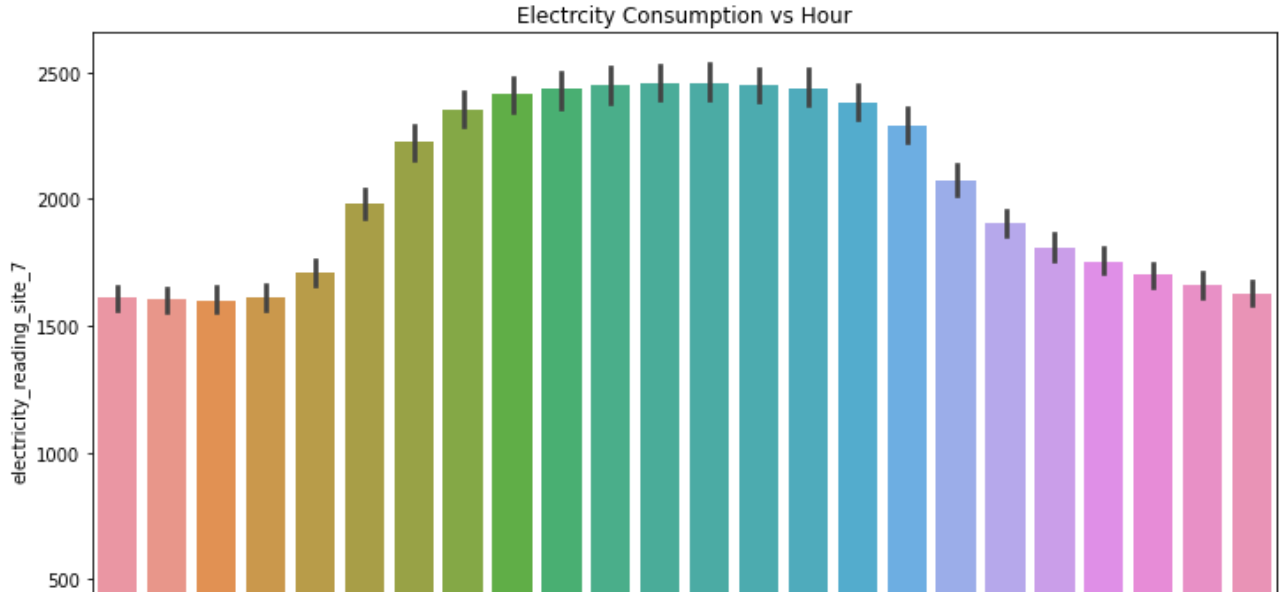
The above plot shows that the electricity consumption varies over the month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_7')
plt.title('Electrcity consumption vs Weekday')
plt.show()
```



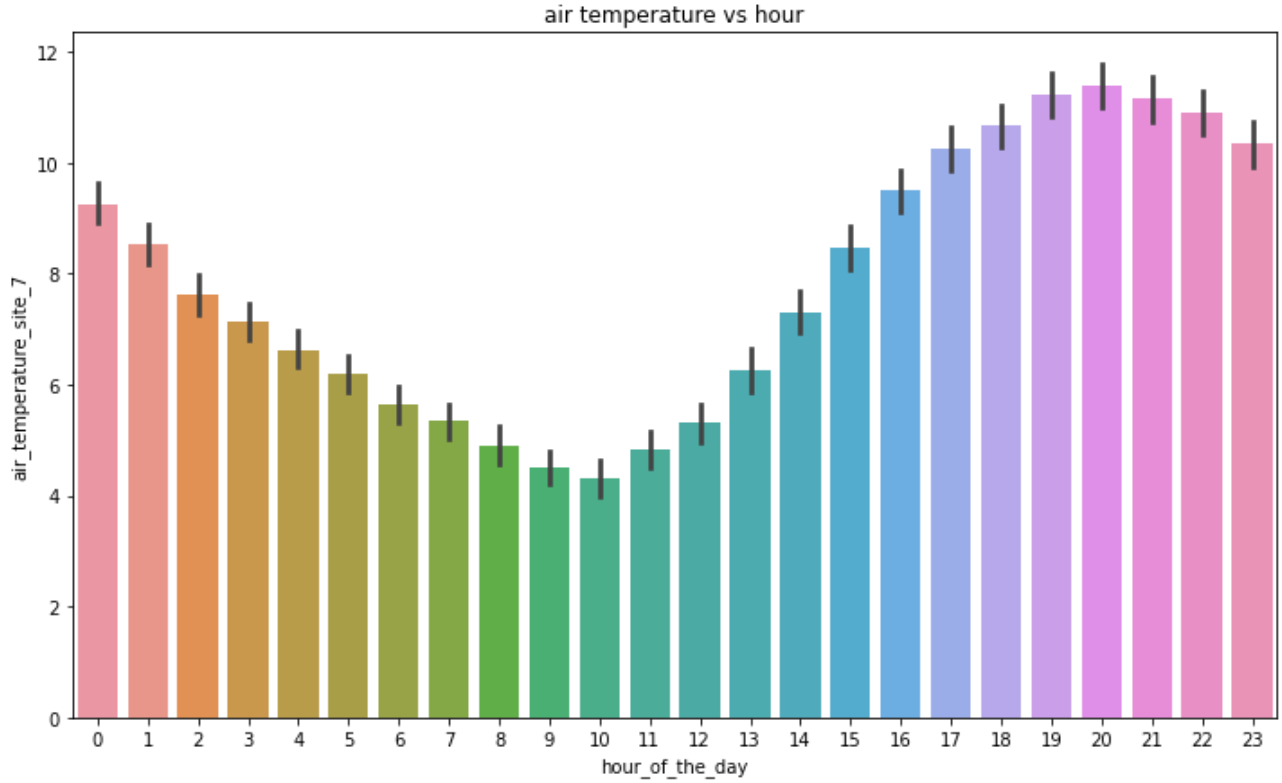
The above plot shows that the electricity consumption is more for the weekday as compared to the weekend

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_7')
plt.title('Electrcity Consumption vs Hour')
plt.show()
```



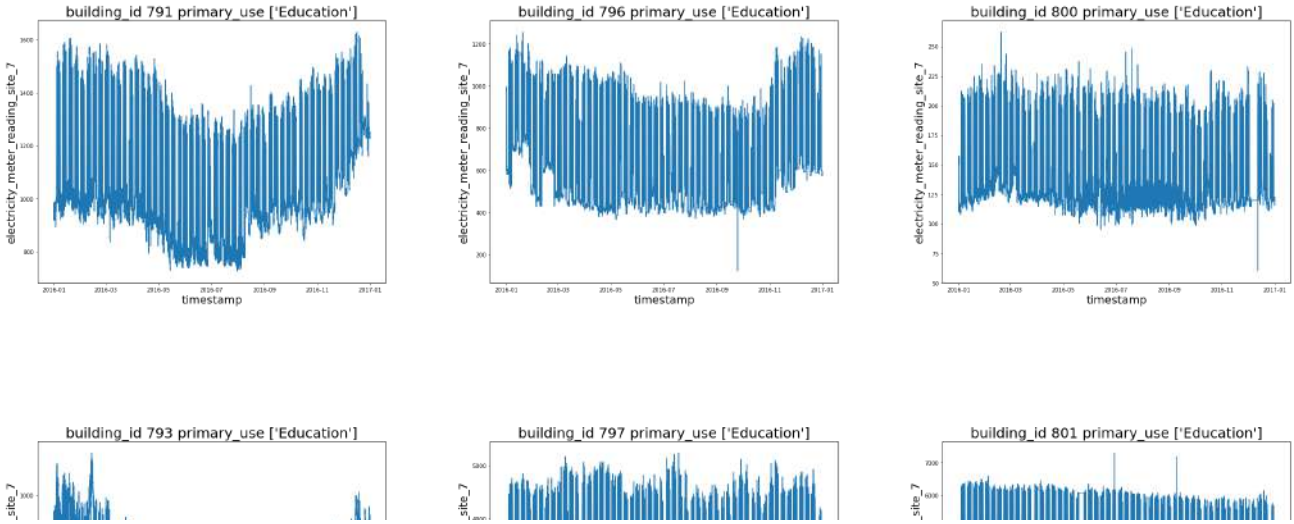
From the above plot we can see that the electricity consumption starts increasing from the morning hour and it becomes constant during the daytime and then starts decreasing after 15:00 pm

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_7_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_7')
plt.title('air temperature vs hour')
plt.show()
```



From the above plot we can see that the weather timestamp was not in alignment with the local timestamp of the hourly meter readings as the temperature peaks around 20:00 pm

```
fig,axs=plt.subplots(figsize=(40,50),nrows=4,ncols=3,squeeze=True)
for i in range(df_train_site_7_meter_0['building_id'].nunique()):
   g=df_train_site_7_meter_0['building_id'].unique()[i]
   axes=axs[i%4][i//4]
   z=df_train_site_7_meter_0.loc[df_train_site_7_meter_0['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp',fontsize=20)
   axes.set_ylabel('electricity_meter_reading_site_7',fontsize=20)
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.6,wspace=0.3)
```



Important Obseravations

- Building 799 800 801 802 803 have streaks of constant zero values that needs to be filtered out.
- Building 801 803 798 contains spike which needs to be taken care of.

```
df_train_site_8=df_train_merge.loc[df_train_merge['site_id']==8]
```

```
df_train_site_8.isnull().sum()/df_train_site_8.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           1.00
floor_count          0.00
air_temperature      0.00
cloud_coverage       0.44
dew_temperature      0.00
precip_depth_1_hr    0.00
sea_level_pressure   0.01
wind_direction       0.03
wind_speed           0.00
dtype: float64
```

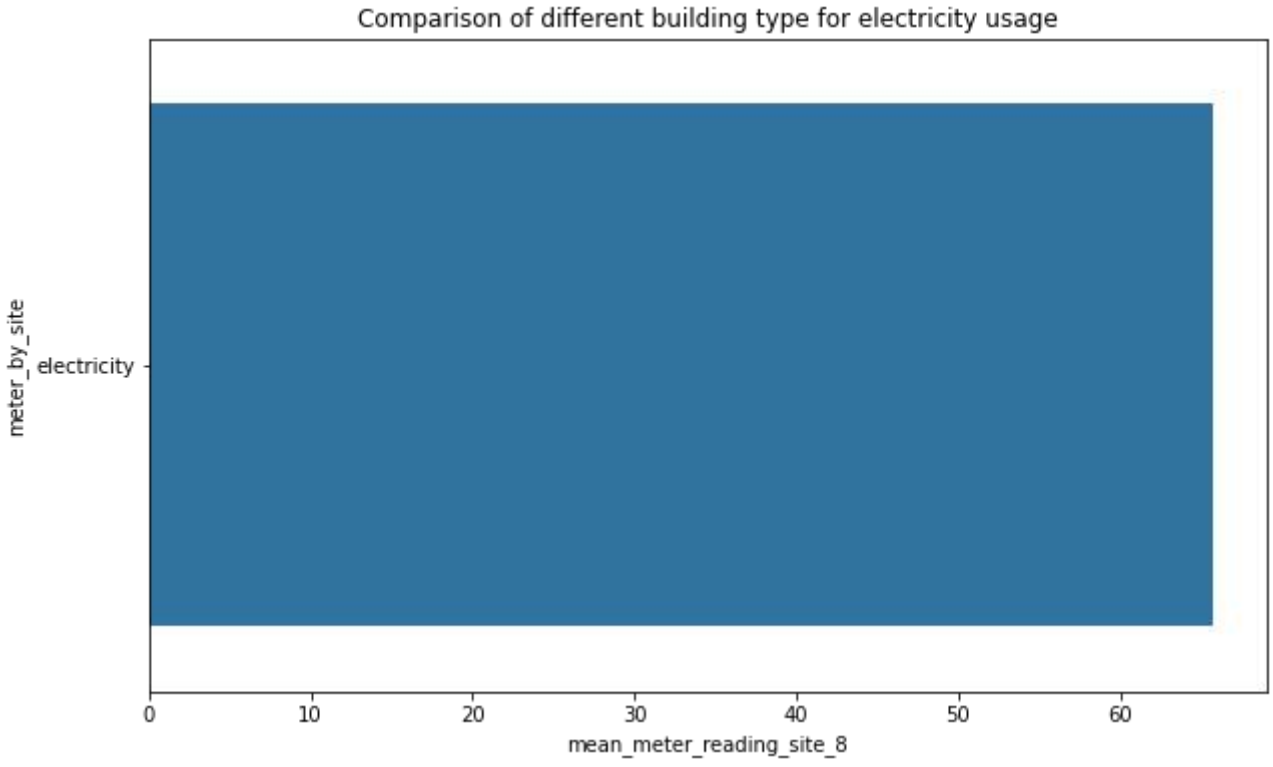
We can see that site 8 have null values which needs to be imputed

```
df_corr_8=df_train_site_8.corr()
df_corr_8.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.21	nan	0.19	nan	-0.20	0.00
meter_reading	0.21	1.00	nan	0.87	nan	-0.02	0.01
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.19	0.87	nan	1.00	nan	-0.00	0.00
year_built	nan	nan	nan	nan	nan	nan	nan
floor_count	-0.20	-0.02	nan	-0.00	nan	1.00	0.00
air_temperature	0.00	0.01	nan	0.00	nan	0.00	1.00
cloud_coverage	-0.00	0.00	nan	0.00	nan	0.00	0.29
dew_temperature	0.00	0.01	nan	0.00	nan	0.00	0.73

From the above correlation plot we can see that the meter reading correlates highly with the square feet of the building

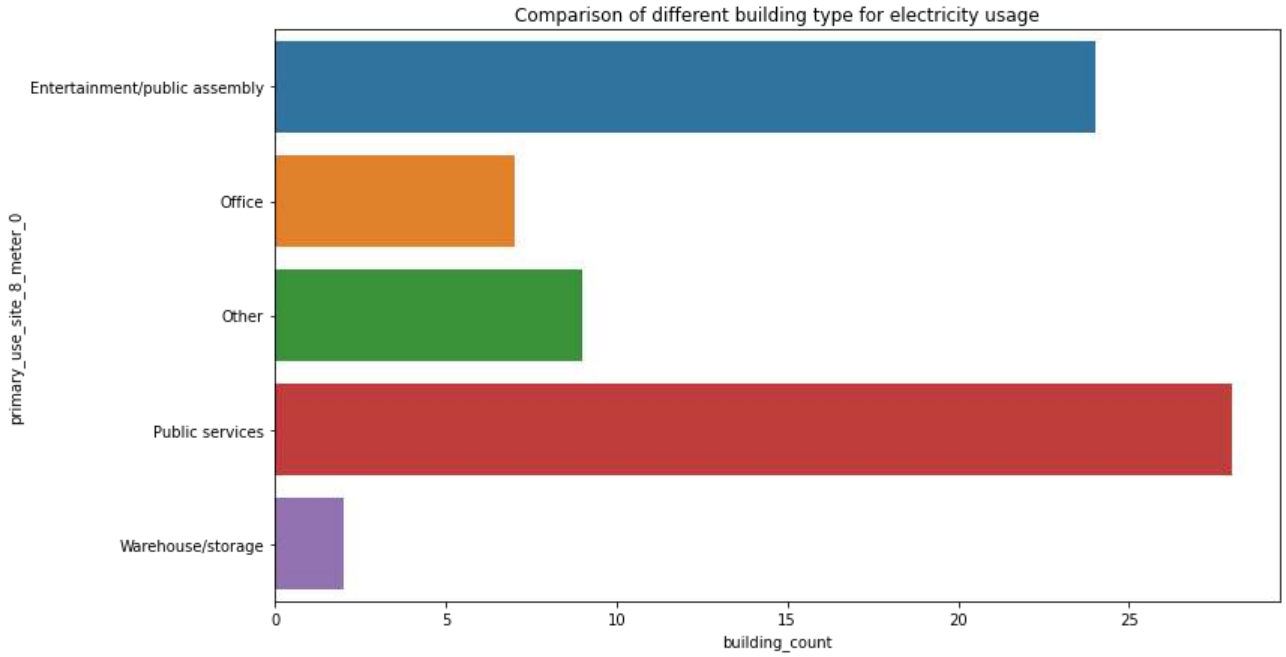
```
z=df_train_site_8.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(10,6))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading_site_8')
plt.ylabel('meter_by_site')
plt.title('Comparison of different building type for electricity usage')
plt.show()
```



At site 8 we can see that only source of energy consumption is electricity

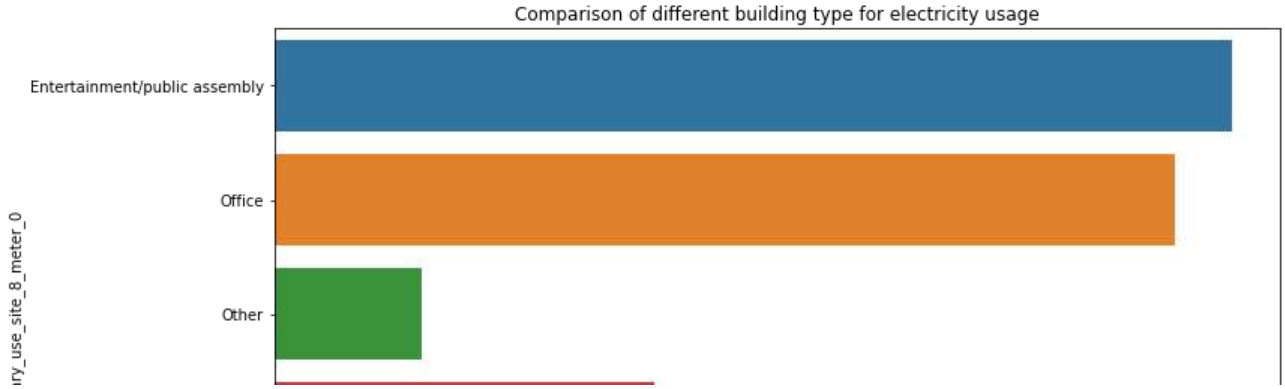
```
z=df_train_site_8.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
```

```
plt.xlabel('building_count')
plt.ylabel('primary_use_site_8_meter_0')
plt.title('Comparison of different building type for electricity usage')
plt.show()
```



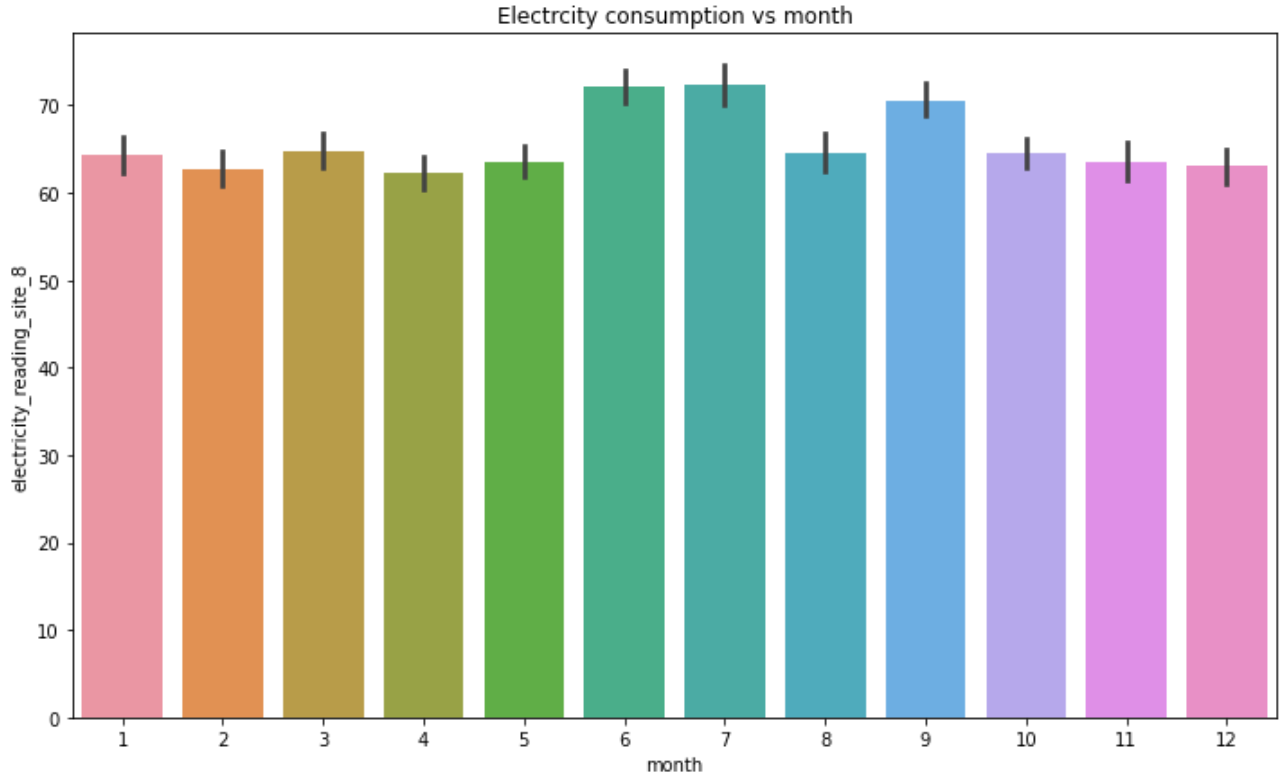
The above plot shows the building count for diffrent building type for electricity usage

```
z=df_train_site_8.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_reading')
plt.ylabel('primary_use_site_8_meter_0')
plt.title('Comparison of different building type for electricity usage')
plt.show()
```



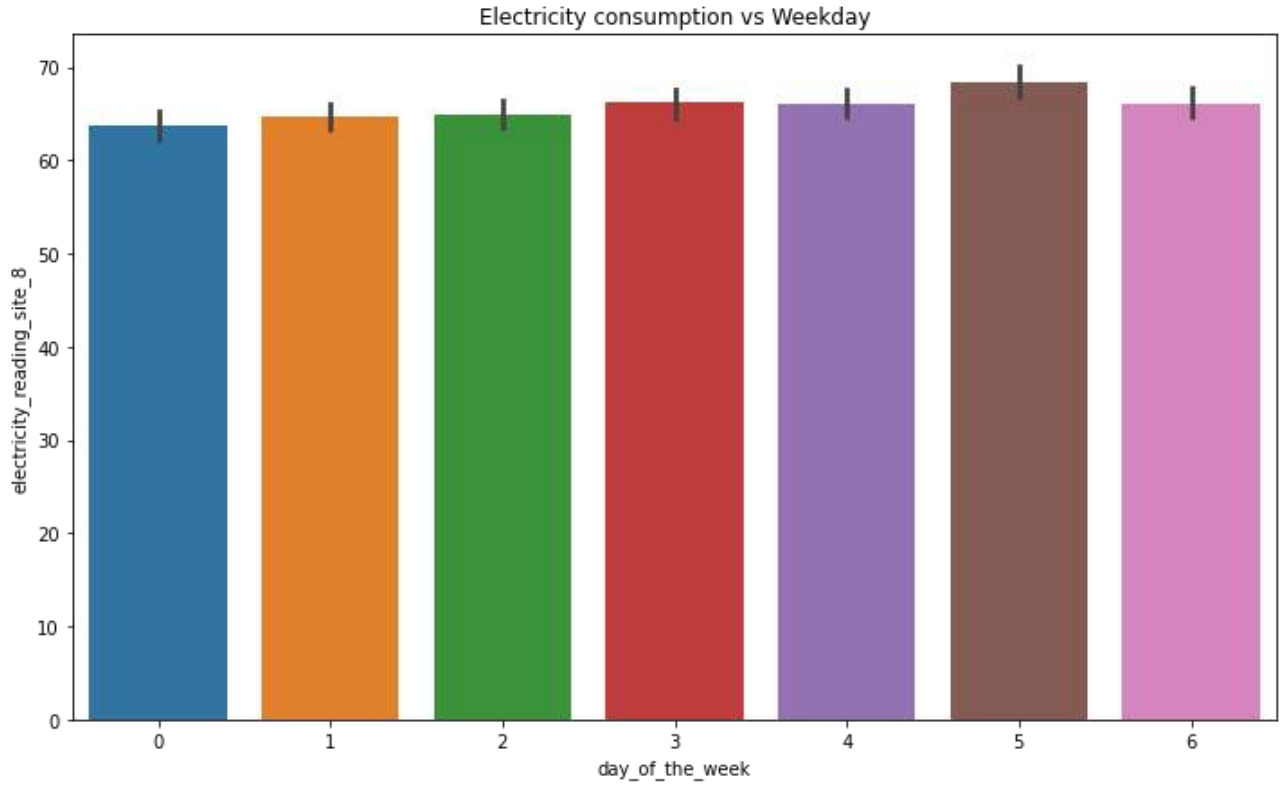
From the above plot we can see that the entertainment and office buildings are consuming highest electricity

```
df_train_site_8['month']=df_train_site_8['timestamp'].dt.month
df_train_site_8['weekday']=df_train_site_8['timestamp'].dt.weekday
df_train_site_8['hour']=df_train_site_8['timestamp'].dt.hour


fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_8
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_reading_site_8')
plt.title('Electrcity consumption vs month')
plt.show()
```



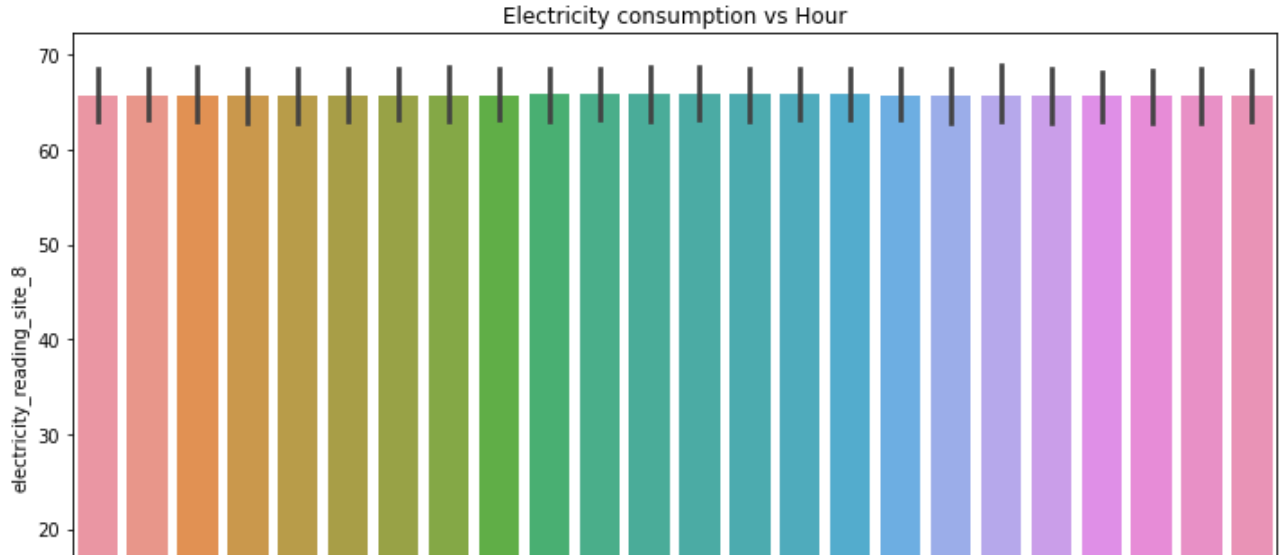
From the above plot we can see that the elctricity consumption varies over the month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_8
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_8')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



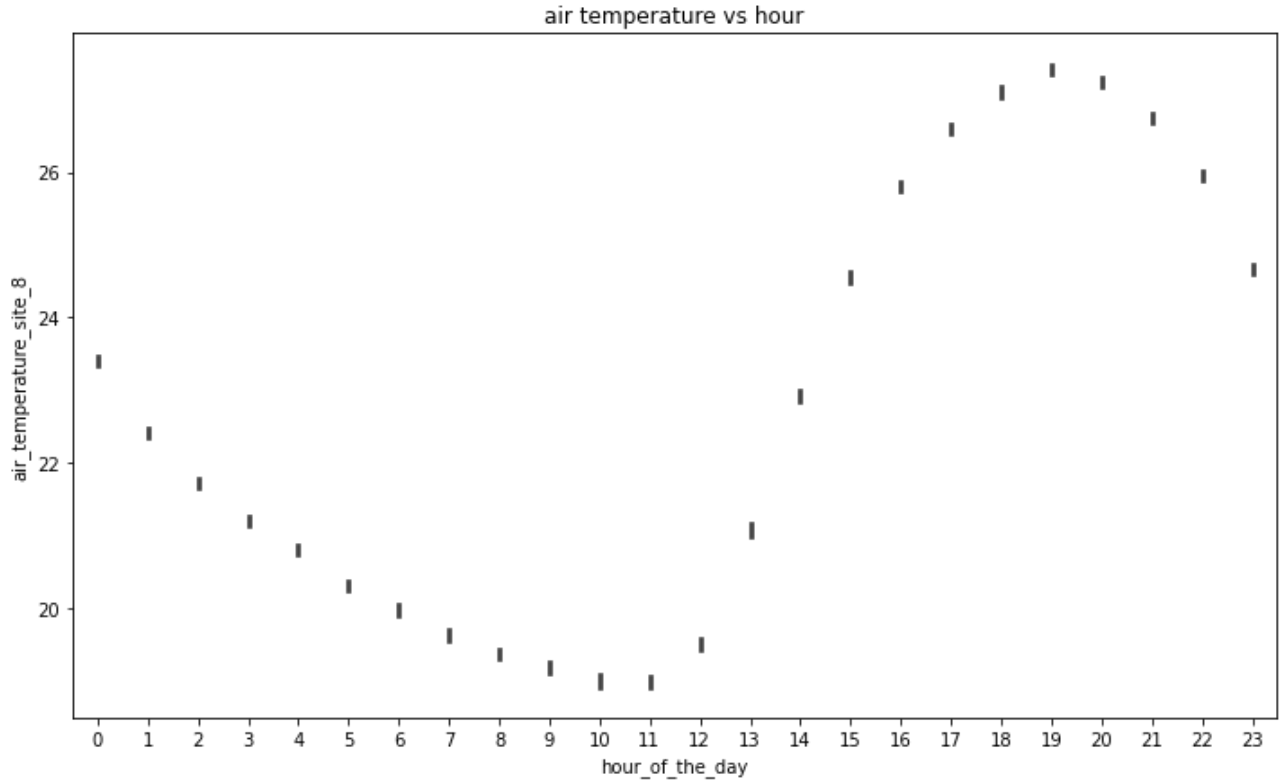
From the above plot we can see that even on weekend the electricity consumption is considerable as compared to the weekday.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_8
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_8')
plt.title('Electricity consumption vs Hour')
plt.show()
```



The above plot shows that the electrcity consumption does not vary over hour of the day which is pretty strange.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_8
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_8')
plt.title('air temperature vs hour')
plt.show()
```



From the above plot we can see that the weather timestamp is not in alignment with the local

```
fig,axs=plt.subplots(figsize=(80,150),nrows=14,ncols=5,squeeze=True)
for i in range(df_train_site_8['building_id'].nunique()):
   g=df_train_site_8['building_id'].unique()[i]
   axes=axs[i%14][i//14]
   z=df_train_site_8.loc[df_train_site_8['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp',fontsize=20)
   axes.set_ylabel('electricity_meter_reading_site_8',fontsize=20)
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.9,wspace=0.4)
```

Important Observations

- Building 818 848 857 845 have constant zero meter readings for certain months which needs to be filtered out.

- Building 869 having a spike which is an entertainment type building and that can be removed.
- Building 853 which is also an entertainment building has many inconsistencies in the meter reading as it contains zero meter readings in between the months consistently and

```
#Starting analysis for site 9
```

```
df_train_site_9=df_train_merge.loc[df_train_merge['site_id']==9]
```

```
df_train_site_9.isnull().sum()/df_train_site_9.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           1.00
floor_count          1.00
air_temperature      0.00
cloud_coverage       0.39
dew_temperature      0.00
precip_depth_1_hr    0.00
sea_level_pressure   0.03
wind_direction       0.29
wind_speed           0.01
dtype: float64
```

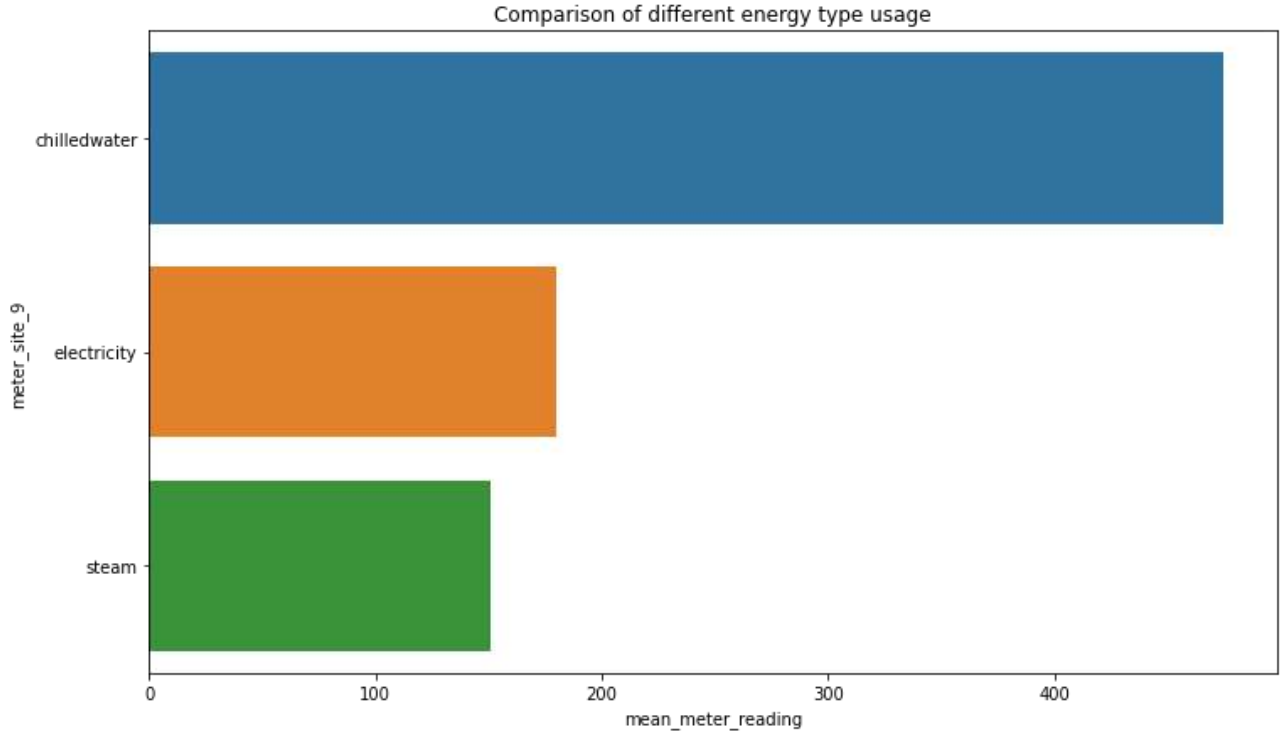
From here we can see that site 9 is having some missing values which needs to be imputed

```
df_corr_9=df_train_site_9.corr()
df_corr_9.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.03	nan	0.08	nan	nan	-0.00
meter_reading	0.03	1.00	nan	0.37	nan	nan	0.07
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.08	0.37	nan	1.00	nan	nan	-0.00
year_built	nan	nan	nan	nan	nan	nan	nan
floor_count	nan	nan	nan	nan	nan	nan	nan
air_temperature	-0.00	0.07	nan	-0.00	nan	nan	1.00
cloud_coverage	-0.00	0.03	nan	-0.00	nan	nan	0.20
dew_temperature	-0.00	0.08	nan	-0.00	nan	nan	0.79
precip_depth_1_hr	-0.00	-0.00	nan	-0.00	nan	nan	-0.03
sea_level_pressure	0.00	-0.03	nan	0.00	nan	nan	-0.55
wind_direction	0.00	-0.01	nan	0.00	nan	nan	0.01
wind_speed	0.00	-0.00	nan	0.00	nan	nan	0.18

From the correlation plot we can see that the meter reading is not strongly correlated with any of the features.

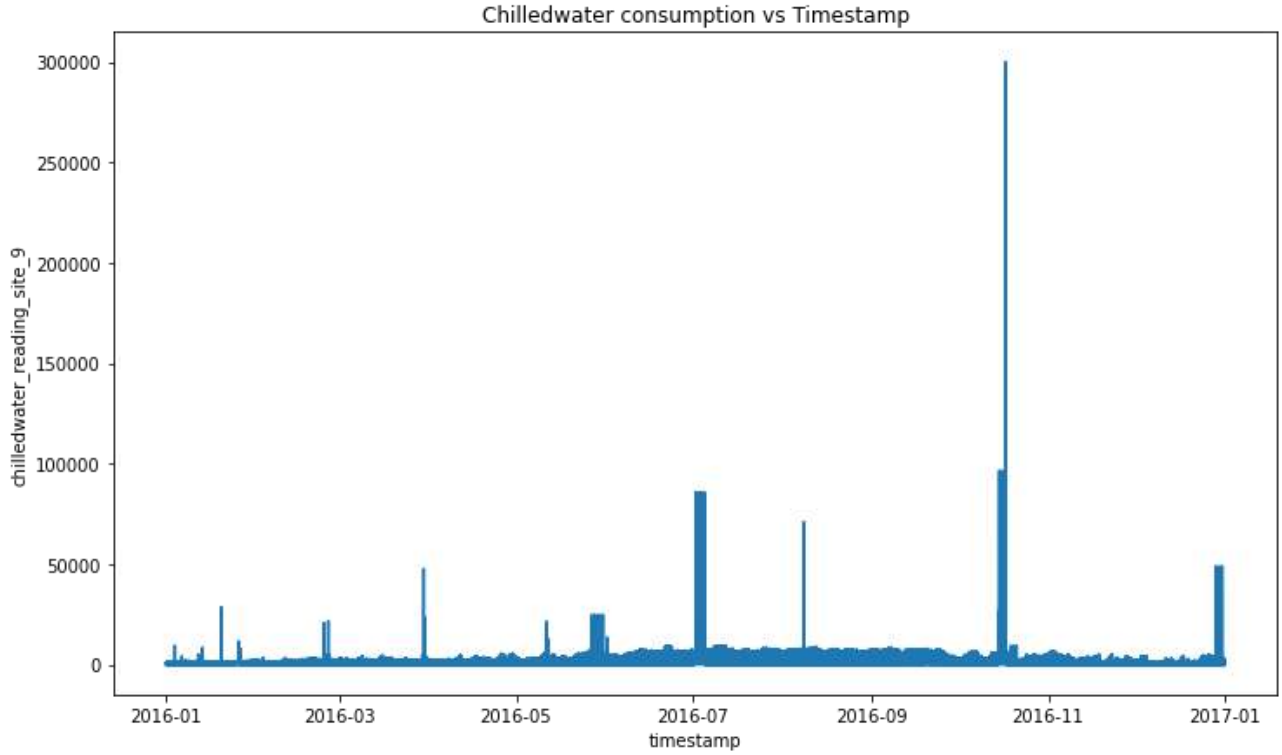
```
z=df_train_site_9.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading')
plt.ylabel('meter_site_9')
plt.title('Comparison of different energy type usage')
plt.show()
```



From the above plot we can see that chilledwater is having the highest energy consumption at site 9.

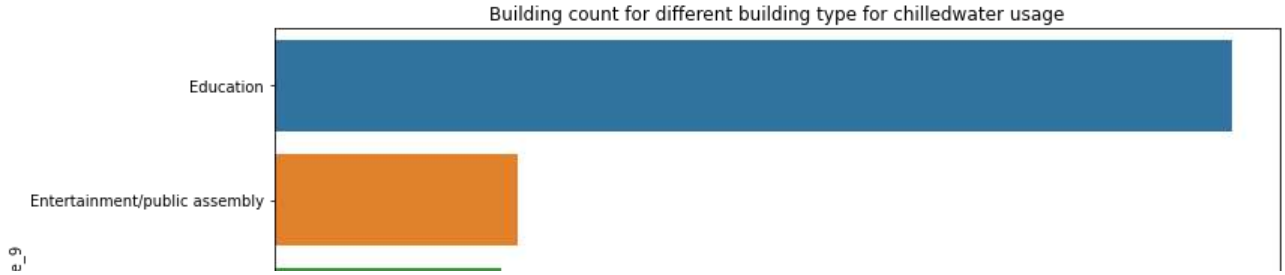
```
df_train_site_9_meter_1=df_train_site_9.loc[df_train_site_9['meter']=='chilledwater']
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_1
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('chilledwater_reading_site_9')
plt.title('Chilledwater consumption vs Timestamp')
plt.show()
```



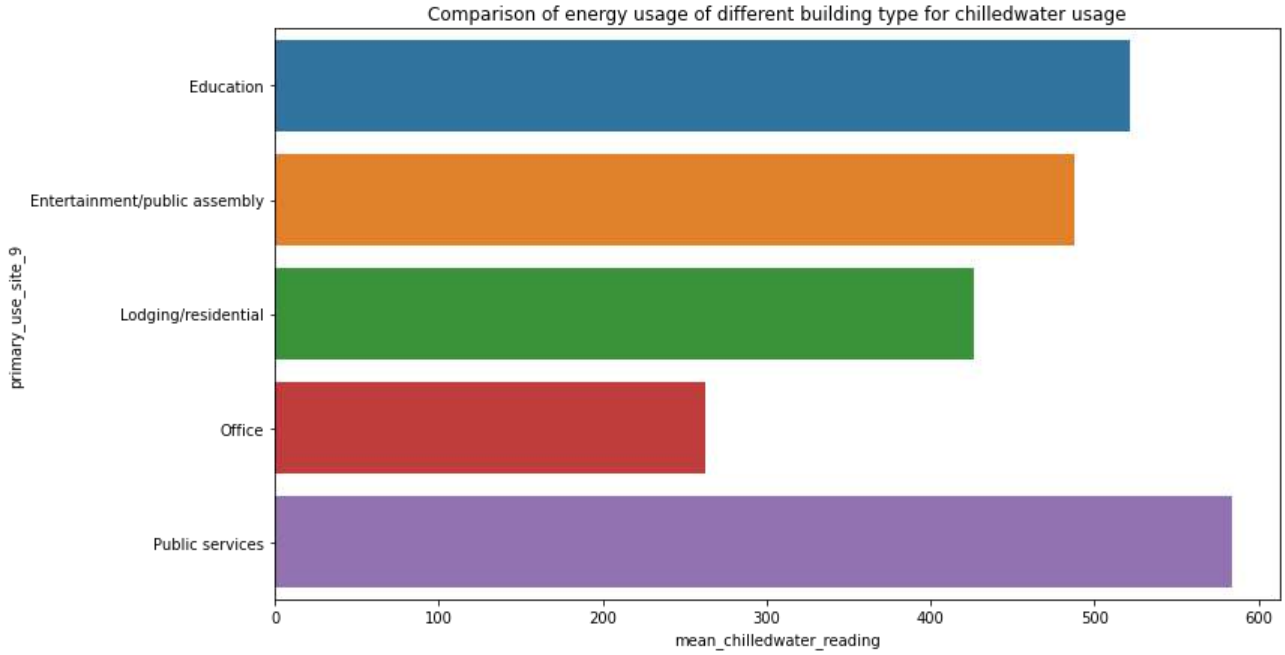
The above plot gives us a rough idea of the chilledwater consumption along with the timestamp.We can observe here that a very high spike occurs in the 10th month whose value is close to 300k.

```
z=df_train_site_9_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_site_9')
plt.title('Building count for different building type for chilledwater usage')
plt.show()
```



The above plot represents the bulding count for different building type for chilledwater usage at site 9

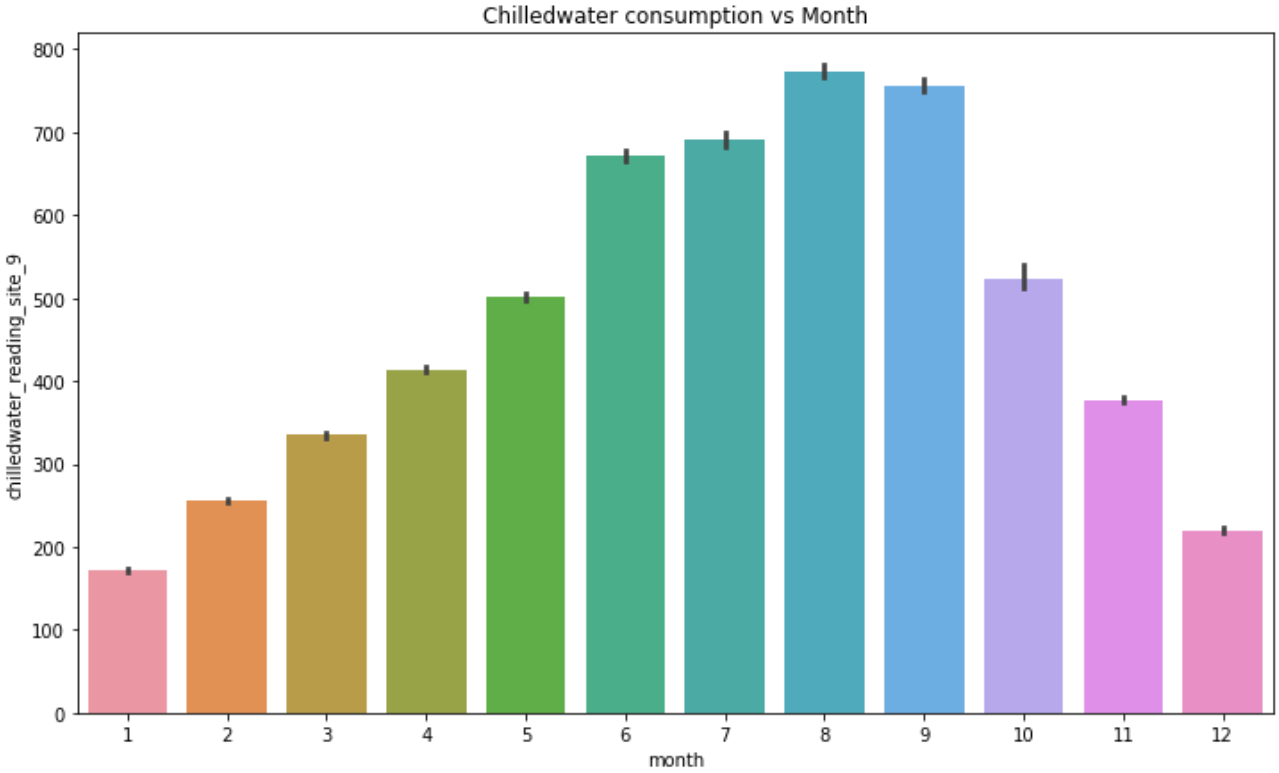
```
z=df_train_site_9_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading')
plt.ylabel('primary_use_site_9')
plt.title('Comparison of energy usage of different building type for chilledwater usage')
plt.show()
```



This plot represents the chilledwater consumption which shows that public services is consuming the highest although they are relatively lesser in number.

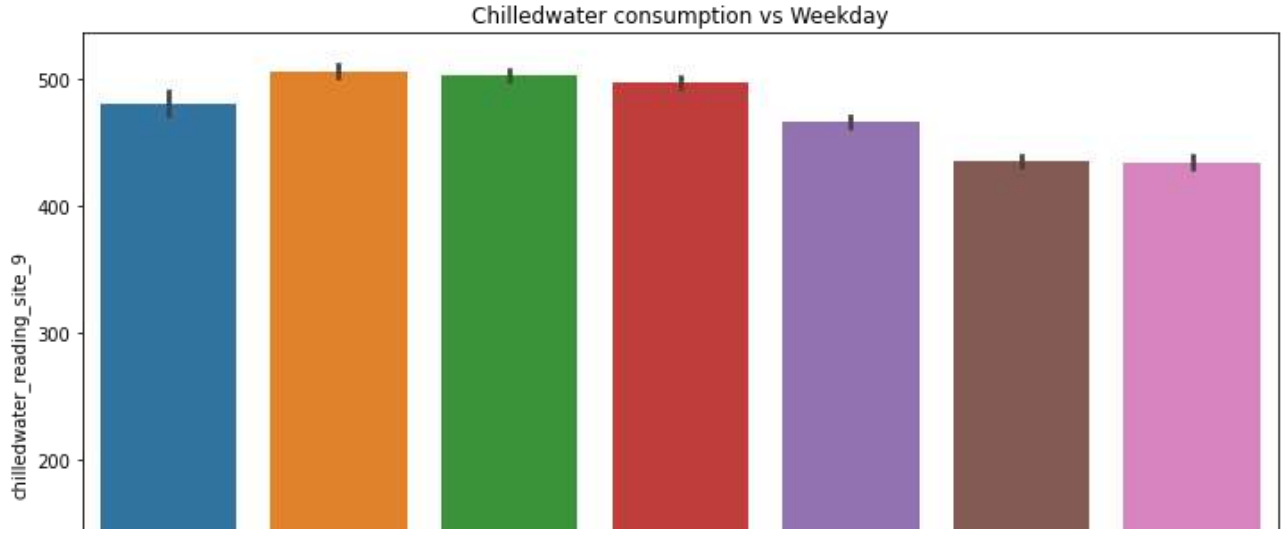
```
df_train_site_9_meter_1['month']=df_train_site_9_meter_1['timestamp'].dt.month
df_train_site_9_meter_1['weekday']=df_train_site_9_meter_1['timestamp'].dt.weekday
df_train_site_9_meter_1['hour']=df_train_site_9_meter_1['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_1
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading_site_9')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



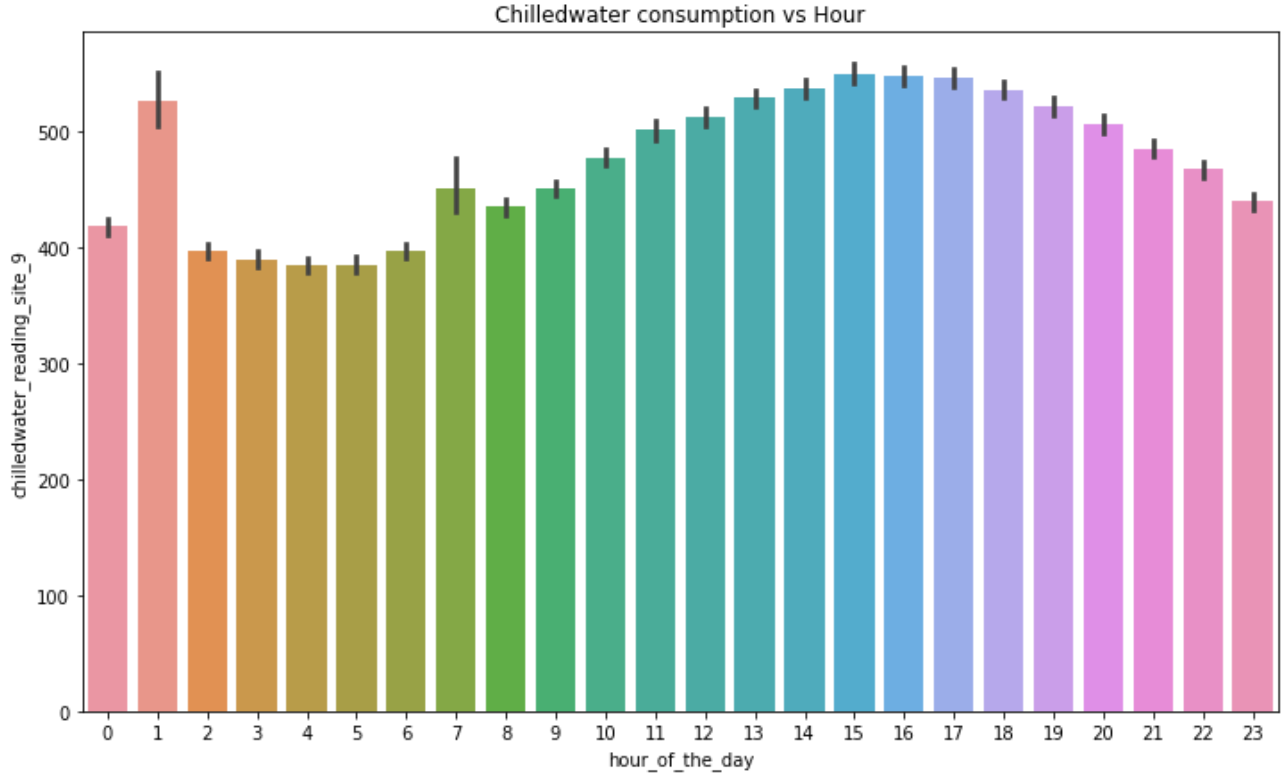
From the above plot we can see that chilledwater is having the highest consumption during the summer month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_1
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading_site_9')
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



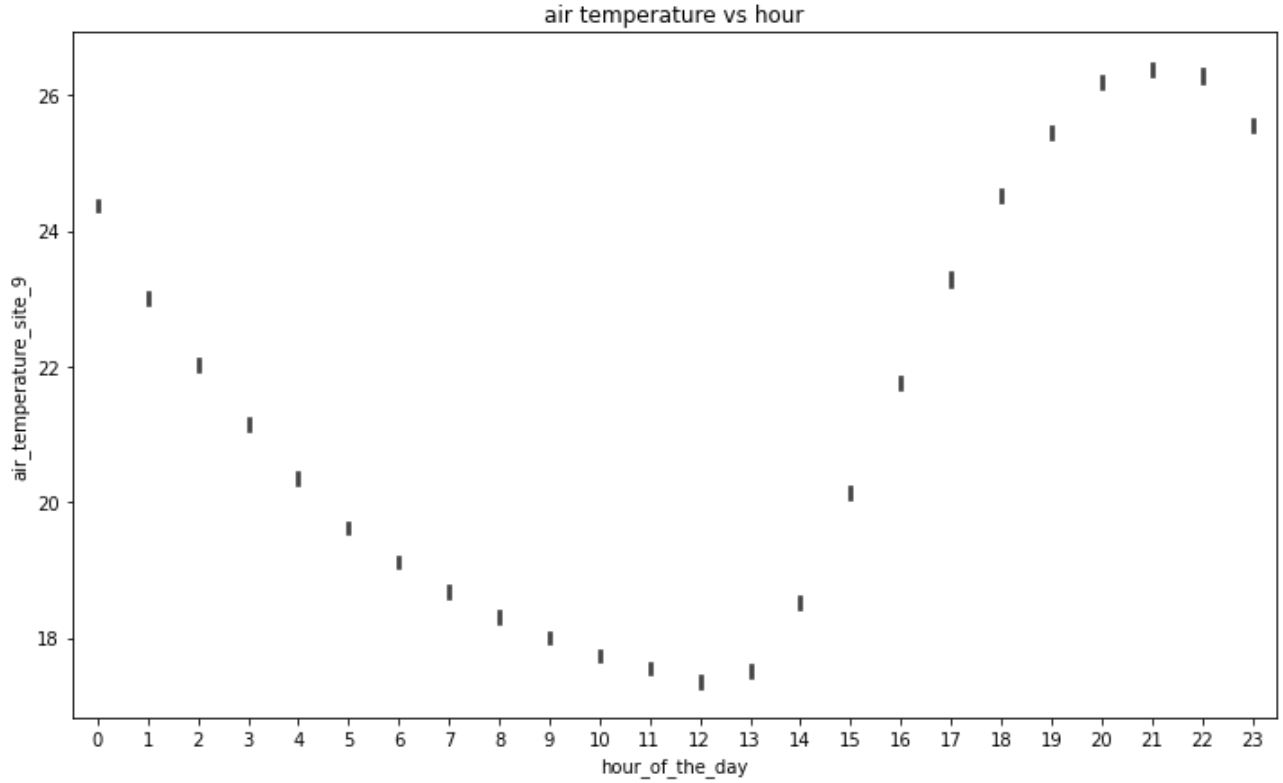
From the above plot we can see that the chilledwater consumption is less over the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilledwater_reading_site_9')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```



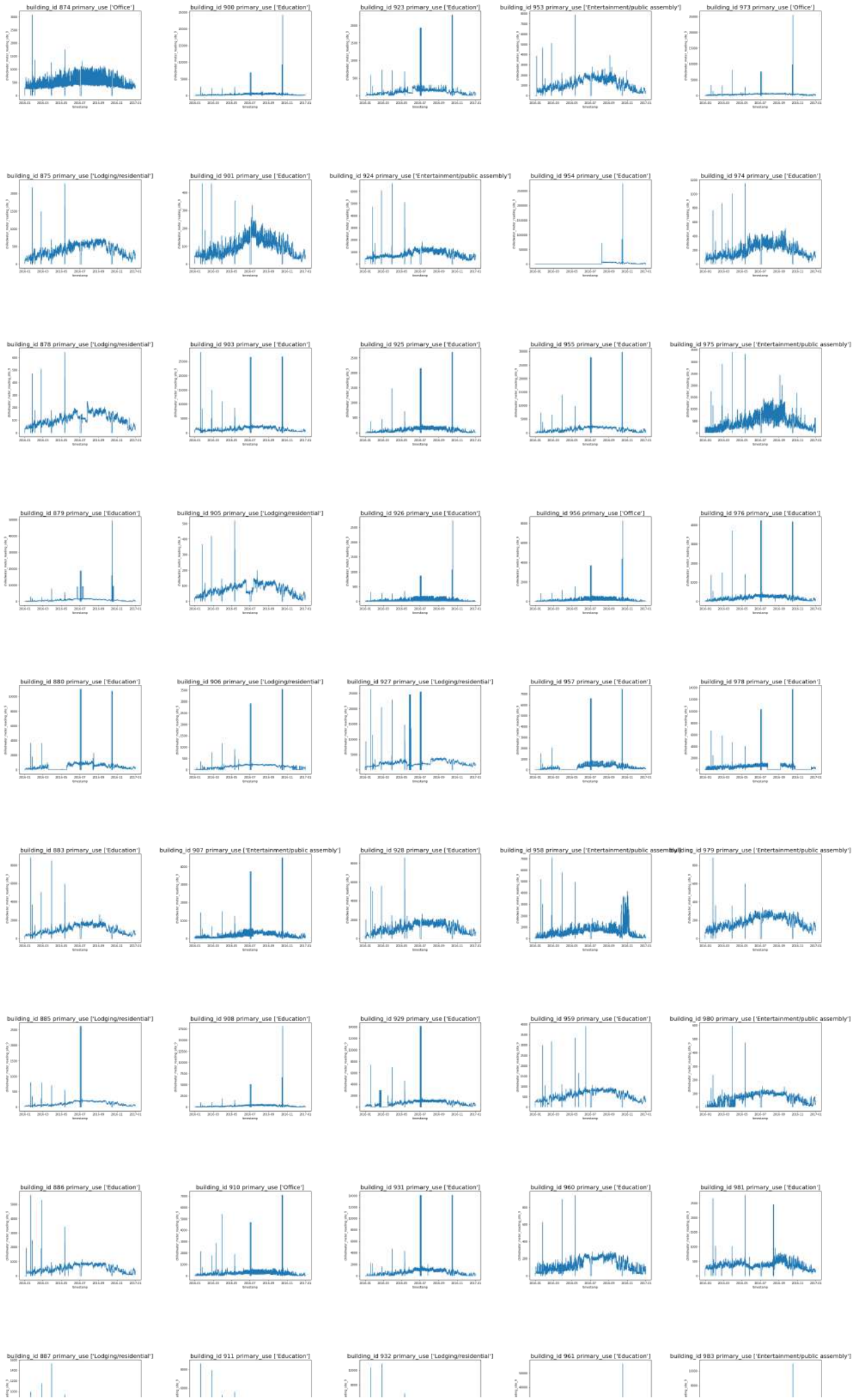
From the above plot we can see that it is showing a spike at 01:00 pm in the night which is an abnormal behaviour.

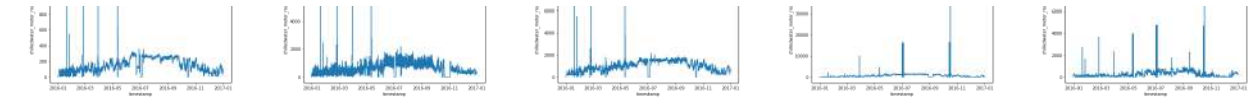
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_9')
plt.title('air temperature vs hour')
plt.show()
```

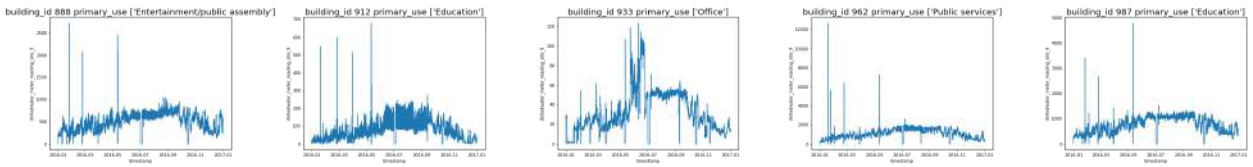


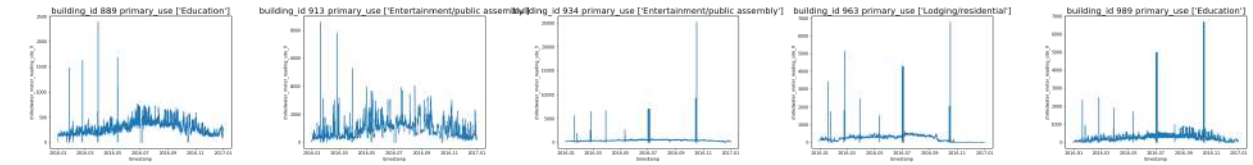
From the above plot we can see that weather timestamp is not in alignment with the local timestamp of the hourly meter reading as the temperature peaks around 21:00 pm

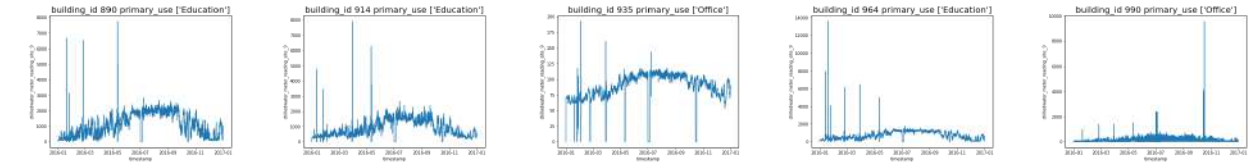
```
fig,axs=plt.subplots(figsize=(50,200),nrows=19,ncols=5,squeeze=True)
for i in range(df_train_site_9_meter_1['building_id'].nunique()):
   g=df_train_site_9_meter_1['building_id'].unique()[i]
   axes=axs[i%19][i//19]
   z=df_train_site_9_meter_1.loc[df_train_site_9_meter_1['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('chilledwater_meter_reading_site_9')
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.9,wspace=0.4)
```


building_id 874 primary_use ['Office']
building_id 900 primary_use ['Education']
building_id 923 primary_use ['Education']
building_id 953 primary_use ['Entertainment/public assembly']
building_id 973 primary_use ['Office']
building_id 875 primary_use ['Lodging/residential']
building_id 901 primary_use ['Education']
building_id 924 primary_use ['Entertainment/public assembly']
building_id 954 primary_use ['Education']
building_id 974 primary_use ['Education']
building_id 878 primary_use ['Lodging/residential']
building_id 903 primary_use ['Education']
building_id 925 primary_use ['Education']
building_id 955 primary_use ['Education']
building_id 975 primary_use ['Entertainment/public assembly']
building_id 879 primary_use ['Education']
building_id 905 primary_use ['Lodging/residential']
building_id 926 primary_use ['Education']
building_id 956 primary_use ['Office']
building_id 976 primary_use ['Education']
building_id 880 primary_use ['Education']
building_id 906 primary_use ['Lodging/residential']
building_id 927 primary_use ['Lodging/residential']
building_id 957 primary_use ['Education']
building_id 978 primary_use ['Education']
building_id 883 primary_use ['Education']
building_id 907 primary_use ['Entertainment/public assembly']
building_id 928 primary_use ['Education']
building_id 958 primary_use ['Entertainment/public assembly']
building_id 979 primary_use ['Entertainment/public assembly']
building_id 885 primary_use ['Lodging/residential']
building_id 908 primary_use ['Education']
building_id 929 primary_use ['Education']
building_id 959 primary_use ['Education']
building_id 980 primary_use ['Entertainment/public assembly']
building_id 886 primary_use ['Education']
building_id 910 primary_use ['Office']
building_id 931 primary_use ['Education']
building_id 960 primary_use ['Education']
building_id 981 primary_use ['Education']
building_id 887 primary_use ['Lodging/residential']
building_id 911 primary_use ['Education']
building_id 932 primary_use ['Lodging/residential']
building_id 961 primary_use ['Education']
building_id 983 primary_use ['Entertainment/public assembly']

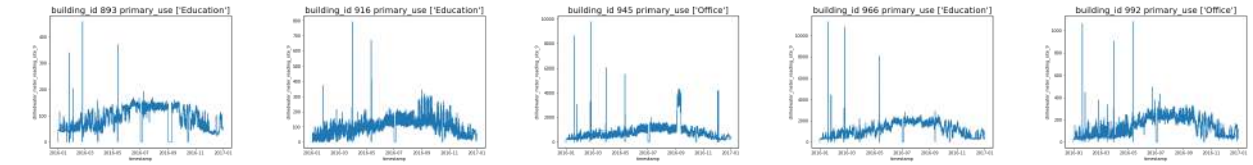


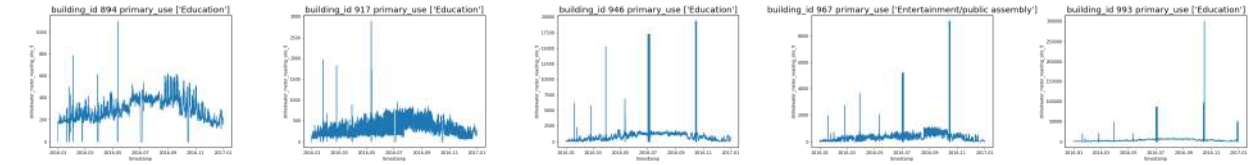

building_id 888 primary_use ['Entertainment/public assembly']
building_id 912 primary_use ['Education']
building_id 933 primary_use ['Office']
building_id 962 primary_use ['Public services']
building_id 987 primary_use ['Education']

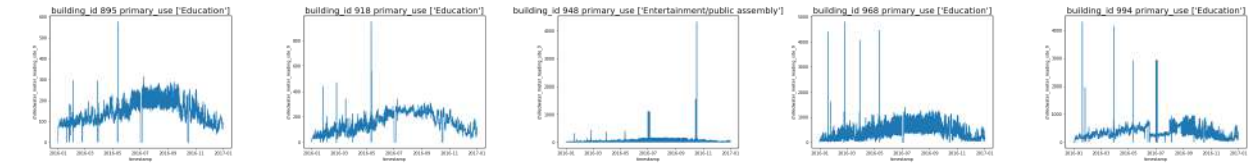

building_id 889 primary_use ['Education']
building_id 913 primary_use ['Entertainment/public assembly']
building_id 934 primary_use ['Entertainment/public assembly']
building_id 963 primary_use ['Lodging/residential']
building_id 989 primary_use ['Education']

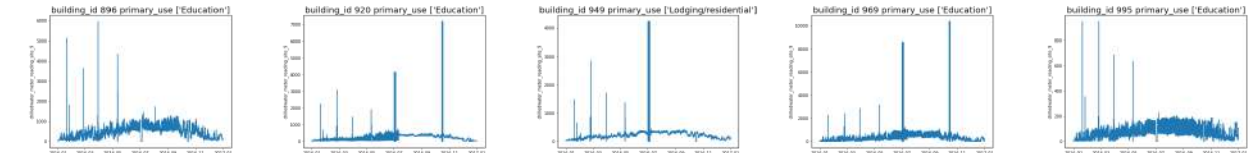

building_id 890 primary_use ['Education']
building_id 914 primary_use ['Education']
building_id 935 primary_use ['Office']
building_id 964 primary_use ['Education']
building_id 990 primary_use ['Office']

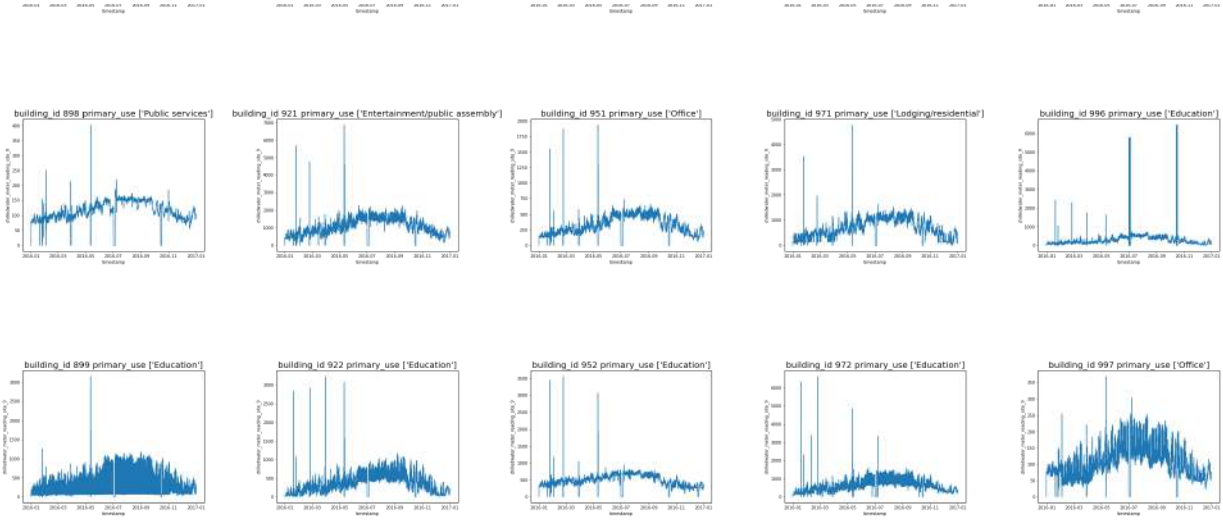

building_id 891 primary_use ['Education']
building_id 915 primary_use ['Education']
building_id 942 primary_use ['Lodging/residential']
building_id 965 primary_use ['Lodging/residential']
building_id 991 primary_use ['Education']


building_id 893 primary_use ['Education']
building_id 916 primary_use ['Education']
building_id 945 primary_use ['Office']
building_id 966 primary_use ['Education']
building_id 992 primary_use ['Office']


building_id 894 primary_use ['Education']
building_id 917 primary_use ['Education']
building_id 946 primary_use ['Education']
building_id 967 primary_use ['Entertainment/public assembly']
building_id 993 primary_use ['Education']


building_id 895 primary_use ['Education']
building_id 918 primary_use ['Education']
building_id 948 primary_use ['Entertainment/public assembly']
building_id 968 primary_use ['Education']
building_id 994 primary_use ['Education']


building_id 896 primary_use ['Education']
building_id 920 primary_use ['Education']
building_id 949 primary_use ['Lodging/residential']
building_id 969 primary_use ['Education']
building_id 995 primary_use ['Education']


building_id 898 primary_use ['Public services']
building_id 921 primary_use ['Entertainment/public assembly']
building_id 951 primary_use ['Office']
building_id 971 primary_use ['Lodging/residential']
building_id 996 primary_use ['Education']
building_id 899 primary_use ['Education']
building_id 922 primary_use ['Education']
building_id 952 primary_use ['Education']
building_id 972 primary_use ['Education']
building_id 997 primary_use ['Office']

Important observations

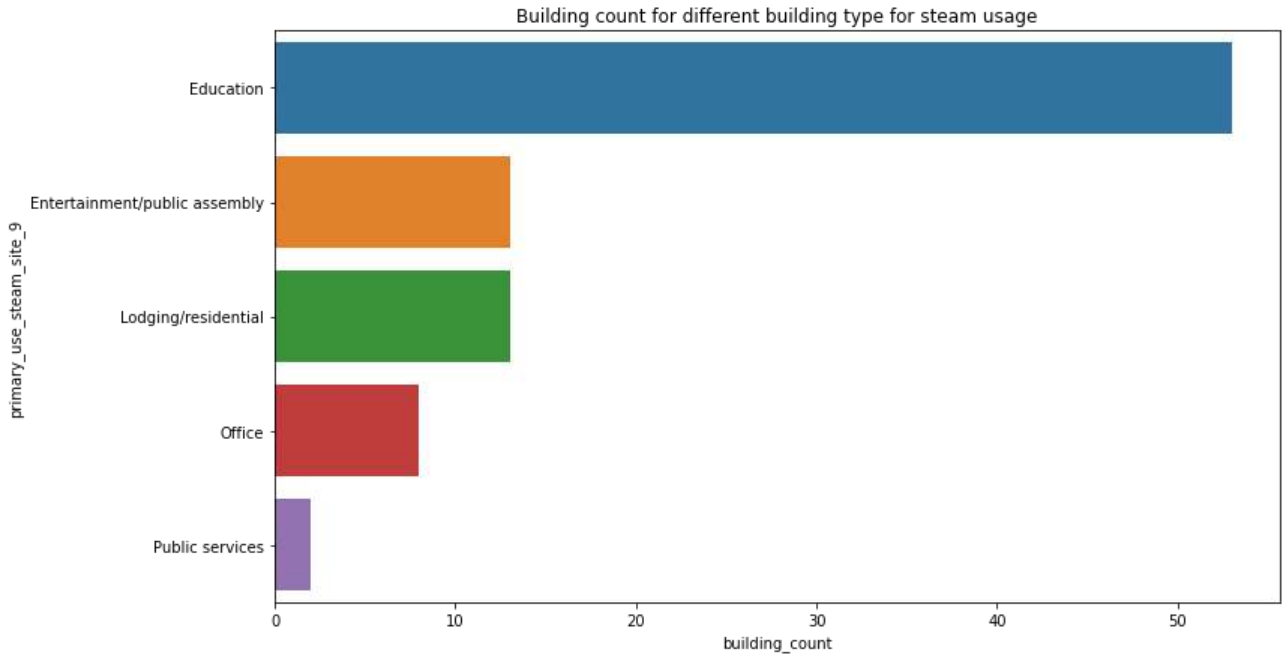
- From the above plot we can observe that many of the buildings are having constant spike whose magnitude is very high and we need to carefully remove those readings as they are definitely an anomaly.
- We are also able to observe that in between the readingd suddenly the meter readings goes to zero which is also a faulty reading and needs to be filtered out.

```
df_train_site_9_meter_2=df_train_site_9.loc[df_train_site_9['meter']=='steam']
```

```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_9_meter_2
ax.plot(z['timestamp'],z['meter_reading'])
plt.ylabel('steam_reading_site_9')
plt.xlabel('timestamp')
plt.ylabel('Steam consumption vs Timestamp')
plt.show()
```

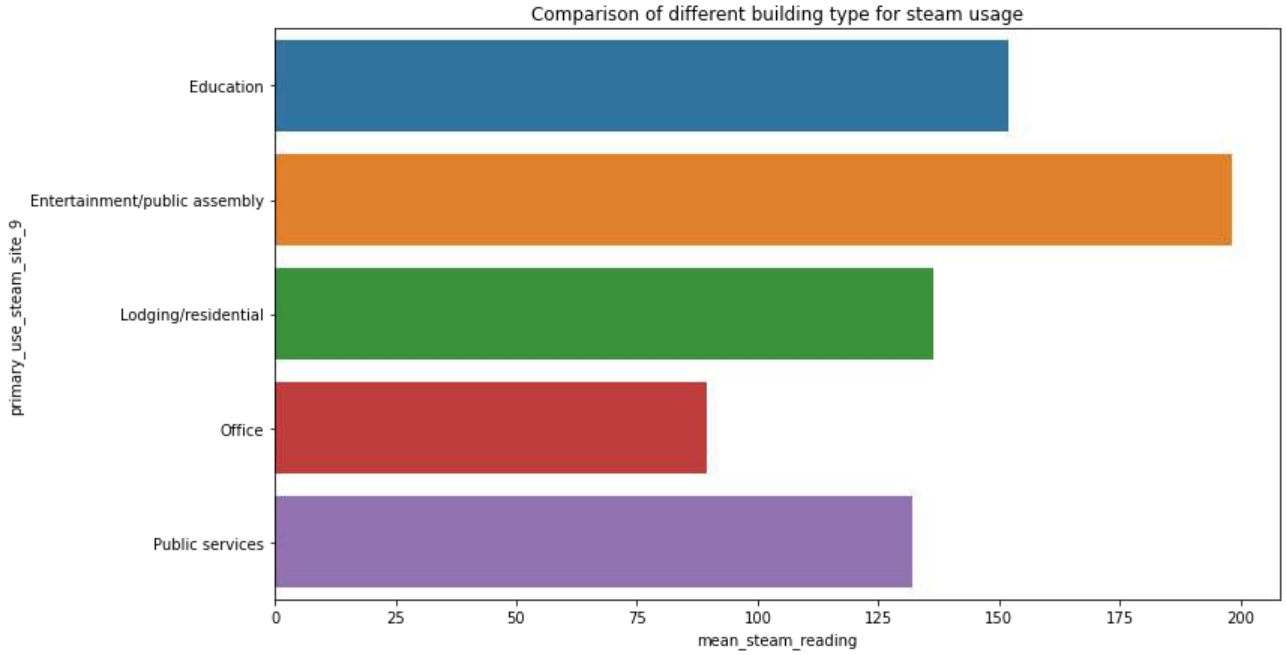
This plot shows the overall steam consumption with the timestamp.We can also notice the high peaked readings in between the months.

```
z=df_train_site_9_meter_2.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_steam_site_9')
plt.title('Building count for different building type for steam usage')
plt.show()
```



The above plot represents the building count for steam usage for the different building type.

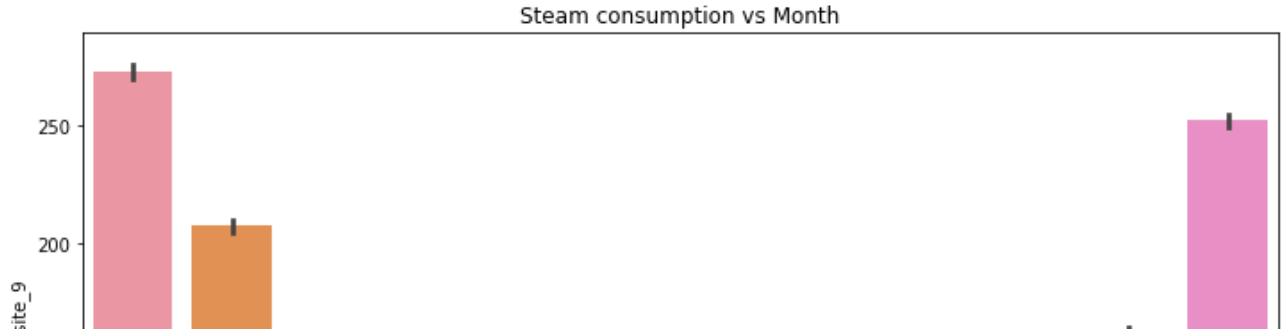
```
z=df_train_site_9_meter_2.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_steam_reading')
plt.ylabel('primary_use_steam_site_9')
plt.title('Comparison of different building type for steam usage')
plt.show()
```



From the above plot we can see that on an average the entertainment buildings are having more steam usage as compared to the other buildings although they are lesser in number.

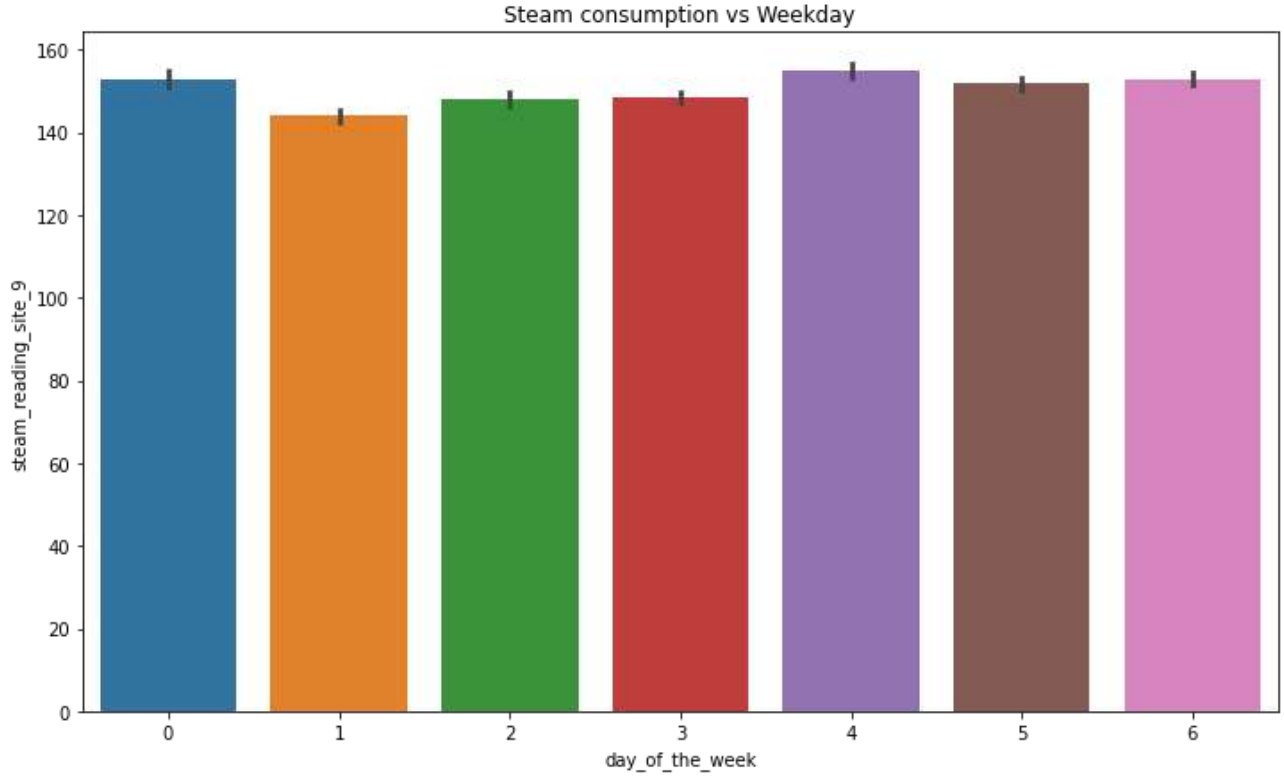
```
df_train_site_9_meter_2['month']=df_train_site_9_meter_2['timestamp'].dt.month
df_train_site_9_meter_2['weekday']=df_train_site_9_meter_2['timestamp'].dt.weekday
df_train_site_9_meter_2['hour']=df_train_site_9_meter_2['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_2
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('steam_reading_site_9')
plt.title('Steam consumption vs Month')
plt.show()
```



From the above plot we can observe that the steam consumption is highest for the winter months and decreases gradually as we approach the summer month

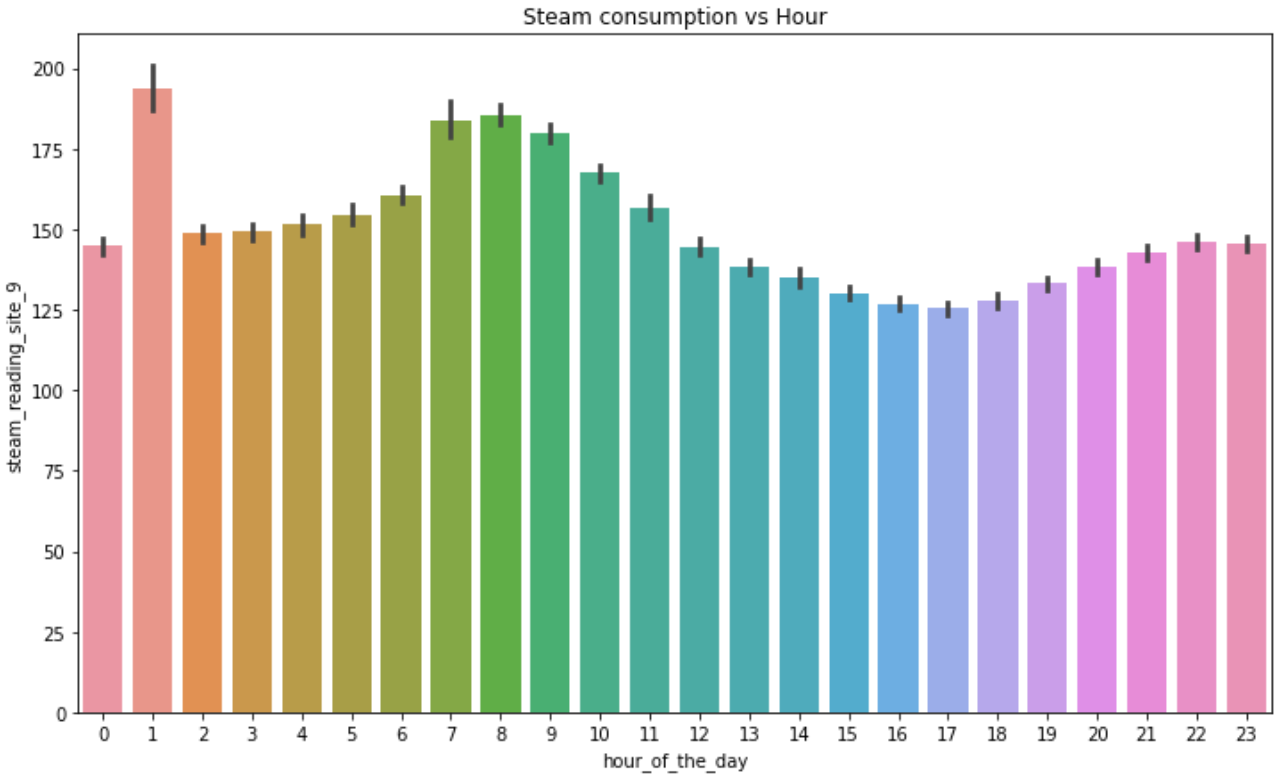

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_2
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('steam_reading_site_9')
plt.title('Steam consumption vs Weekday')
plt.show()
```



Here from the above plot we can see that steam consumption shows variation over all the days in a week

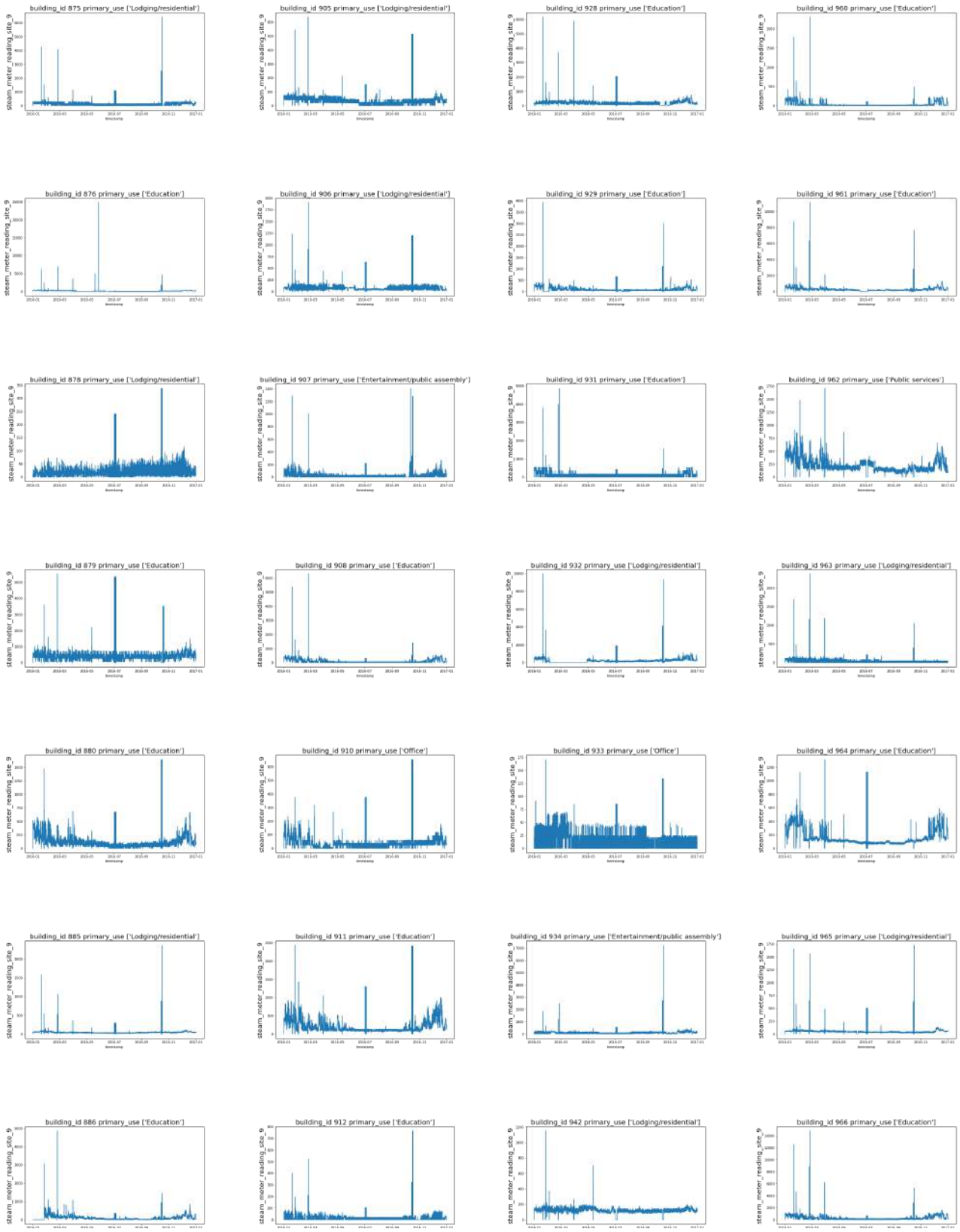
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
```

```
plt.ylabel('steam_reading_site_9')
plt.title('Steam consumption vs Hour')
plt.show()
```

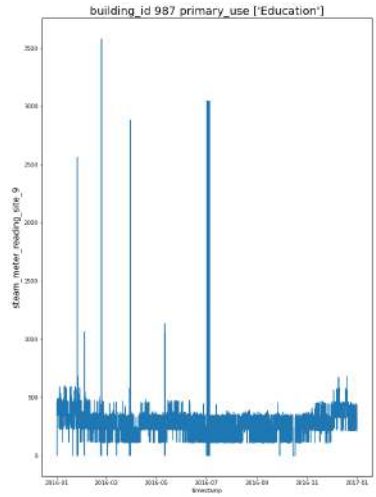


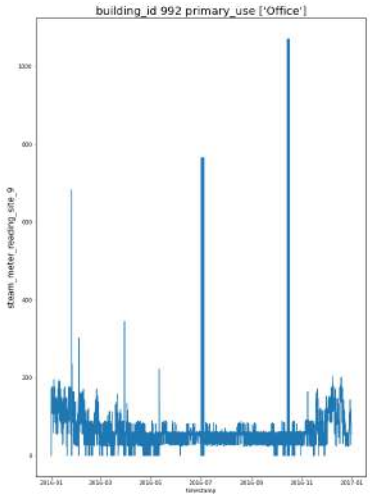
From the above plot we can observe that steam consumption also shows peak around 01:00 pm on the night.WE can also observe that it shows high consumption during the morning hours and decreases gradually as the day proceeds.

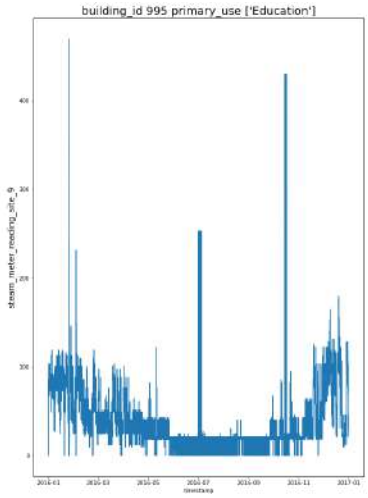
```
fig,axs=plt.subplots(figsize=(50,200),nrows=20,ncols=4,squeeze=True)
for i in range(80):
   g=df_train_site_9_meter_2['building_id'].unique()[i]
   axes=axs[i%20][i//20]
   z=df_train_site_9_meter_2.loc[df_train_site_9_meter_2['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('steam_meter_reading_site_9',fontsize=20)
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.9,wspace=0.4)
```

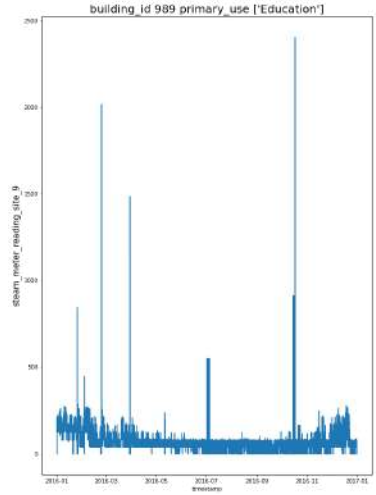


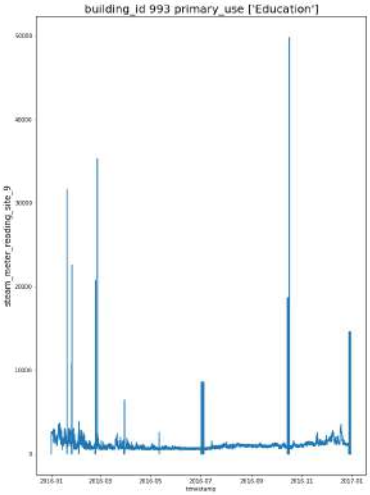

```
fig,axs=plt.subplots(figsize=(40,60),nrows=3,ncols=3,squeeze=True)
for i in range(9):
    g=df_train_site_9_meter_2['building_id'].unique()[80:90][i]
    axes=axs[i%3][i//3]
    z=df_train_site_9_meter_2.loc[df_train_site_9_meter_2['building_id']==g]
    axes.plot(z['timestamp'],z['meter_reading'])
    axes.set_xlabel('timestamp')
    axes.set_ylabel('steam_meter_reading_site_9',fontsize=15)
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
    plt.subplots_adjust(hspace=0.5,wspace=0.4)
```

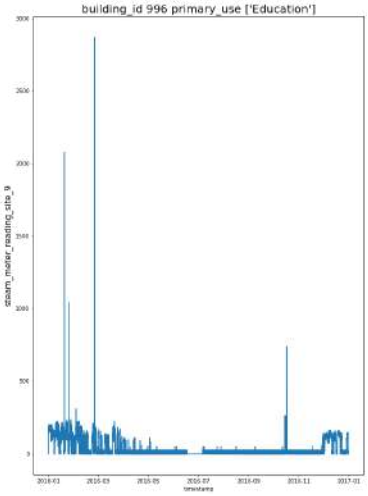

building_id 987 primary_use ['Education']
steam_meter_reading_site_9

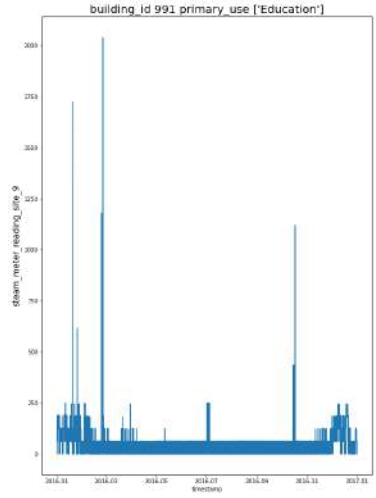

building_id 992 primary_use ['Office']
steam_meter_reading_site_9

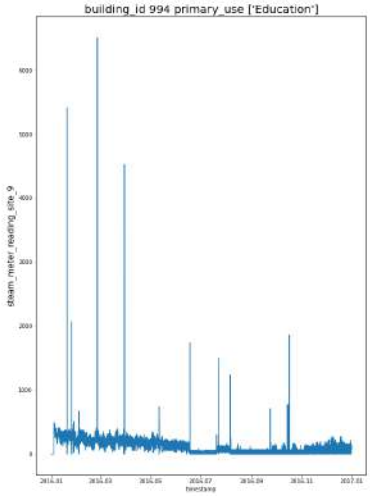

building_id 995 primary_use ['Education']
steam_meter_reading_site_9

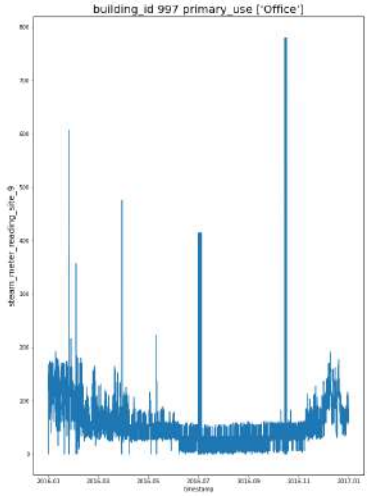

building_id 989 primary_use ['Education']
steam_meter_reading_site_9


building_id 993 primary_use ['Education']
steam_meter_reading_site_9


building_id 996 primary_use ['Education']
steam_meter_reading_site_9


building_id 991 primary_use ['Education']
steam_meter_reading_site_9


building_id 994 primary_use ['Education']
steam_meter_reading_site_9


building_id 997 primary_use ['Office']
steam_meter_reading_site_9

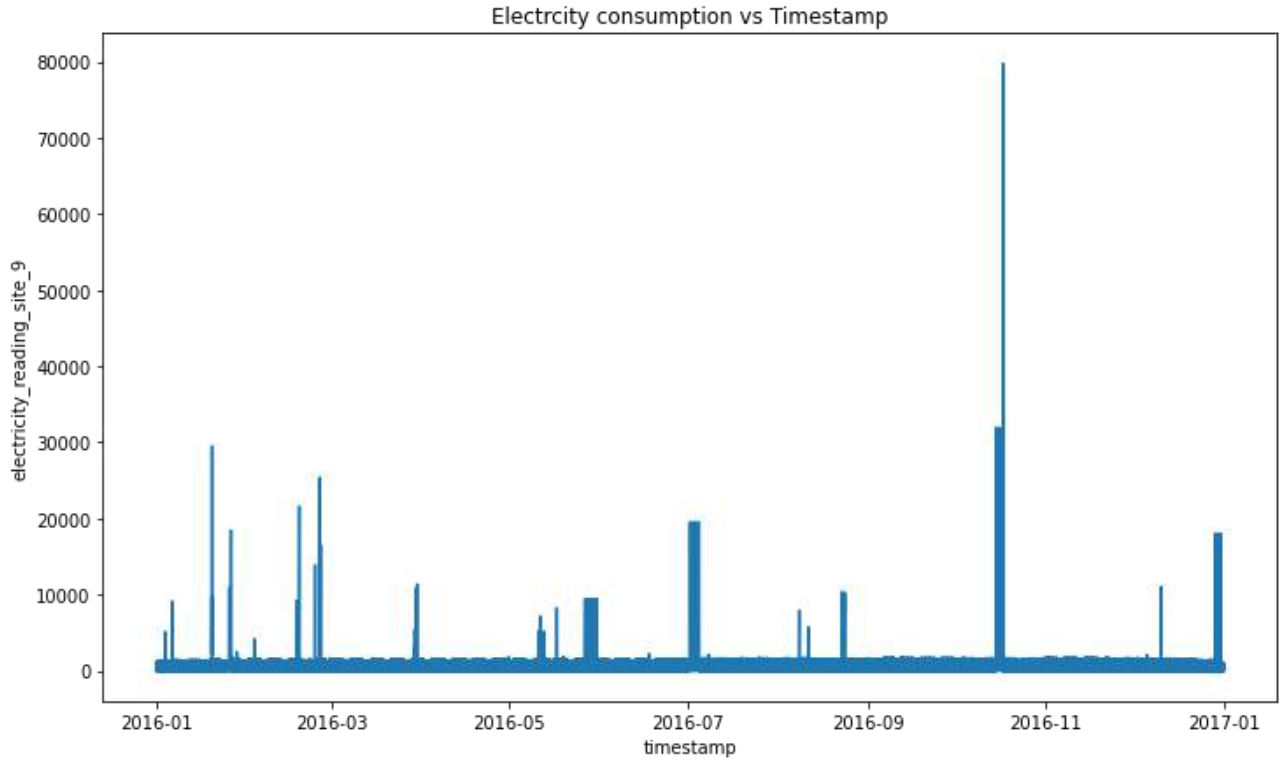
The total number of buildings which are using steam at site 9 is 89 therefore I divided it into 80 and 9 for a better representation.

Important Observations

- Similarly as we observed for the chilledwater readings here also the readings are filled with spikes and constant zero meter readings inbetween the months which needs to be

```
df_train_site_9_meter_0=df_train_site_9.loc[df_train_site_9['meter']=='electricity']
```

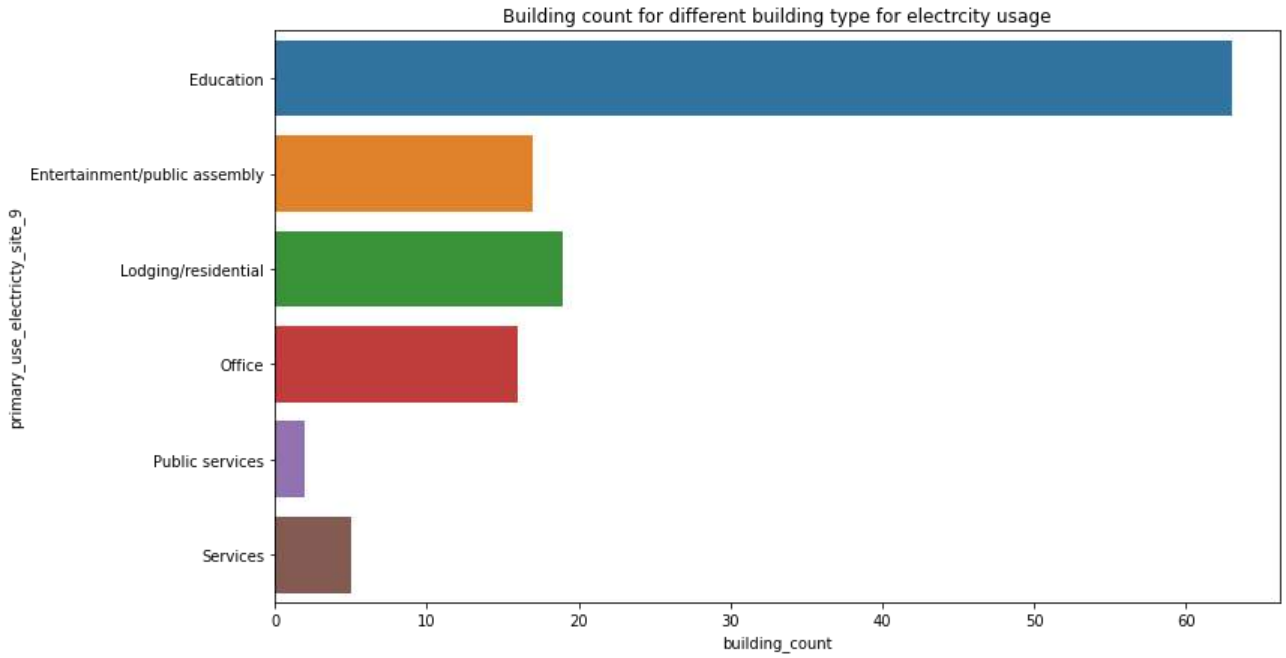
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_0
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('electricity_reading_site_9')
plt.title('Electrcity consumption vs Timestamp')
plt.show()
```



Now from the above plot we get a rough estimation of the electrcity usage for all the buildings over the timestamp.We can also observe the peaks similar to what we observed for steam and chilledwater readings.

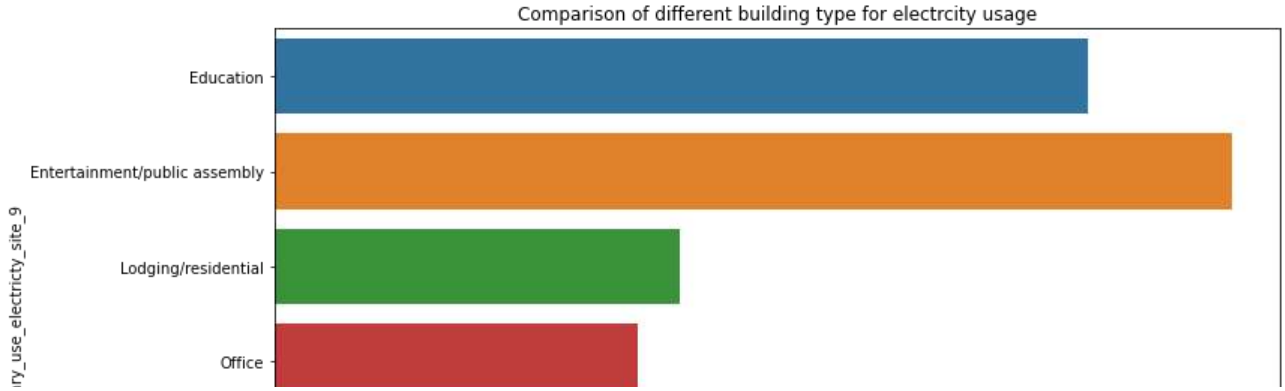
```
z=df_train_site_9_meter_0.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count')
plt.ylabel('primary_use_electricty_site_9')
```

```
plt.title('Building count for different building type for electrcity usage')
plt.show()
```



The above plot shows the building count for different building type for electrcity usage at site 9.Education is having the highest number of buildings which are consuming electricity.

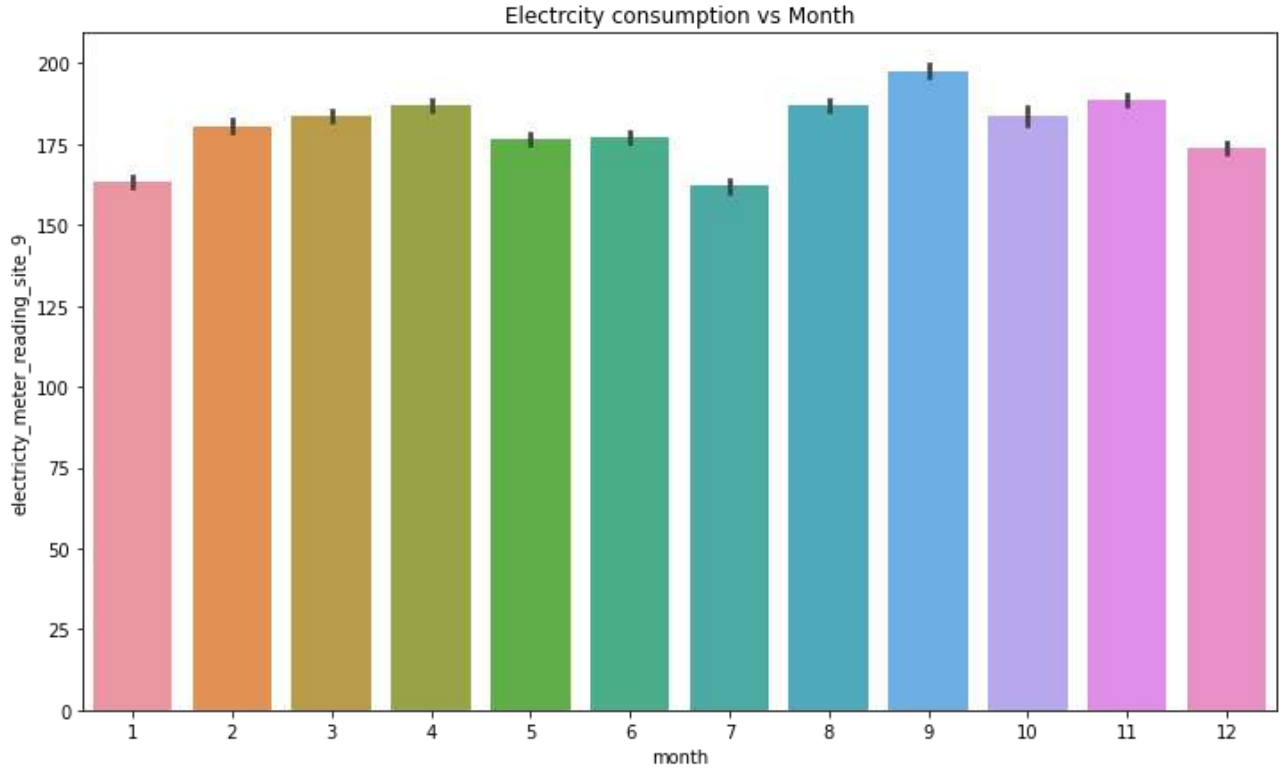
```
z=df_train_site_9_meter_0.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electrcity_reading')
plt.ylabel('primary_use_electricty_site_9')
plt.title('Comparison of different building type for electrcity usage')
plt.show()
```



From the above plot we can see that on an average entertainment and public services building which are lesser in number are consuming more electrcity than the educational buildings which are present in higher numbers as comapared to the other building.

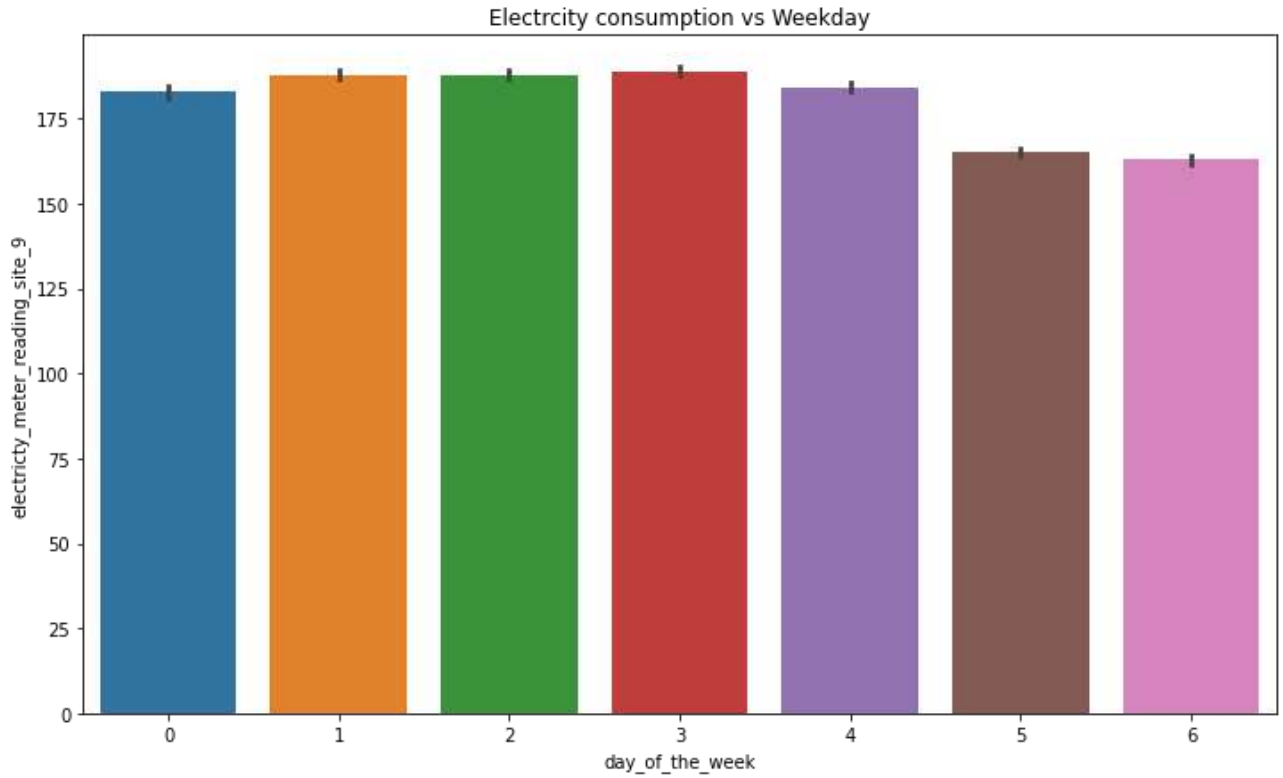
```
df_train_site_9_meter_0['month']=df_train_site_9_meter_0['timestamp'].dt.month
df_train_site_9_meter_0['weekday']=df_train_site_9_meter_0['timestamp'].dt.weekday
df_train_site_9_meter_0['hour']=df_train_site_9_meter_0['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricty_meter_reading_site_9')
plt.title('Electrcity consumption vs Month')
plt.show()
```



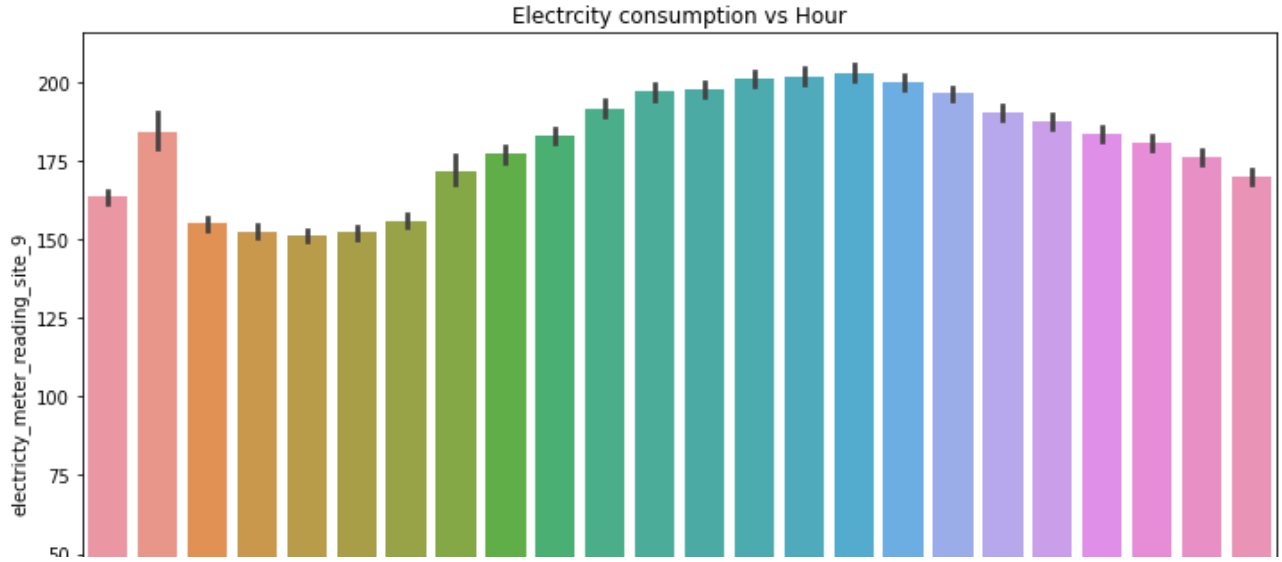
Here we can see that the elctrcity consumption varies differently according to the month.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricty_meter_reading_site_9')
plt.title('Electrcity consumption vs Weekday')
plt.show()
```



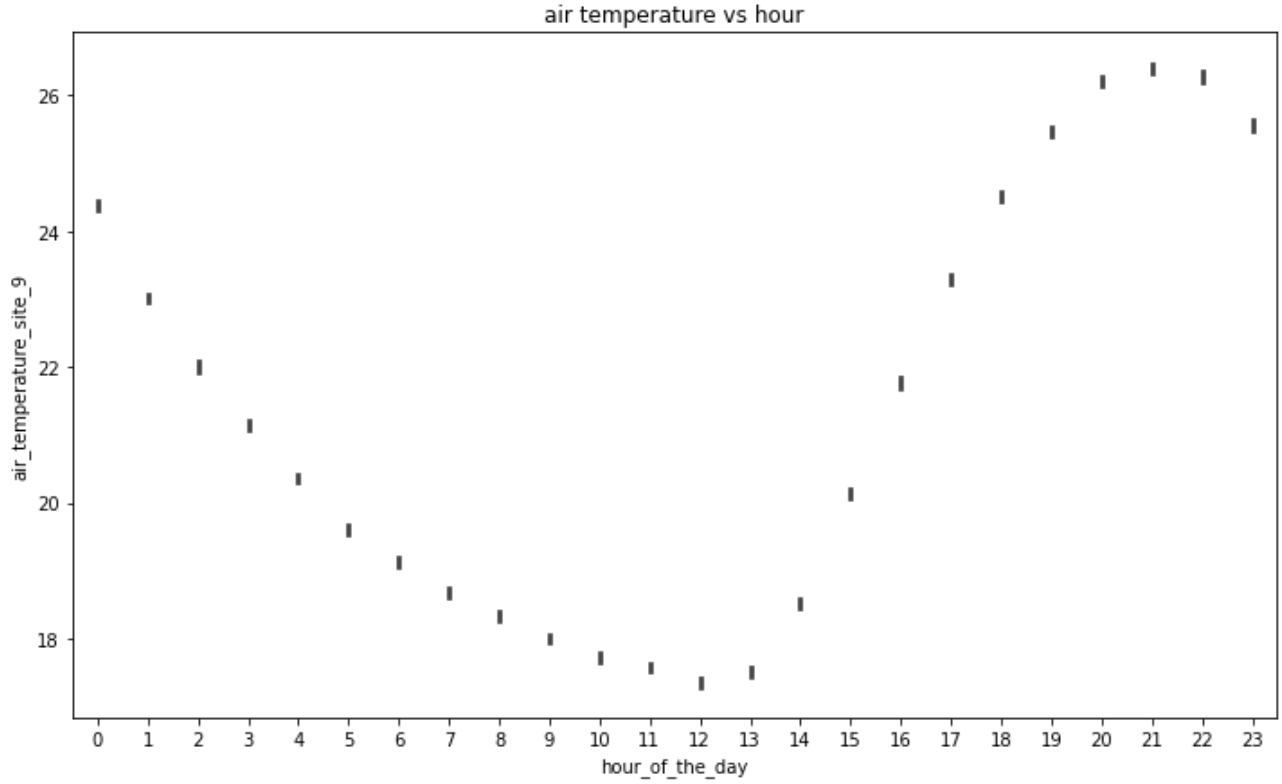
Electrcity consumption is more for the weekday as compared to the weekend.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricty_meter_reading_site_9')
plt.title('Electrcity consumption vs Hour')
plt.show()
```



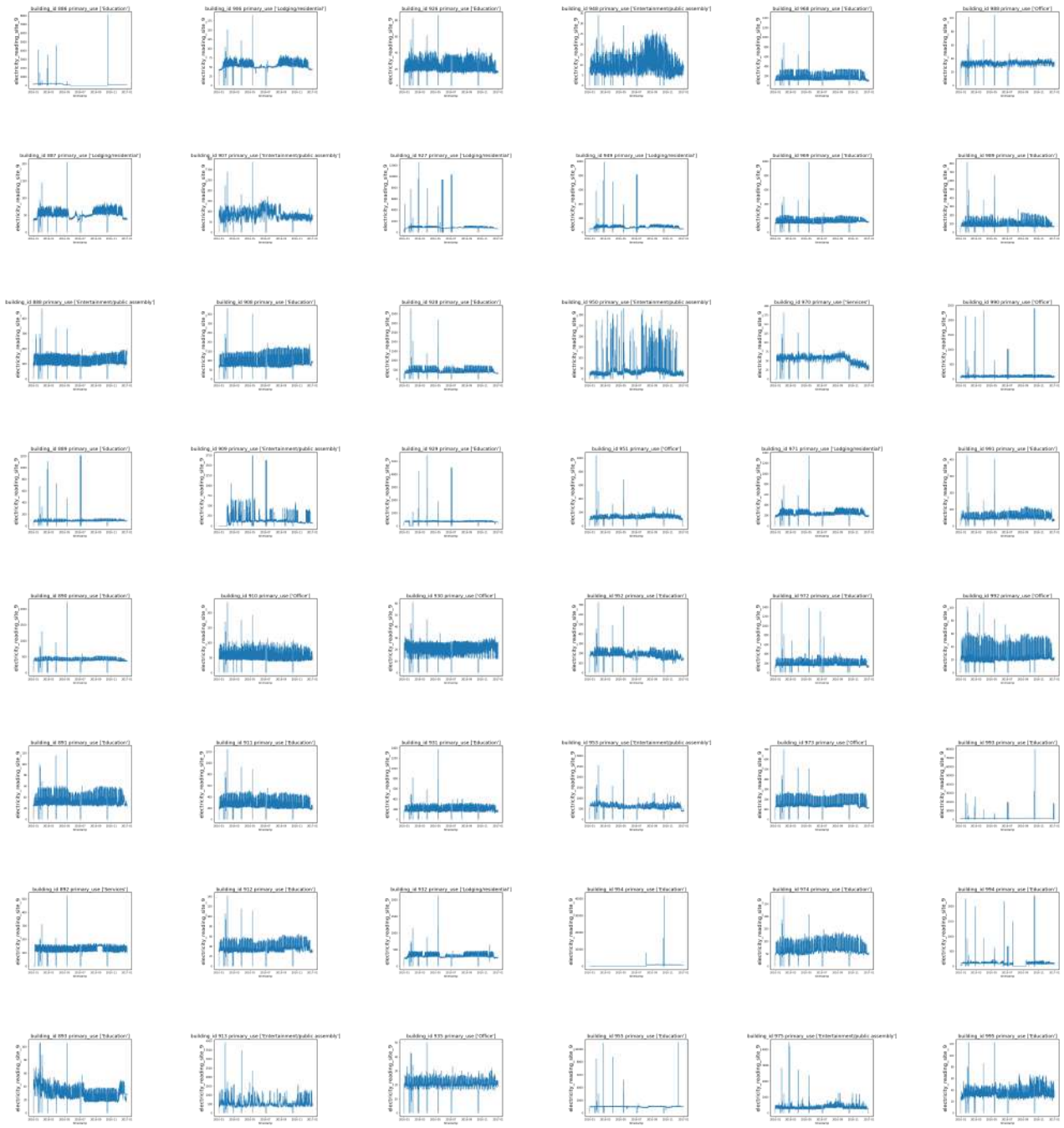
From the above plot we can observe that similar to the steam and chilledwater consumption it is also showing peak at 01:00 pm in the night.Now as regular the consumption odf electricity peaks during the afternoon hours and decreases gradually as we approach towards the night time

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_9_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_9')
plt.title('air temperature vs hour')
plt.show()
```



From the above plot we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings.Here the temperature peaks around 21:00 pm.

```
fig,axs=plt.subplots(figsize=(70,200),nrows=20,ncols=6,squeeze=True)
for i in range(120):
   g=df_train_site_9_meter_0['building_id'].unique()[i]
   axes=axs[i%20][i//20]
   z=df_train_site_9_meter_0.loc[df_train_site_9_meter_0['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('electricity_reading_site_9',fontsize=20)
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.9,wspace=0.8)
```

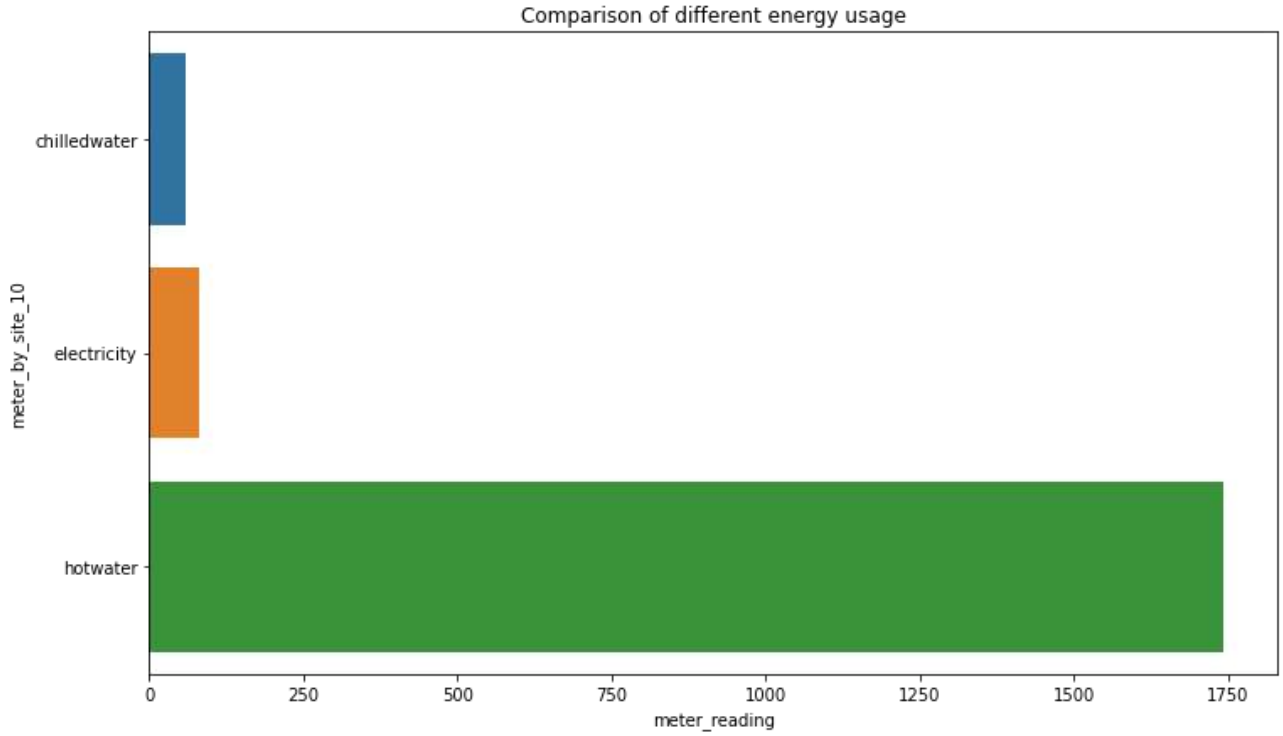


Important Observations

- Similar to the chilledwater and steam readings here the electricity readings are filled with spikes and constant zero readings which needs to be filtered out.

```
#Starting analysis for site 10


df_train_site_10=df_train_merge.loc[df_train_merge['site_id']==10]


z=df_train_site_10.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
```

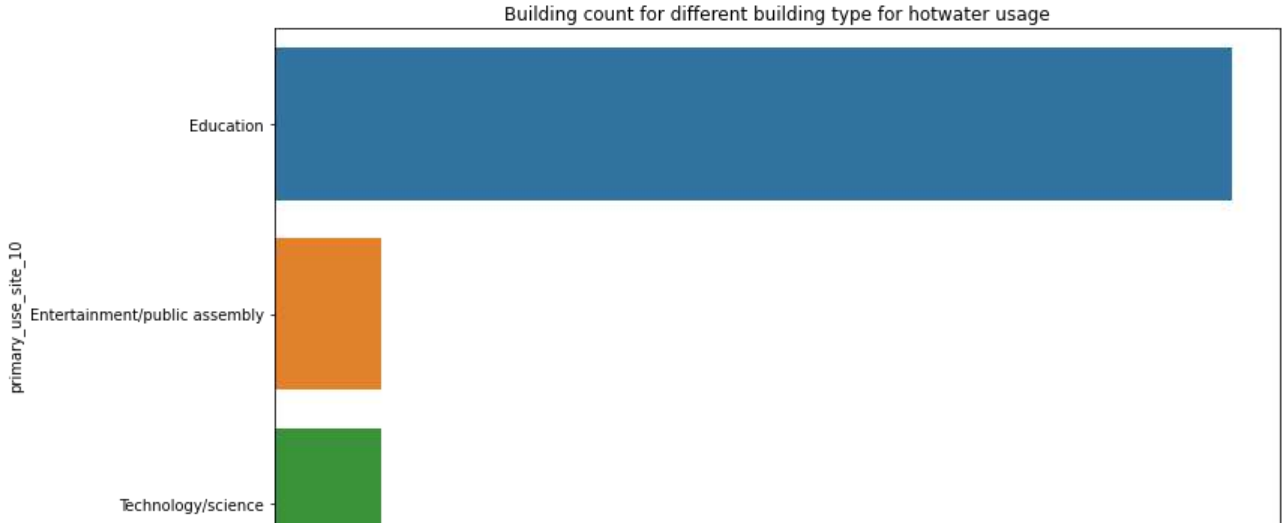
```
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('meter_reading')
plt.ylabel('meter_by_site_10')
plt.title('Comparison of different energy usage')
plt.show()
```



From site 10 we can observe that hotwater is having the highest energy consumption

```
df_train_site_10_meter_3=df_train_site_10.loc[df_train_site_10['meter']=='hotwater']
```

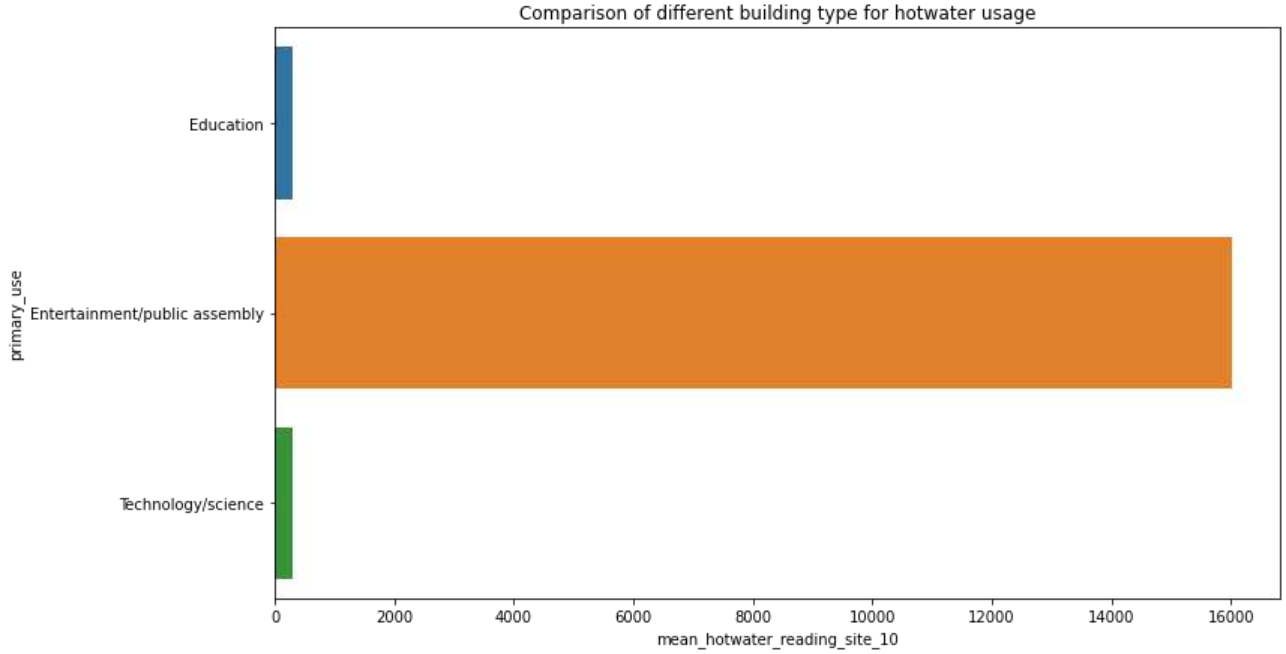
```
z=df_train_site_10_meter_3.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_hotwater_usage')
plt.ylabel('primary_use_site_10')
plt.title('Building count for different building type for hotwater usage')
plt.show()
```



The above plot represents the building count for different building type for hotwater usage

0 2 4 6 8

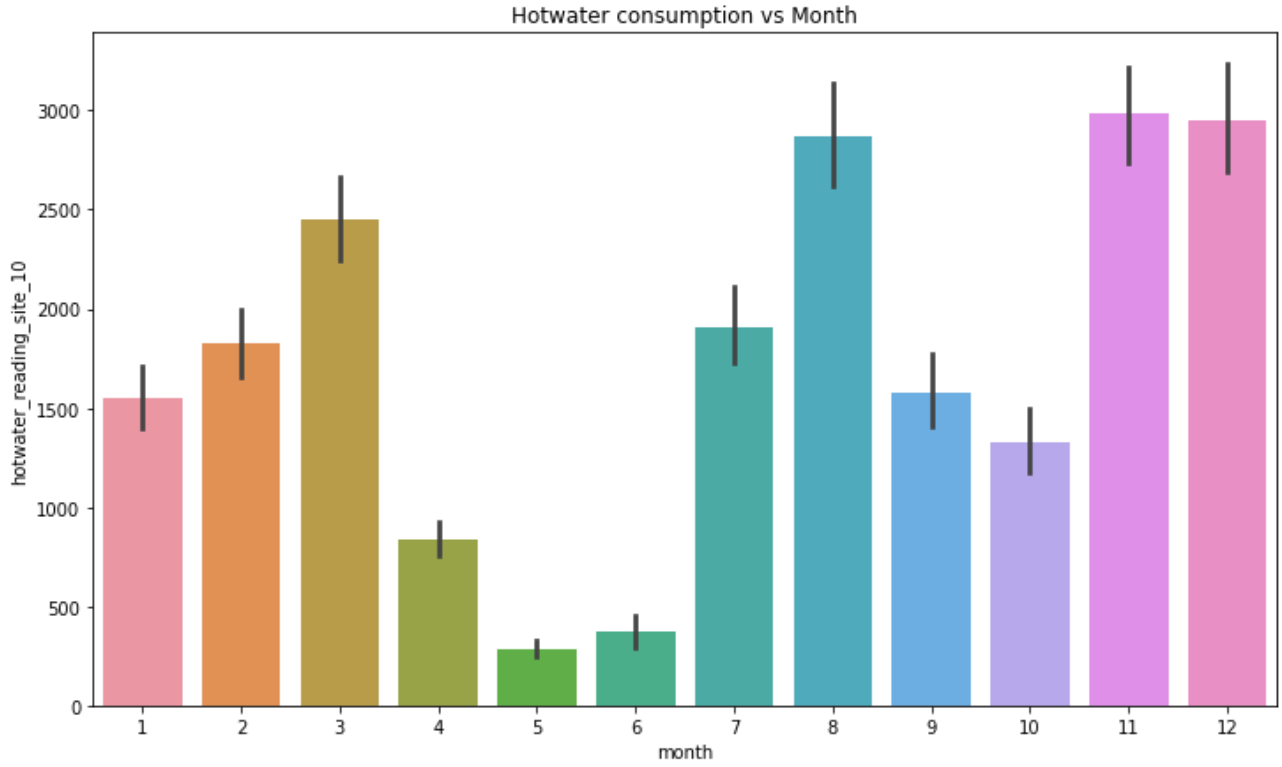
```
z=df_train_site_10_meter_3.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_hotwater_reading_site_10')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for hotwater usage')
plt.show()
```



From the above plot we can see that on an average entertainment type building is consuming the highest hotwater at site 9

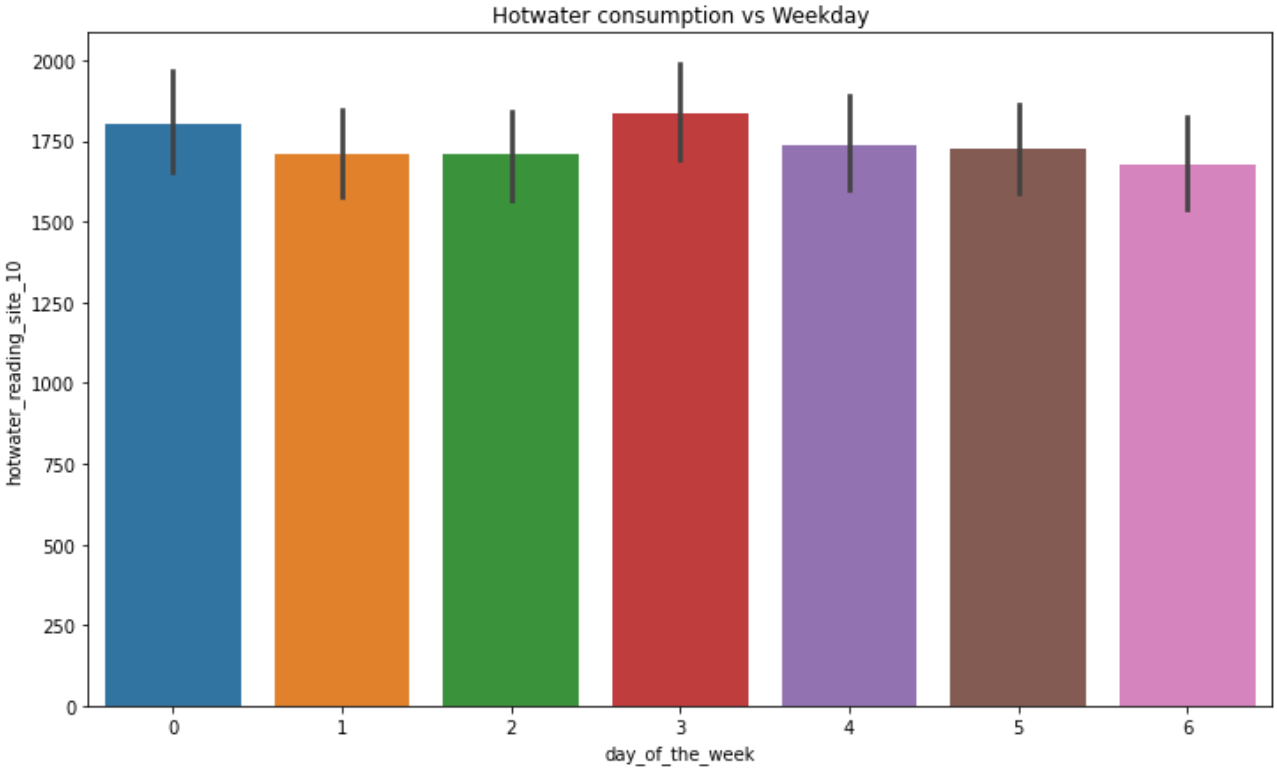
```
df_train_site_10_meter_3['month']=df_train_site_10_meter_3['timestamp'].dt.month
df_train_site_10_meter_3['weekday']=df_train_site_10_meter_3['timestamp'].dt.weekday
df_train_site_10_meter_3['hour']=df_train_site_10_meter_3['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_3
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('hotwater_reading_site_10')
plt.title('Hotwater consumption vs Month')
plt.show()
```



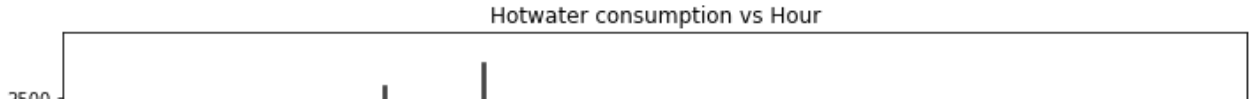
From the above plot we can see that hotwater consumption is showing very irregular usage over the month.We can observe that in the 8th month it is having higher consumption which is kind of strange.We can analyze this further when we plot the readings for all the buildings.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_3
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('hotwater_reading_site_10')
plt.title('Hotwater consumption vs Weekday')
plt.show()
```



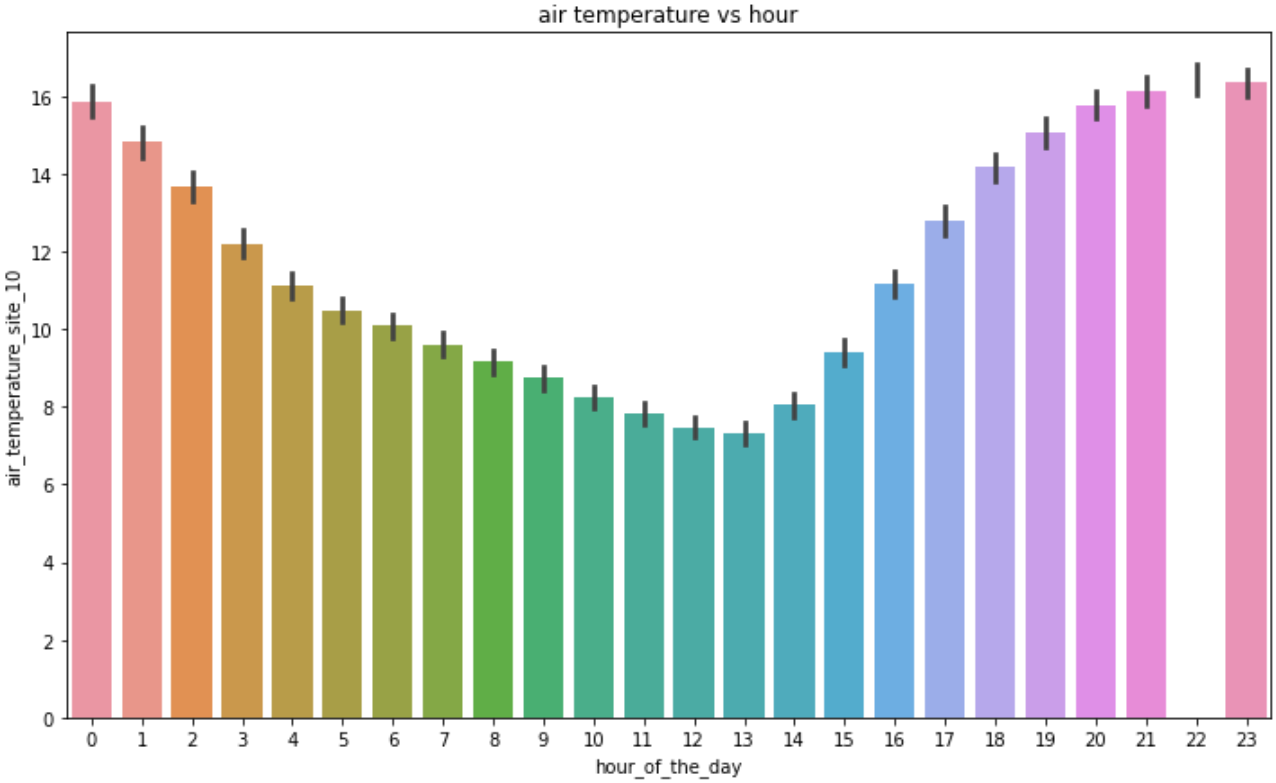
Hotwater consumption is having different variations over the week and we cannot see a specific pattern

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('hotwater_reading_site_10')
plt.title('Hotwater consumption vs Hour')
plt.show()
```



Hotwater consumption is having higher usage at the morning hours and we can also see that it gets higher usage at the night time

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_10')
plt.title('air temperature vs hour')
plt.show()
```



From the above plot we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings,The air temperature peaks around 22:00 pm

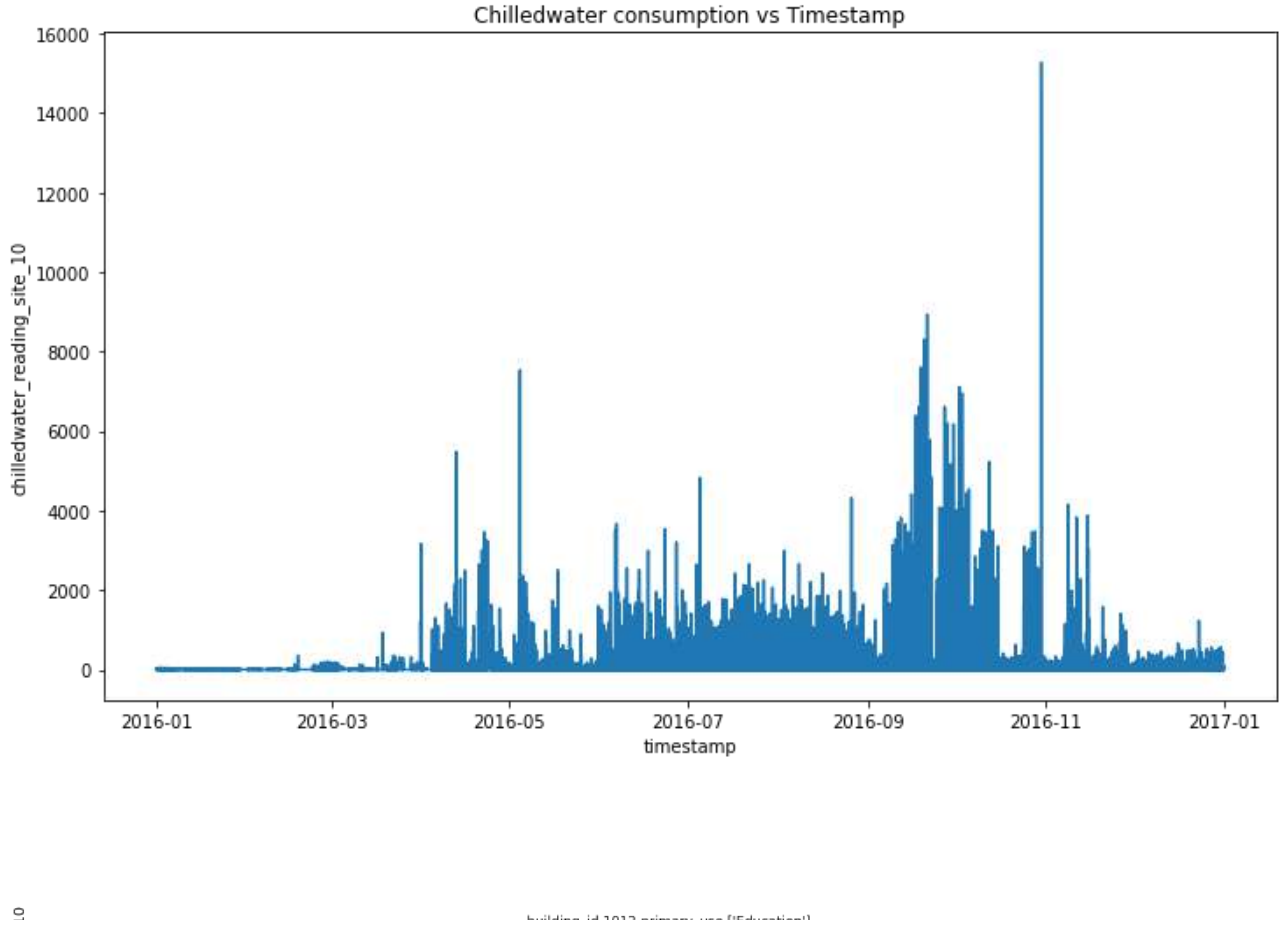
```
fig,axs=plt.subplots(figsize=(30,70),nrows=11,ncols=1,squeeze=False)
for i in range(df_train_site_10_meter_3['building_id'].nunique()):
   g=df_train_site_10_meter_3['building_id'].unique()[i]
   axes=axs[i%11][i//11]
   z=df_train_site_10_meter_3.loc[df_train_site_10_meter_3['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('hotwater_reading_site_10',fontsize=20)
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.9,wspace=0.8)
```

Important obseravations

- Building 1000,1003,1012,1017,1018 are showing higher spike for the last month which needs to be removed.
- Now the hotwater readings which we have observed for the 7th 8th and 9th month comes from the entertainment building 1021.

```
df_train_site_10_meter_1=df_train_site_10.loc[df_train_site_10['meter']=='chilledwater']
```

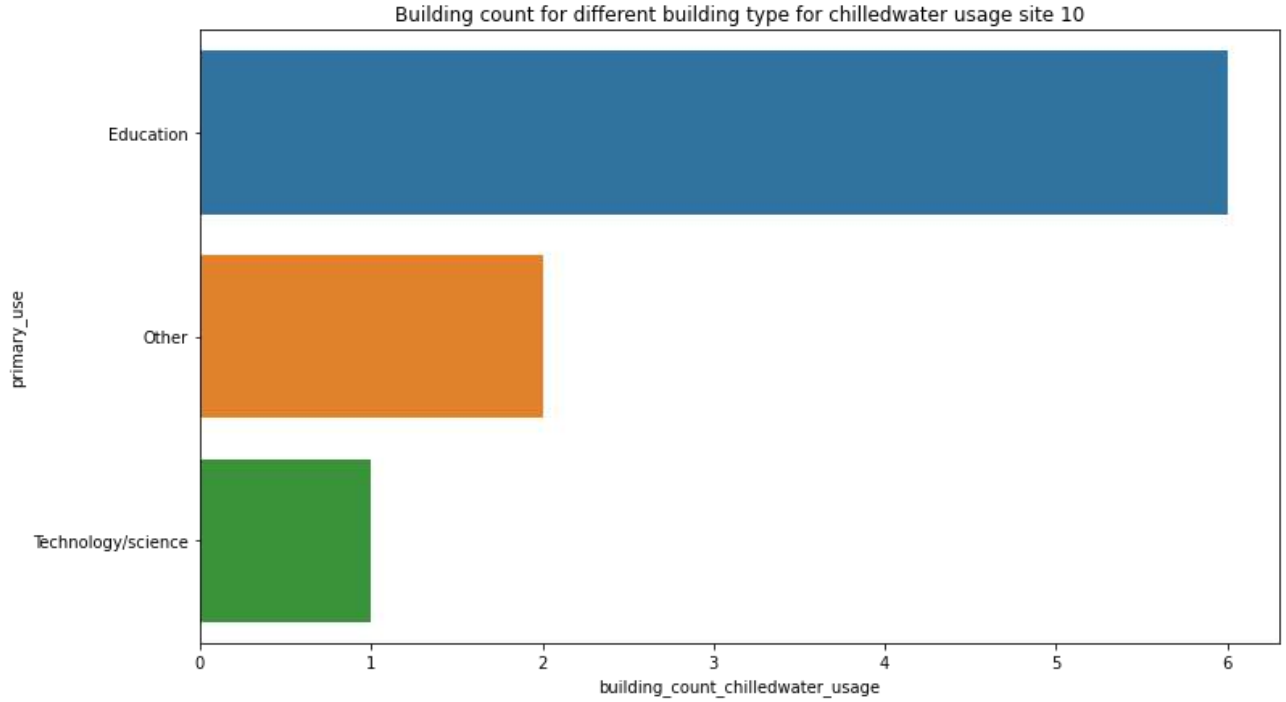
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_1
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('chilledwater_reading_site_10')
plt.title('Chilledwater consumption vs Timestamp')
plt.show()
```



The above plot shows the rough consumption of chilledwater of all the buildings over the timestamp

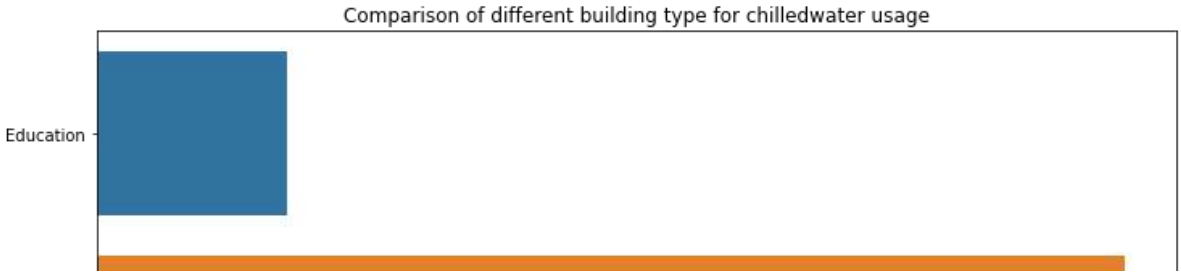
```
z=df_train_site_10_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_chilledwater_usage')
plt.ylabel('primary_use')
plt.title('Building count for different building type for chilledwater usage site 10')
plt.show()
```

```
plt.show()
```



This plot shows the building count for differebt building type for chilledwater usage

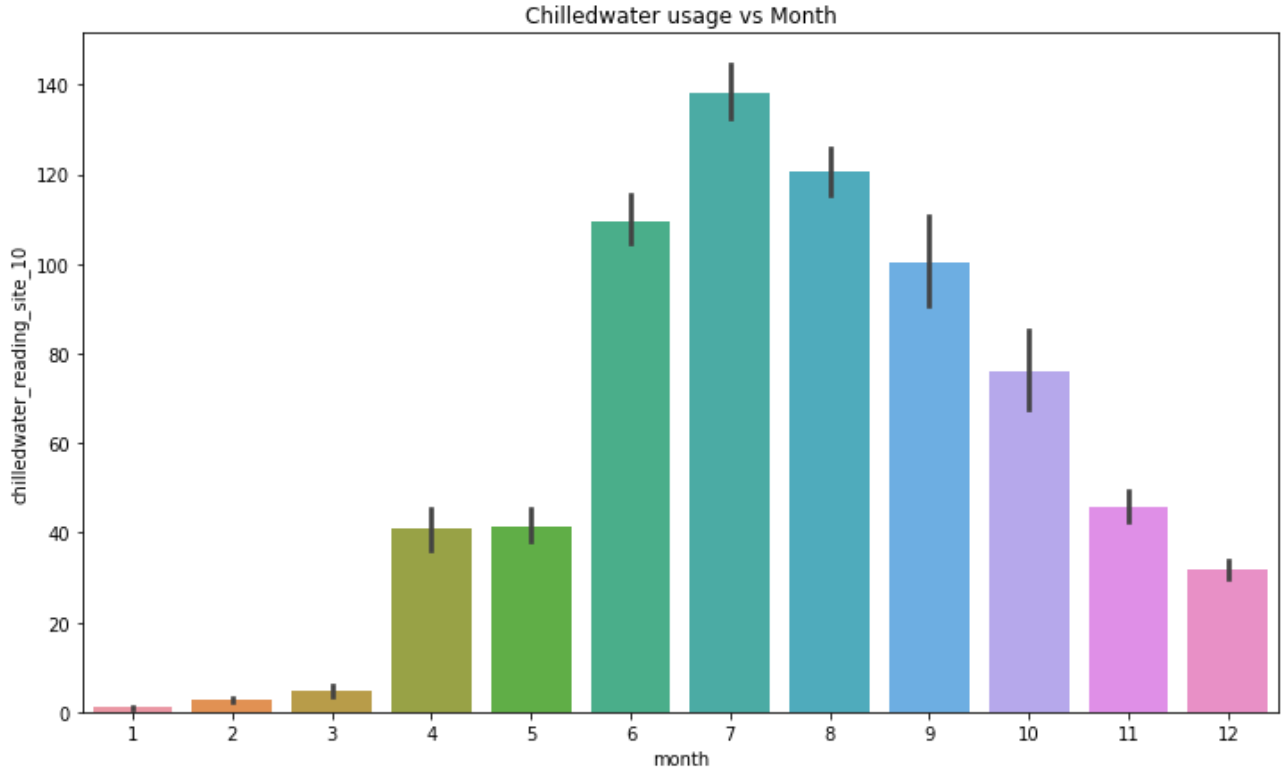
```
z=df_train_site_10_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading_site_10')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for chilledwater usage')
plt.show()
```



From the above plot we can see that on an average other building type is consuming highest chilledwater

```
df_train_site_10_meter_1['month']=df_train_site_10_meter_1['timestamp'].dt.month
df_train_site_10_meter_1['weekday']=df_train_site_10_meter_1['timestamp'].dt.weekday
df_train_site_10_meter_1['hour']=df_train_site_10_meter_1['timestamp'].dt.hour
```

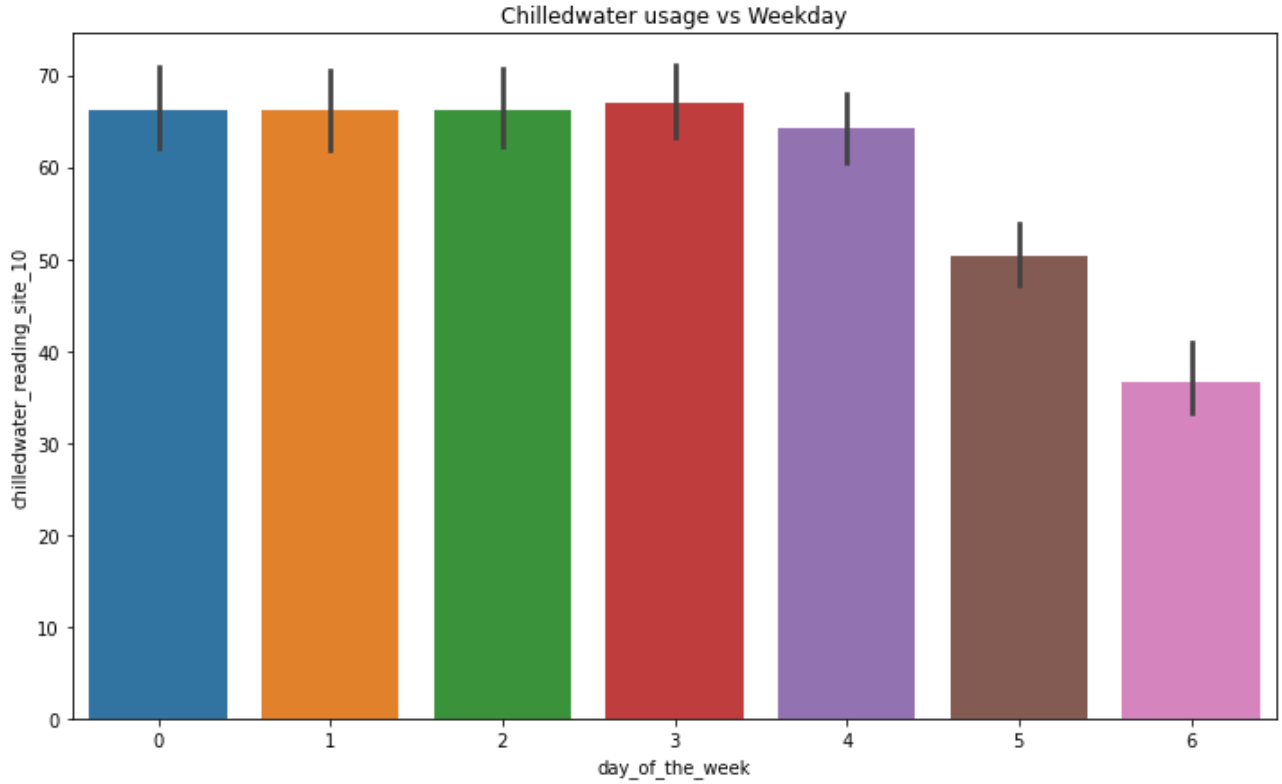
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_1
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading_site_10')
plt.title('Chilledwater usage vs Month')
plt.show()
```



From the above plot we can see that chilledwater usage shows higher consumption for the summer month

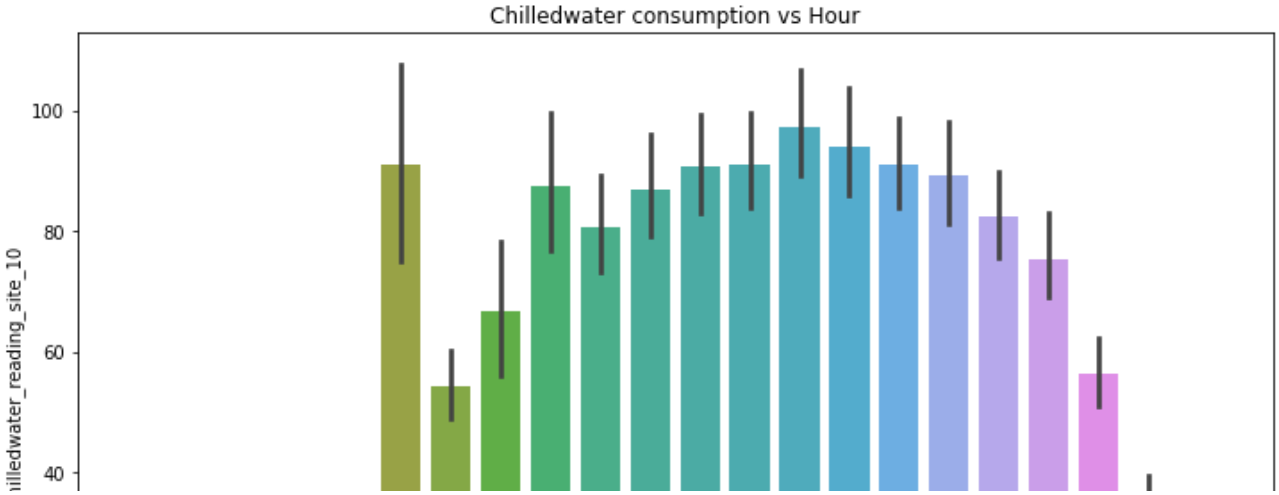
```
fig,ax=plt.subplots(figsize=(12,7))
```

```
z=df_train_site_10_meter_1
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading_site_10')
plt.title('Chilledwater usage vs Weekday')
plt.show()
```



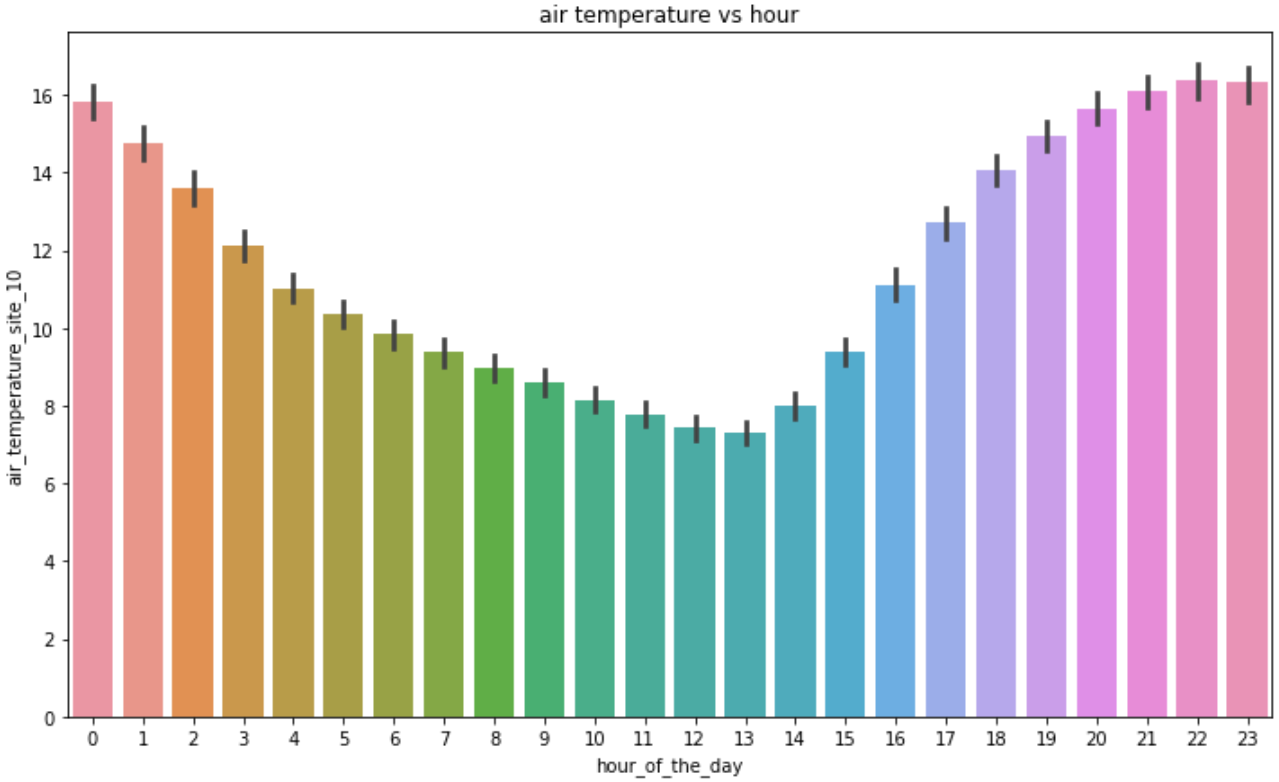
the above plot shows that chilledwater consumption is lesser over the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilledwater_reading_site_10')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```



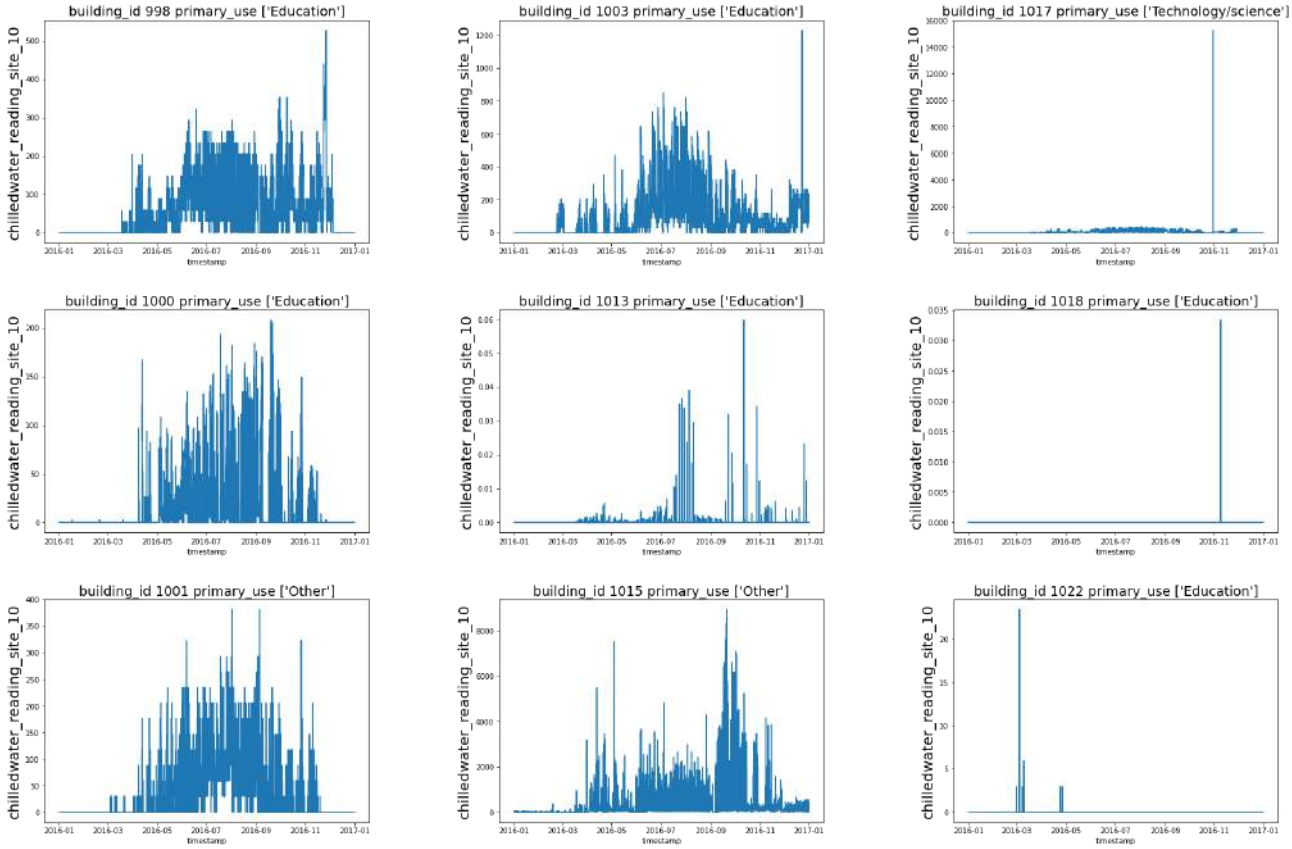
Chilledwater consumption suddenly spikes up at 6:00 am in the morning and then we can see that it is peaking during the afternoon and then decreases gradually

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_10')
plt.title('air temperature vs hour')
plt.show()
```



From the above plot we can see that the weather timestamp is not in alignment with the local timestamp with the hourly meter reading.here the temperature peaks around at 22:00 pm

```
fig,axs=plt.subplots(figsize=(30,20),nrows=3,ncols=3,squeeze=True)
for i in range(df_train_site_10_meter_1['building_id'].nunique()):
    g=df_train_site_10_meter_1['building_id'].unique()[i]
    axes=axs[i%3][i//3]
    z=df_train_site_10_meter_1.loc[df_train_site_10_meter_1['building_id']==g]
    axes.plot(z['timestamp'],z['meter_reading'])
    axes.set_xlabel('timestamp')
    axes.set_ylabel('chilledwater_reading_site_10',fontsize=20)
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
    plt.subplots_adjust(hspace=0.3,wspace=0.4)
```

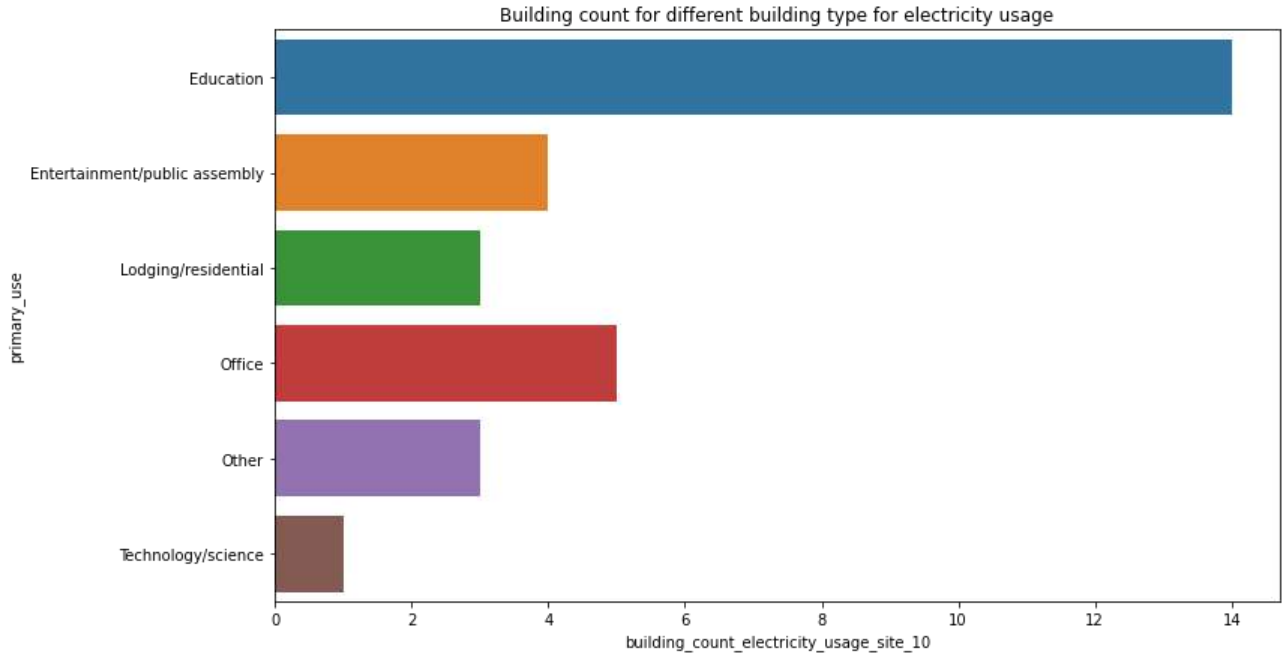


Important Observations

- Building 1018 1022 are showing constant zero meter readings over the whole timestamp so we can definitely drop these buildings.
- Building 1017 1003 are showing a high spike at the last month which needs to be filtered out.

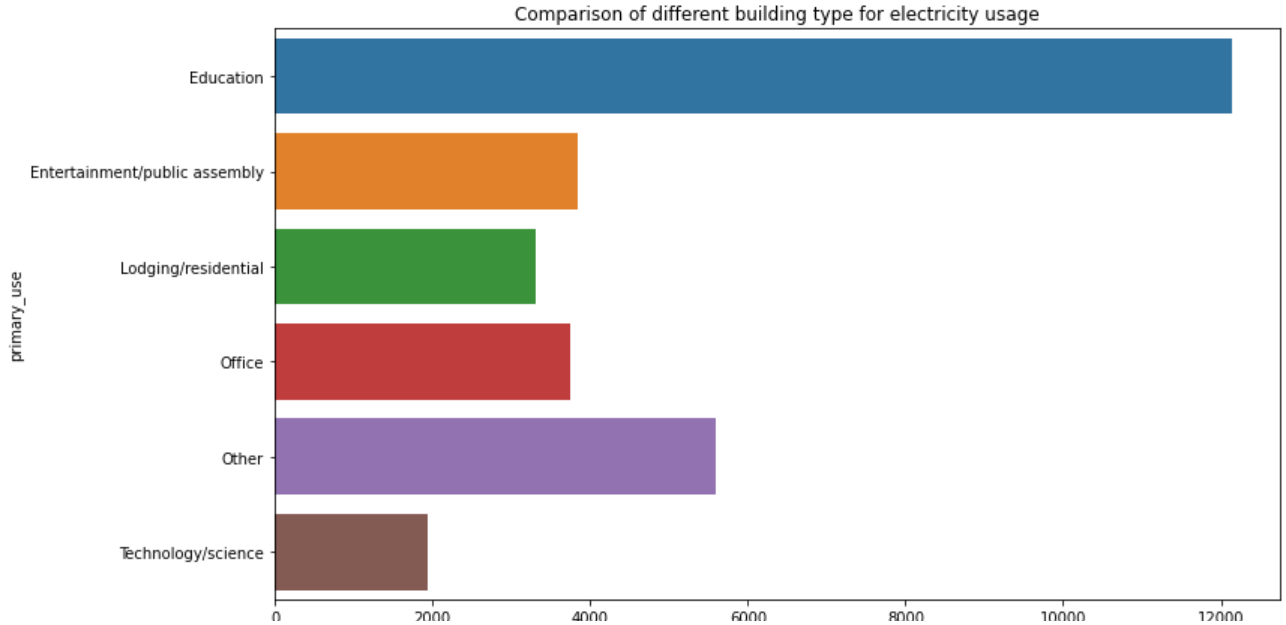
```
df_train_site_10_meter_0=df_train_site_10.loc[df_train_site_10['meter']=='electricity']
```

```
z=df_train_site_10_meter_0.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_electricity_usage_site_10')
plt.ylabel('primary_use')
plt.title('Building count for different building type for electricity usage')
plt.show()
```



The above plot shows the building count for different building type for electricity usage

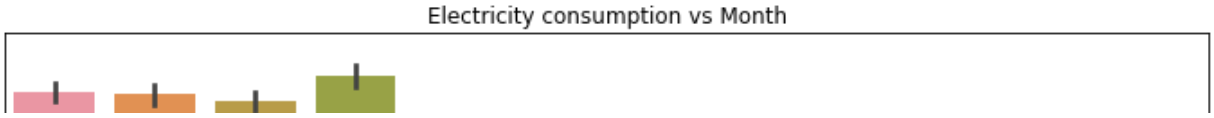
```
z=df_train_site_10_meter_0.groupby(['primary_use'])
z=z['meter_reading'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_usage_site_10')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for electricity usage')
plt.show()
```



From the above plot we can see that education is having the highest electricicty consumption at site 10

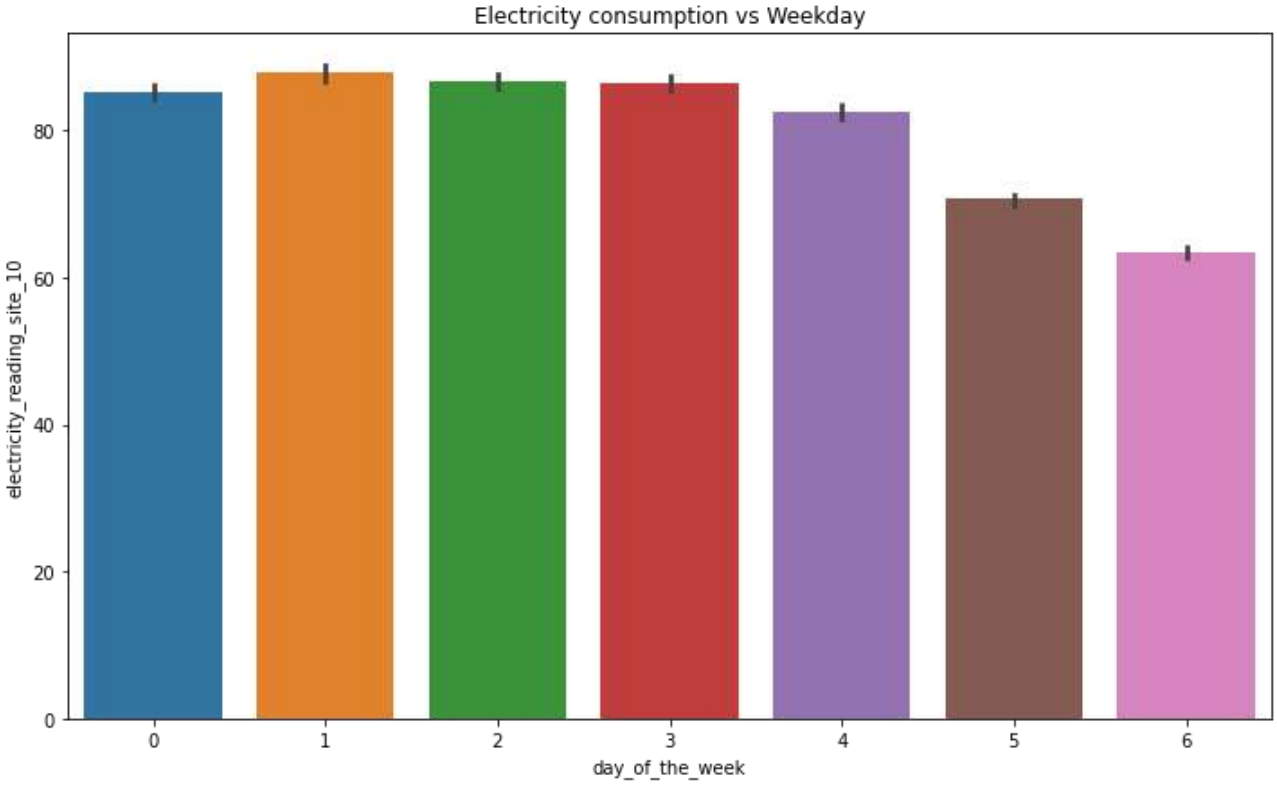
```
df_train_site_10_meter_0['month']=df_train_site_10_meter_0['timestamp'].dt.month
df_train_site_10_meter_0['weekday']=df_train_site_10_meter_0['timestamp'].dt.weekday
df_train_site_10_meter_0['hour']=df_train_site_10_meter_0['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_reading_site_10')
plt.title('Electricity consumption vs Month')
plt.show()
```



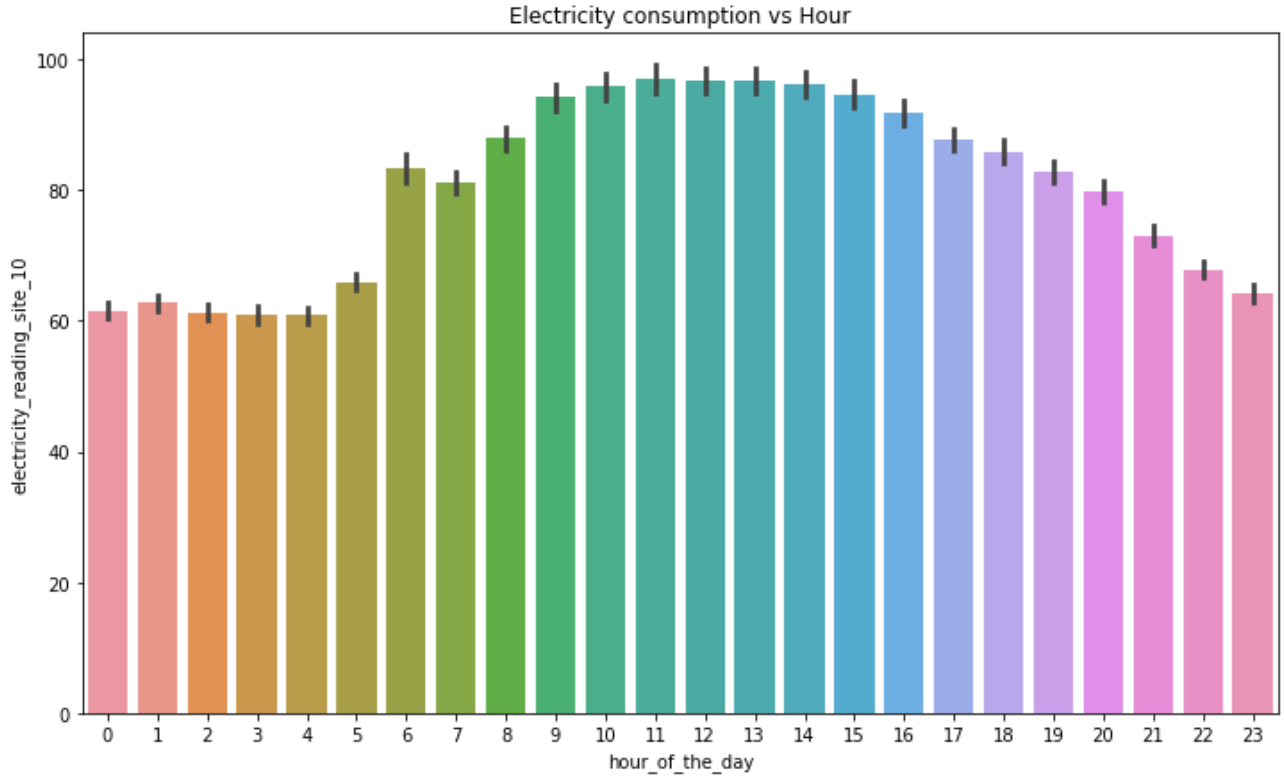
From the above plot we can see that the elctricity consumption is varying accordingly to the month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_10')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



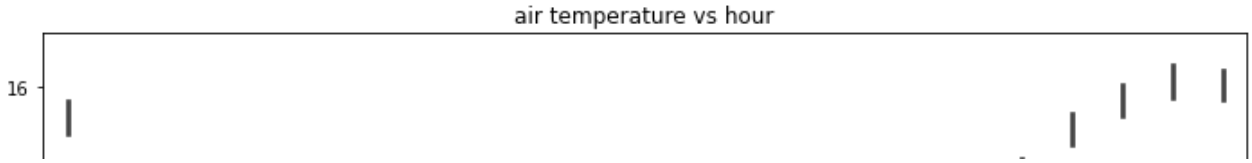
From the above plot we can see that the electricity consumption is less for the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_10')
plt.title('Electricity consumption vs Hour')
plt.show()
```



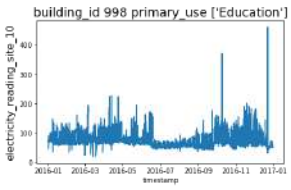
Now we can see that electricity consumption peaks at 13:00 pm and then starts falling over the hour of the day

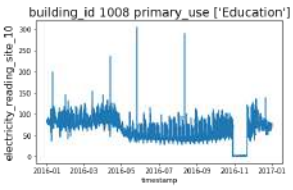
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_10')
plt.title('air temperature vs hour')
plt.show()
```

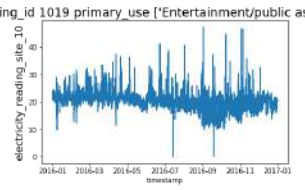


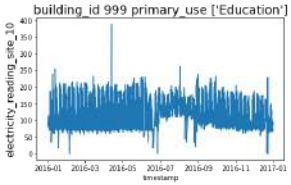
Here we can observe that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings.The air temperature peaks around 22:00 pm

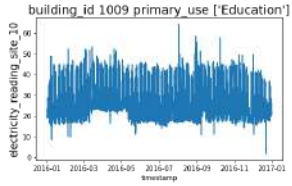
```
fig,axs=plt.subplots(figsize=(30,70),nrows=10,ncols=3,squeeze=True)
for i in range(df_train_site_10_meter_0['building_id'].nunique()):
   g=df_train_site_10_meter_0['building_id'].unique()[i]
   axes=axs[i%10][i//10]
   z=df_train_site_10_meter_0.loc[df_train_site_10_meter_0['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('electricity_reading_site_10',fontsize=15)
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.9,wspace=0.8)
```

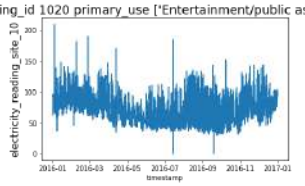


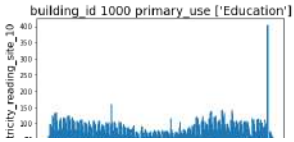


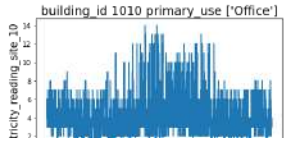


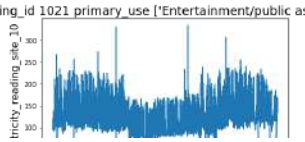












Important Observations

- Building 998 1000 1006 1019 1025 have readings which needs to be filtered out.

```
df_train_site_11=df_train_merge.loc[df_train_merge['site_id']==11]
```

```
df_train_site_11.isnull().sum()/df_train_site_11.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           1.00
floor_count          1.00
air_temperature      0.02
cloud_coverage       1.00
dew_temperature      0.02
precip_depth_1_hr    0.92
sea_level_pressure   0.02
wind_direction       0.02
wind_speed           0.02
dtype: float64
```

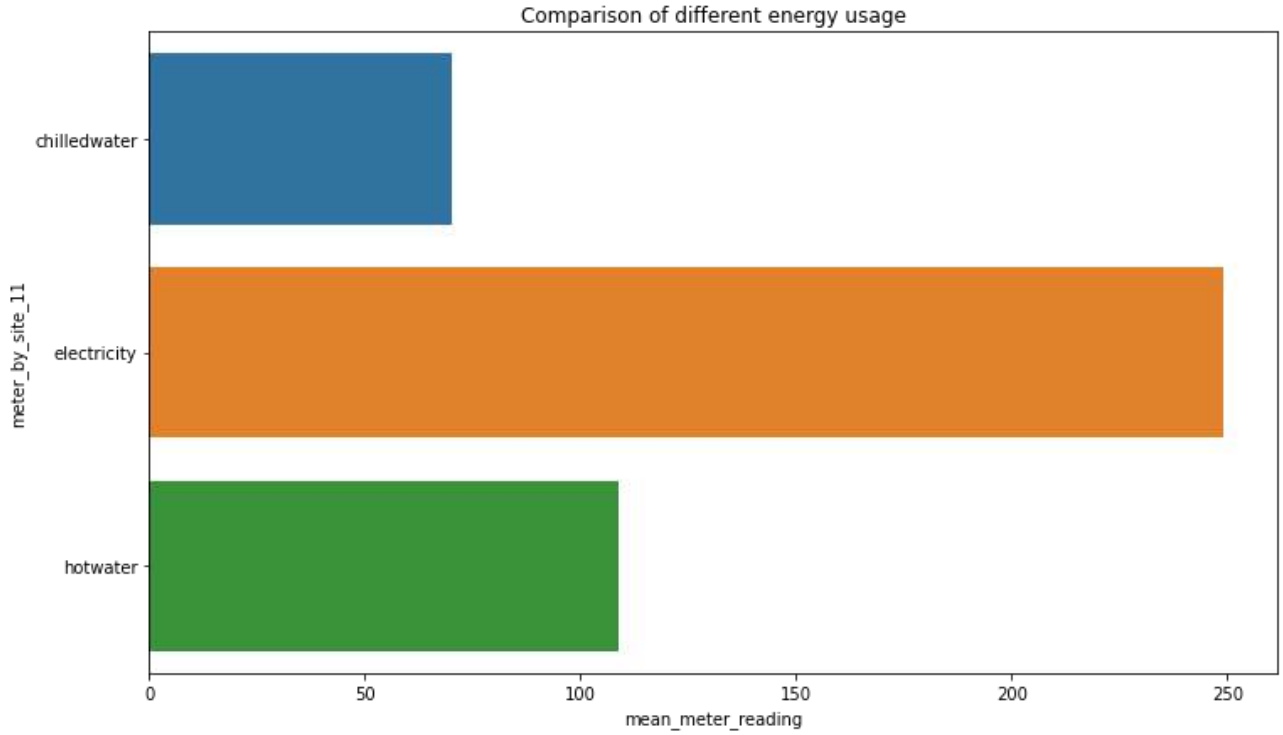
Here we can see the missing values which needs to be imputed at site 11

```
df_corr_11=df_train_site_11.corr()
df_corr_11.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.06	nan	0.24	nan	nan	0.01
meter_reading	0.06	1.00	nan	0.20	nan	nan	0.02
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.24	0.20	nan	1.00	nan	nan	0.01
year_built	nan	nan	nan	nan	nan	nan	nan
floor_count	nan	nan	nan	nan	nan	nan	nan
air_temperature	0.01	0.02	nan	0.01	nan	nan	1.00
cloud_coverage	nan	nan	nan	nan	nan	nan	nan
dew_temperature	0.01	0.04	nan	0.01	nan	nan	0.91
precip_depth_1_hr	0.00	0.01	nan	0.00	nan	nan	0.31
sea_level_pressure	-0.00	-0.00	nan	0.00	nan	nan	-0.21

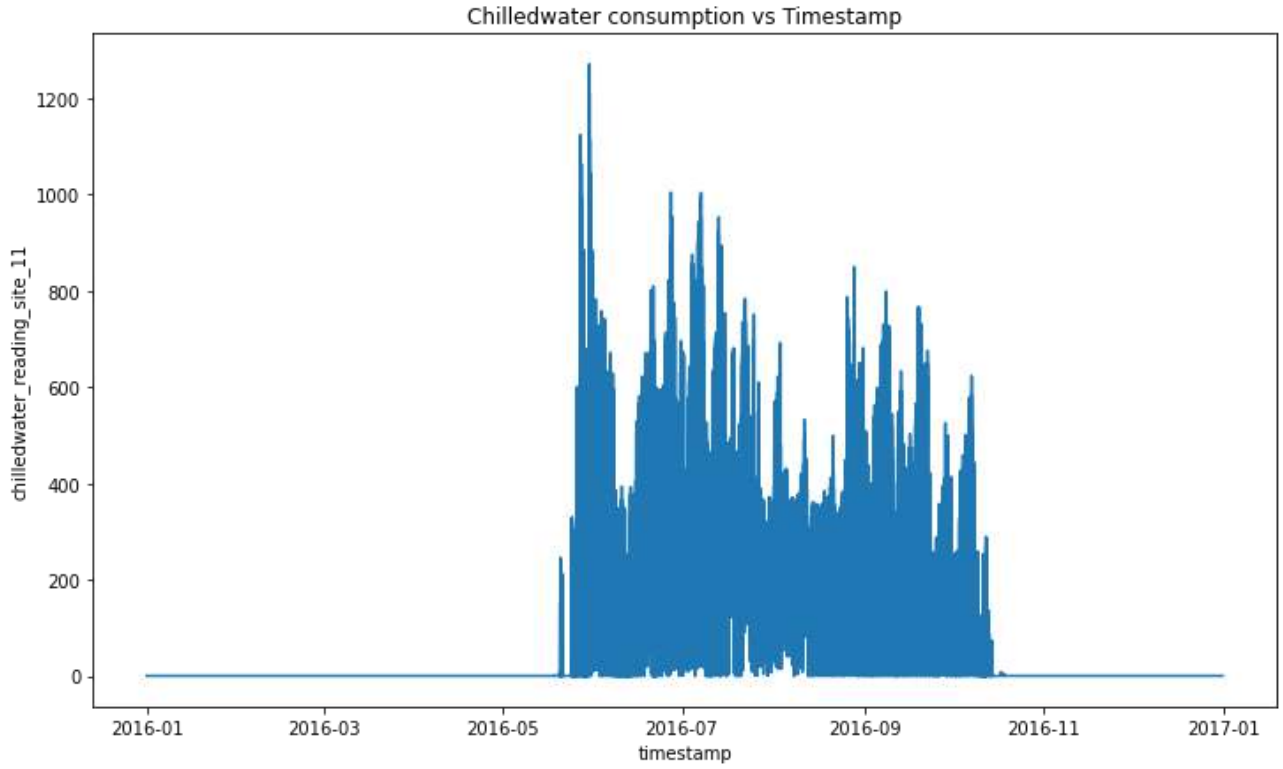
From the correlation plot we can see that the meter reading is not showing strong correlation with any of the features

```
z=df_train_site_11.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading')
plt.ylabel('meter_by_site_11')
plt.title('Comparison of different energy usage')
plt.show()
```



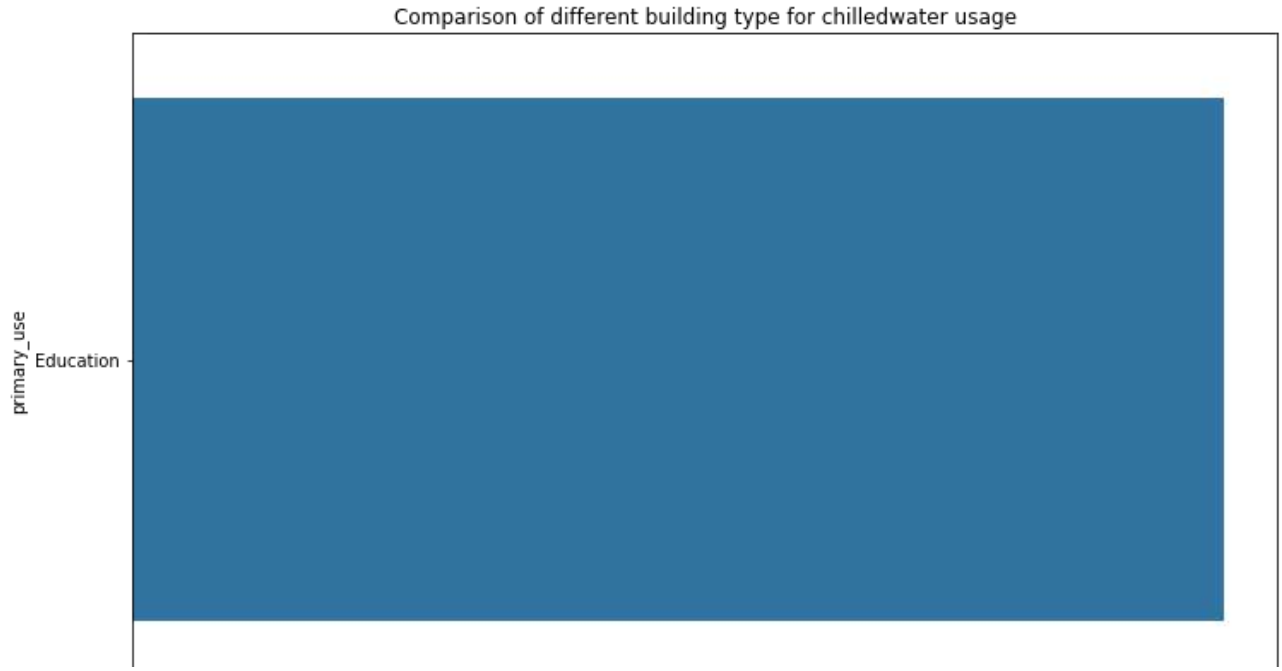
From the above plot we can see that electrcity is having the higher energy usage at site 11

```
df_train_site_11_meter_1=df_train_site_11.loc[df_train_site_11['meter']=='chilledwater']


fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_1
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('chilledwater_reading_site_11')
plt.title('Chilledwater consumption vs Timestamp')
plt.show()
```



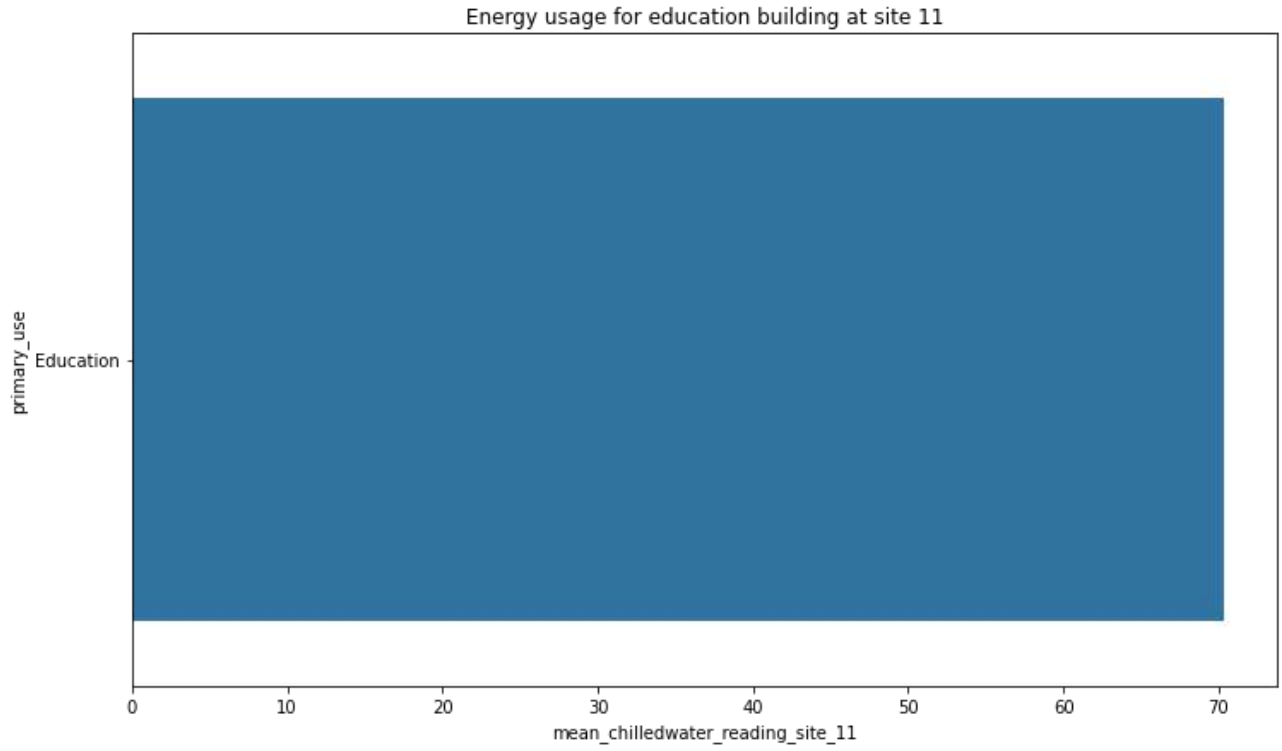
The above plot shows an overall chilledwater consumption for all the building over the timestamp

```
z=df_train_site_11_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_chilledwater_usage_site_11')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for chilledwater usage')
plt.show()
```



The above plot shows the count for different building type for chilledwater usage at site 11

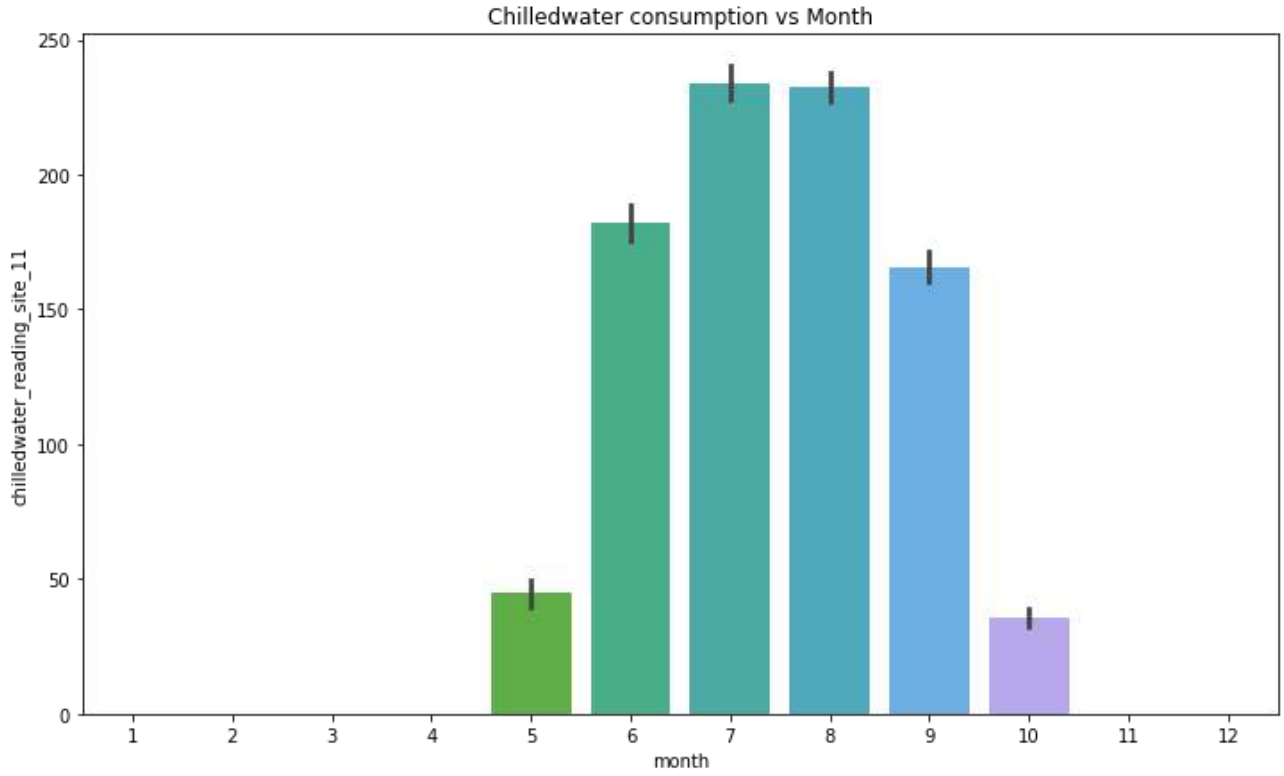
```
z=df_train_site_11_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading_site_11')
plt.ylabel('primary_use')
plt.title('Energy usage for education building at site 11')
plt.show()
```



This plot represents the energy consumption of the educational building at site 11

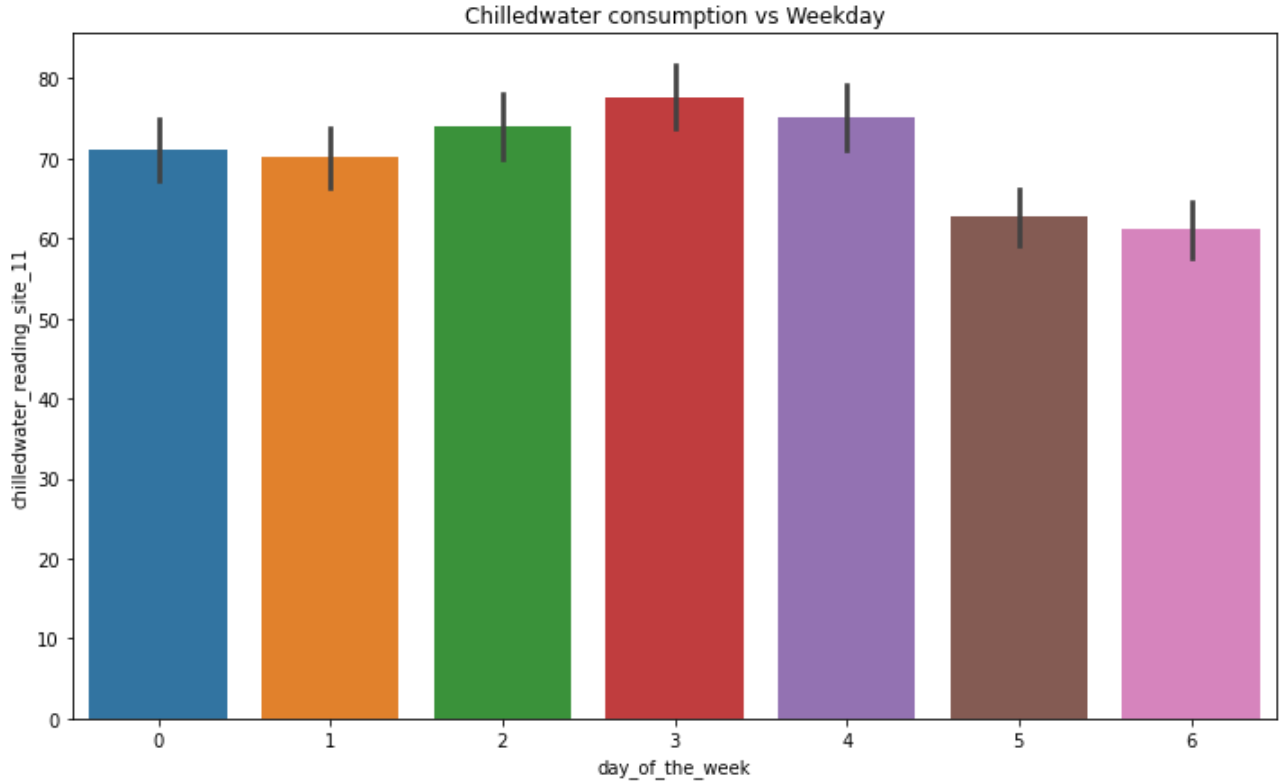
```
df_train_site_11_meter_1['month']=df_train_site_11_meter_1['timestamp'].dt.month
df_train_site_11_meter_1['weekday']=df_train_site_11_meter_1['timestamp'].dt.weekday
df_train_site_11_meter_1['hour']=df_train_site_11_meter_1['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_1
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading_site_11')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



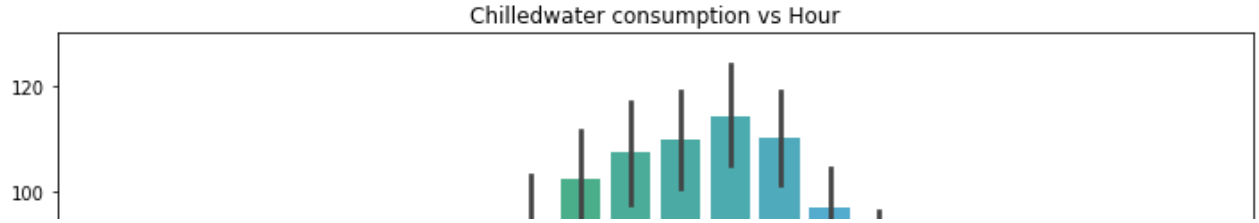
The abov eplot shows the chilledwater consumption over the month and it shows that the consumption is highly over the summer month and shows zero readings for the winter and transition month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_1
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading_site_11')
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



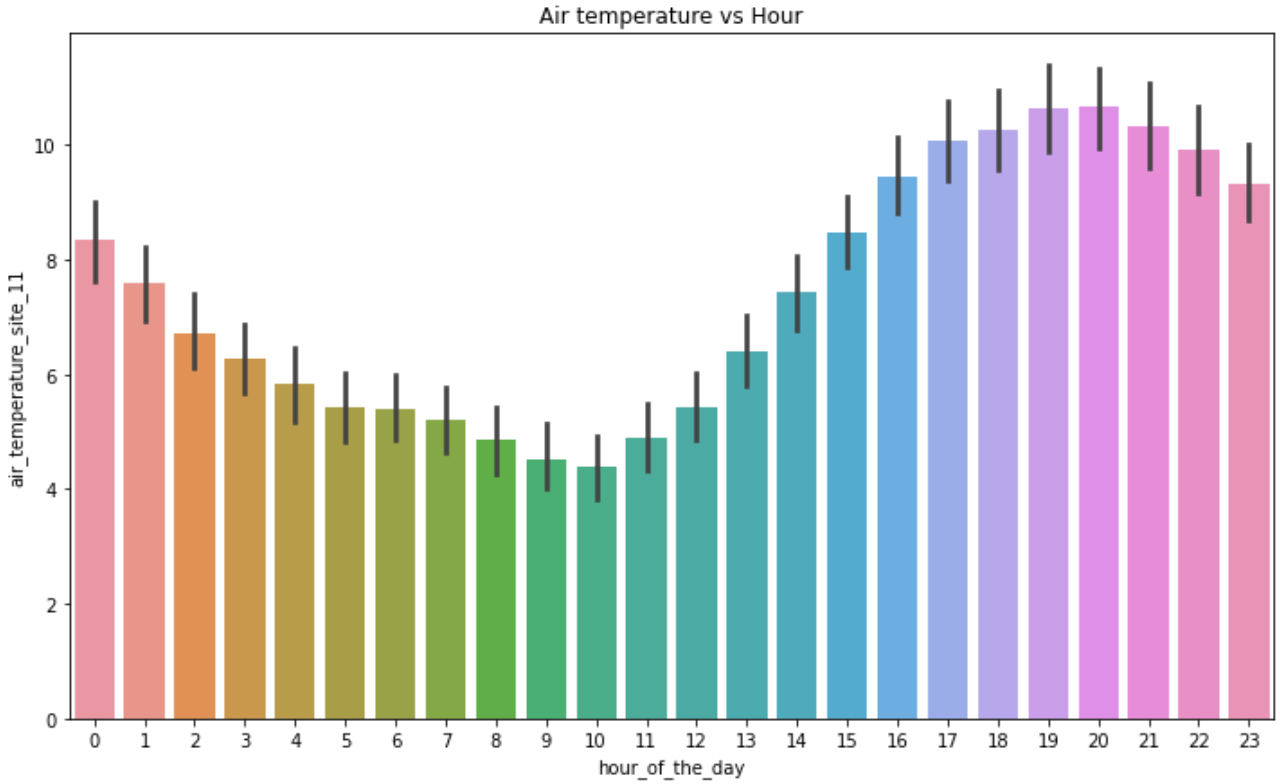
The above plot shows that the chilledwater consumption is lesser on the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilledwater_reading_site_11')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```



Chilledwater consumption starts increasing from 6:00 am in the morning and peaks around 13:00 pm and thenstarts to increase gradually

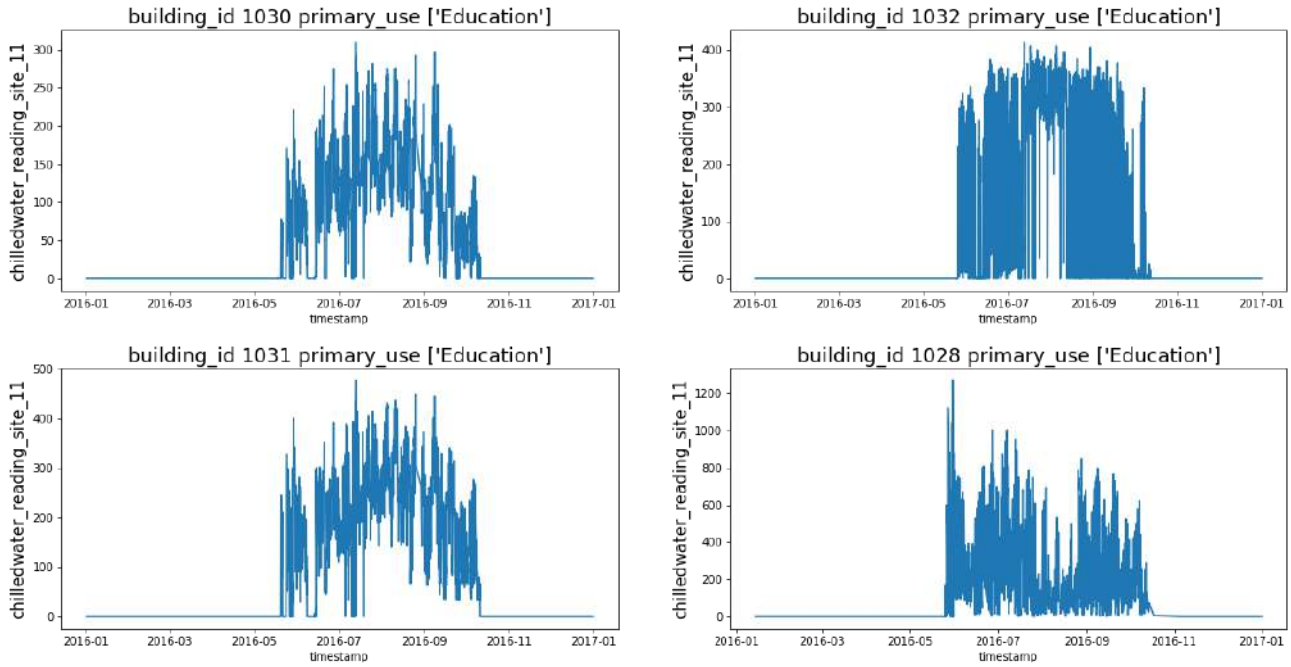
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_11')
plt.title('Air temperature vs Hour')
plt.show()
```



From the above plot we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter reading as the temperature peaks around 20:00 pm

```
fig,axs=plt.subplots(figsize=(20,10),nrows=2,ncols=2,squeeze=True)
for i in range(df_train_site_11_meter_1['building_id'].nunique()):
   g=df_train_site_11_meter_1['building_id'].unique()[i]
   axes=axs[i%2][i//2]
   z=df_train_site_11_meter_1.loc[df_train_site_11_meter_1['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
```

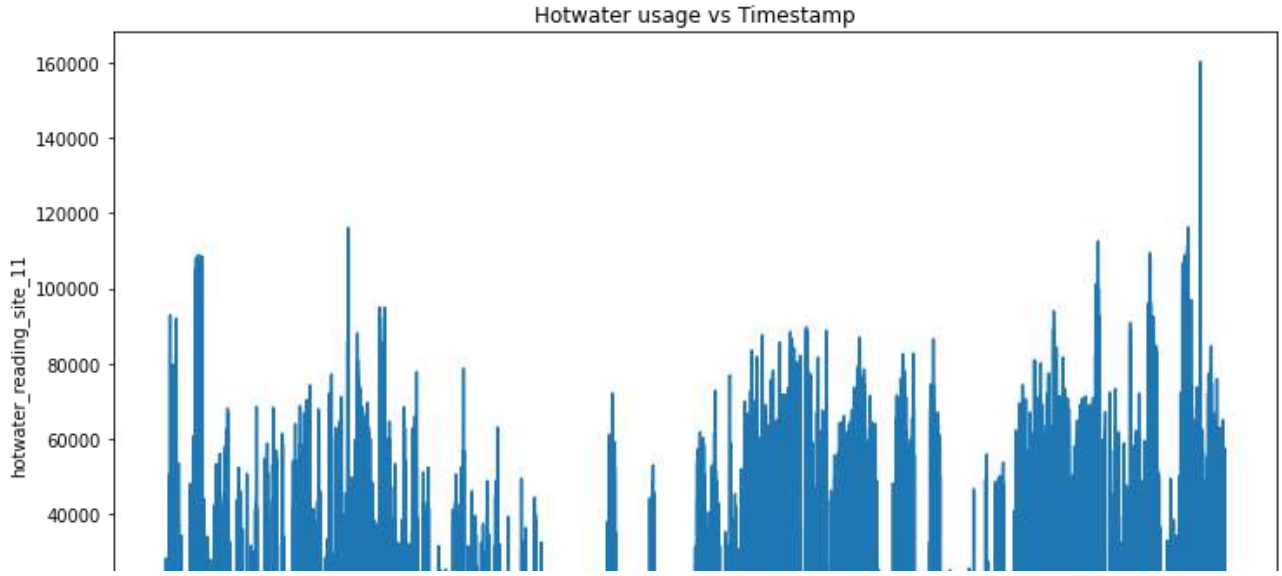
```
axes.set_ylabel('chilledwater_reading_site_11',fontsize=15)
axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
plt.subplots_adjust(hspace=0.3,wspace=0.2)
```



Here we do not need to remove the zero readings as our model needs to learn the seasonal transitions

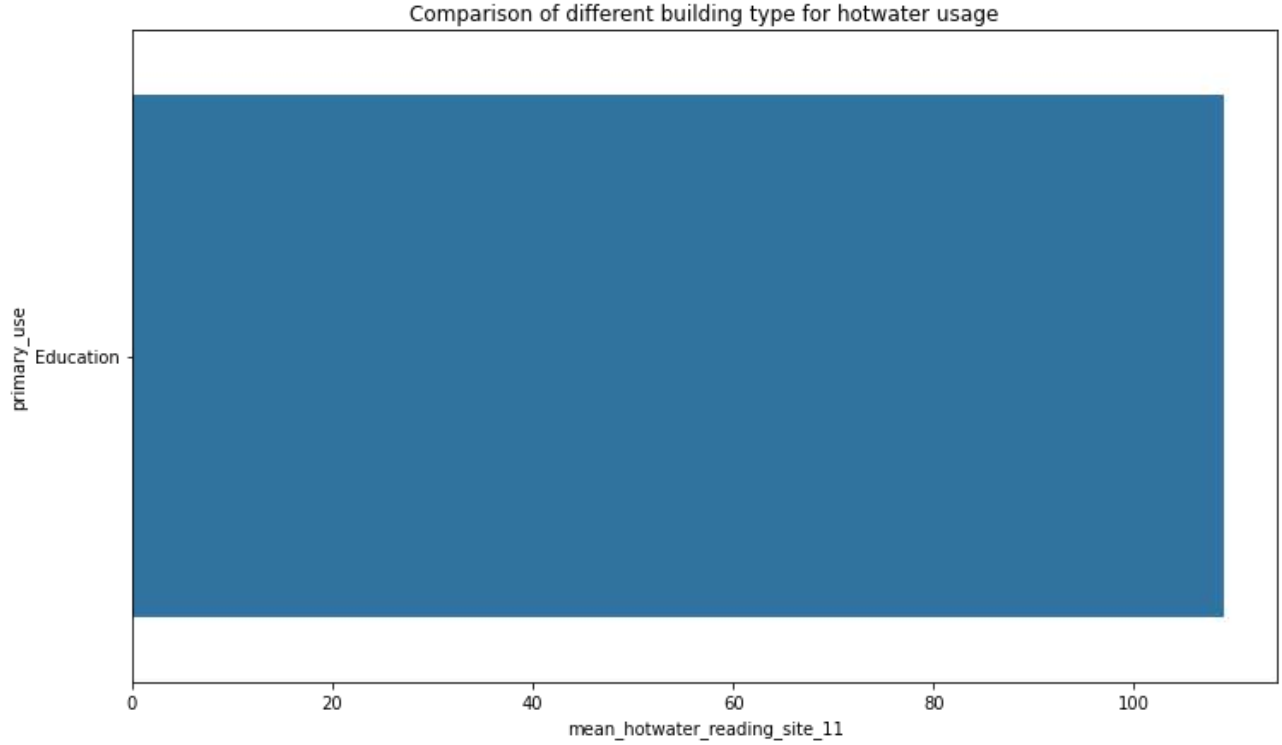
```
df_train_site_11_meter_3=df_train_site_11.loc[df_train_site_11['meter']=='hotwater']
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_10_meter_3
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('hotwater_reading_site_11')
plt.title('Hotwater usage vs Timestamp')
plt.show()
```



This plot shows the rough approximation of all the buildings for hotwater usage over the timestamp

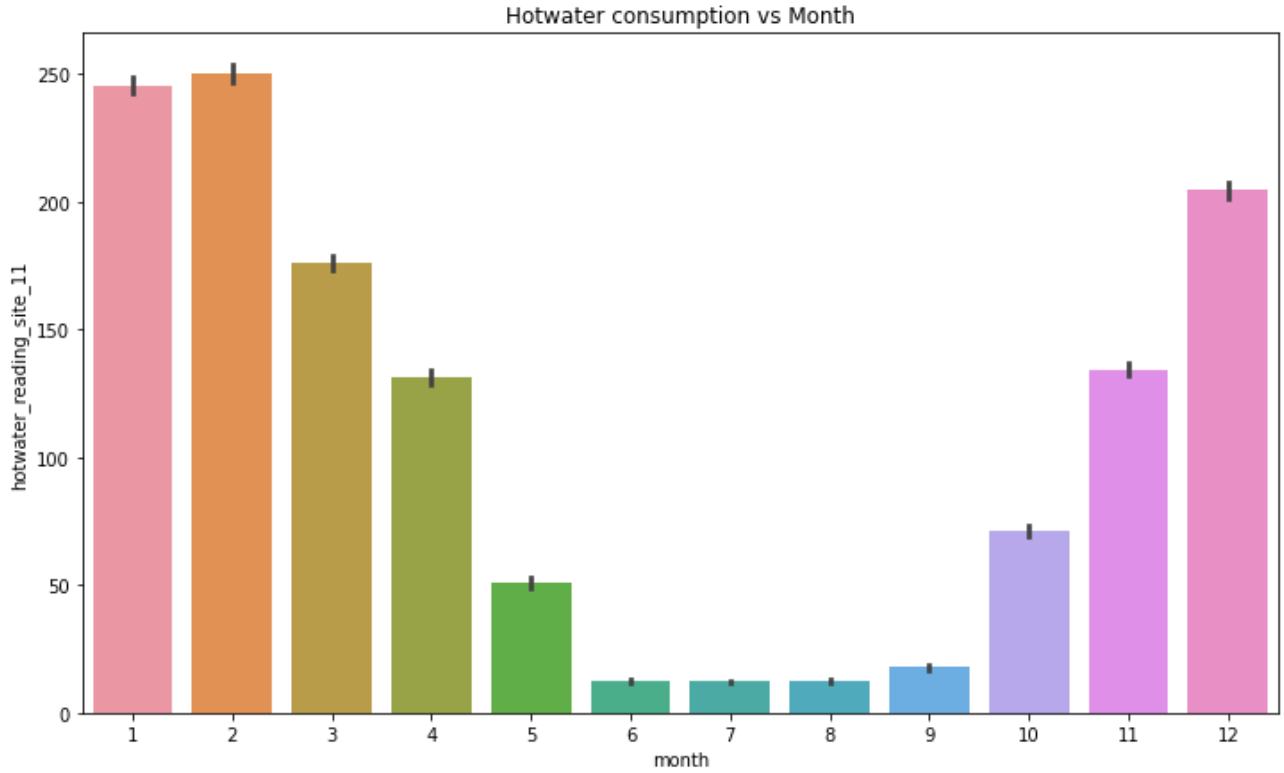
```
z=df_train_site_11_meter_3.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_hotwater_reading_site_11')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for hotwater usage')
plt.show()
```



At site 11 we can see that only educational building is consumong hotwater

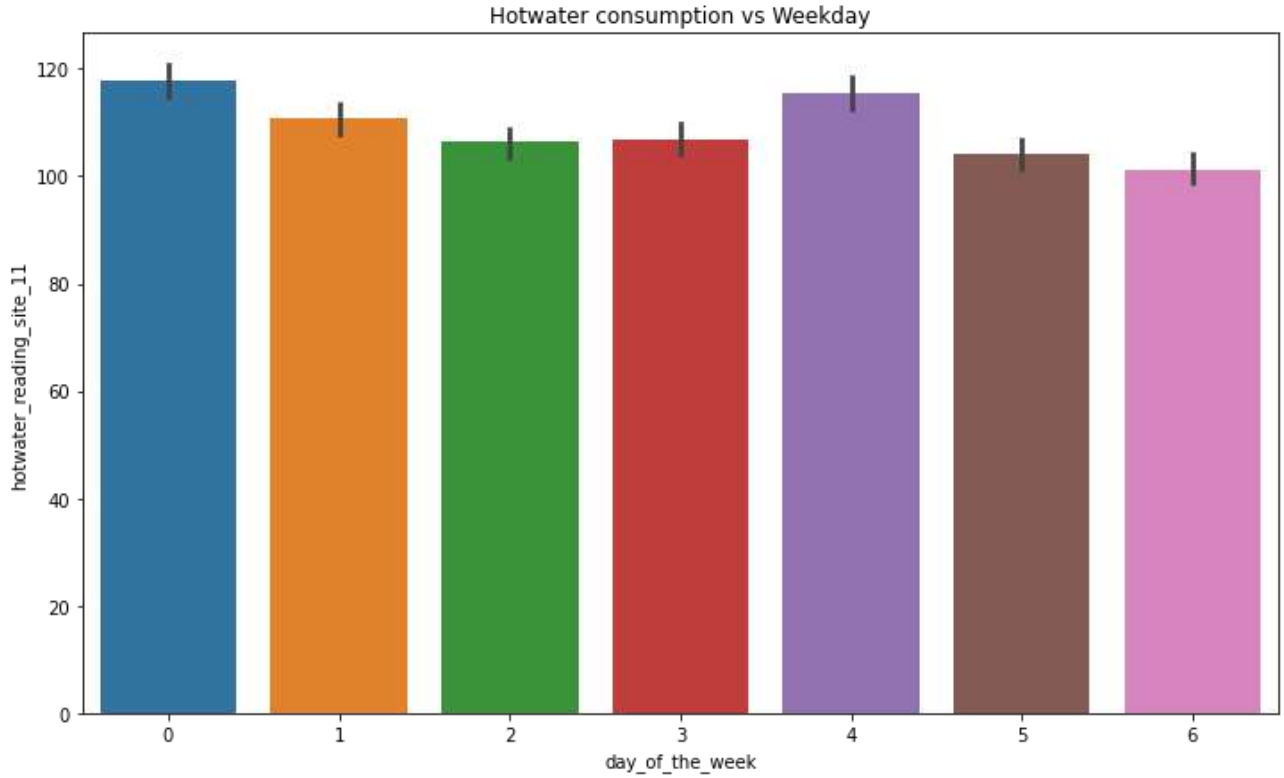
```
df_train_site_11_meter_3['month']=df_train_site_11_meter_3['timestamp'].dt.month
df_train_site_11_meter_3['weekday']=df_train_site_11_meter_3['timestamp'].dt.weekday
df_train_site_11_meter_3['hour']=df_train_site_11_meter_3['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_3
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('hotwater_reading_site_11')
plt.title('Hotwater consumption vs Month')
plt.show()
```



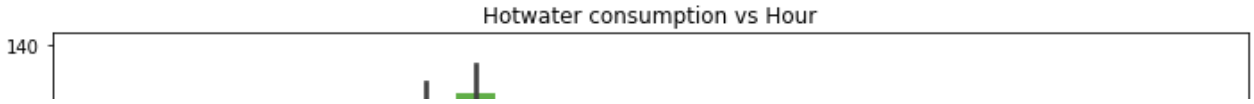
The above plot shows the hotwater consumption over the month and we can see that the consumption is higher for the winter month and becomes less as we transition over the month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_3
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('hotwater_reading_site_11')
plt.title('Hotwater consumption vs Weekday')
plt.show()
```



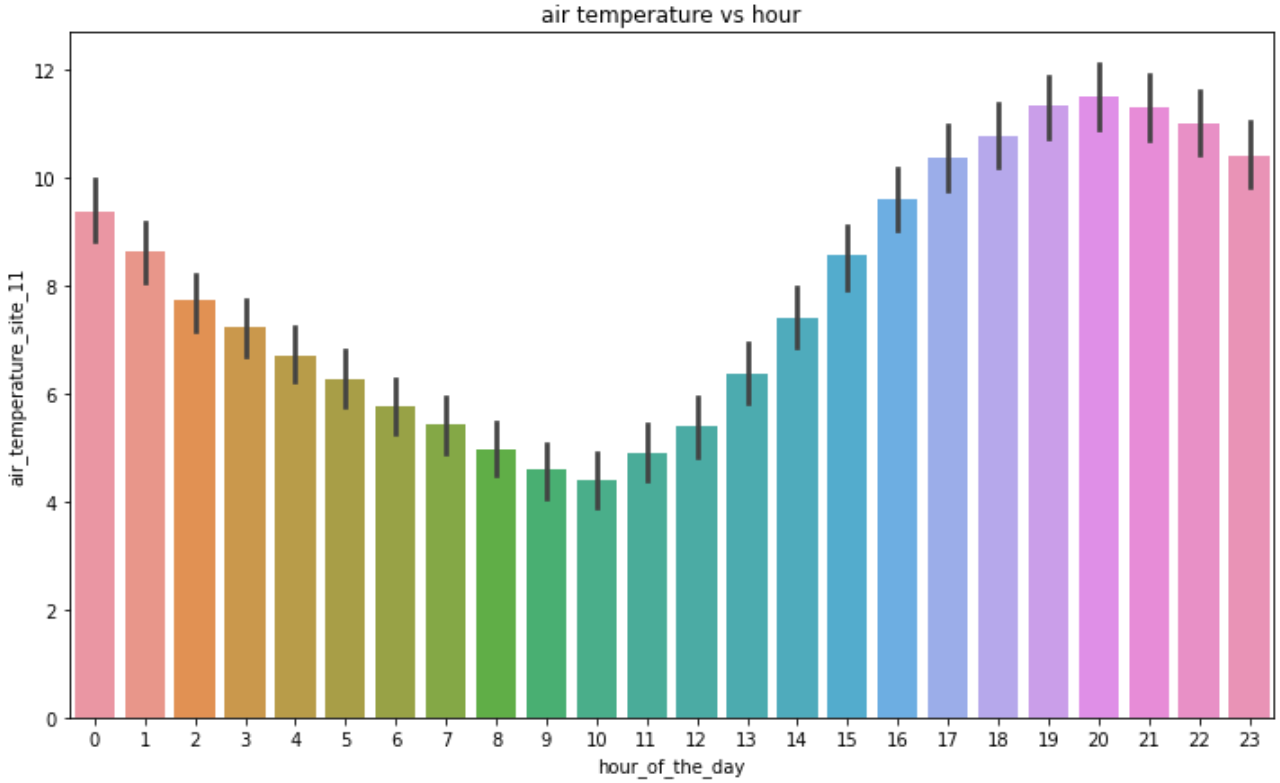
The above plot shows that the hotwater consumption is showing variations over the week but does not follow any specific pattern

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('hotwater_reading_site_11')
plt.title('Hotwater consumption vs Hour')
plt.show()
```



From the above plot we can see that the hotwater consumption is higher during the morning hours

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_11')
plt.title('air temperature vs hour')
plt.show()
```



The weather timestamp is not in alignment with the local timestamp of the hourly meter readings.The temperature peaks around 20:00 pm

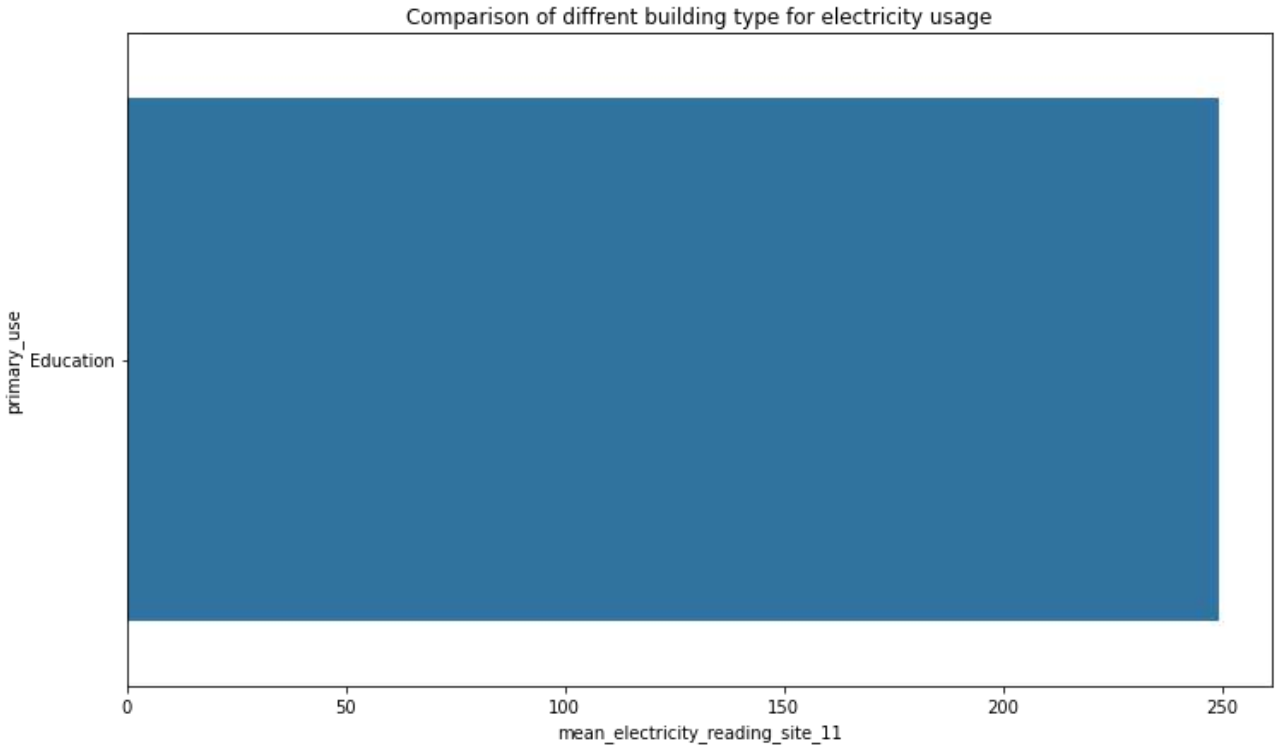
```
fig,axs=plt.subplots(figsize=(14,20),nrows=5,ncols=1,squeeze=False)
for i in range(df_train_site_11_meter_3['building_id'].nunique()):
   g=df_train_site_11_meter_3['building_id'].unique()[i]
   axes=axs[i%5][i//5]
   z=df_train_site_11_meter_3.loc[df_train_site_11_meter_3['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('hotwater_reading_site_11')
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
   plt.subplots_adjust(hspace=0.5)
```

Important Observations

- Here we can see the zero meter and low meter readings for the months starting from the 5th month upto the 10th month which is actually seasonal variation which the model needs to learn.

```
df_train_site_11_meter_0=df_train_site_11.loc[df_train_site_11['meter']=='electricity']
```

```
z=df_train_site_11_meter_0.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_reading_site_11')
plt.ylabel('primary_use')
plt.title('Comparison of diffrent building type for electricity usage')
plt.show()
```

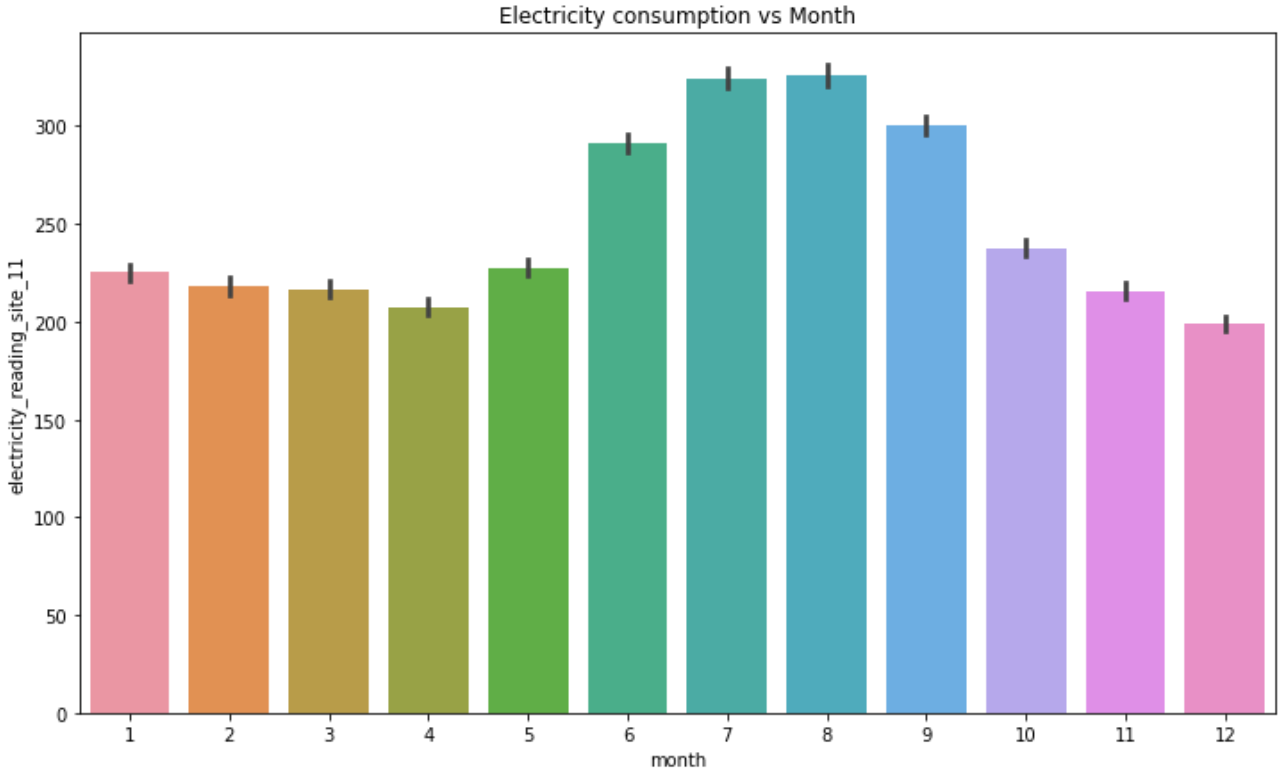


At site 11 we can see that only educational building is consuming electricity

```
df_train_site_11_meter_0['month']=df_train_site_11_meter_0['timestamp'].dt.month
df_train_site_11_meter_0['weekday']=df_train_site_11_meter_0['timestamp'].dt.weekday
df_train_site_11_meter_0['hour']=df_train_site_11_meter_0['timestamp'].dt.hour
```

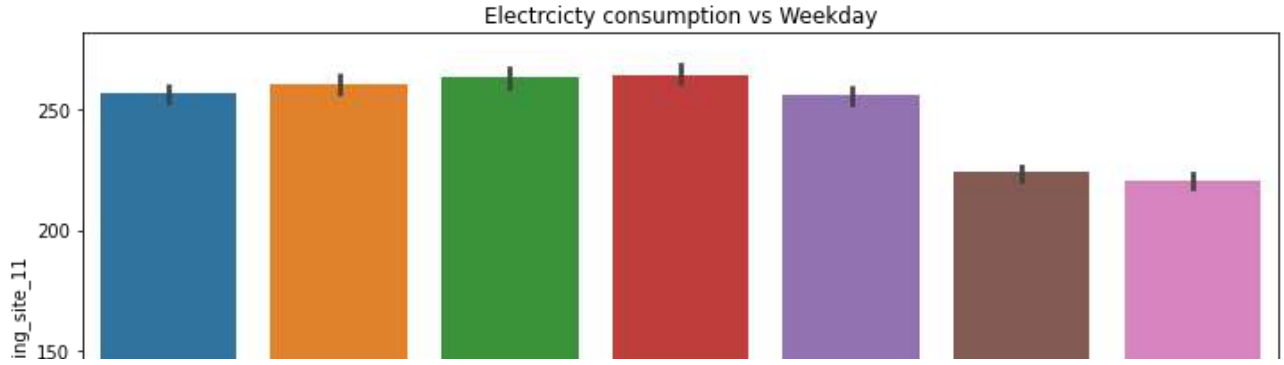
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
```

```
plt.ylabel('electricity_reading_site_11')
plt.title('Electricity consumption vs Month')
plt.show()
```



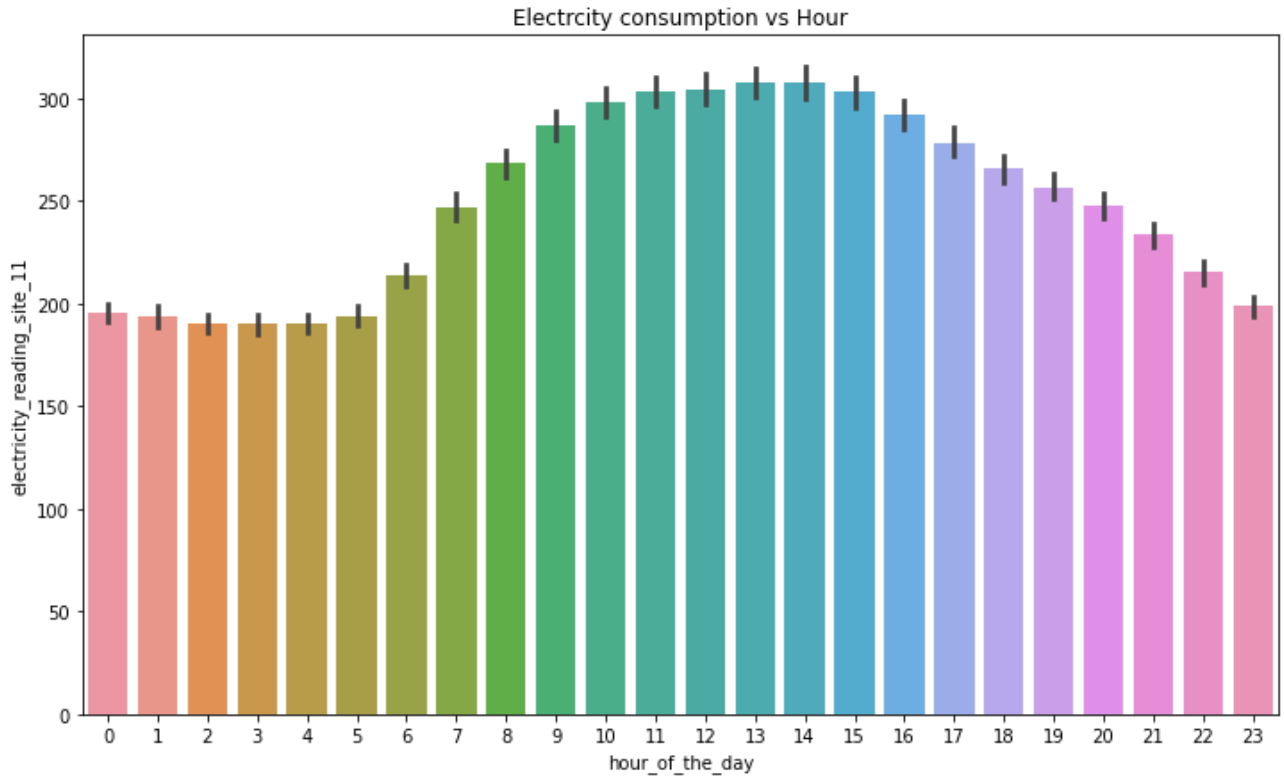
The above plot shows that the electrcity consumption is relatively higher for the summer months

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_11')
plt.title('Electrcicty consumption vs Weekday')
plt.show()
```



Here we can see that the electricity consumption is less over the weekend as compared to the weekday

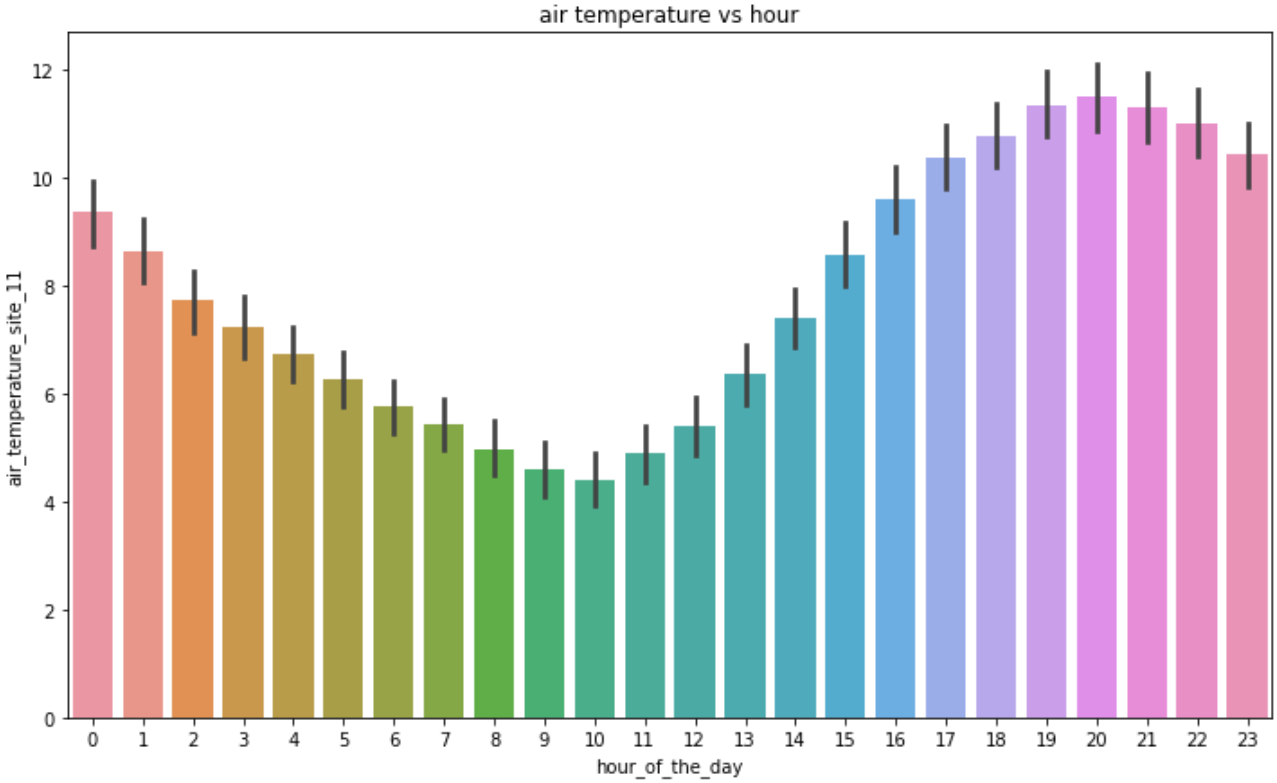
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_11')
plt.title('Electrcity consumption vs Hour')
plt.show()
```



The electricity consumption starts increasing from 6:00 am and peaks around 14:00 pm and then after that starts to fall gradually

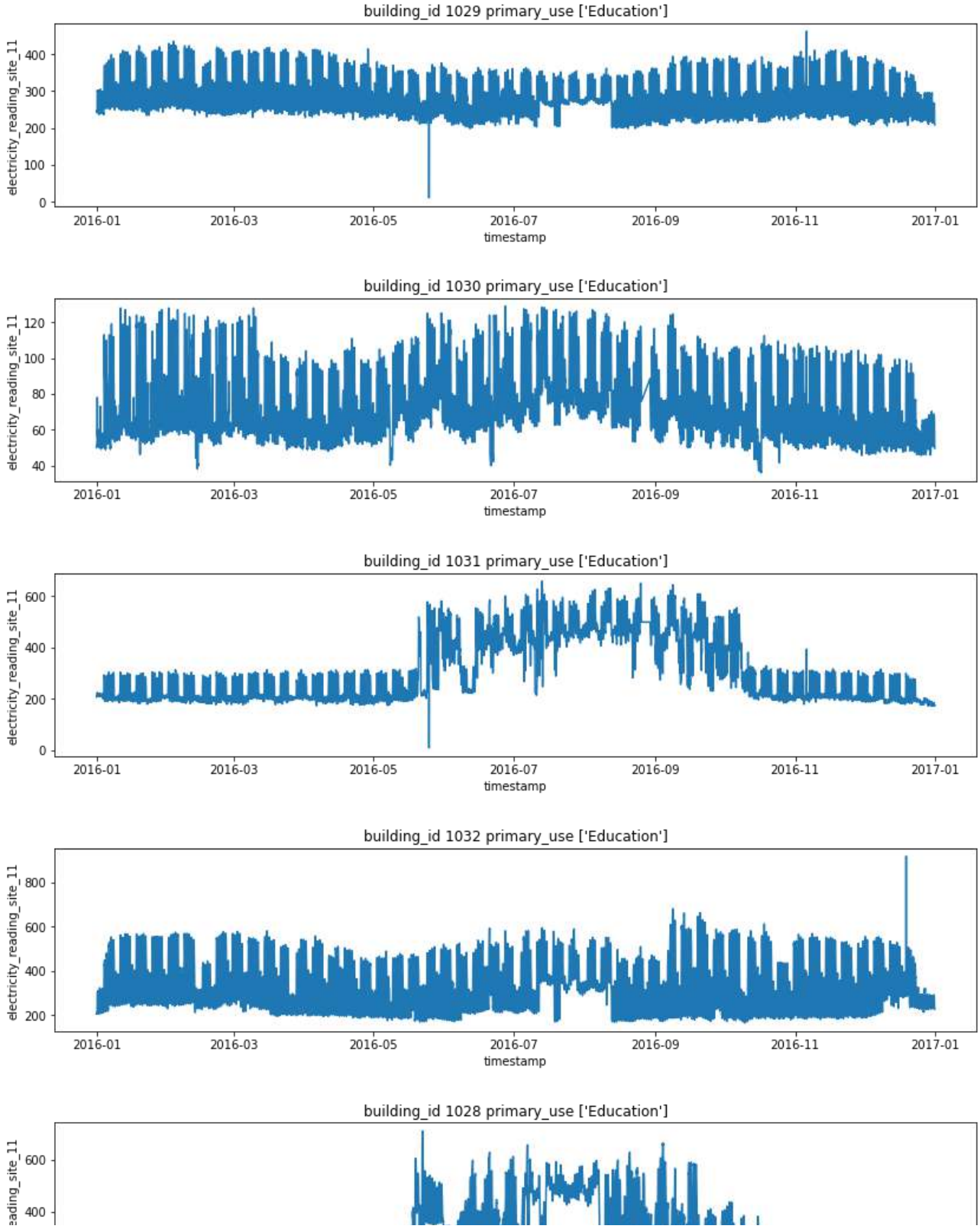
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_11_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
```

```
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_11')
plt.title('air temperature vs hour')
plt.show()
```



From the above plot we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter reading.The temperature peaks around 20:00 pm

```
fig,axs=plt.subplots(figsize=(14,20),nrows=5,ncols=1,squeeze=False)
for i in range(df_train_site_11_meter_0['building_id'].nunique()):
   g=df_train_site_11_meter_0['building_id'].unique()[i]
   axes=axs[i%5][i//5]
   z=df_train_site_11_meter_0.loc[df_train_site_11_meter_0['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('electricity_reading_site_11')
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()))
   plt.subplots_adjust(hspace=0.5)
```



Important Observations

- Building 1029 1031 1032 contains spike and zero meter reading which can be filtered out.

```
df_train_site_12=df_train_merge.loc[df_train_merge['site_id']==12]
```

```
df_train_site_12.isnull().sum()/df_train_site_12.shape[0]
```

```
building_id          0.00
meter                0.00
```

```
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           1.00
floor_count          0.75
air_temperature      0.00
cloud_coverage       0.01
dew_temperature      0.00
precip_depth_1_hr    1.00
sea_level_pressure   0.01
wind_direction       0.00
wind_speed           0.00
dtype: float64
```

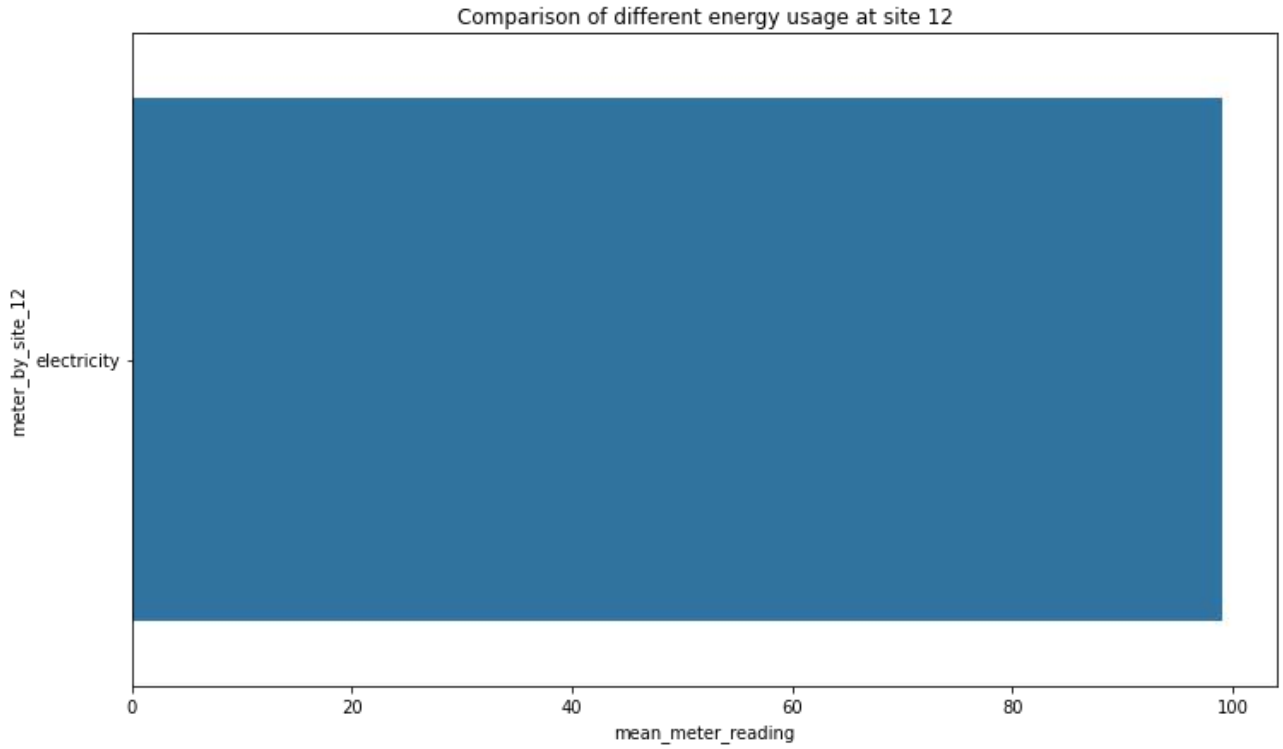
Here we can see the missing values at site 12 and we need to impute them for training

```
df_corr_12=df_train_site_12.corr()
df_corr_12.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.09	nan	0.14	nan	0.43	0.00
meter_reading	0.09	1.00	nan	0.61	nan	0.52	0.03
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.14	0.61	nan	1.00	nan	0.64	-0.00
year_built	nan	nan	nan	nan	nan	nan	nan
floor_count	0.43	0.52	nan	0.64	nan	1.00	0.00
air_temperature	0.00	0.03	nan	-0.00	nan	0.00	1.00
cloud_coverage	-0.00	0.01	nan	-0.00	nan	-0.00	0.25
dew_temperature	0.00	-0.00	nan	-0.00	nan	-0.00	0.90
precip_depth_1_hr	nan	nan	nan	nan	nan	nan	nan
sea_level_pressure	0.00	-0.00	nan	0.00	nan	-0.00	0.10
wind_direction	-0.00	-0.01	nan	-0.00	nan	0.00	-0.14
wind_speed	0.00	0.04	nan	0.00	nan	0.00	0.07

From the above correlation plot we can see that the meter reading has some correlation with the square feet and the floor count

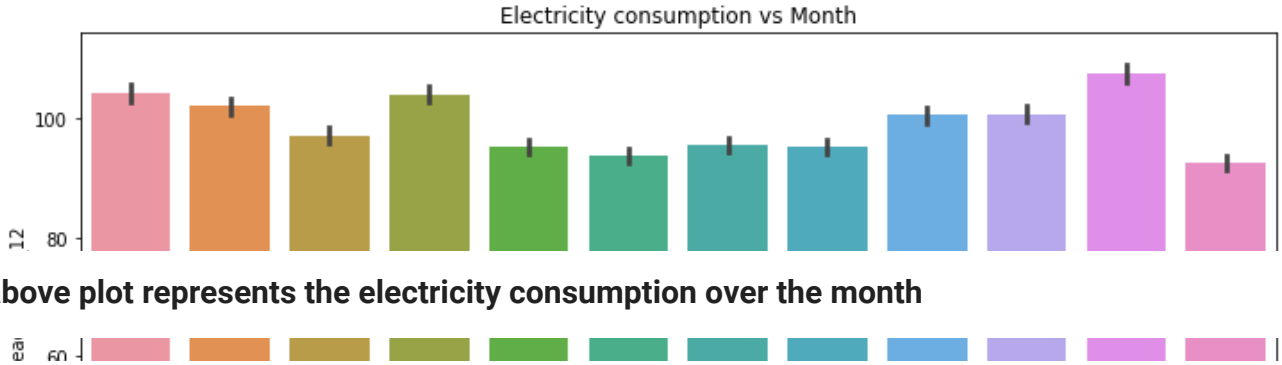
```
z=df_train_site_12.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading')
plt.ylabel('meter_by_site_12')
plt.title('Comparison of different energy usage at site 12')
plt.show()
```



From the above plot we can see that the only energy usage at site 12 is electricity

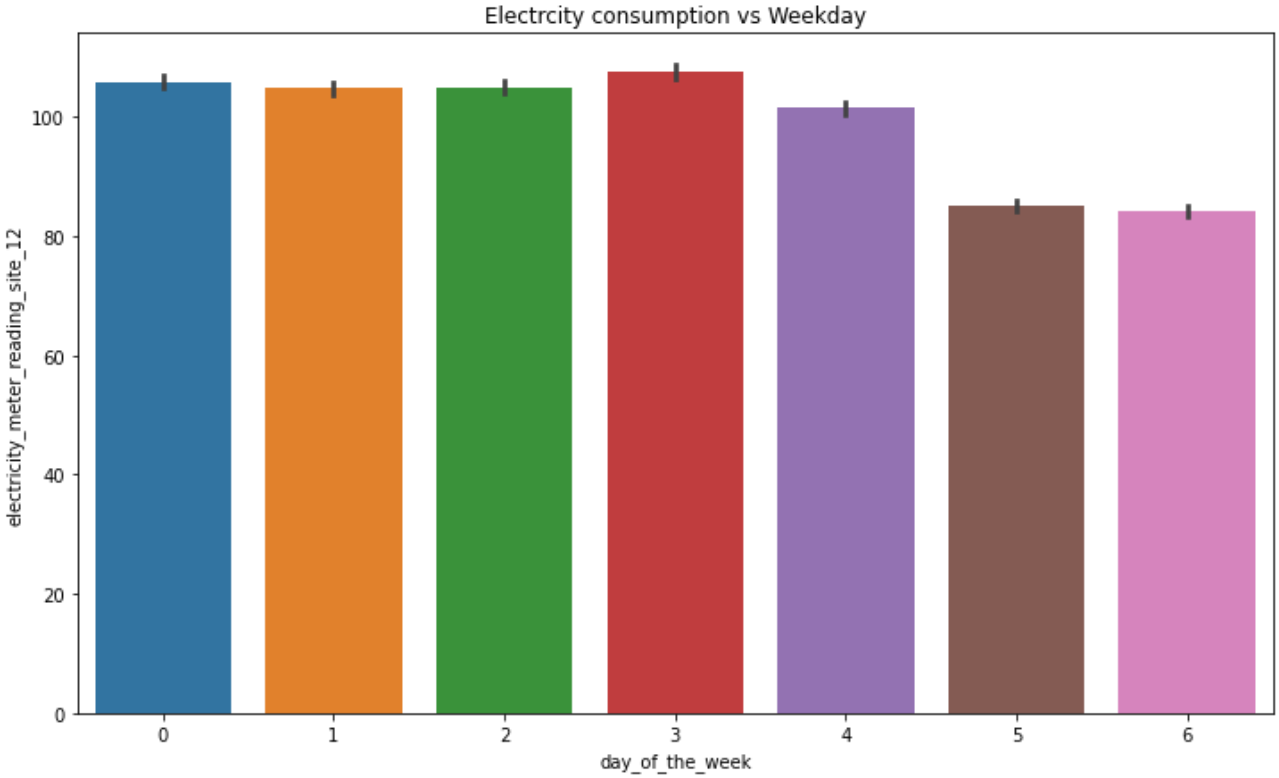
```
df_train_site_12['month']=df_train_site_12['timestamp'].dt.month
df_train_site_12['weekday']=df_train_site_12['timestamp'].dt.weekday
df_train_site_12['hour']=df_train_site_12['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_12
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_meter_reading_site_12')
plt.title('Electricity consumption vs Month')
plt.show()
```



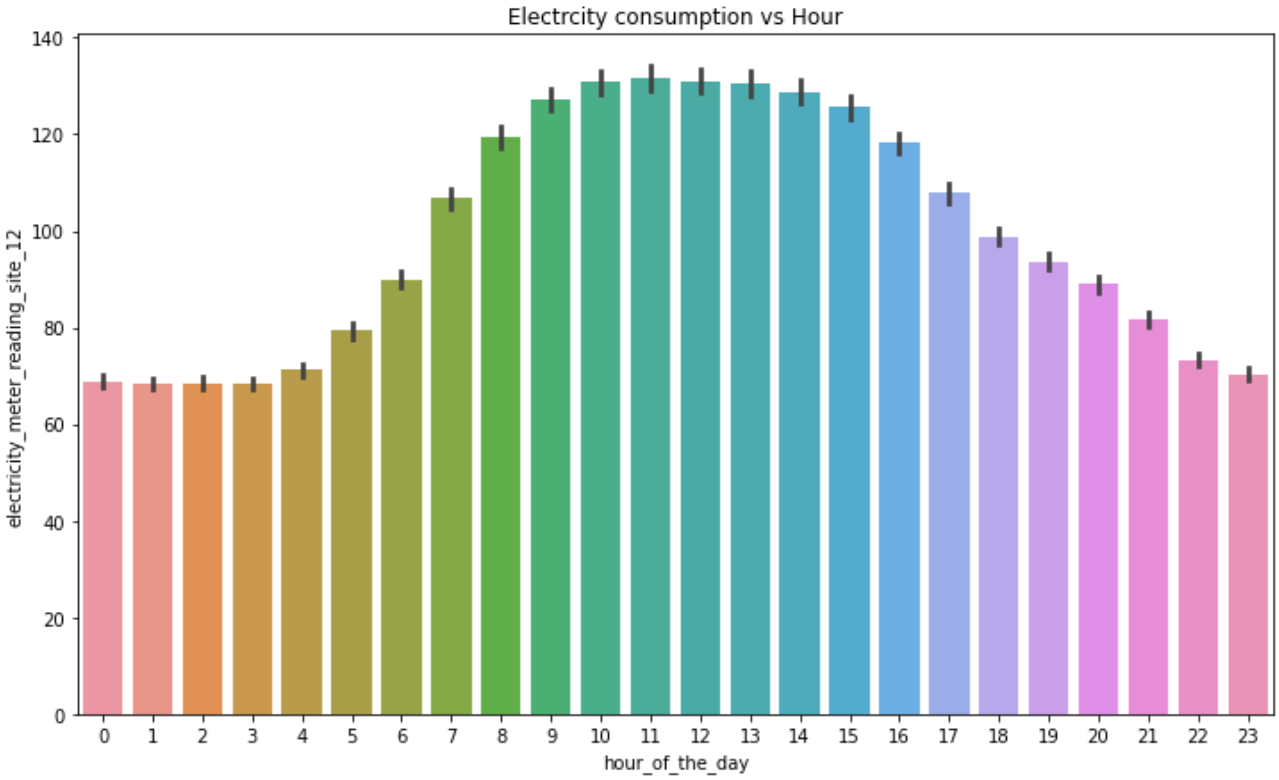
The above plot represents the electricity consumption over the month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_12
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_meter_reading_site_12')
plt.title('Electrcity consumption vs Weekday')
plt.show()
```



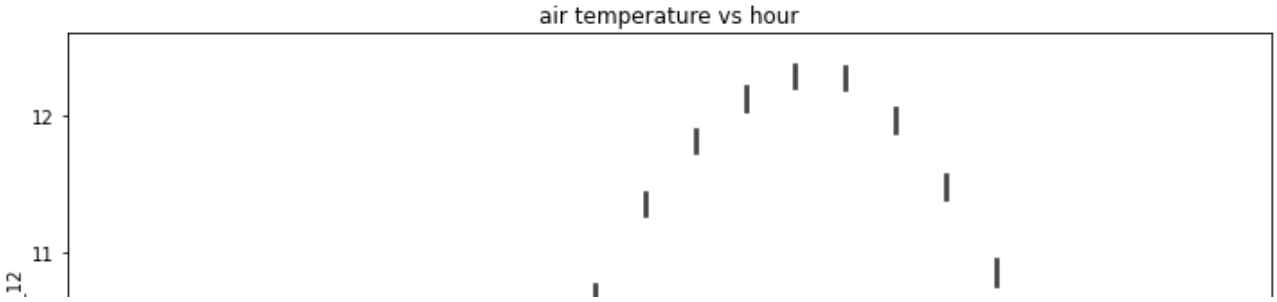
From the above plot we can see that the eelctrcity consumption is less over the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_12
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_meter_reading_site_12')
plt.title('Electrcity consumption vs Hour')
plt.show()
```



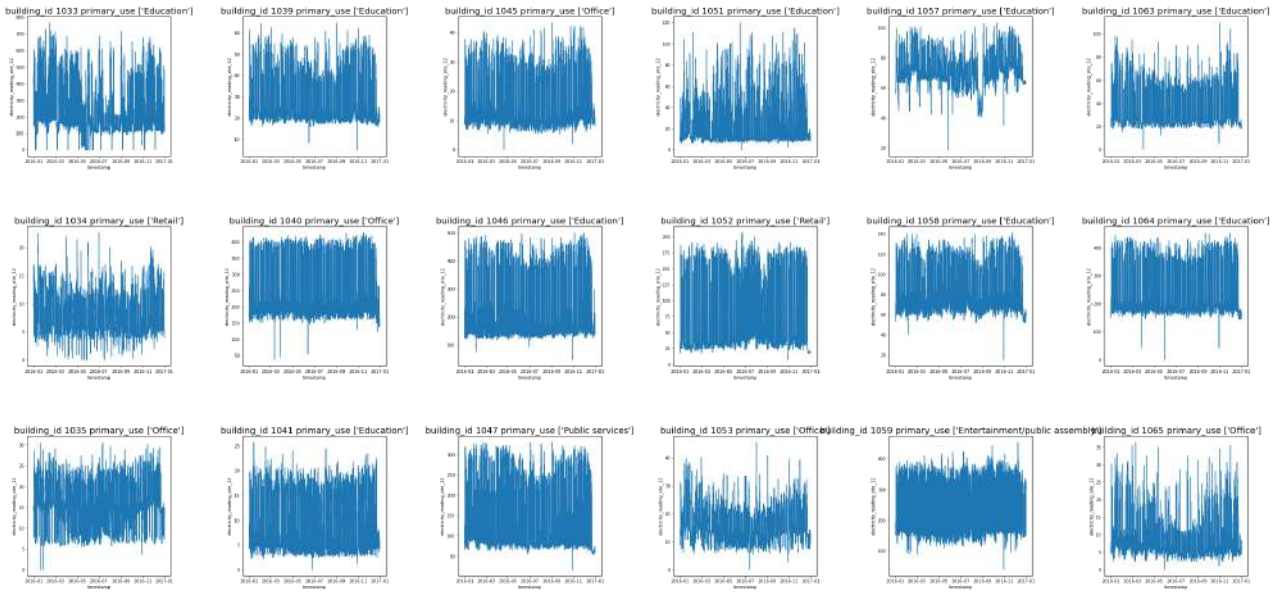
From the above plot we can see that the electrcity consumption starts increasing from 6:00 am in the morning peaks around 11:00 am in the morning and starts gradually decreasing after 15:00 pm

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_12
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_12')
plt.title('air temperature vs hour')
plt.show()
```



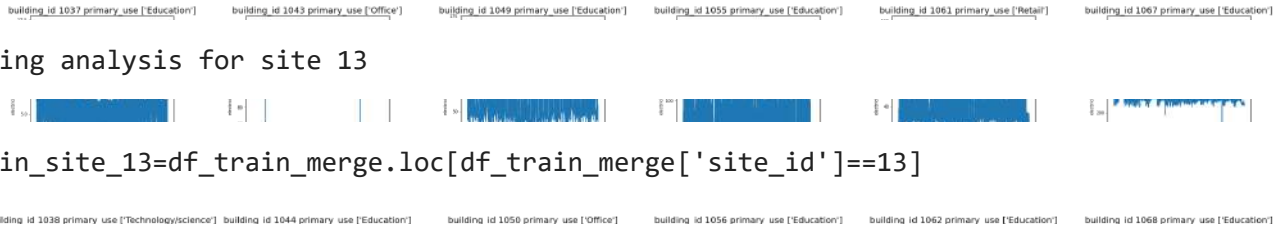
Here we can see that the weather timestamp is in alignment for the local timestamp of the hourly meter reading.The temperature peaks around 14:00 pm

```
fig,axs=plt.subplots(figsize=(50,50),nrows=6,ncols=6,squeeze=True)
for i in range(df_train_site_12['building_id'].nunique()):
   g=df_train_site_12['building_id'].unique()[i]
   axes=axs[i%6][i//6]
   z=df_train_site_12.loc[df_train_site_12['building_id']==g]
   axes.plot(z['timestamp'],z['meter_reading'])
   axes.set_xlabel('timestamp')
   axes.set_ylabel('electricity_reading_site_12')
   axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
   plt.subplots_adjust(hspace=0.5,wspace=0.5)
```



Important Observations

- Building 1066 contains zero meter readings during a certain period which needs to be filtered out.




```
#Starting analysis for site 13
```



```
df_train_site_13=df_train_merge.loc[df_train_merge['site_id']==13]
```



```
df_train_site_13.isnull().sum()/df_train_site_13.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           1.00
floor_count          1.00
air_temperature      0.00
cloud_coverage       0.49
dew_temperature      0.00
precip_depth_1_hr    0.00
sea_level_pressure   0.01
wind_direction       0.02
wind_speed           0.00
dtype: float64
```

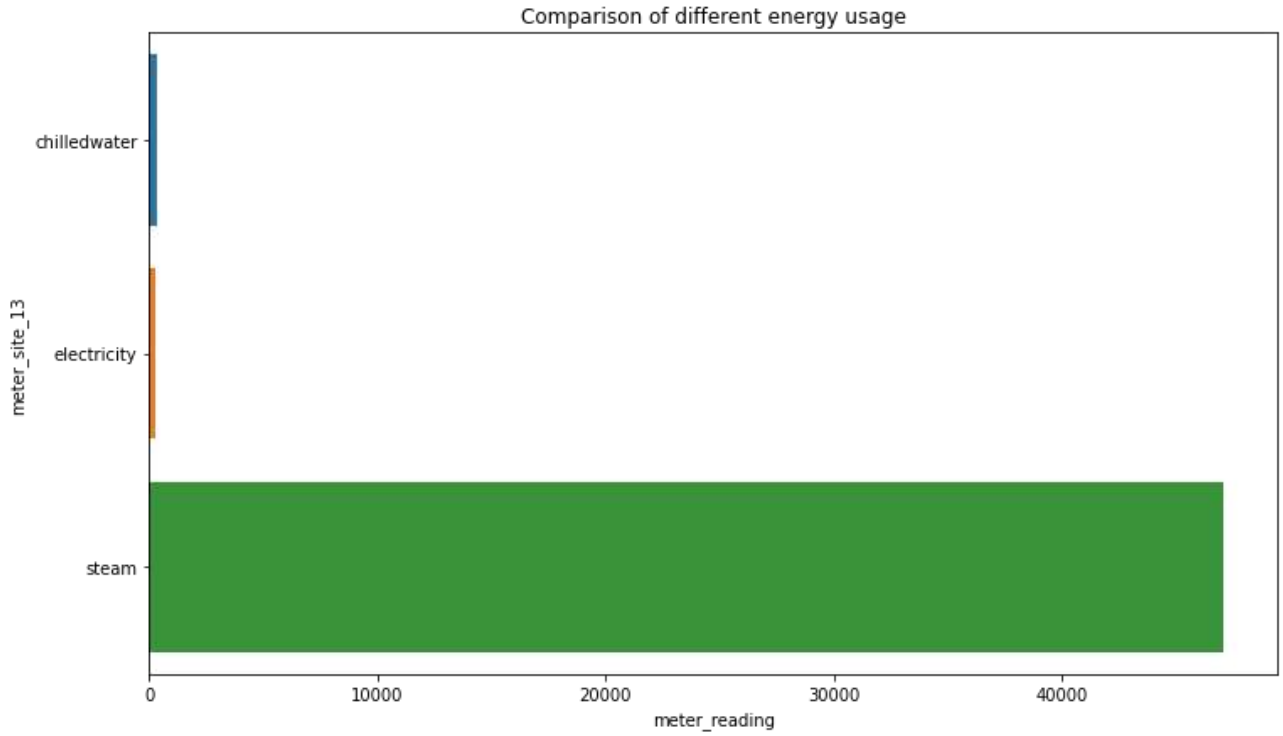
Here we can see some missing values and we need to impute those values

```
df_corr_13=df_train_site_13.corr()
df_corr_13.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	-0.03	nan	0.02	nan	nan	0.00
meter_reading	-0.03	1.00	nan	0.04	nan	nan	0.01
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.02	0.04	nan	1.00	nan	nan	0.00
year_built	nan	nan	nan	nan	nan	nan	nan
floor_count	nan	nan	nan	nan	nan	nan	nan
air_temperature	0.00	0.01	nan	0.00	nan	nan	1.00
cloud_coverage	-0.00	0.01	nan	0.00	nan	nan	0.02
dew_temperature	0.00	0.00	nan	0.00	nan	nan	0.94
precip_depth_1_hr	-0.00	0.00	nan	0.00	nan	nan	0.05
sea_level_pressure	0.00	-0.01	nan	-0.00	nan	nan	-0.32
wind_direction	-0.00	-0.00	nan	-0.00	nan	nan	-0.13
wind_speed	-0.00	0.01	nan	0.00	nan	nan	-0.02

From the above correlation plot we can see that the meter reading is not correlated with any of the features

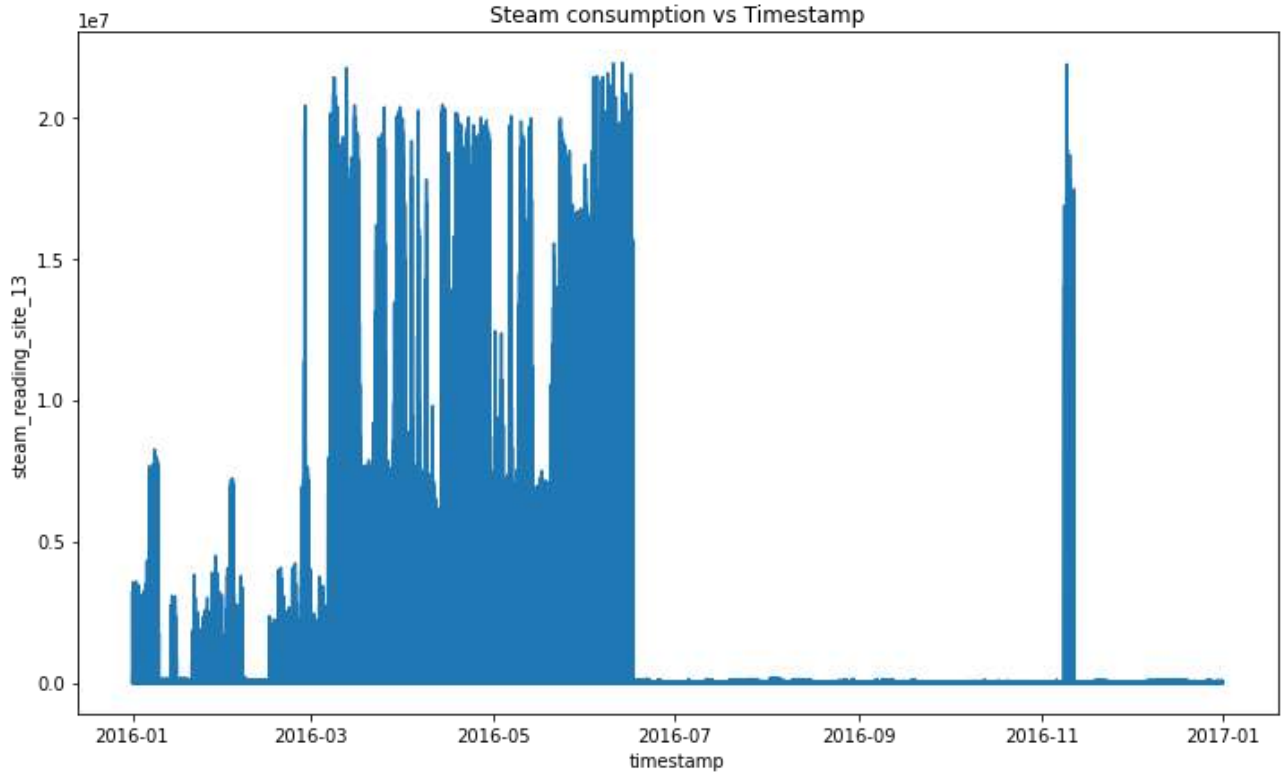
```
z=df_train_site_13.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('meter_reading')
plt.ylabel('meter_site_13')
plt.title('Comparison of different energy usage')
plt.show()
```



From the above plot we can see that at site 13 steam is having very high energy consumption.Lets investigate it further about the energy consumption of steam.

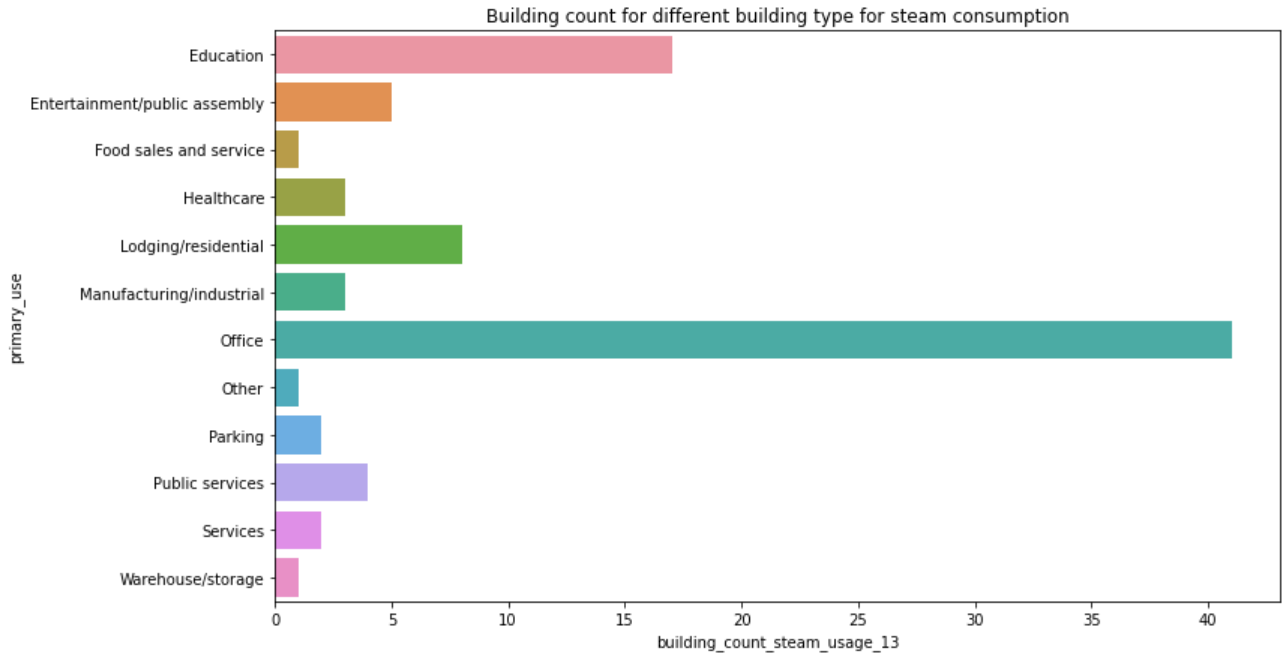
```
df_train_site_13_meter_2=df_train_site_13.loc[df_train_site_13['meter']=='steam']
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_2
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('steam_reading_site_13')
plt.title('Steam consumption vs Timestamp')
plt.show()
```



The above plot represents the steam consumption of all the buildings over the timestamp

```
z=df_train_site_13_meter_2.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_steam_usage_13')
plt.ylabel('primary_use')
plt.title('Building count for different building type for steam consumption')
plt.show()
```



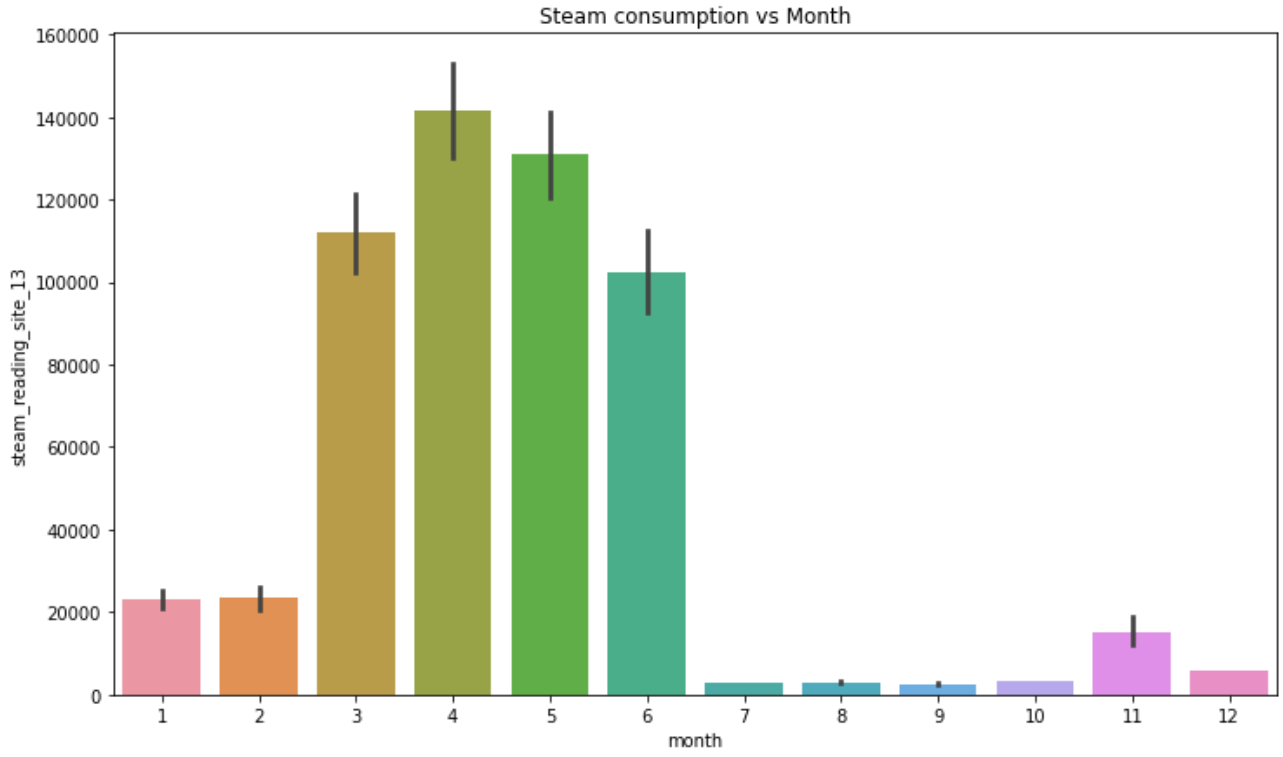
This plot represents the building count for different building type for steam consumption at site 13

```
z=df_train_site_13_meter_2.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_steam_reading_site_13')
plt.ylabel('primary_use')
plt.title('Comparison for different building type fro steam consumption')
plt.show()
```

The above plot represents the steam consumption for different building type.Education is the highest consumption of steam relative as compared to the other building type.

```
df_train_site_13_meter_2['month']=df_train_site_13_meter_2['timestamp'].dt.month
df_train_site_13_meter_2['weekday']=df_train_site_13_meter_2['timestamp'].dt.weekday
df_train_site_13_meter_2['hour']=df_train_site_13_meter_2['timestamp'].dt.hour
```

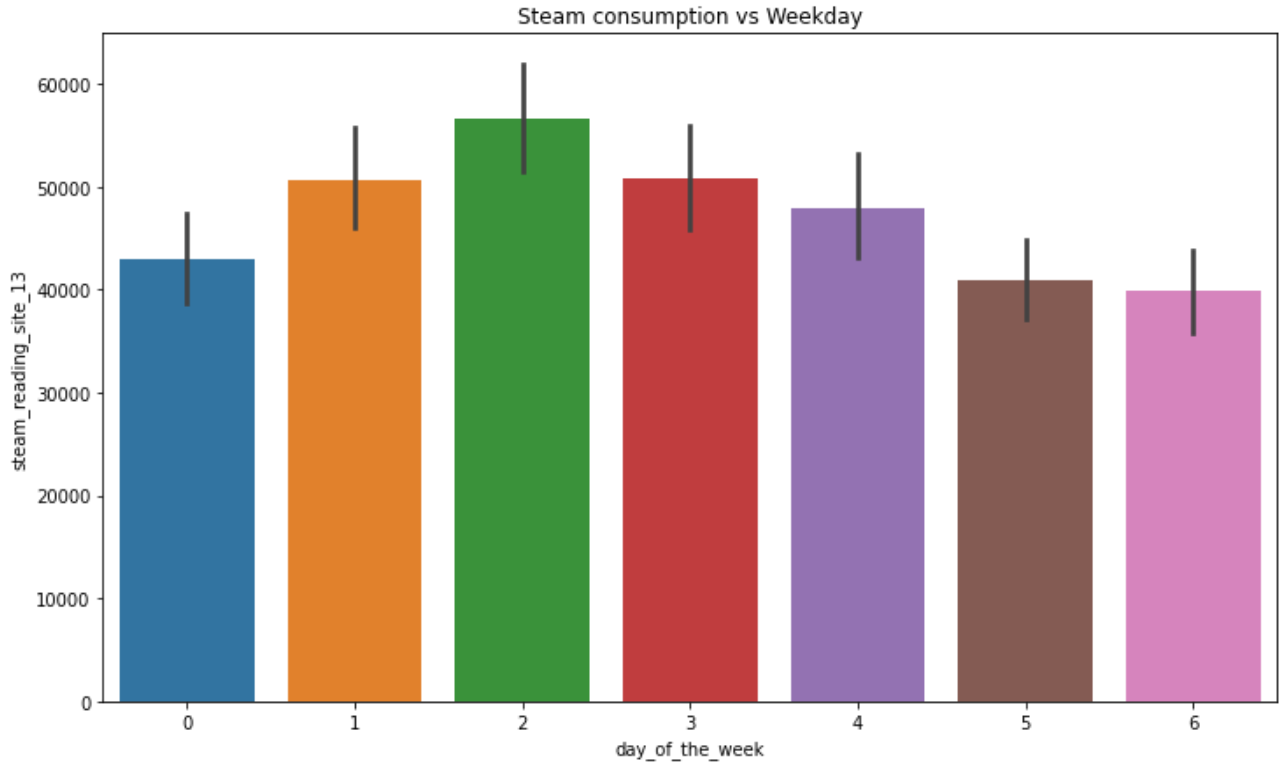
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_2
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('steam_reading_site_13')
plt.title('Steam consumption vs Month')
plt.show()
```



We can see a unusual pattern for the steam consumption as it rises for the transition month and remains low for the other month of the year.

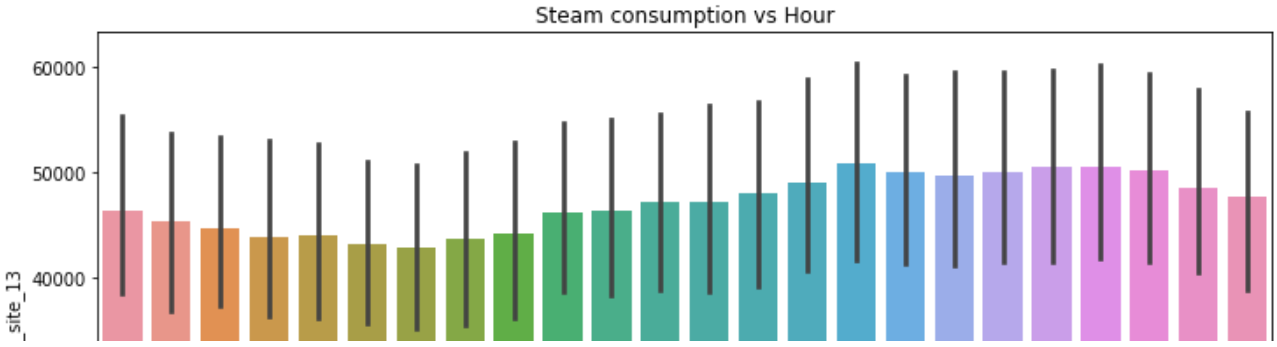
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_2
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day of the week')
```

```
plt.xlabel('day_of_the_week')
plt.ylabel('steam_reading_site_13')
plt.title('Steam consumption vs Weekday')
plt.show()
```



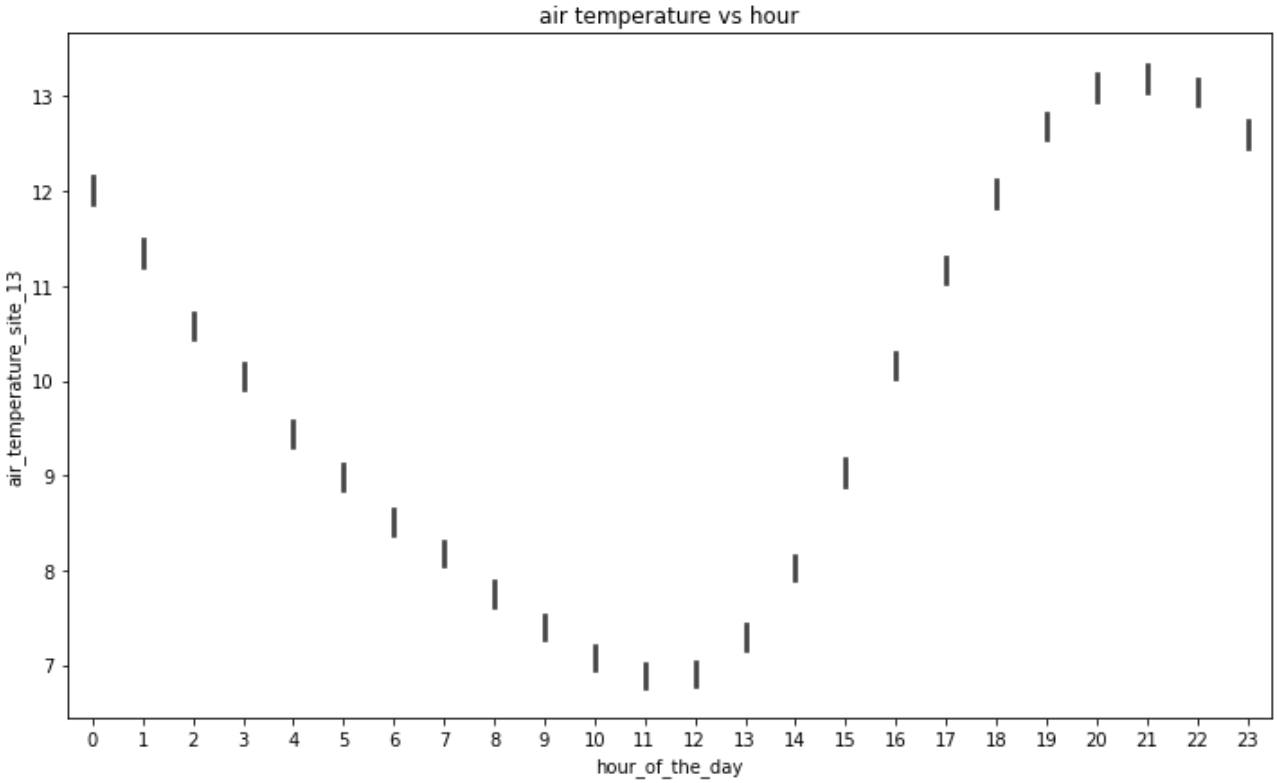
From the above plot we can see the variation of steam over different days of the week.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('steam_reading_site_13')
plt.title('Steam consumption vs Hour')
plt.show()
```



Steam consumption shows different usage over the hour of the day and we cannot see a definite pattern

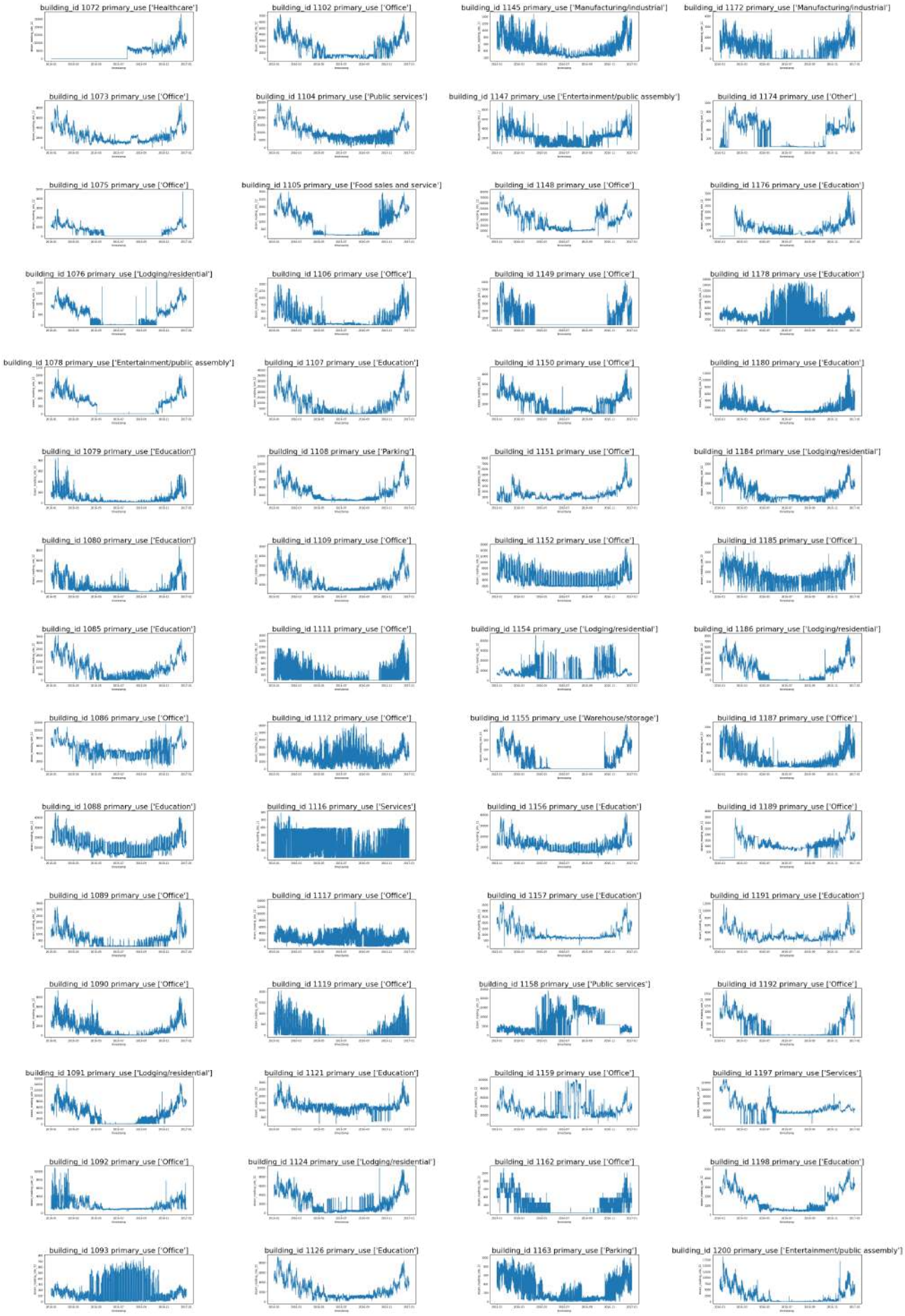

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_13')
plt.title('air temperature vs hour')
plt.show()
```



From the above plot we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings.

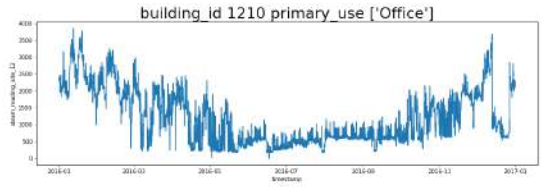
```
fig,axs=plt.subplots(figsize=(55,120),nrows=20,ncols=4,squeeze=True)
for i in range(df_train_site_13_meter_2['building_id'].nunique()-8):
   g=df_train_site_13_meter_2['building_id'].unique()[i]
   axes=axs[i%20][i//20]
```

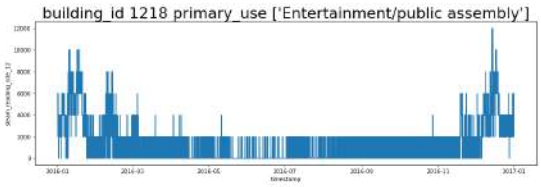
```
    z=df_train_site_13_meter_2.loc[df_train_site_13_meter_2['building_id']==g]
    axes.plot(z['timestamp'],z['meter_reading'])
    axes.set_xlabel('timestamp')
    axes.set_ylabel('steam_reading_site_12')
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
    plt.subplots_adjust(hspace=0.8,wspace=0.5)
```

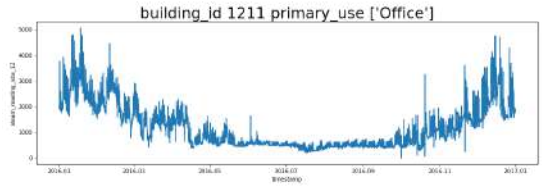


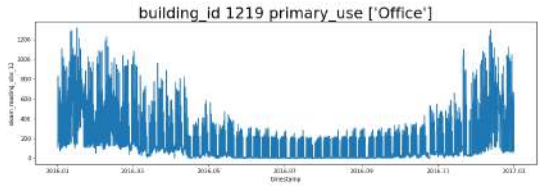

```
fig,axs=plt.subplots(figsize=(40,30),nrows=4,ncols=2,squeeze=True)
for i in range(df_train_site_13_meter_2['building_id'].nunique()-80):
    g=df_train_site_13_meter_2['building_id'].unique()[80:89][i]
    axes=axs[i%4][i//4]
    z=df_train_site_13_meter_2.loc[df_train_site_13_meter_2['building_id']==g]
```

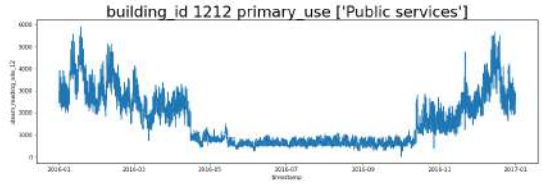
```
axes.plot(z['timestamp'],z['meter_reading'])
axes.set_xlabel('timestamp')
axes.set_ylabel('steam_reading_site_12')
axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
plt.subplots_adjust(hspace=0.8,wspace=0.5)
```

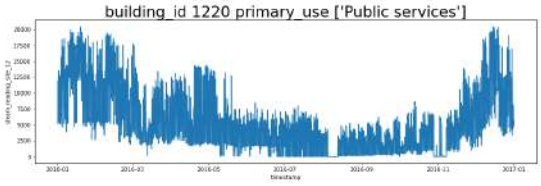


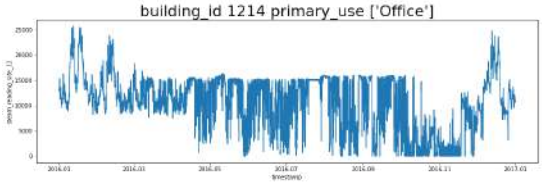


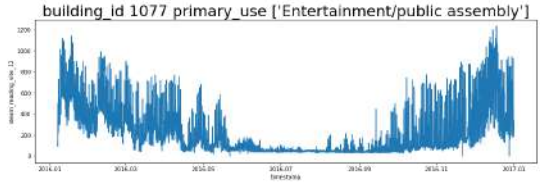












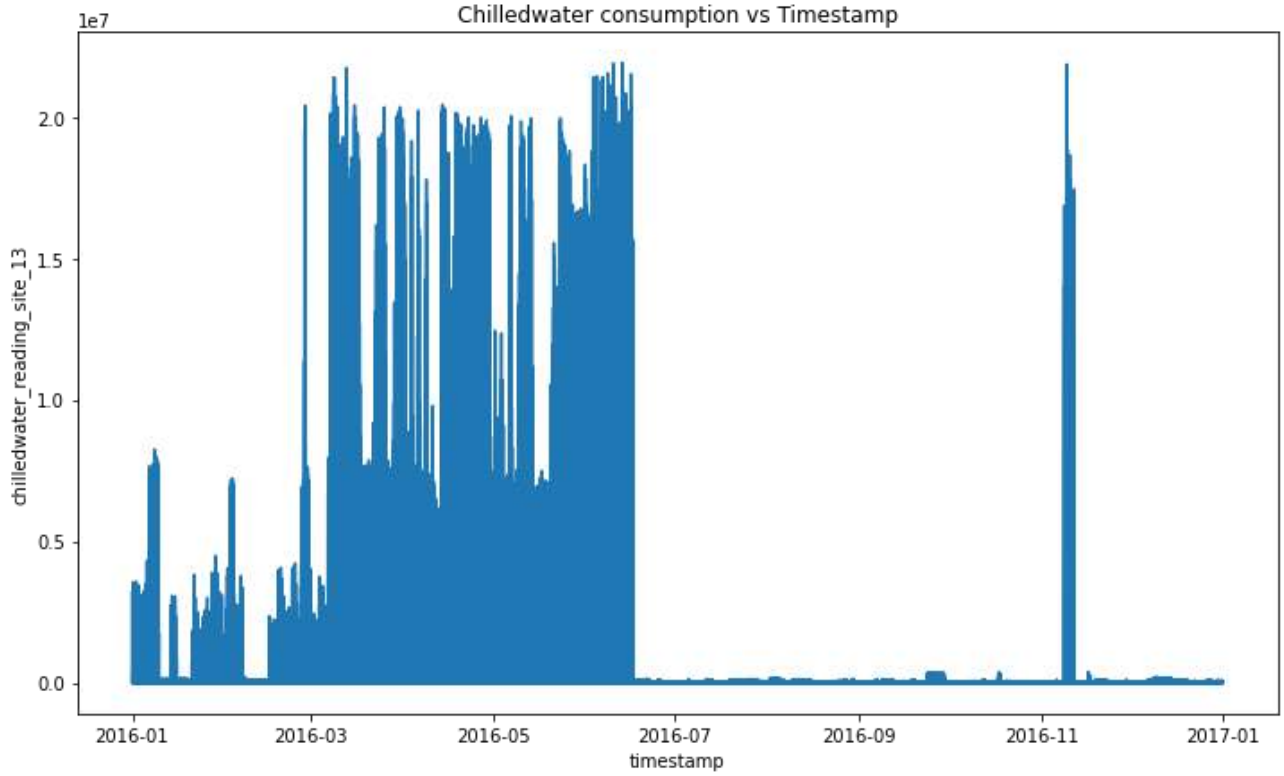
There are total of 88 buildings so for representing it clearly I have divided it into 80 and 8.

Important Observations

- Building 1075 1076 1099 1124 1150 1184 1200 1203 have spike which needs to be filtered out.
- Building 1072 1098 1158 1129 1176 1189 1111 have certain zero meter readings during ceratin months which should be removed.As we do not want our model to fit on these values.

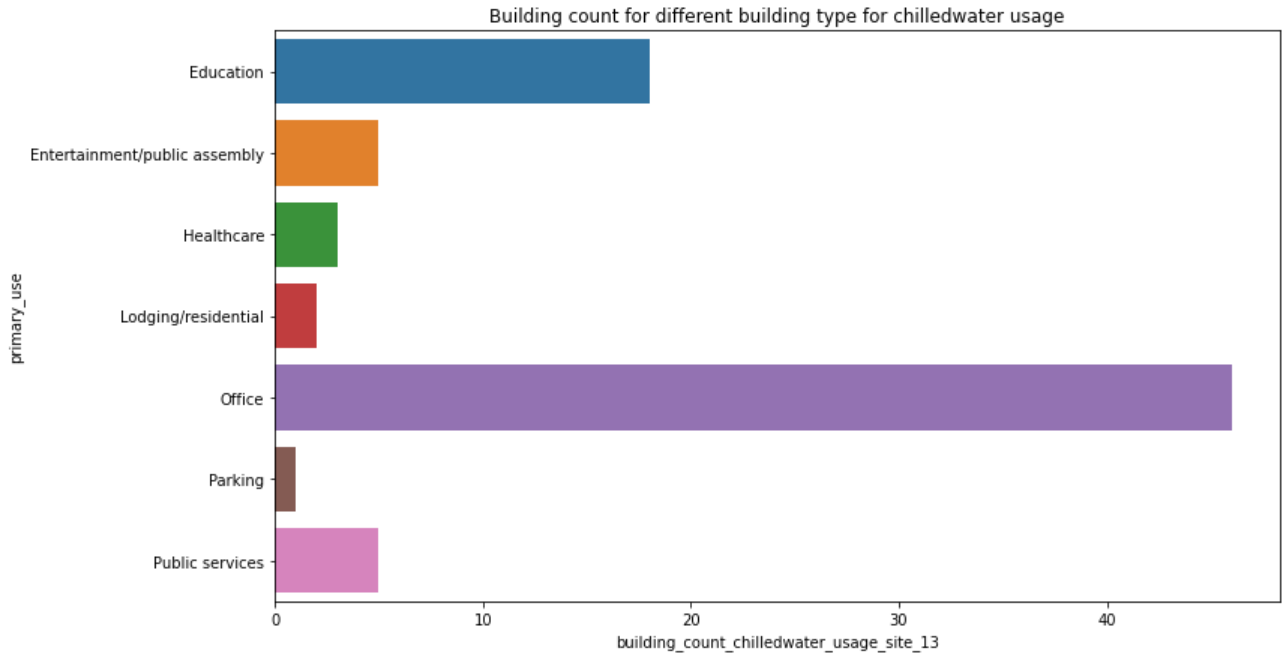
```
df_train_site_13_meter_1=df_train_site_13.loc[df_train_site_13['meter']=='chilledwater']
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('chilledwater_reading_site_13')
plt.title('Chilledwater consumption vs Timestamp')
plt.show()
```



From the above plot we can see that the overall consumption of chilledwater for all the buildings over the timestamp

```
z=df_train_site_13_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_chilledwater_usage_site_13')
plt.ylabel('primary_use')
plt.title('Building count for different building type for chilledwater usage')
plt.show()
```



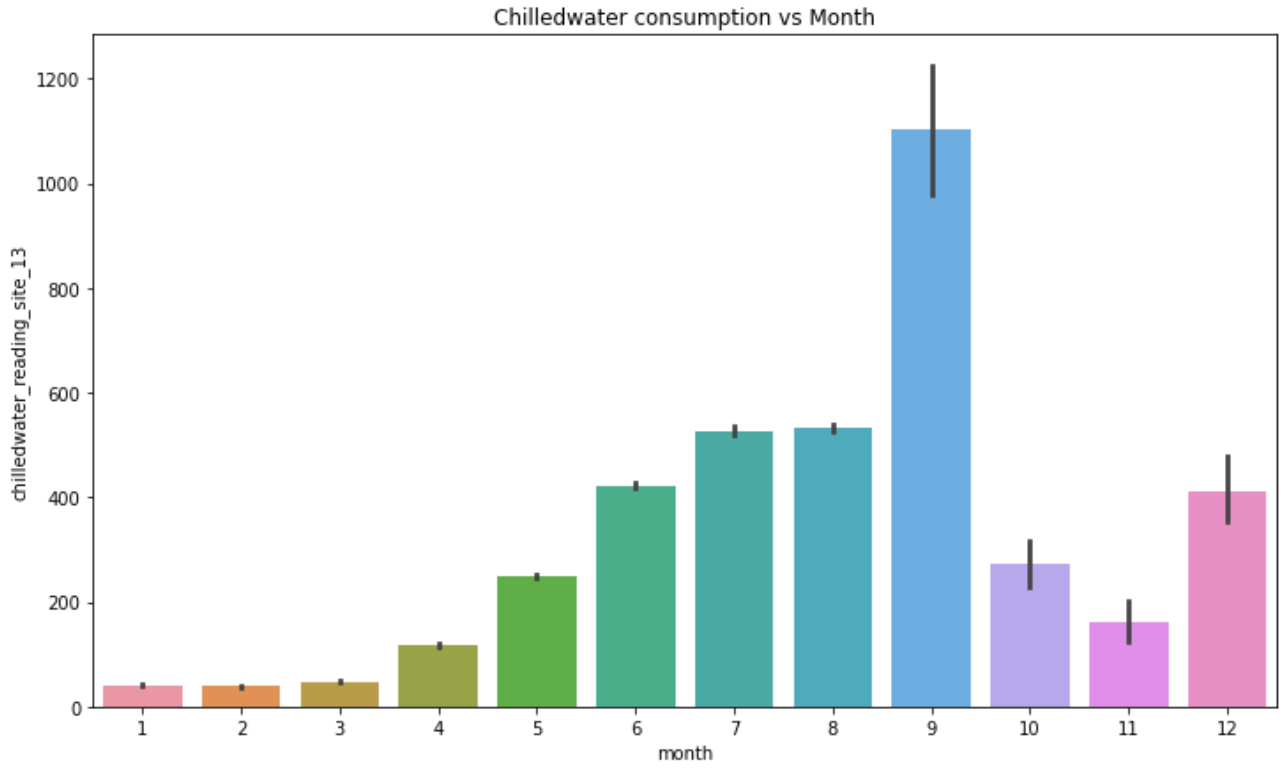
From the above plot we can check the building count for different building type for chilledwater usage.

```
z=df_train_site_13_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading_site_13')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for chilledwater usage')
plt.show()
```

From the above plot we can see that Education is having the highest chilledwater consumption at site 13

```
df_train_site_13_meter_1['month']=df_train_site_13_meter_1['timestamp'].dt.month
df_train_site_13_meter_1['weekday']=df_train_site_13_meter_1['timestamp'].dt.weekday
df_train_site_13_meter_1['hour']=df_train_site_13_meter_1['timestamp'].dt.hour
```

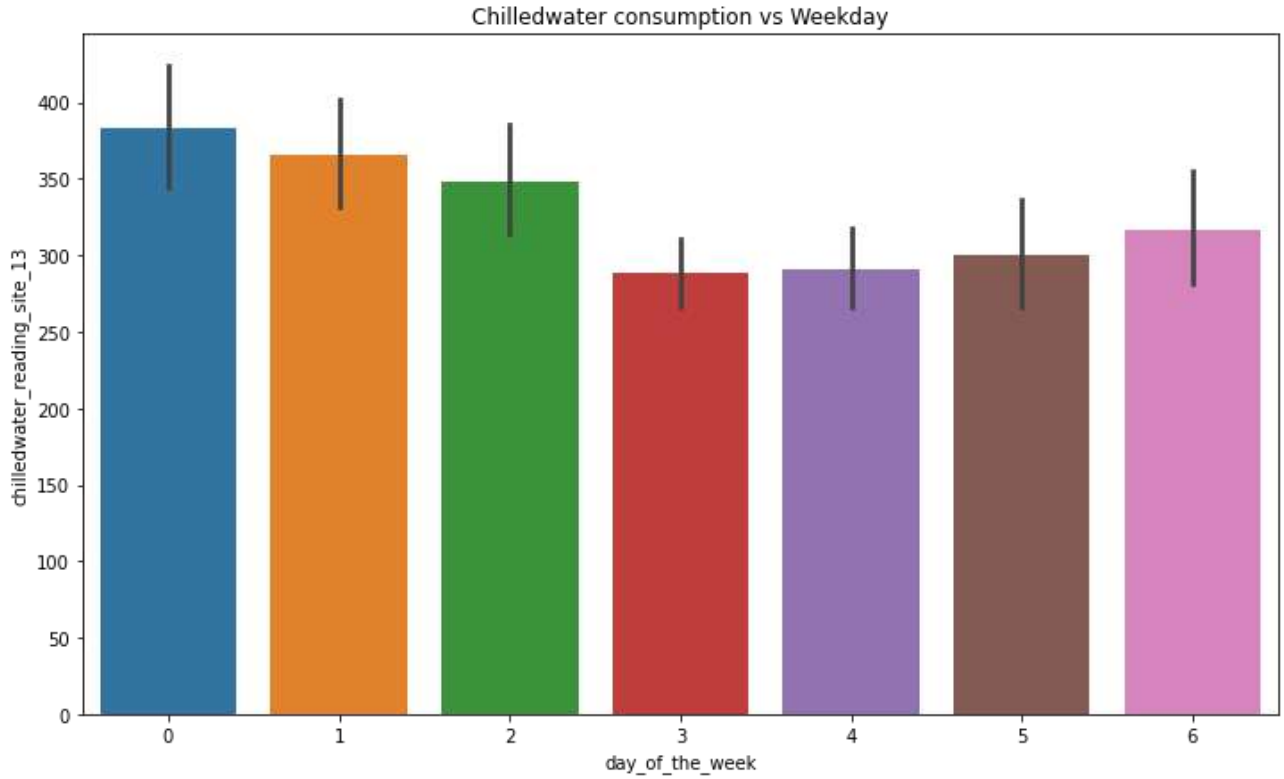

```
z=df_train_site_13_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading_site_13')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



Here we can see that chilledwater consumption is significant for the summer month and it is very significant for the 9th month.We will investigate it further when I will plot the meter readings from all the building with the timestamp.

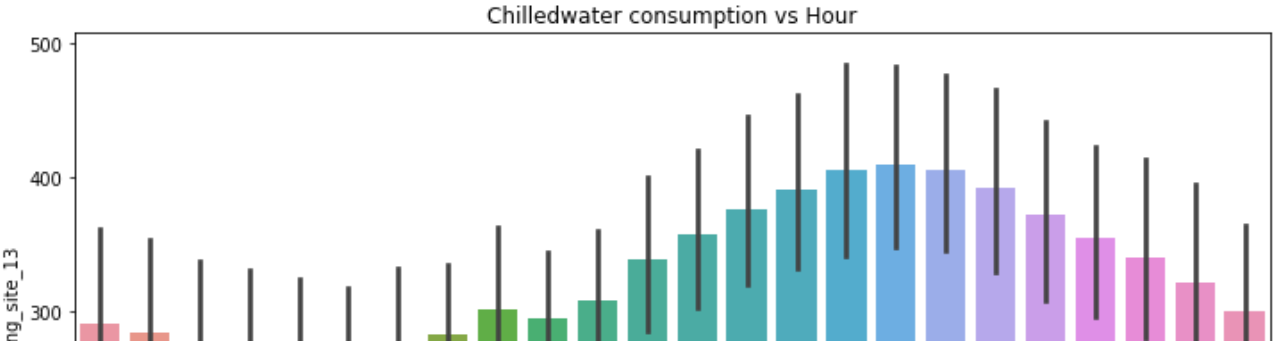
```
z=df_train_site_13_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading_site_13')
```

```
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



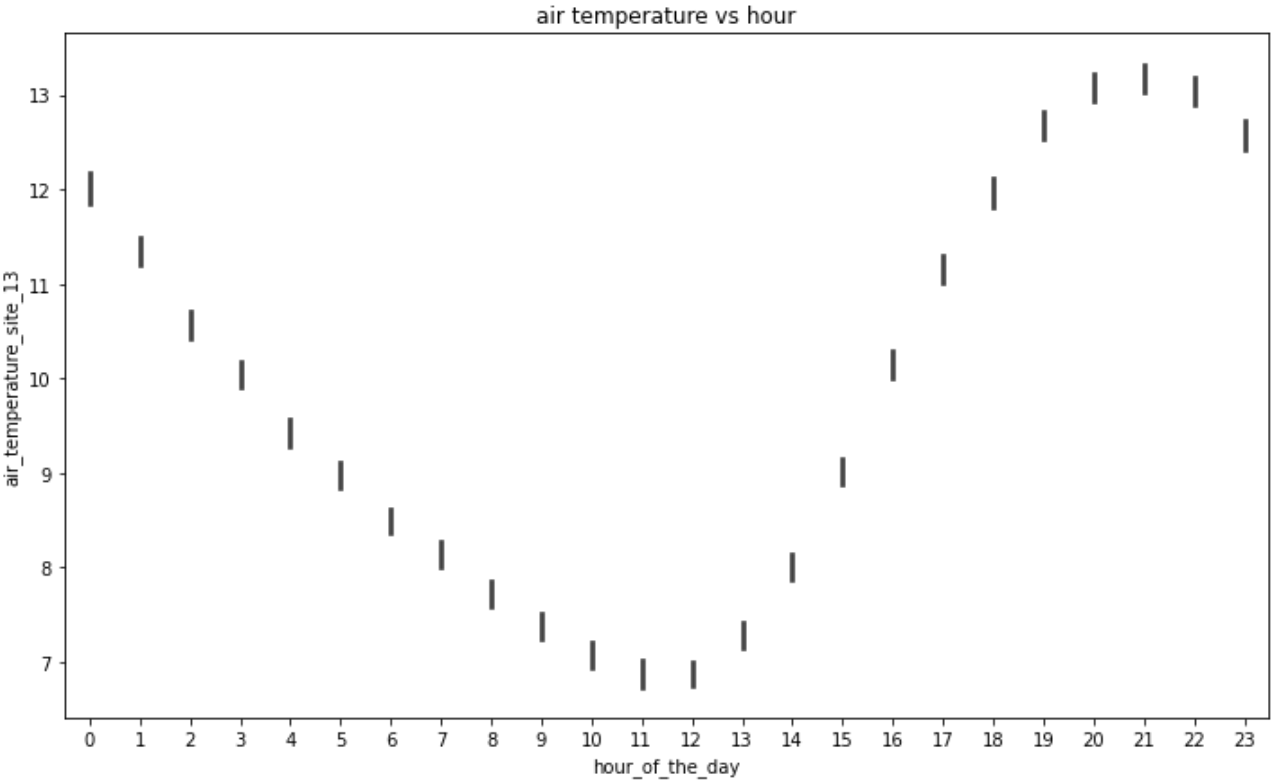
From the above plot we can see that chilledwater consumption varies differently over the week and we do not see any specific pattern.

```
z=df_train_site_13_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilledwater_reading_site_13')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```



From the above plot we can see that chilledwater consumption starts increasing around 6:00 am in the morning and then peaks around 16:00 pm and then decreases rapidly over the day

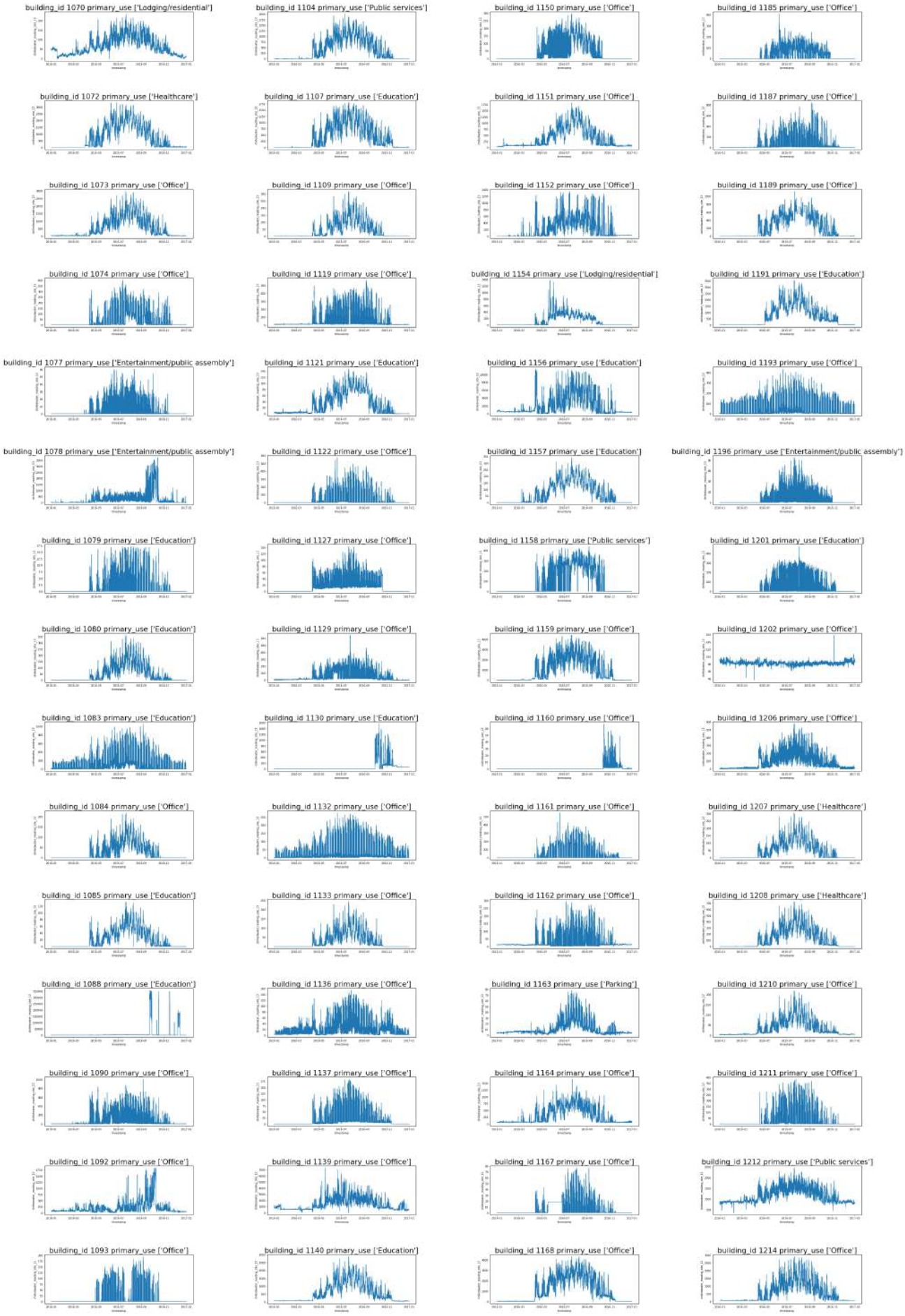
```
z=df_train_site_13_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_13')
plt.title('air temperature vs hour')
plt.show()
```



here we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter reading.The temperature peaks around 21:00 pm

```
fig,axs=plt.subplots(figsize=(55,120),nrows=20,ncols=4,squeeze=True)
for i in range(df_train_site_13_meter_1['building_id'].nunique()):
    g=df_train_site_13_meter_1['building_id'].unique()[i]
    axes=axs[i%20][i//20]
```

```
    z=df_train_site_13_meter_1.loc[df_train_site_13_meter_1['building_id']==g]
    axes.plot(z['timestamp'],z['meter_reading'])
    axes.set_xlabel('timestamp')
    axes.set_ylabel('chilledwater_reading_site_13')
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),font
    plt.subplots_adjust(hspace=0.8,wspace=0.5)
```



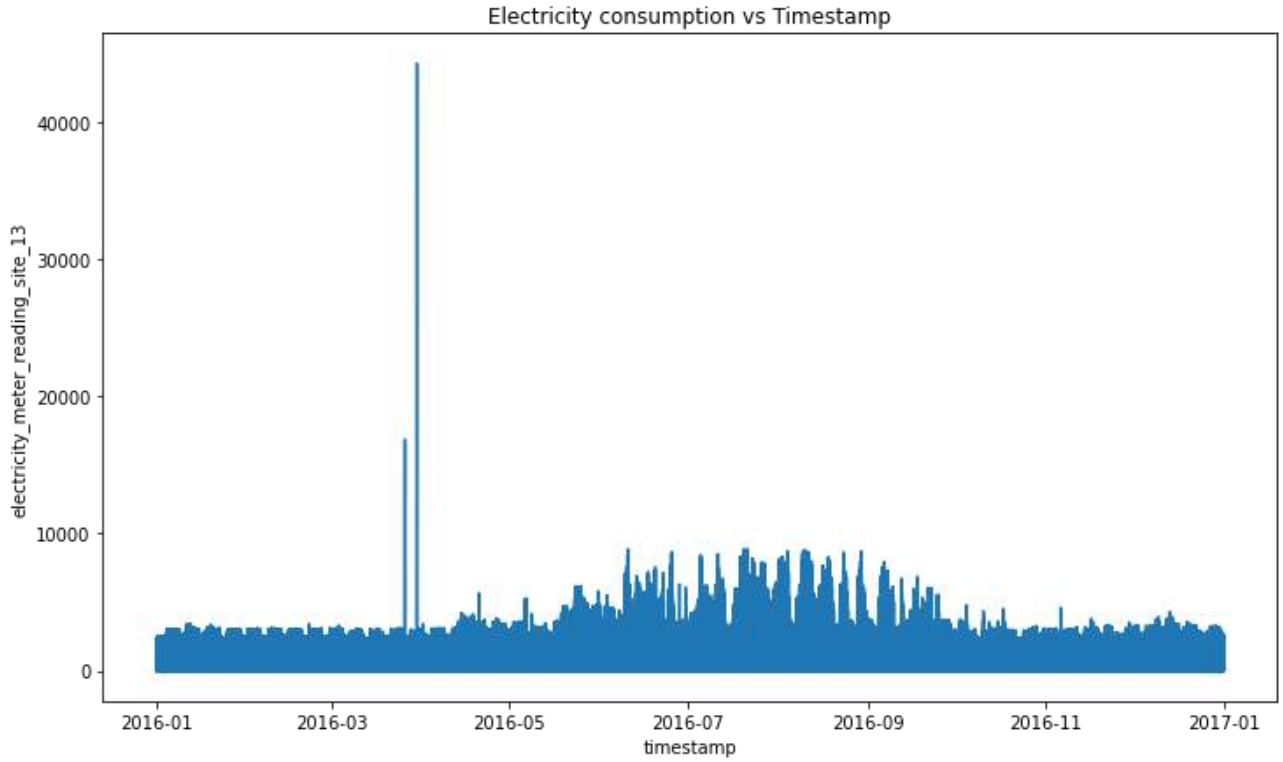
Important Observations

- Building 1088 1220 havelarge spikes that needs to be removed

- Building 1130 1160 have constant zero meter readings that needs to be filtered out.Building 1167 have a constant meter reading in between the normal readings which should be taken care of

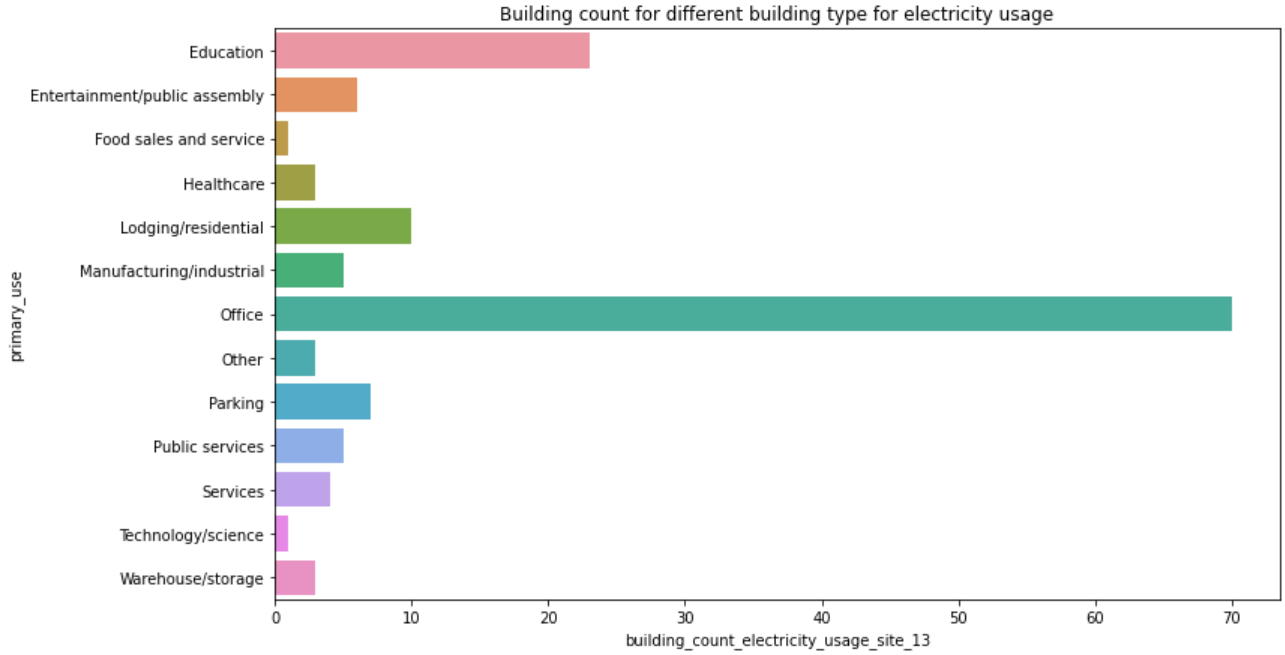
```
df_train_site_13_meter_0=df_train_site_13.loc[df_train_site_13['meter']=='electricity']
```


```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_0
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('electricity_meter_reading_site_13')
plt.title('Electricity consumption vs Timestamp')
plt.show()
```



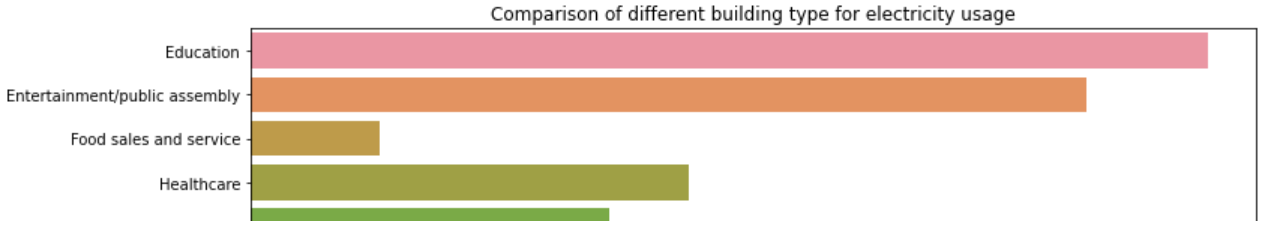
The above plot shows the consumption of overall buildings fro electrcity consumption at site 13.Here we can also see a large spike which is an anomaly and we can further remove the anomalies if we plot it for every building.

```
z=df_train_site_13_meter_0.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_electricity_usage_site_13')
plt.ylabel('primary_use')
plt.title('Building count for different building type for electricity usage')
plt.show()
```



The above plot shows the building count for different building type for eelctricity usage at site 13

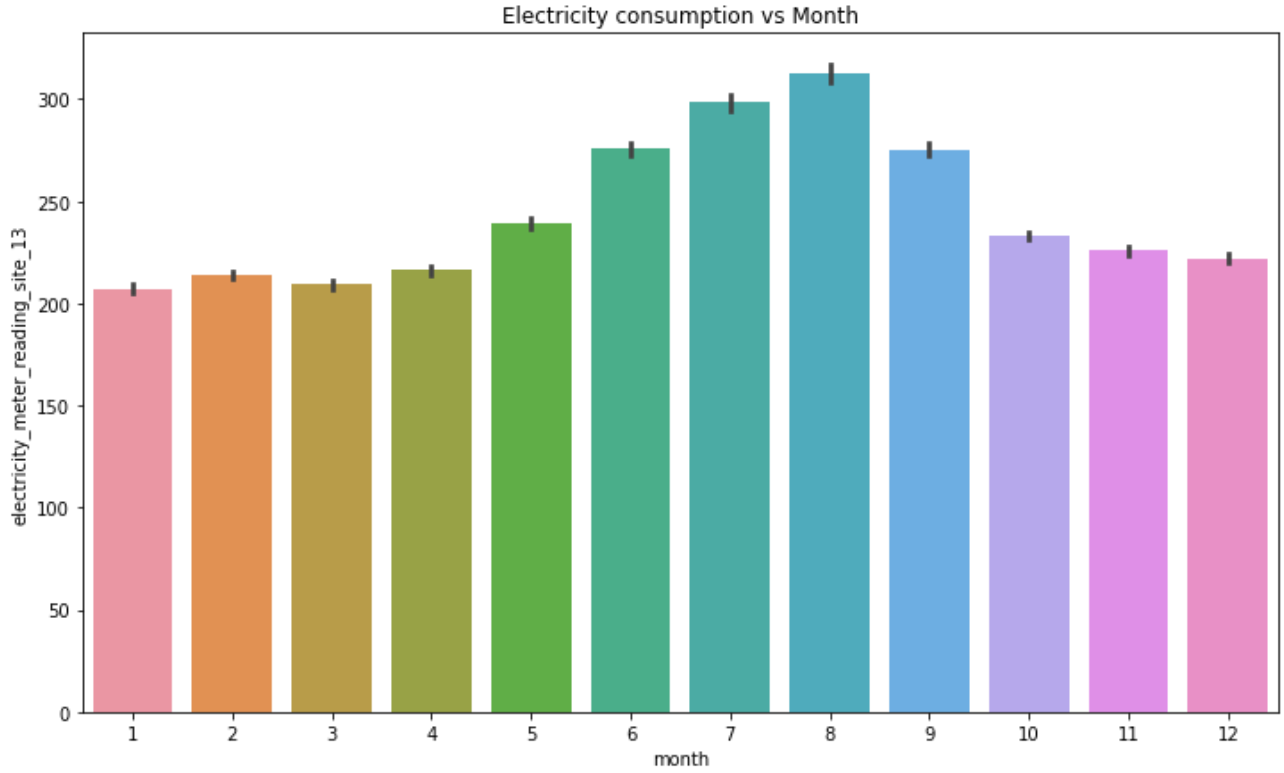
```
z=df_train_site_13_meter_0.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_reading_site_13')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for electricity usage')
plt.show()
```



From the above plot we are observing that Education is ahving the highest electrical consumption.Entertainment and public serivices are also having higher electrical consumption.

```
df_train_site_13_meter_0['month']=df_train_site_13_meter_0['timestamp'].dt.month
df_train_site_13_meter_0['weekday']=df_train_site_13_meter_0['timestamp'].dt.weekday
df_train_site_13_meter_0['hour']=df_train_site_13_meter_0['timestamp'].dt.hour
```

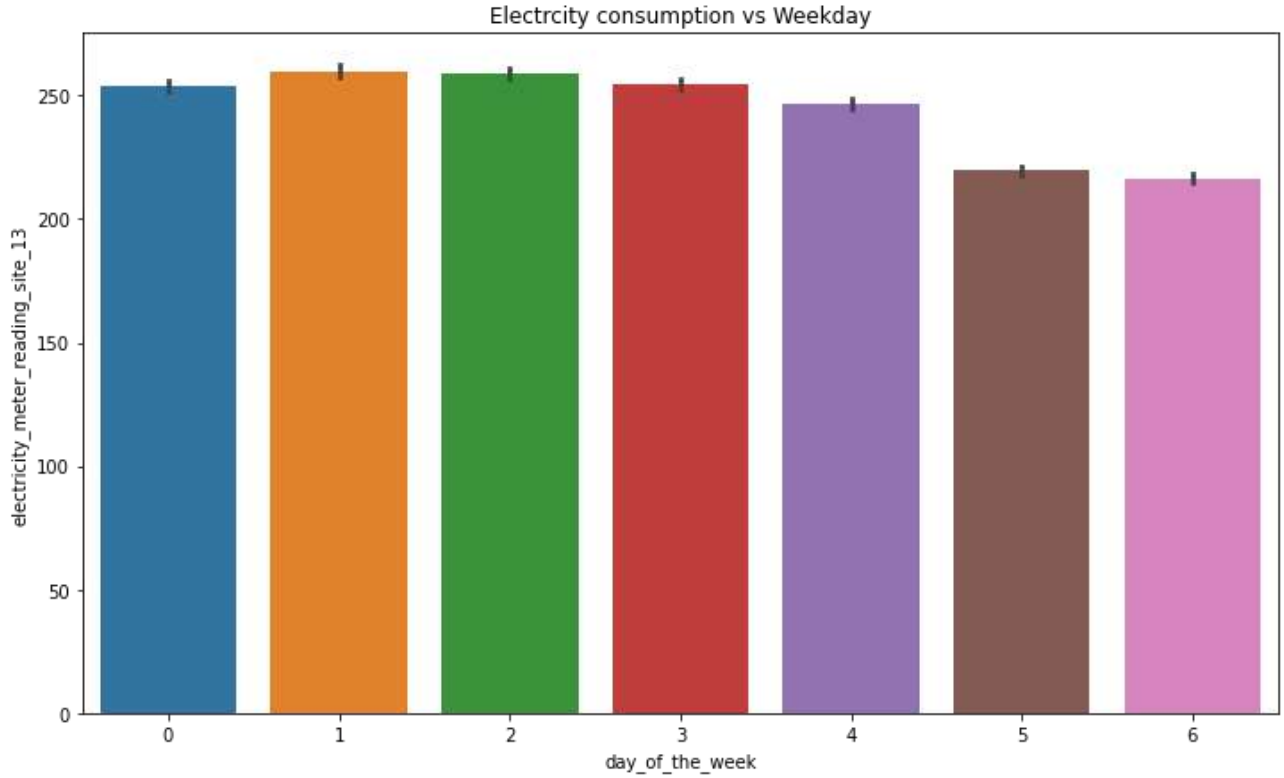
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_meter_reading_site_13')
plt.title('Electricity consumption vs Month')
plt.show()
```



From the above plot we can see that the electricity consumption starts increasing from the 4th month and peaks around 8th month and then again starts decreasing gradually.

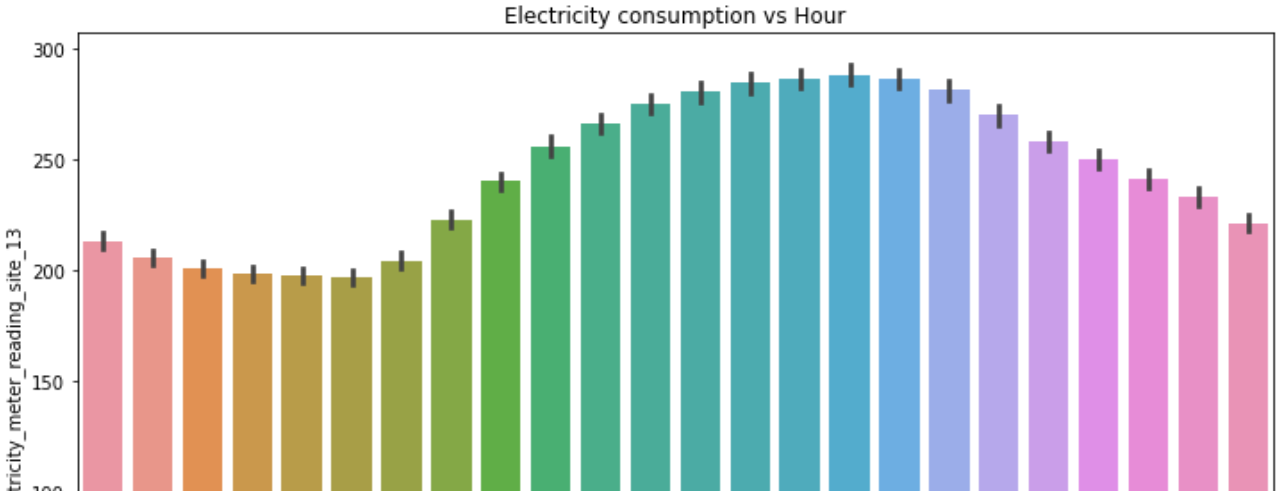
```
fig,ax=plt.subplots(figsize=(12,7))
```

```
z=df_train_site_13_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_meter_reading_site_13')
plt.title('Electrcity consumption vs Weekday')
plt.show()
```



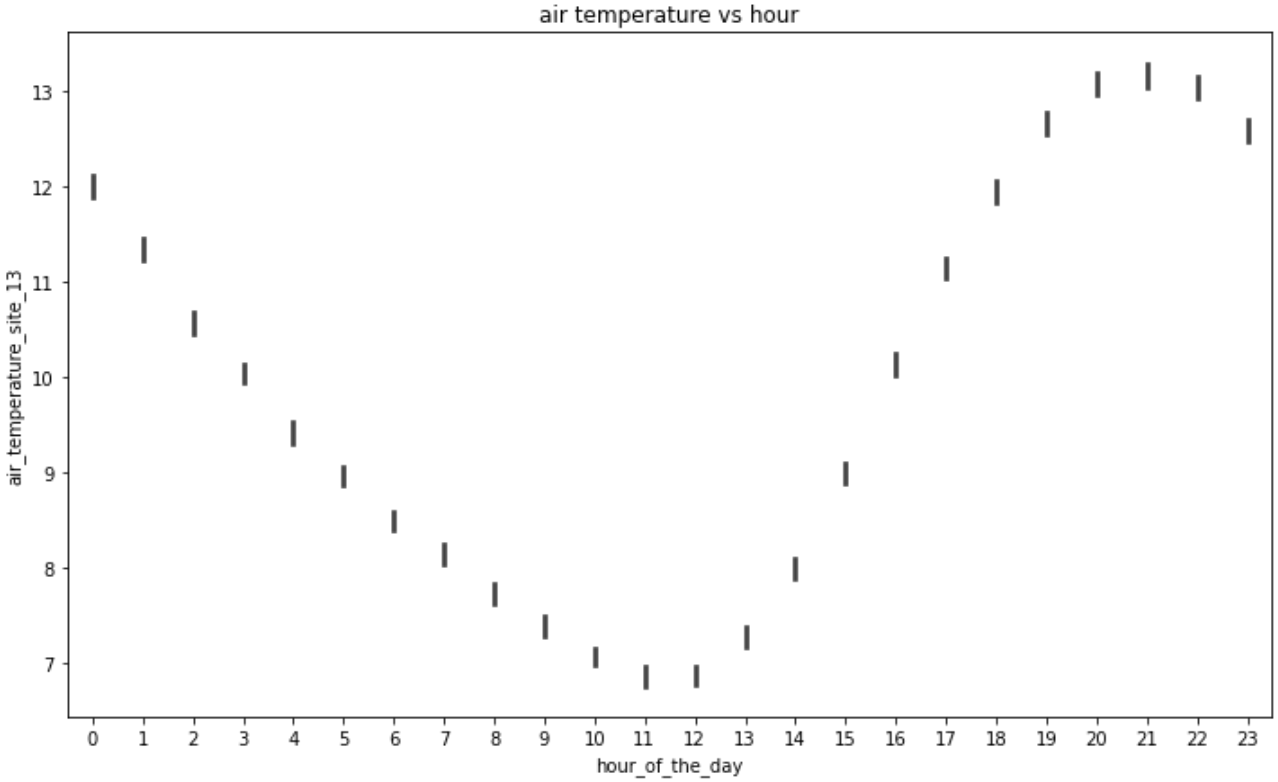
From the above plot we can see that elctricity consumption is less for the weekend as compared to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_meter_reading_site_13')
plt.title('Electricity consumption vs Hour')
plt.show()
```



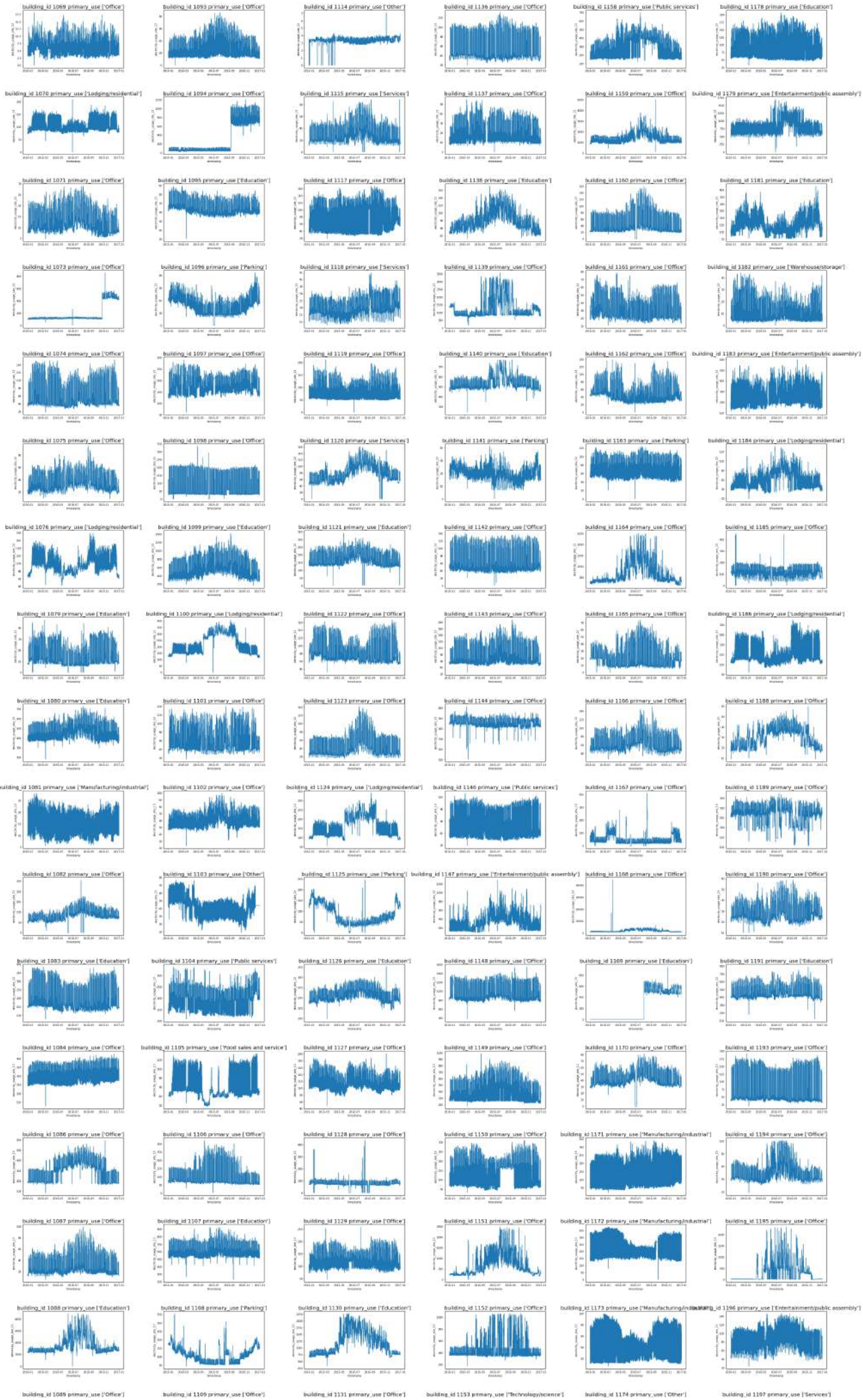
Here we are observing that the electricity consumption starts increasing from 6:00 am in the morning and peaks around 15:00 pm and then starts decreasing gradually.

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_13_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_13')
plt.title('air temperature vs hour')
plt.show()
```

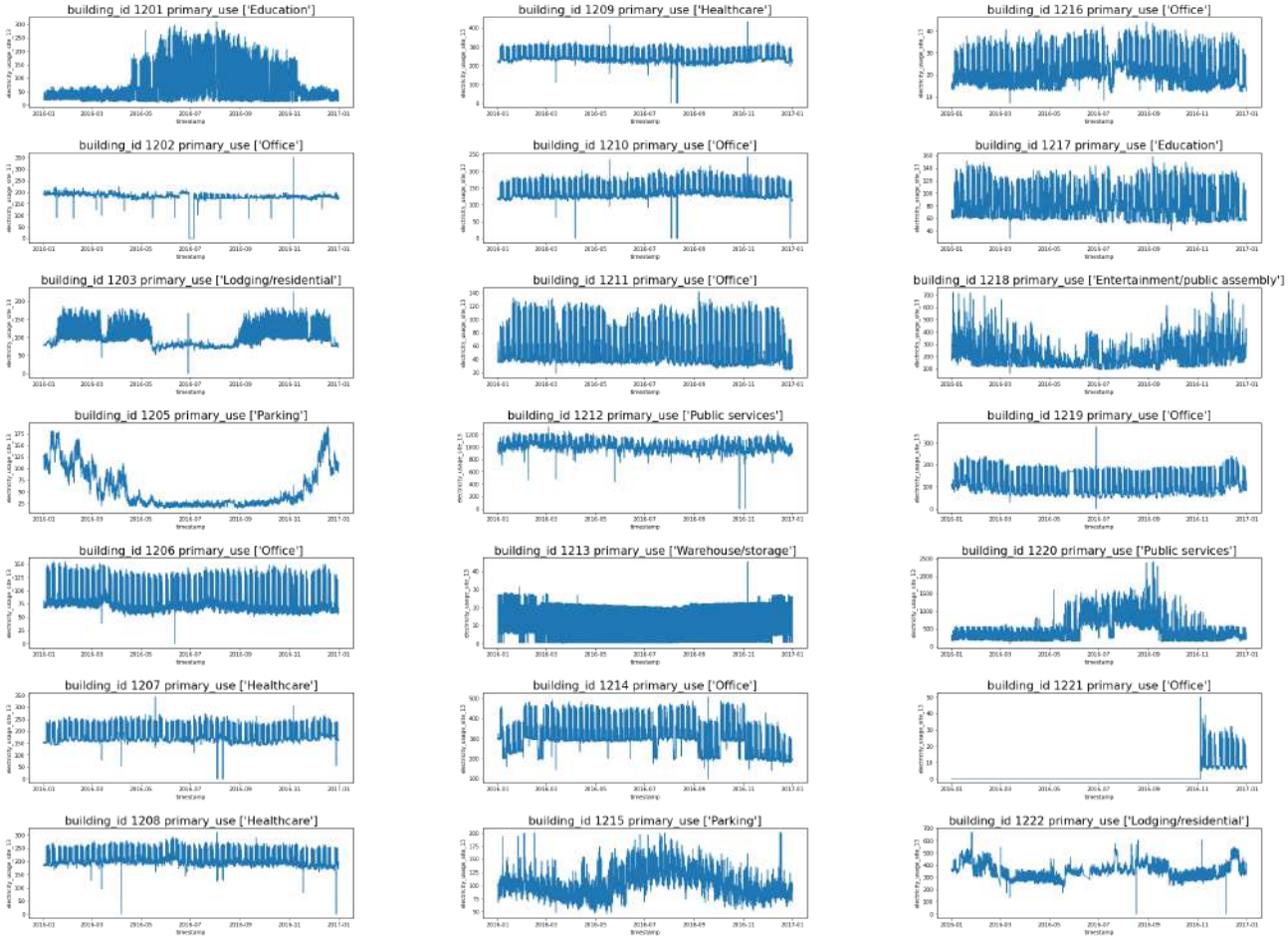


Here we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter reading.The tempearure peaks around 21:00 pm

```
fig,ax=plt.subplots(figsize=(55,120),nrows=20,ncols=6,squeeze=True)
for i in range(120):
  g=df_train_site_13_meter_0['building_id'].unique()[i]
  z=df_train_site_13_meter_0.loc[df_train_site_13_meter_0['building_id']==g]
  axes=ax[i%20][i//20]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_usage_site_13')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.5,wspace=0.4)
```


building_id 1069 primary_use ['Office']
building_id 1093 primary_use ['Office']
building_id 1114 primary_use ['Other']
building_id 1136 primary_use ['Office']
building_id 1158 primary_use ['Public services']
building_id 1178 primary_use ['Education']
building_id 1070 primary_use ['Lodging/residential']
building_id 1094 primary_use ['Office']
building_id 1115 primary_use ['Services']
building_id 1137 primary_use ['Office']
building_id 1159 primary_use ['Office']
building_id 1179 primary_use ['Entertainment/public assembly']
building_id 1071 primary_use ['Office']
building_id 1095 primary_use ['Education']
building_id 1117 primary_use ['Office']
building_id 1138 primary_use ['Education']
building_id 1160 primary_use ['Office']
building_id 1181 primary_use ['Education']
building_id 1073 primary_use ['Office']
building_id 1096 primary_use ['Parking']
building_id 1118 primary_use ['Services']
building_id 1139 primary_use ['Office']
building_id 1161 primary_use ['Office']
building_id 1182 primary_use ['Warehouse/storage']
building_id 1074 primary_use ['Office']
building_id 1097 primary_use ['Office']
building_id 1119 primary_use ['Office']
building_id 1140 primary_use ['Education']
building_id 1162 primary_use ['Office']
building_id 1183 primary_use ['Entertainment/public assembly']
building_id 1075 primary_use ['Office']
building_id 1098 primary_use ['Office']
building_id 1120 primary_use ['Services']
building_id 1141 primary_use ['Parking']
building_id 1163 primary_use ['Parking']
building_id 1184 primary_use ['Lodging/residential']
building_id 1076 primary_use ['Lodging/residential']
building_id 1099 primary_use ['Education']
building_id 1121 primary_use ['Education']
building_id 1142 primary_use ['Office']
building_id 1164 primary_use ['Office']
building_id 1185 primary_use ['Office']
building_id 1079 primary_use ['Education']
building_id 1100 primary_use ['Lodging/residential']
building_id 1122 primary_use ['Office']
building_id 1145 primary_use ['Office']
building_id 1165 primary_use ['Office']
building_id 1186 primary_use ['Lodging/residential']
building_id 1080 primary_use ['Education']
building_id 1101 primary_use ['Office']
building_id 1123 primary_use ['Office']
building_id 1144 primary_use ['Office']
building_id 1166 primary_use ['Office']
building_id 1188 primary_use ['Office']
building_id 1081 primary_use ['Manufacturing/industrial']
building_id 1102 primary_use ['Office']
building_id 1124 primary_use ['Lodging/residential']
building_id 1146 primary_use ['Public services']
building_id 1167 primary_use ['Office']
building_id 1189 primary_use ['Office']
building_id 1082 primary_use ['Office']
building_id 1103 primary_use ['Other']
building_id 1125 primary_use ['Parking']
building_id 1147 primary_use ['Entertainment/public assembly']
building_id 1168 primary_use ['Office']
building_id 1190 primary_use ['Office']
building_id 1083 primary_use ['Education']
building_id 1104 primary_use ['Public services']
building_id 1126 primary_use ['Education']
building_id 1148 primary_use ['Office']
building_id 1169 primary_use ['Education']
building_id 1191 primary_use ['Education']
building_id 1084 primary_use ['Office']
building_id 1105 primary_use ['Food sales and service']
building_id 1127 primary_use ['Office']
building_id 1149 primary_use ['Office']
building_id 1170 primary_use ['Office']
building_id 1193 primary_use ['Office']
building_id 1086 primary_use ['Office']
building_id 1106 primary_use ['Office']
building_id 1128 primary_use ['Office']
building_id 1150 primary_use ['Office']
building_id 1171 primary_use ['Manufacturing/industrial']
building_id 1194 primary_use ['Office']
building_id 1087 primary_use ['Office']
building_id 1107 primary_use ['Education']
building_id 1129 primary_use ['Office']
building_id 1151 primary_use ['Office']
building_id 1172 primary_use ['Manufacturing/industrial']
building_id 1195 primary_use ['Office']
building_id 1088 primary_use ['Education']
building_id 1108 primary_use ['Parking']
building_id 1130 primary_use ['Education']
building_id 1152 primary_use ['Office']
building_id 1173 primary_use ['Manufacturing/industrial']
building_id 1196 primary_use ['Entertainment/public assembly']
building_id 1089 primary_use ['Office']
building_id 1109 primary_use ['Office']
building_id 1131 primary_use ['Office']
building_id 1153 primary_use ['Technology/science']
building_id 1174 primary_use ['Other']
building_id 1197 primary_use ['Services']

```
fig,ax=plt.subplots(figsize=(40,30),nrows=7,ncols=3,squeeze=True)
for i in range(21):
  g=df_train_site_13_meter_0['building_id'].unique()[120:142][i]
  z=df_train_site_13_meter_0.loc[df_train_site_13_meter_0['building_id']==g]
  axes=ax[i%7][i//7]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_usage_site_13')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.5,wspace=0.4)
```



The total number of buildings consuming electricity reading at site 13 is 141 therefore I divided that into 120 and 21 buildings for better representation.

Important Observations

- Building 1079 1096 1113 1154 1160 1169 1170 1189 1221 have constant meter readings which needs to be filtered out.
- There are also many buildings which have spikes that might be due to faulty reading and it needs to be removed.

```
#Starting analysis for site 14
```

```
df_train_site_14=df_train_merge.loc[df_train_merge['site_id']==14]
```

```
df_train_site_14.isnull().sum()/df_train_site_14.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           1.00
floor_count          1.00
air_temperature      0.00
cloud_coverage       0.38
dew_temperature      0.00
precip_depth_1_hr    0.00
sea_level_pressure   0.01
wind_direction       0.03
wind_speed           0.00
dtype: float64
```

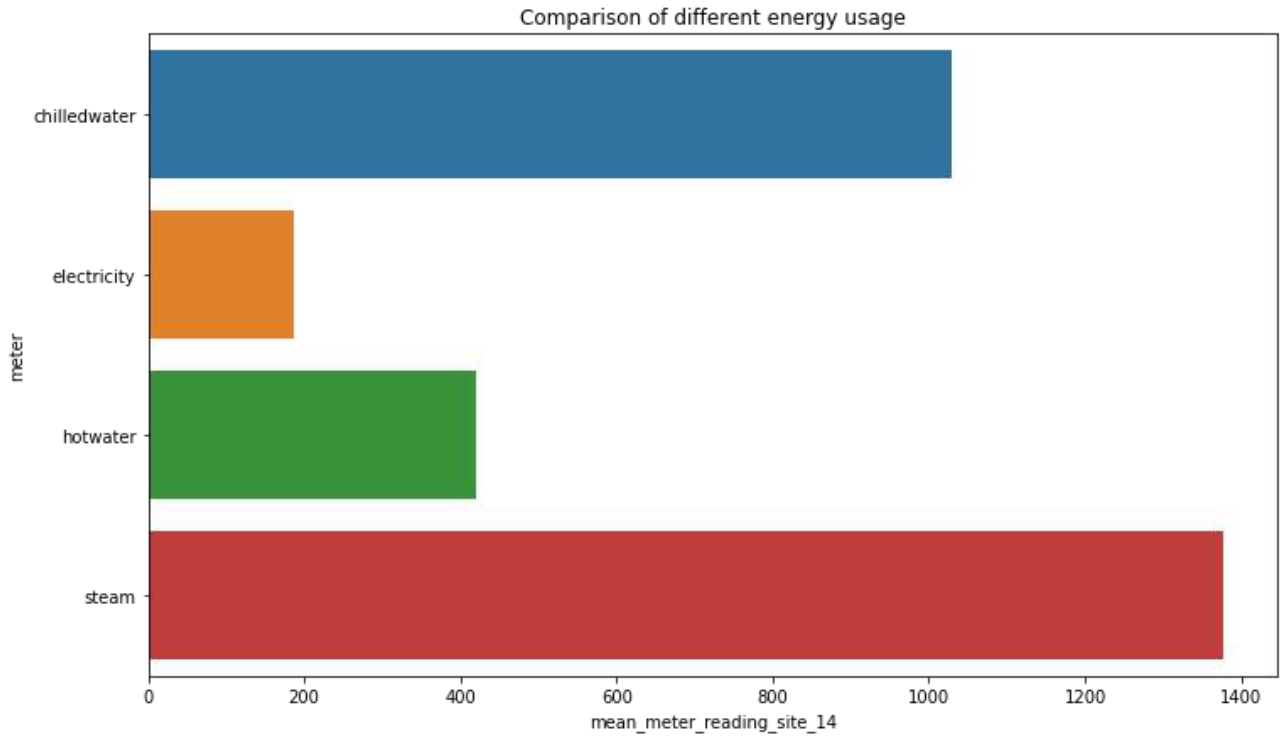
Here we can see that site 14 has some null values which needs to be imputed

```
df_corr_14=df_train_site_14.corr()
df_corr_14.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	0.05	nan	0.10	nan	nan	0.00
meter_reading	0.05	1.00	nan	0.46	nan	nan	0.04
site_id	nan	nan	nan	nan	nan	nan	nan
square_feet	0.10	0.46	nan	1.00	nan	nan	-0.00
year_built	nan	nan	nan	nan	nan	nan	nan
floor_count	nan	nan	nan	nan	nan	nan	nan
air_temperature	0.00	0.04	nan	-0.00	nan	nan	1.00
cloud_coverage	-0.00	0.01	nan	0.00	nan	nan	0.06
dew_temperature	0.00	0.03	nan	-0.00	nan	nan	0.89
precip_depth_1_hr	-0.00	-0.00	nan	-0.00	nan	nan	0.02
sea_level_pressure	0.00	-0.01	nan	0.00	nan	nan	-0.25
wind_direction	-0.00	0.00	nan	0.00	nan	nan	-0.04
wind_speed	-0.00	0.00	nan	-0.00	nan	nan	-0.04

From the above correlarion plot we can see that the meter reading is not strongly correlated with any of the features

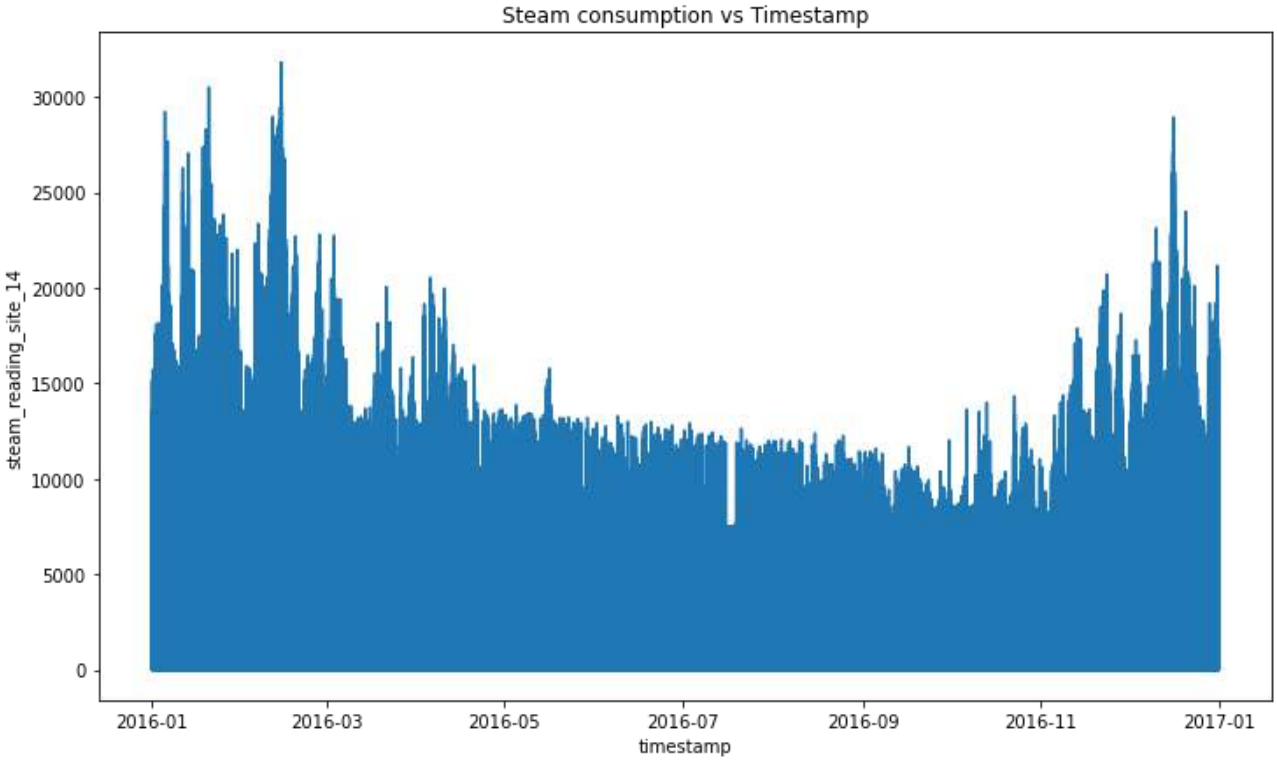
```
z=df_train_site_14.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading_site_14')
plt.ylabel('meter')
plt.title('Comparison of different energy usage')
plt.show()
```



At site 15 Steam is having the highest energy consumption.

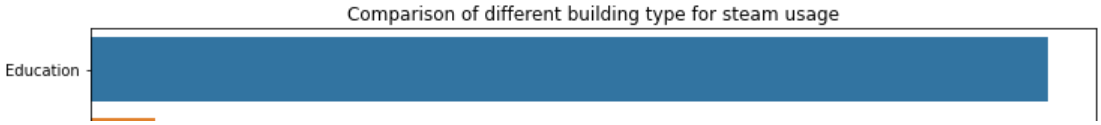
```
df_train_site_14_meter_2=df_train_site_14.loc[df_train_site_14['meter']=='steam']
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_2
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('steam_reading_site_14')
plt.title('Steam consumption vs Timestamp')
plt.show()
```



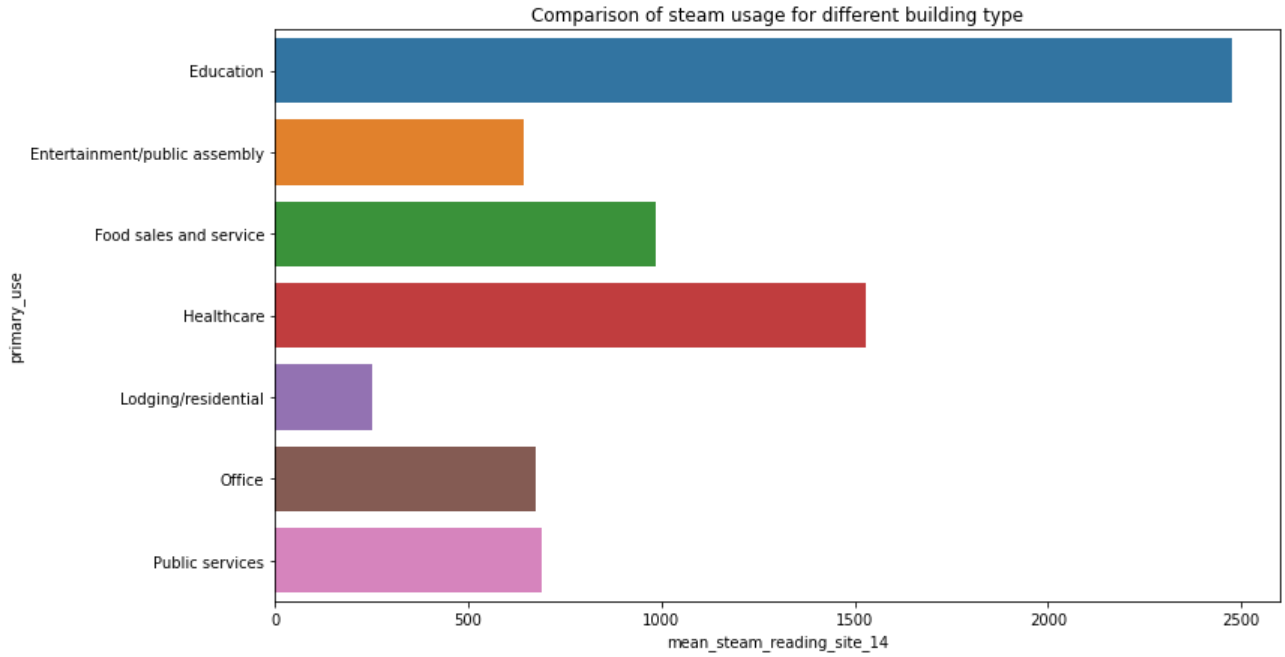
The above plot shows an overall steam consumption of all the buildings over the timestamp

```
z=df_train_site_14_meter_2.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_steam_usage_site_14')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for steam usage')
plt.show()
```



The above plot shows the building count for different building type for steam consumption.

```
z=df_train_site_14_meter_2.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_steam_reading_site_14')
plt.ylabel('primary_use')
plt.title('Comparison of steam usage for different building type')
plt.show()
```

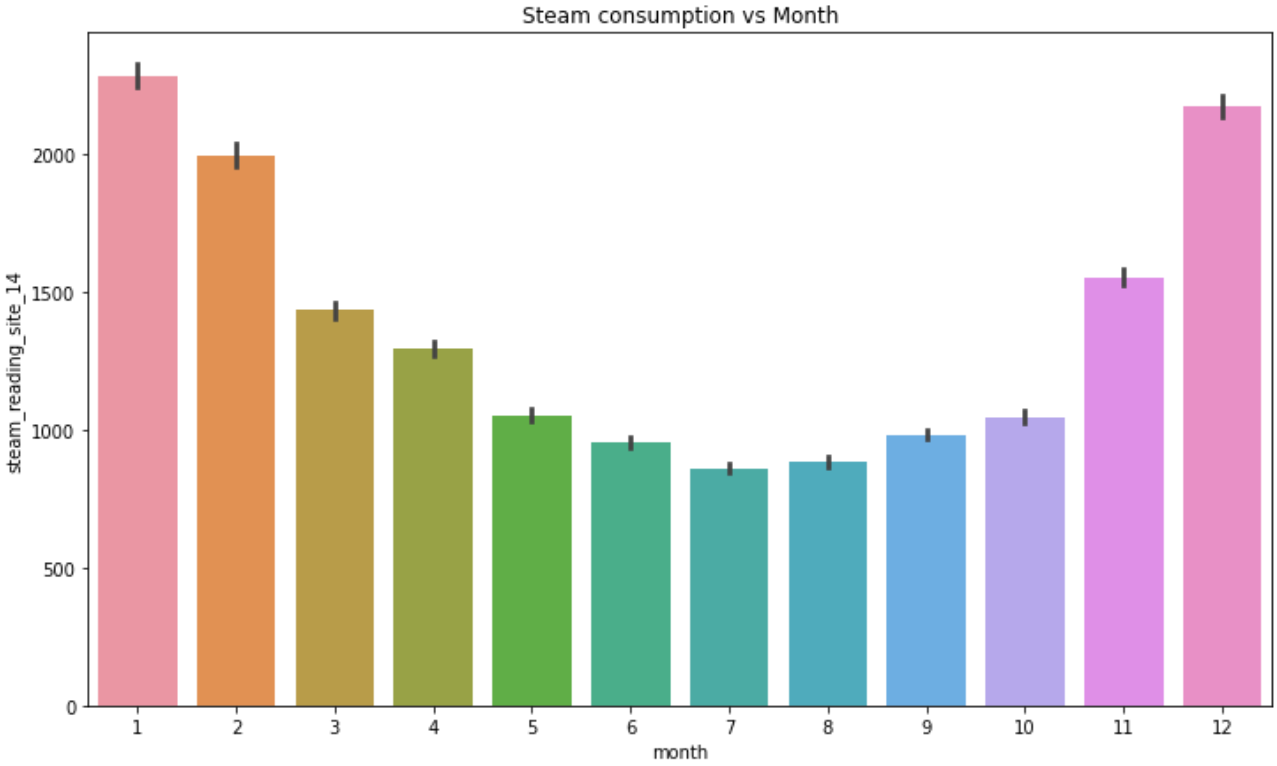


From the above plot we can see that on an average the educational buildings are having the highest steam consumption and from the count plot we can also notice that these buildings are largest in number.

```
df_train_site_14_meter_2['month']=df_train_site_14_meter_2['timestamp'].dt.month
df_train_site_14_meter_2['weekday']=df_train_site_14_meter_2['timestamp'].dt.weekday
df_train_site_14_meter_2['hour']=df_train_site_14_meter_2['timestamp'].dt.hour
```

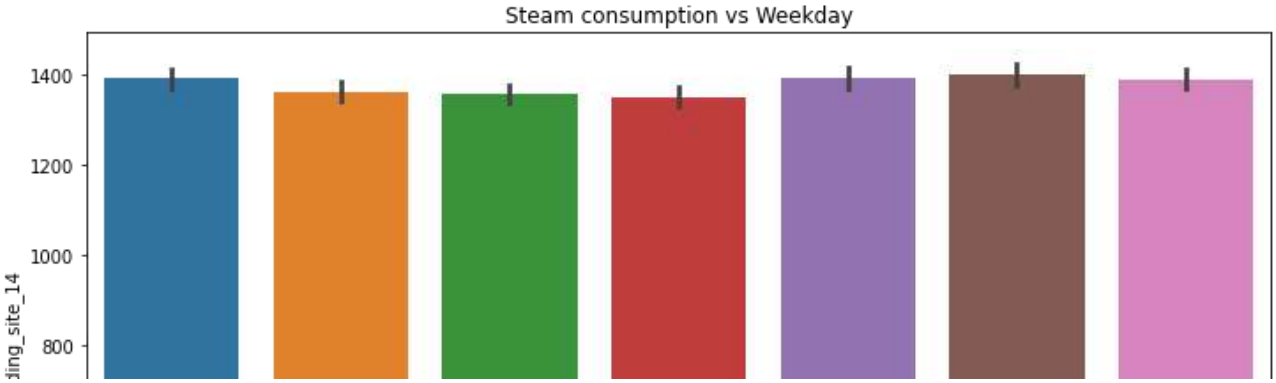
```
z=df_train_site_14_meter_2
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
```

```
plt.ylabel('steam_reading_site_14')
plt.title('Steam consumption vs Month')
plt.show()
```



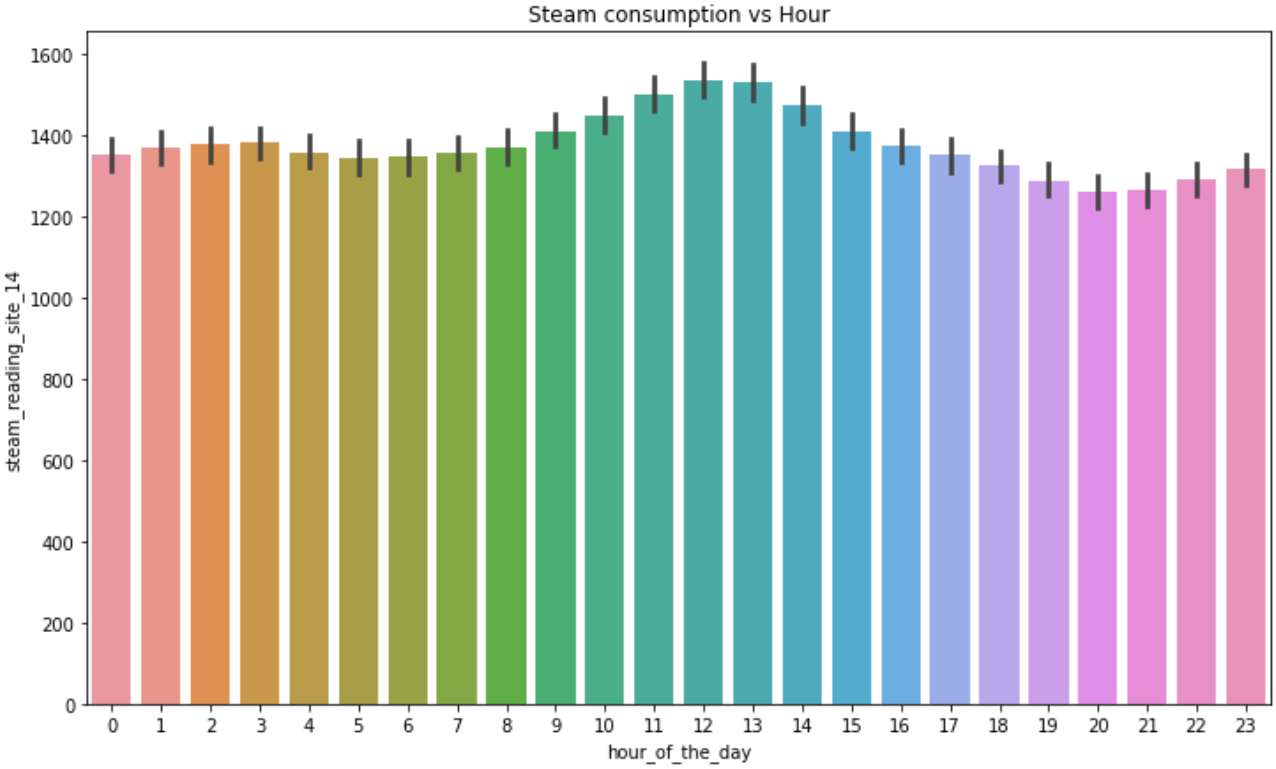
Here we can see that steam consumption is more towards the winter months and decreases gradually as we approach the summer month

```
z=df_train_site_14_meter_2
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('steam_reading_site_14')
plt.title('Steam consumption vs Weekday')
plt.show()
```



Here we can see that steam consumption does not show a specific pattern over the week

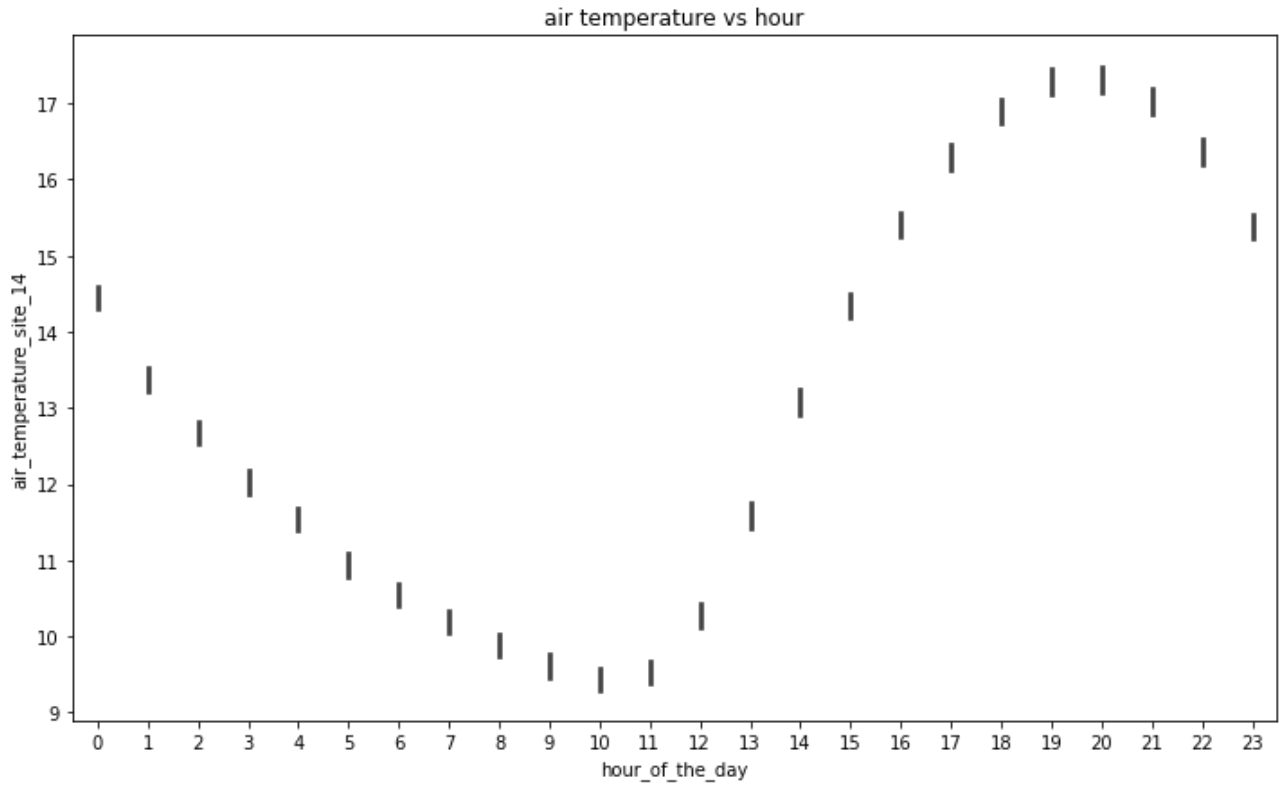
```
z=df_train_site_14_meter_2
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('steam_reading_site_14')
plt.title('Steam consumption vs Hour')
plt.show()
```



From the above plot we can see that steam consumption starts increasing from 6:00 am in the morning and peaks around 12:00 am in the daytime and after which it starts to decrease gradually.

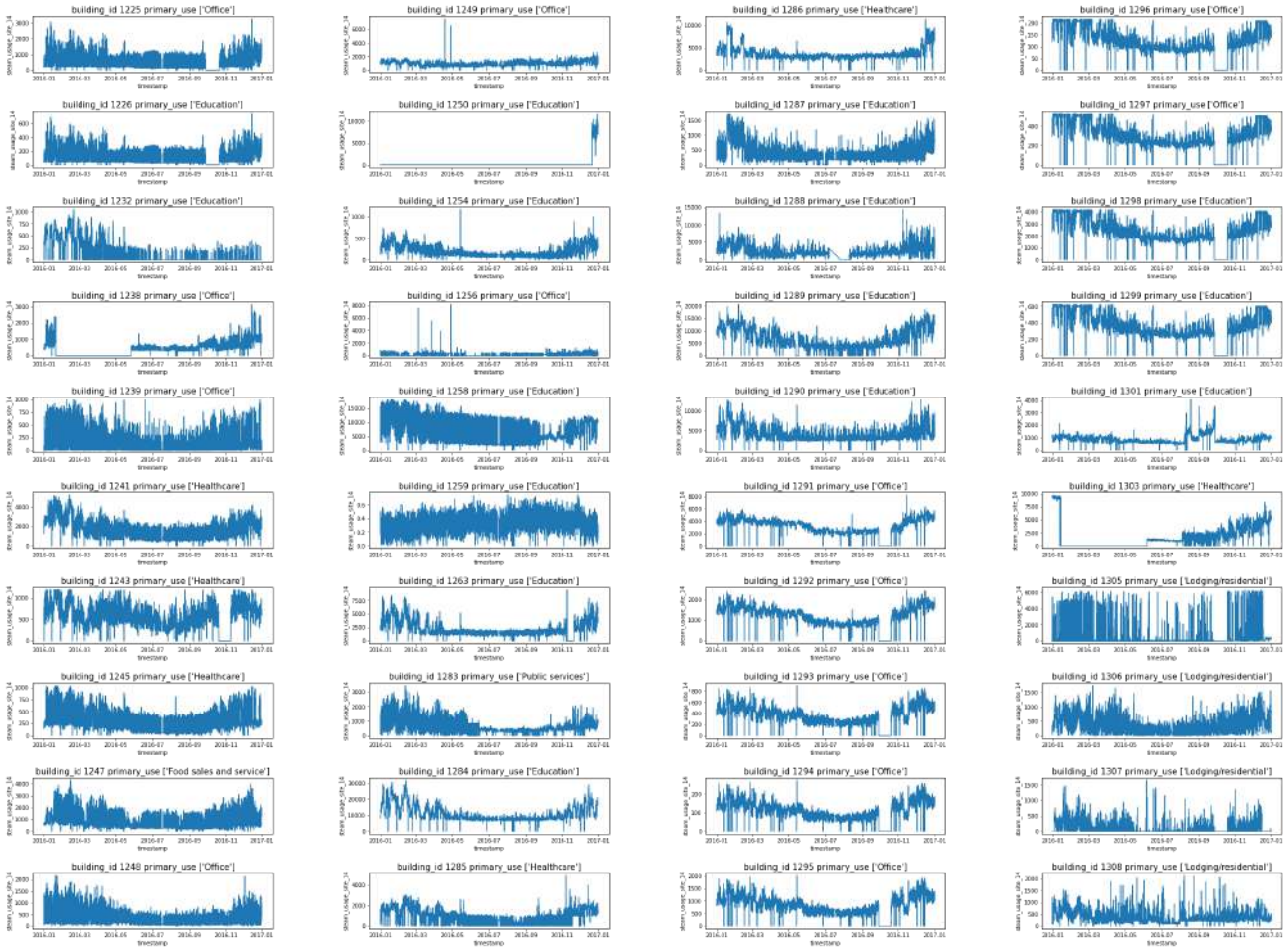
```
z=df_train_site_14_meter_2
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
```

```
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_14')
plt.title('air temperature vs hour')
plt.show()
```

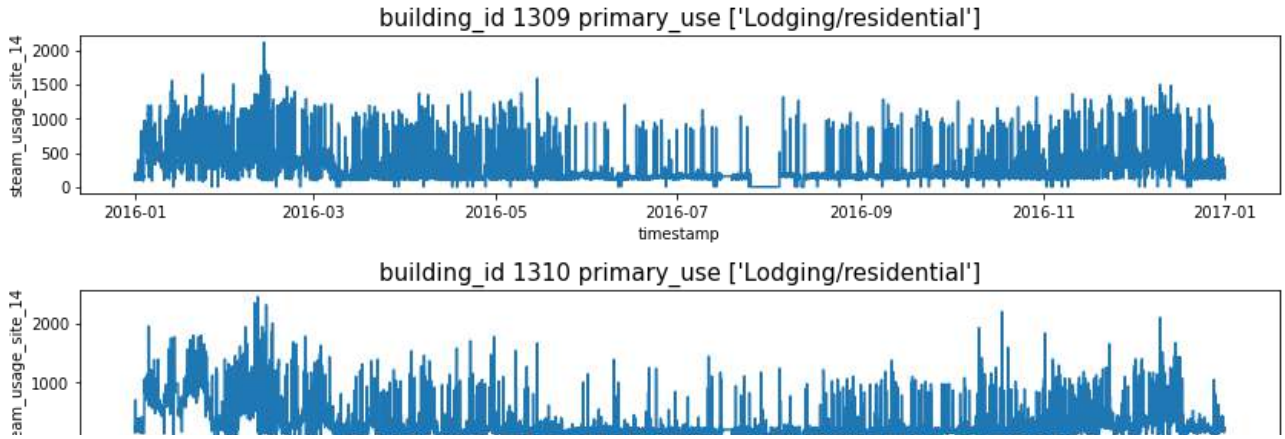


Here we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings.The temperature peaks around 19:00 pm

```
fig,ax=plt.subplots(figsize=(40,30),nrows=10,ncols=4,squeeze=True)
for i in range(40):
  g=df_train_site_14_meter_2['building_id'].unique()[i]
  z=df_train_site_14_meter_2.loc[df_train_site_14_meter_2['building_id']==g]
  axes=ax[i%10][i//10]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('steam_usage_site_14')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.7,wspace=0.4)
```




```
fig,ax=plt.subplots(figsize=(14,8),nrows=3,ncols=1,squeeze=False)
for i in range(3):
  g=df_train_site_14_meter_2['building_id'].unique()[40:44][i]
  z=df_train_site_14_meter_2.loc[df_train_site_14_meter_2['building_id']==g]
  axes=ax[i%10][i//10]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('steam_usage_site_14')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.6)
```



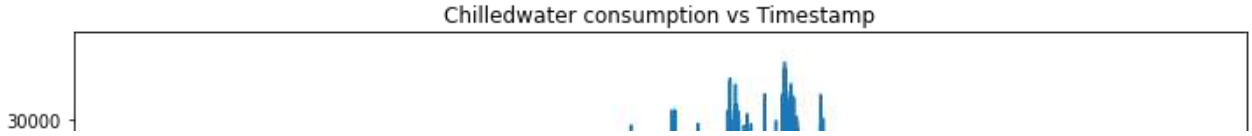
There are total of 43 buildings at site 14 which are consuming steam therefore for better representation I divided that into 40 and 3 buildings.

Important Observations

- There are many buildings which are having constant zero meter readings which needs to be filtered out.
- These builings are also filled with spikes which needs to be removed as it migh be due to faulty readings.

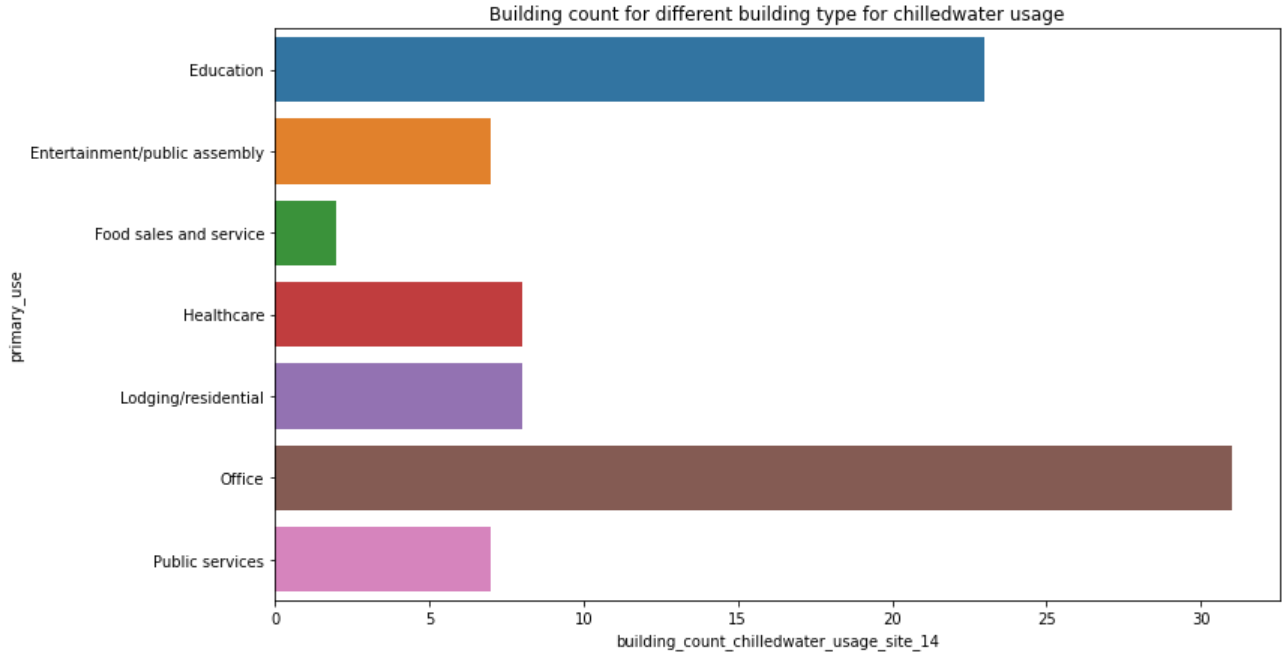
```
df_train_site_14_meter_1=df_train_site_14.loc[df_train_site_14['meter']=='chilledwater']
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_1
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('chilledwater_site_14')
plt.title('Chilledwater consumption vs Timestamp')
plt.show()
```



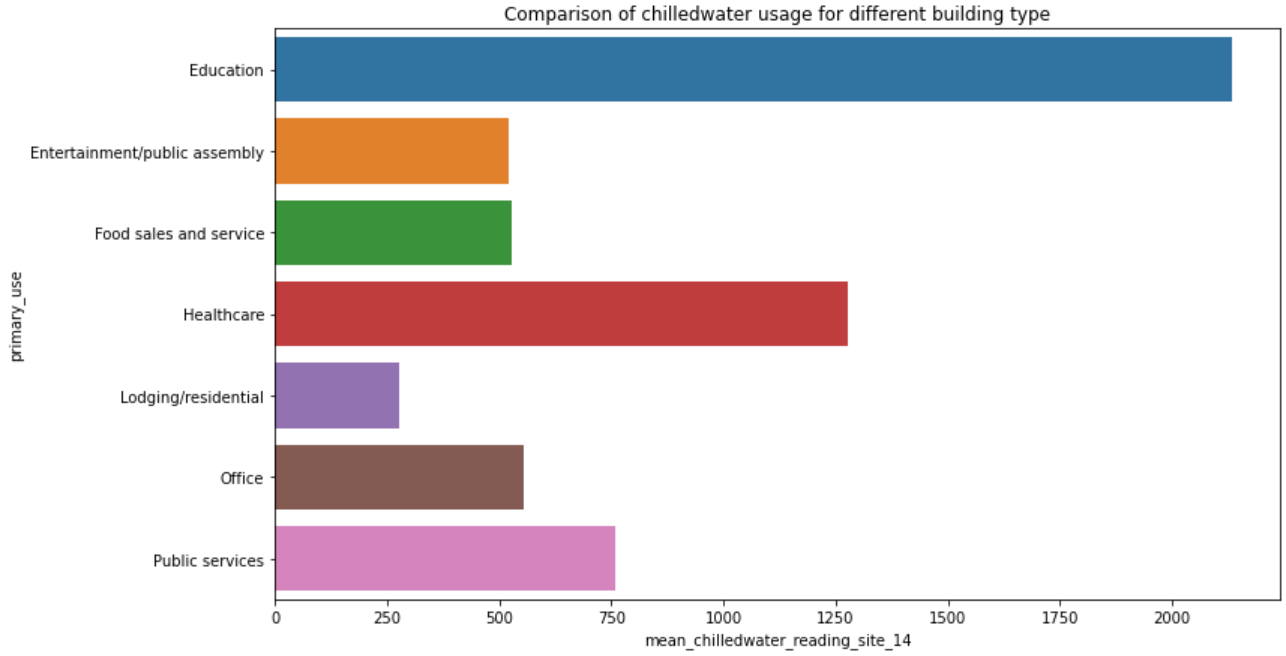
The above plot shows the chilledwater consumption for the overall buildings and we can observe that it is having higher consumption during the summer months

```
z=df_train_site_14_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_chilledwater_usage_site_14')
plt.ylabel('primary_use')
plt.title('Building count for different building type for chilledwater usage')
plt.show()
```



The above plot represents the building count for different building type for chilledwater usage.Here office is having the highest number after that educational buildings are present.

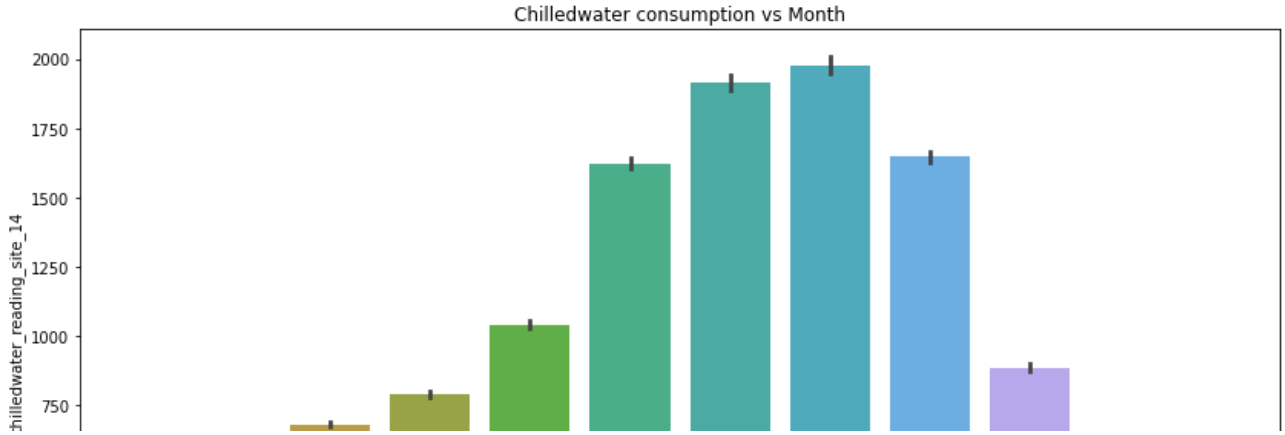
```
z=df_train_site_14_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading_site_14')
plt.ylabel('primary_use')
plt.title('Comparison of chilledwater usage for different building type')
plt.show()
```



Here we can see that educational buildings are showing the highest consumption of chilledwater.Although office buildings were large in number it is consumong less chilledwater

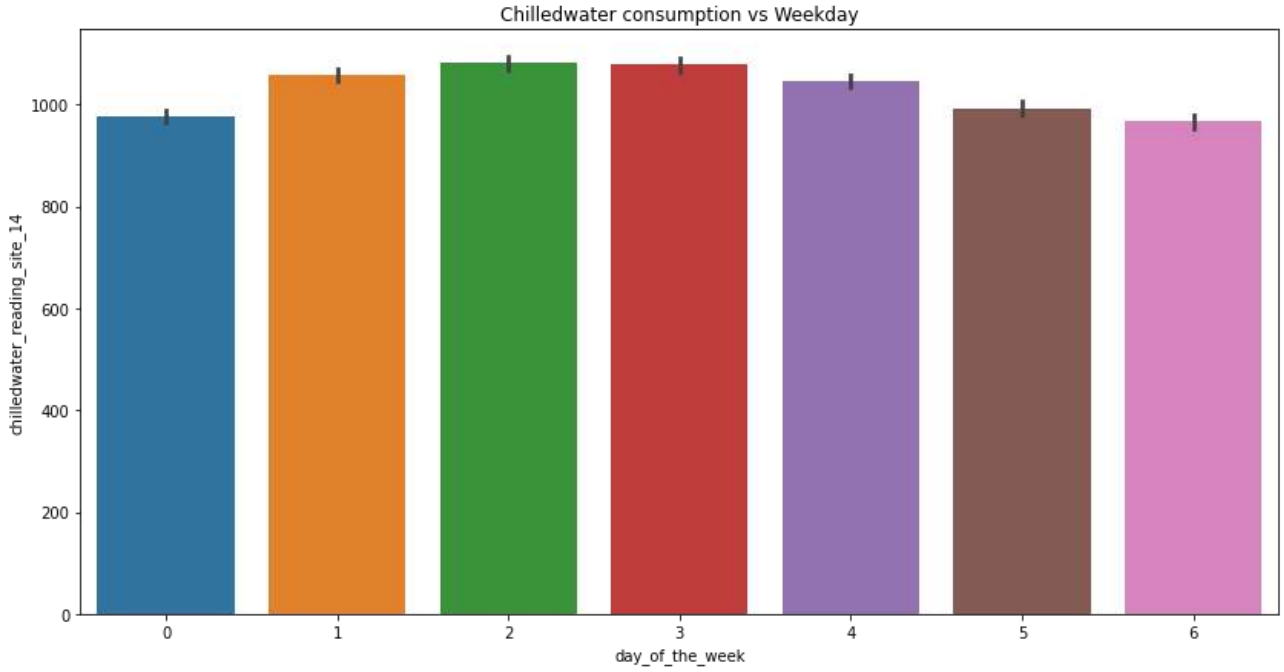
```
df_train_site_14_meter_1['month']=df_train_site_14_meter_1['timestamp'].dt.month
df_train_site_14_meter_1['weekday']=df_train_site_14_meter_1['timestamp'].dt.weekday
df_train_site_14_meter_1['hour']=df_train_site_14_meter_1['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_14_meter_1
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading_site_14')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



Here we can see that chilledwater consumption shows bhigher usage during the summer month

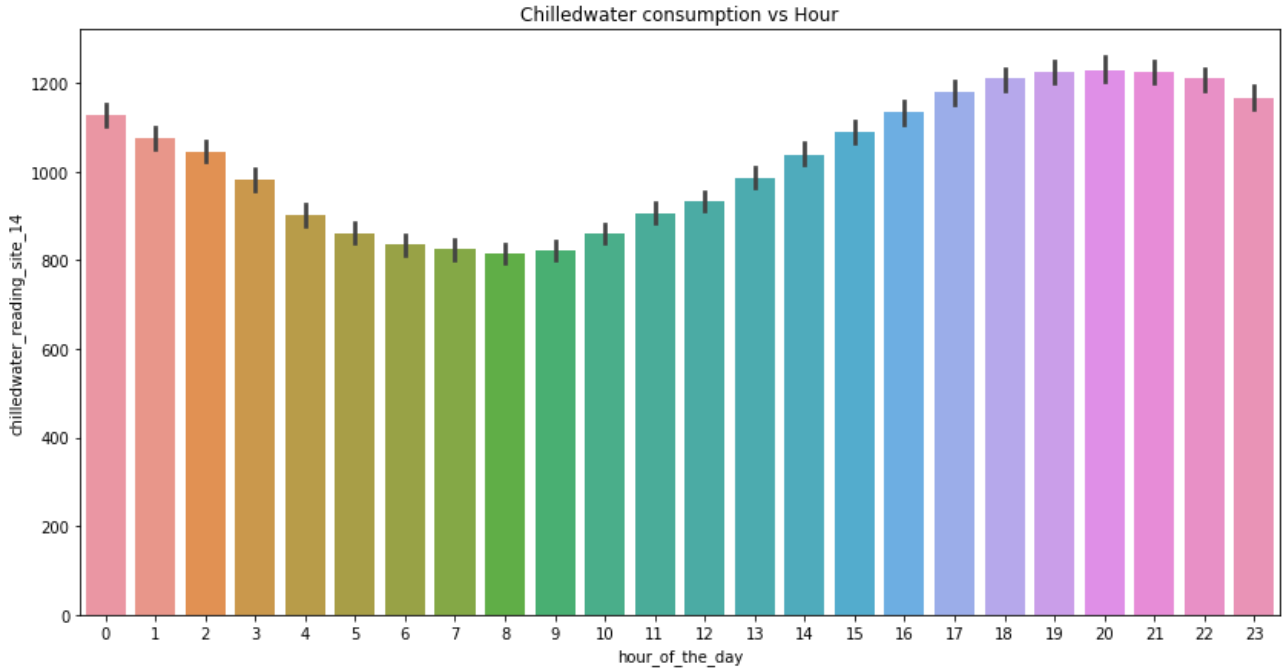
```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_14_meter_1
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading_site_14')
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



Chilledwater consumption varies over the week but it shows a little less consumption over the weekend

```
fig,ax=plt.subplots(figsize=(14,7))
```

```
z=df_train_site_14_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilledwater_reading_site_14')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```

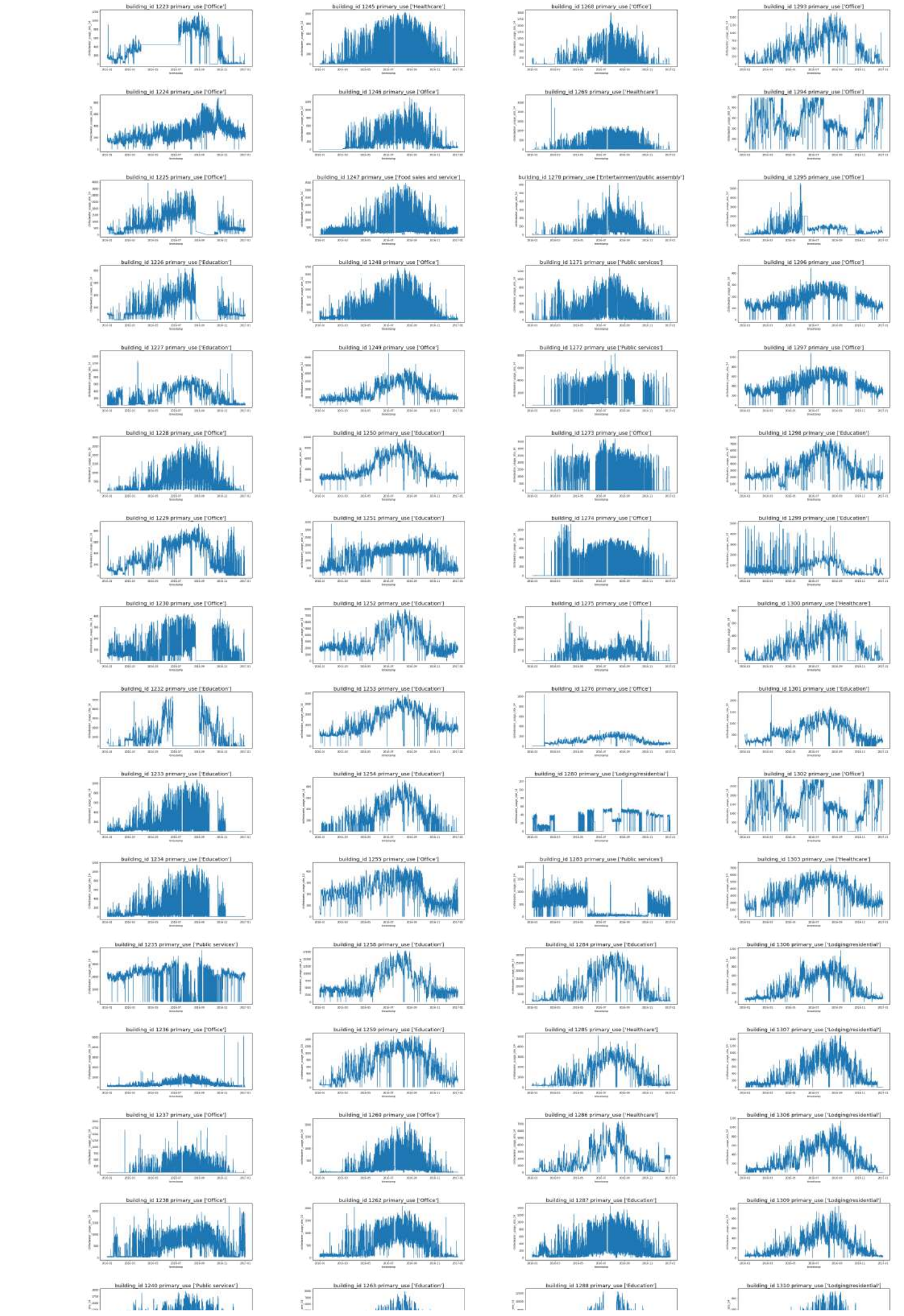


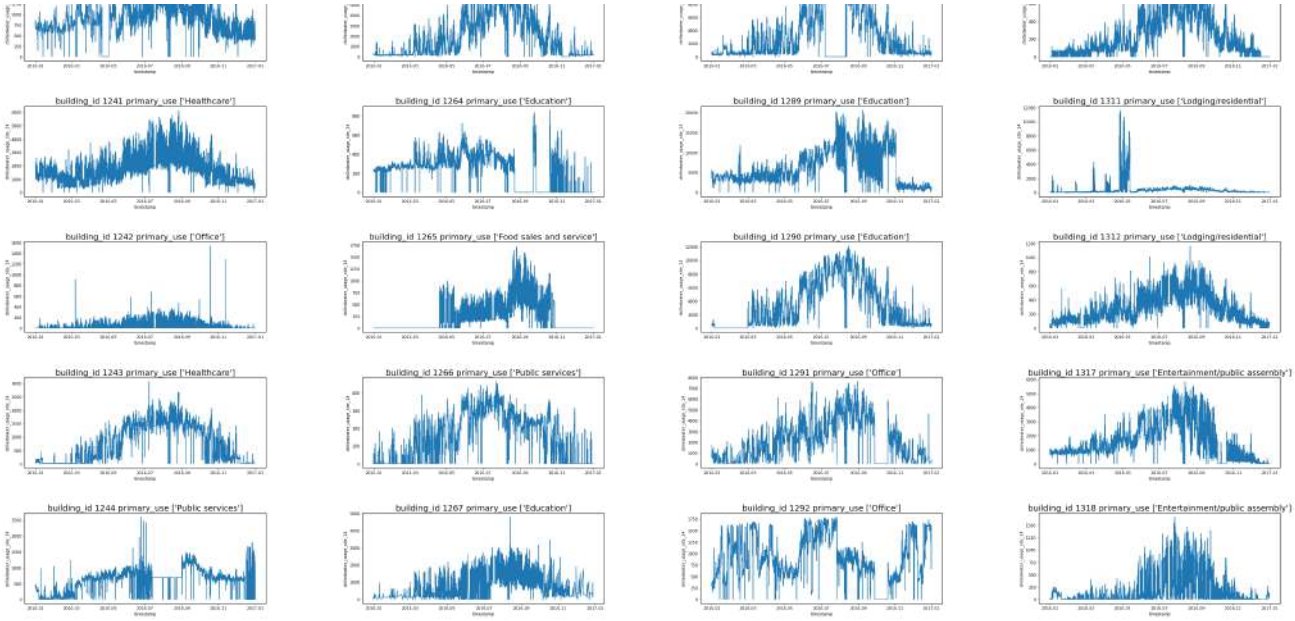
Here we are observing that the chilledwater consumption peaks at 20:00 pm which is a strange thing.

```
fig,ax=plt.subplots(figsize=(14,7))
z=df_train_site_14_meter_1
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_14')
plt.show()
```

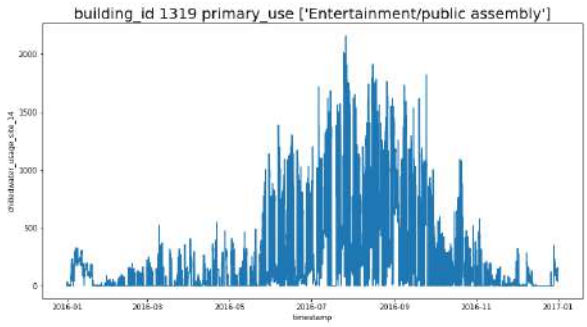
Here we can see that the weather timestamp is aligned with the timestamp of the hourly meter reading for chilledwater but they might not be aligned with the local timestamp od the site.This thing I have only observed for this particular site.

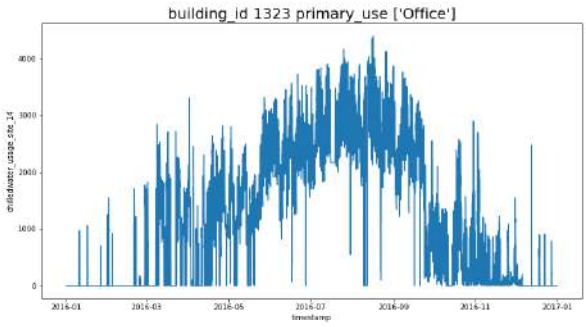

```
fig,ax=plt.subplots(figsize=(55,120),nrows=20,ncols=4,squeeze=True)
for i in range(80):
  g=df_train_site_14_meter_1['building_id'].unique()[i]
  z=df_train_site_14_meter_1.loc[df_train_site_14_meter_1['building_id']==g]
  axes=ax[i%20][i//20]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('chilledwater_usage_site_14')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.5,wspace=0.4)
```

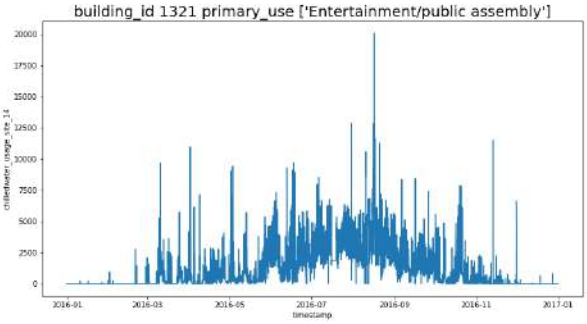

building_id 1223 primary_use ['Office']
building_id 1245 primary_use ['Healthcare']
building_id 1268 primary_use ['Office']
building_id 1293 primary_use ['Office']
building_id 1224 primary_use ['Office']
building_id 1246 primary_use ['Office']
building_id 1269 primary_use ['Healthcare']
building_id 1294 primary_use ['Office']
building_id 1225 primary_use ['Office']
building_id 1247 primary_use ['Food sales and service']
building_id 1270 primary_use ['Entertainment/public assembly']
building_id 1295 primary_use ['Office']
building_id 1226 primary_use ['Education']
building_id 1248 primary_use ['Office']
building_id 1271 primary_use ['Public services']
building_id 1296 primary_use ['Office']
building_id 1227 primary_use ['Education']
building_id 1249 primary_use ['Office']
building_id 1272 primary_use ['Public services']
building_id 1297 primary_use ['Office']
building_id 1228 primary_use ['Office']
building_id 1250 primary_use ['Education']
building_id 1273 primary_use ['Office']
building_id 1298 primary_use ['Education']
building_id 1229 primary_use ['Office']
building_id 1251 primary_use ['Education']
building_id 1274 primary_use ['Office']
building_id 1299 primary_use ['Education']
building_id 1230 primary_use ['Office']
building_id 1252 primary_use ['Education']
building_id 1275 primary_use ['Office']
building_id 1300 primary_use ['Healthcare']
building_id 1232 primary_use ['Education']
building_id 1253 primary_use ['Education']
building_id 1276 primary_use ['Office']
building_id 1301 primary_use ['Education']
building_id 1233 primary_use ['Education']
building_id 1254 primary_use ['Education']
building_id 1280 primary_use ['Lodging/residential']
building_id 1302 primary_use ['Office']
building_id 1234 primary_use ['Education']
building_id 1255 primary_use ['Office']
building_id 1283 primary_use ['Public services']
building_id 1303 primary_use ['Healthcare']
building_id 1235 primary_use ['Public services']
building_id 1256 primary_use ['Education']
building_id 1284 primary_use ['Education']
building_id 1306 primary_use ['Lodging/residential']
building_id 1236 primary_use ['Office']
building_id 1259 primary_use ['Education']
building_id 1285 primary_use ['Healthcare']
building_id 1307 primary_use ['Lodging/residential']
building_id 1237 primary_use ['Office']
building_id 1260 primary_use ['Office']
building_id 1286 primary_use ['Healthcare']
building_id 1308 primary_use ['Lodging/residential']
building_id 1238 primary_use ['Office']
building_id 1262 primary_use ['Office']
building_id 1287 primary_use ['Education']
building_id 1309 primary_use ['Lodging/residential']
building_id 1240 primary_use ['Public services']
building_id 1263 primary_use ['Education']
building_id 1288 primary_use ['Education']
building_id 1310 primary_use ['Lodging/residential']

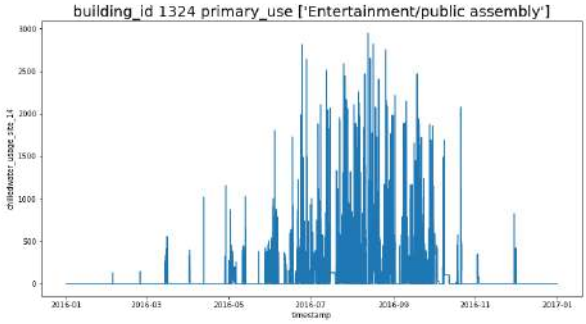


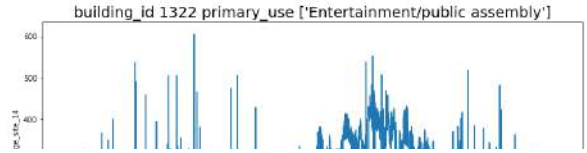

```
fig,ax=plt.subplots(figsize=(30,30),nrows=3,ncols=2,squeeze=True)
for i in range(6):
  g=df_train_site_14_meter_1['building_id'].unique()[80:87][i]
  z=df_train_site_14_meter_1.loc[df_train_site_14_meter_1['building_id']==g]
  axes=ax[i%3][i//3]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('chilledwater_usage_site_14')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.7,wspace=0.3)
```

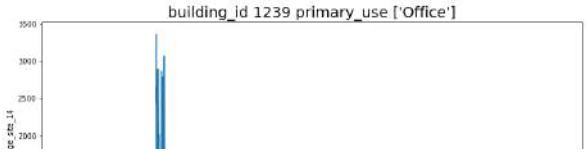












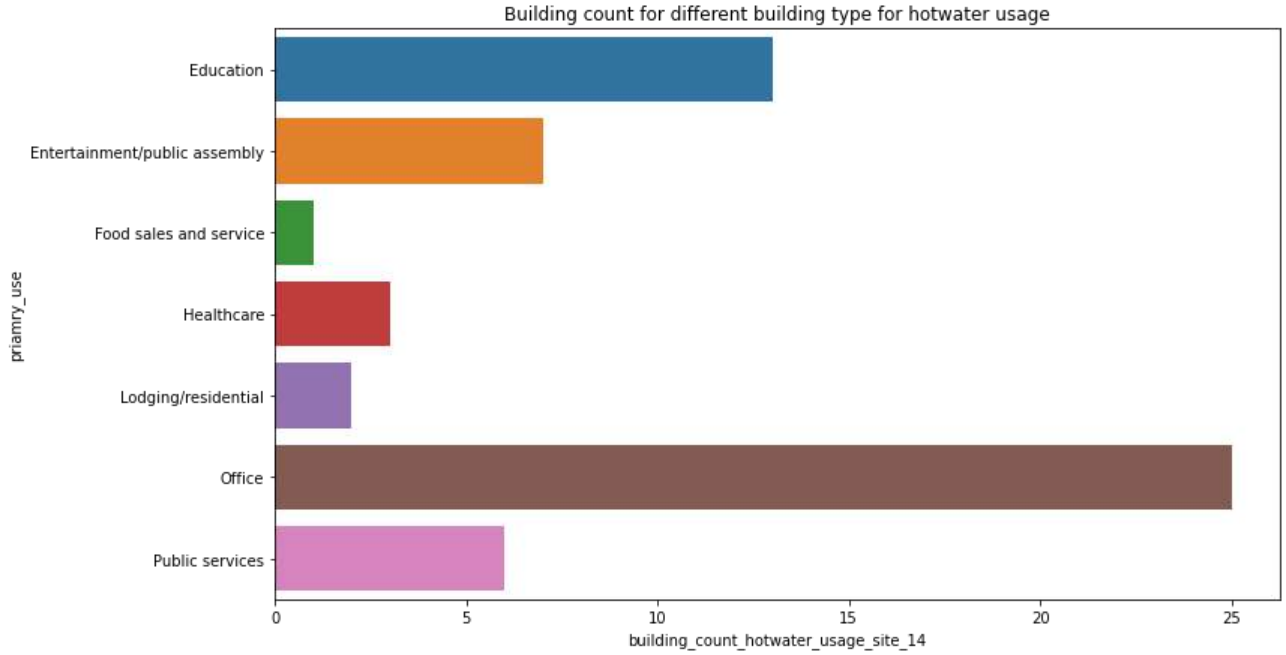
Here at site 14 there are total of 86 buildings which are divided further into 80 and 6 buildings for better visualization.

Important Observations

- The buildings are filled with constant meter values ans spikes which definitely needs to be filtered out.

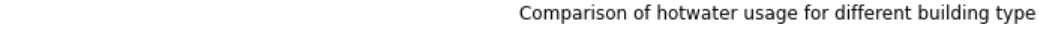
```
df_train_site_14_meter_3=df_train_site_14.loc[df_train_site_14['meter']=='hotwater']
```

```
z=df_train_site_14_meter_3.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_hotwater_usage_site_14')
plt.ylabel('priamry_use')
plt.title('Building count for different building type for hotwater usage')
plt.show()
```



The above plot shows the building count for hotwater usage

```
z=df_train_site_14_meter_3.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_hotwater_reading_site_14')
plt.ylabel('priamry_use')
plt.title('Comparison of hotwater usage for different building type')
plt.show()
```

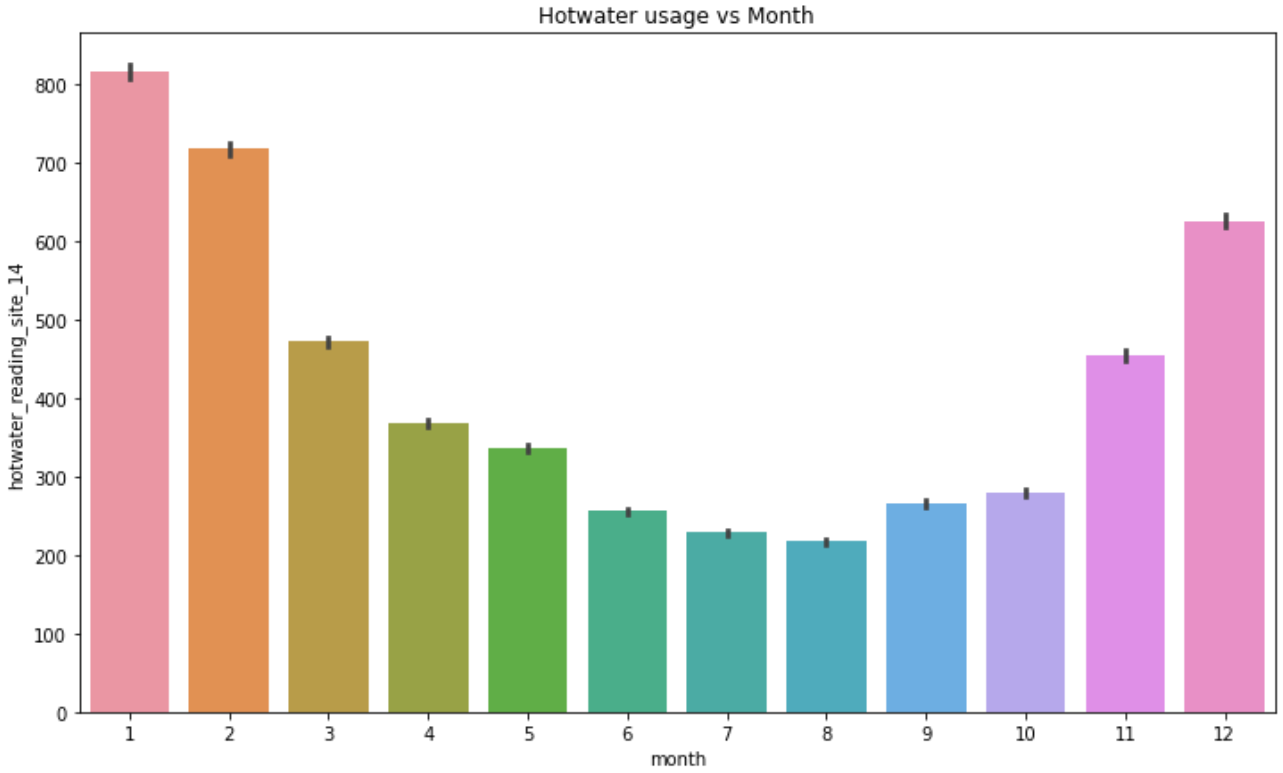


Here we can see that on an average public serivices is having the highest energy consumption although they are less in number.

Entertainment and Educational building are also showing higher consumption of chilledwater although entertainment building are lesser in number as compared to the educational

```
df_train_site_14_meter_3['month']=df_train_site_14_meter_3['timestamp'].dt.month
df_train_site_14_meter_3['weekday']=df_train_site_14_meter_3['timestamp'].dt.weekday
df_train_site_14_meter_3['hour']=df_train_site_14_meter_3['timestamp'].dt.hour
```

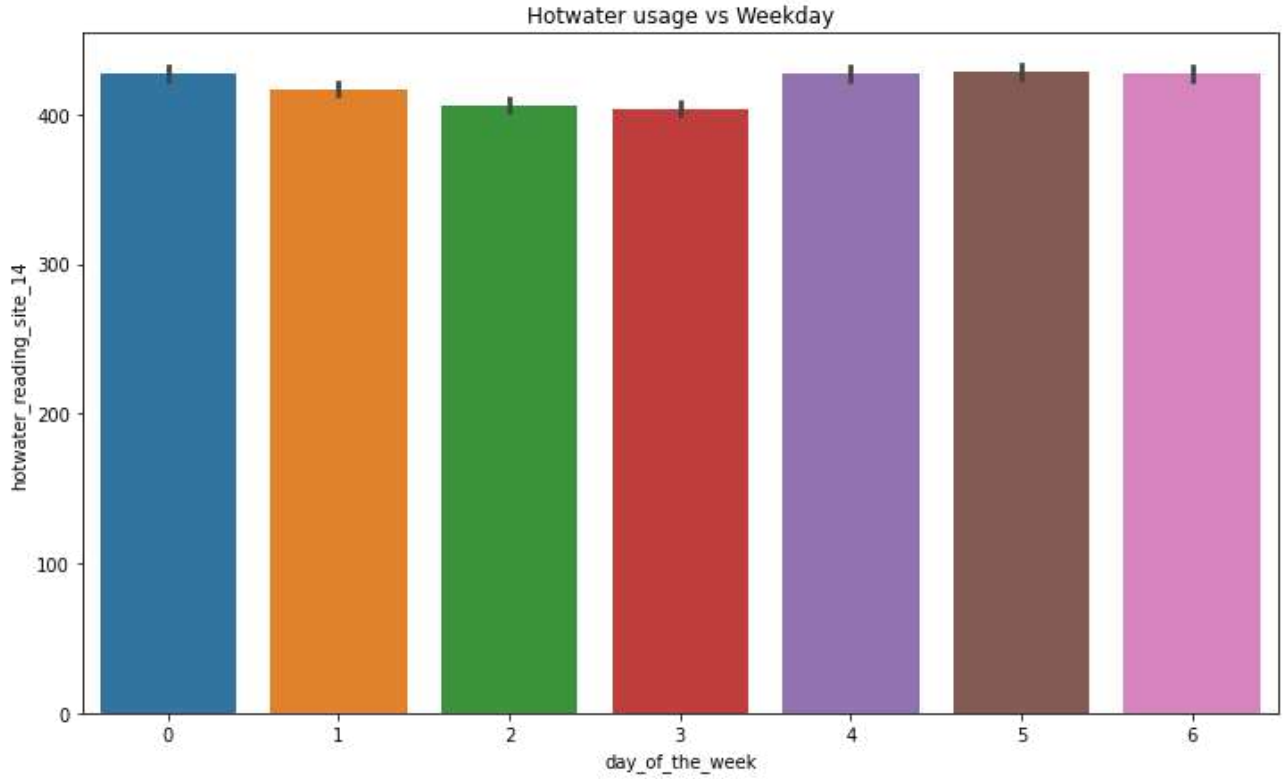
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_3
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('hotwater_reading_site_14')
plt.title('Hotwater usage vs Month')
plt.show()
```



Here we can observe that hotwater has higher consumption during the winter month and it decreases gradually as we approach the summer months

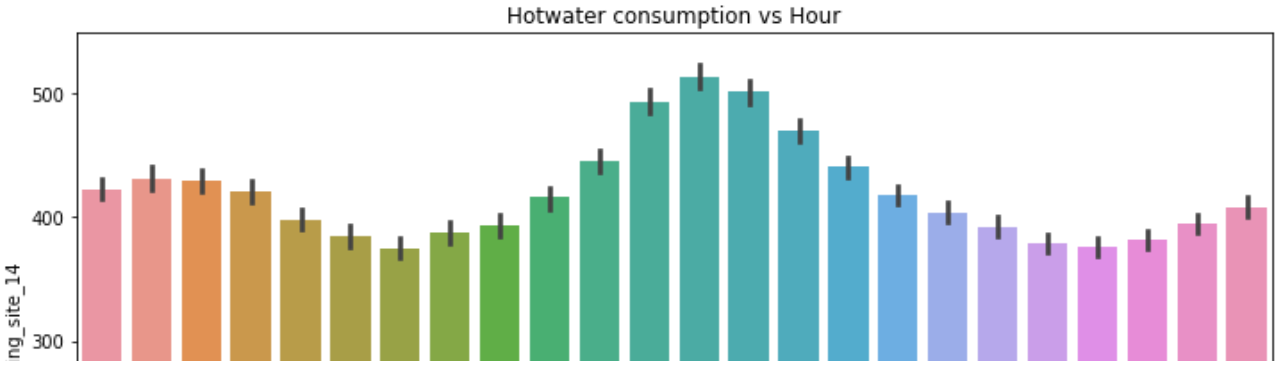
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_3
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
```

```
plt.ylabel('hotwater_reading_site_14')
plt.title('Hotwater usage vs Weekday')
plt.show()
```



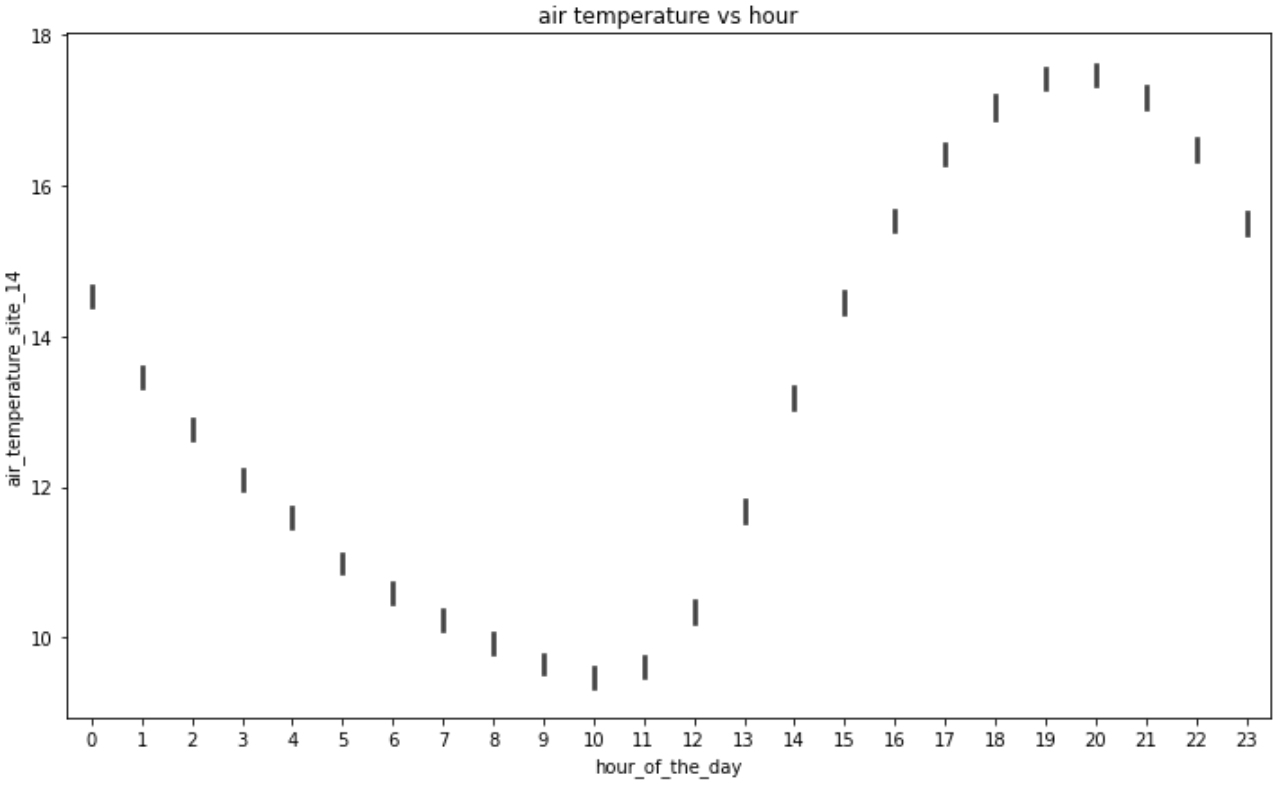
Here we can see that hotwater consumption also has higher consumption on the weekend.It is not showing any consumption pattern

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('hotwater_reading_site_14')
plt.title('Hotwater consumption vs Hour')
plt.show()
```



The hotwater consumption peaks around 12:00 am in the daytime and then starts to decrease gradually after that.Here we can also see that it increases a litle after 20:00 pm and again starts decreasing after 01:00 pm in the night which is a little strange behaviour

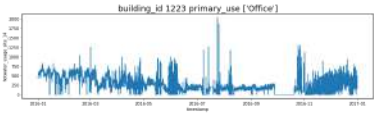
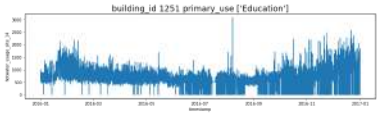
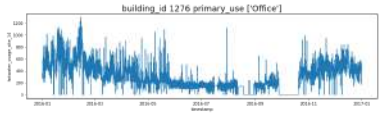
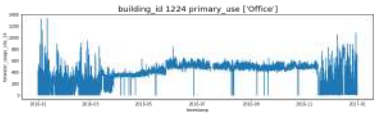
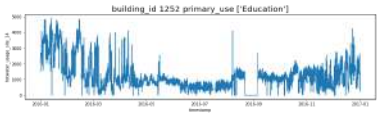
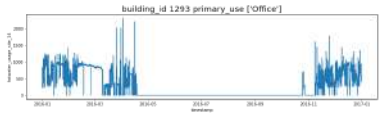
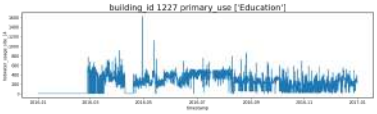
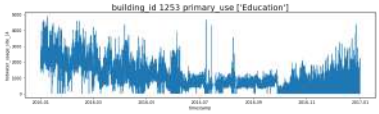
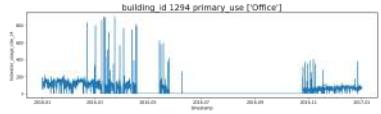
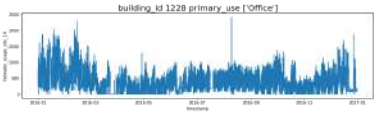
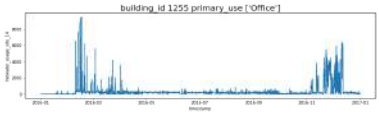
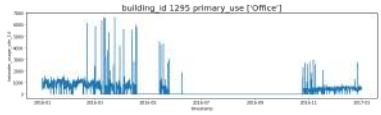
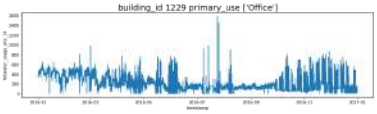
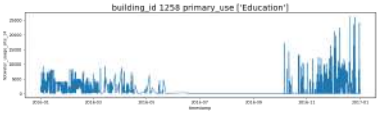
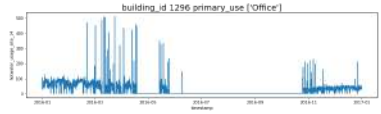
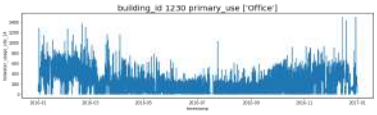
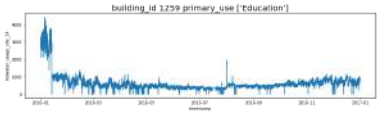
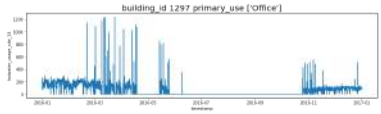
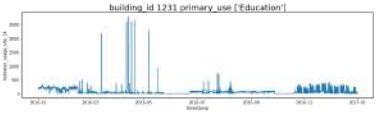
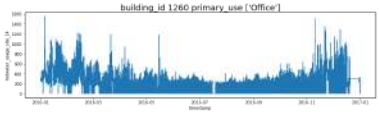
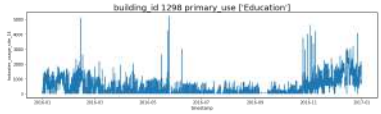
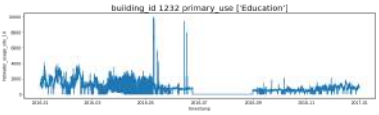
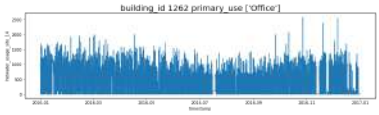
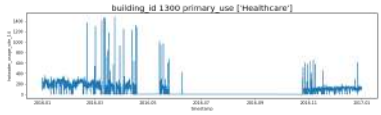
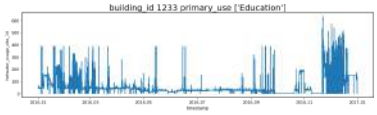
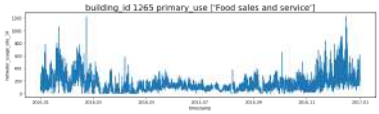
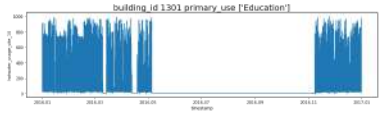
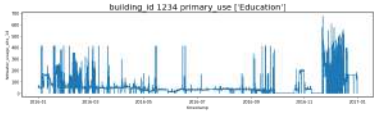
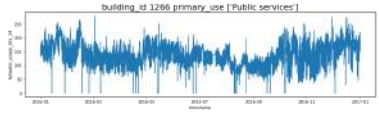
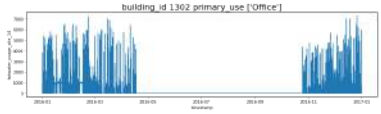
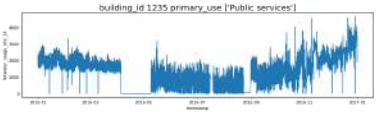
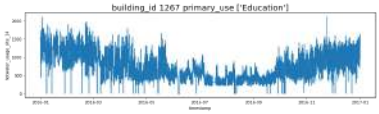
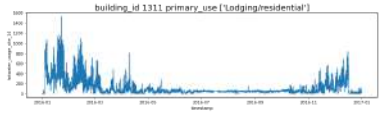
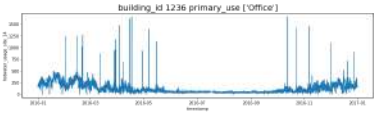
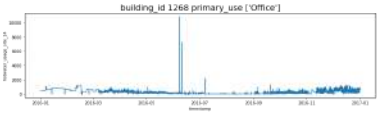
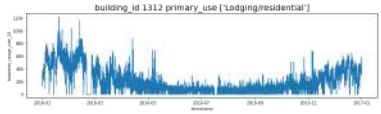
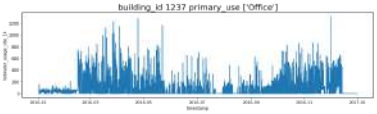
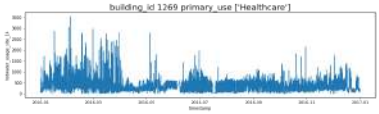
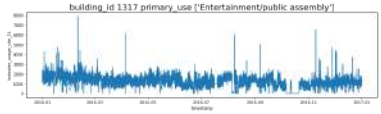
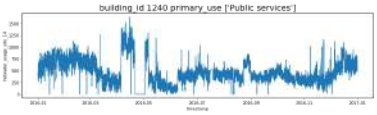
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_14')
plt.title('air temperature vs hour')
plt.show()
```



The weather timestamp is not aligned with the local timestamp of the hourly hotwater meter readings.The temperature peaks at 19:00 pm

```
fig,ax=plt.subplots(figsize=(55,120),nrows=19,ncols=3,squeeze=True)
for i in range(df_train_site_14_meter_3['building_id'].nunique()):
```

```
    g=df_train_site_14_meter_3['building_id'].unique()[i]
    z=df_train_site_14_meter_3.loc[df_train_site_14_meter_3['building_id']==g]
    axes=ax[i%19][i//19]
    axes.plot(z['timestamp'],z['meter_reading'])
    axes.set_xlabel('timestamp')
    axes.set_ylabel('hotwater_usage_site_14')
    axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
    plt.subplots_adjust(hspace=0.7,wspace=0.3)
```


building_id 1223 primary_use ['Office']

building_id 1251 primary_use ['Education']

building_id 1276 primary_use ['Office']

building_id 1224 primary_use ['Office']

building_id 1252 primary_use ['Education']

building_id 1293 primary_use ['Office']

building_id 1227 primary_use ['Education']

building_id 1253 primary_use ['Education']

building_id 1294 primary_use ['Office']

building_id 1228 primary_use ['Office']

building_id 1255 primary_use ['Office']

building_id 1295 primary_use ['Office']

building_id 1229 primary_use ['Office']

building_id 1258 primary_use ['Education']

building_id 1296 primary_use ['Office']

building_id 1230 primary_use ['Office']

building_id 1259 primary_use ['Education']

building_id 1297 primary_use ['Office']

building_id 1231 primary_use ['Education']

building_id 1260 primary_use ['Office']

building_id 1298 primary_use ['Education']

building_id 1232 primary_use ['Education']

building_id 1262 primary_use ['Office']

building_id 1300 primary_use ['Healthcare']

building_id 1233 primary_use ['Education']

building_id 1265 primary_use ['Food sales and service']

building_id 1301 primary_use ['Education']

building_id 1234 primary_use ['Education']

building_id 1266 primary_use ['Public services']

building_id 1302 primary_use ['Office']

building_id 1235 primary_use ['Public services']

building_id 1267 primary_use ['Education']

building_id 1311 primary_use ['Lodging/residential']

building_id 1236 primary_use ['Office']

building_id 1268 primary_use ['Office']

building_id 1312 primary_use ['Lodging/residential']

building_id 1237 primary_use ['Office']

building_id 1269 primary_use ['Healthcare']

building_id 1317 primary_use ['Entertainment/public assembly']

building_id 1240 primary_use ['Public services']

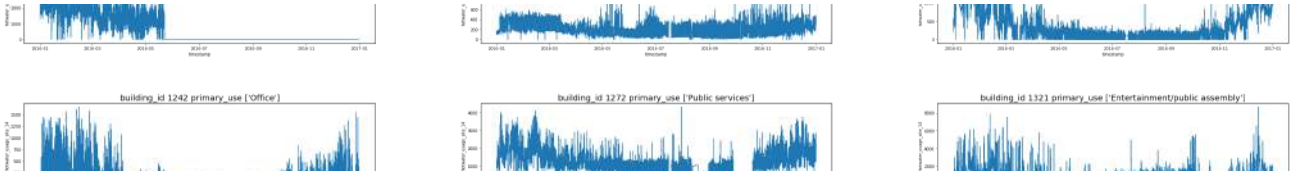
building_id 1270 primary_use ['Entertainment/public assembly']

building_id 1318 primary_use ['Entertainment/public assembly']

building_id 1241 primary_use ['Healthcare']

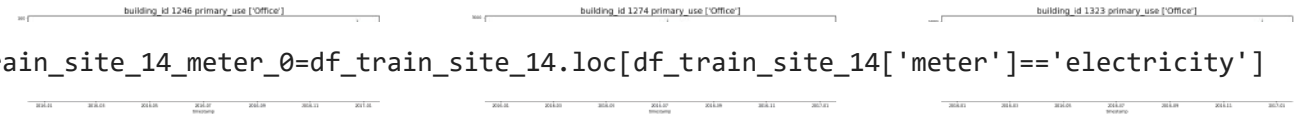
building_id 1271 primary_use ['Public services']

building_id 1319 primary_use ['Entertainment/public assembly']



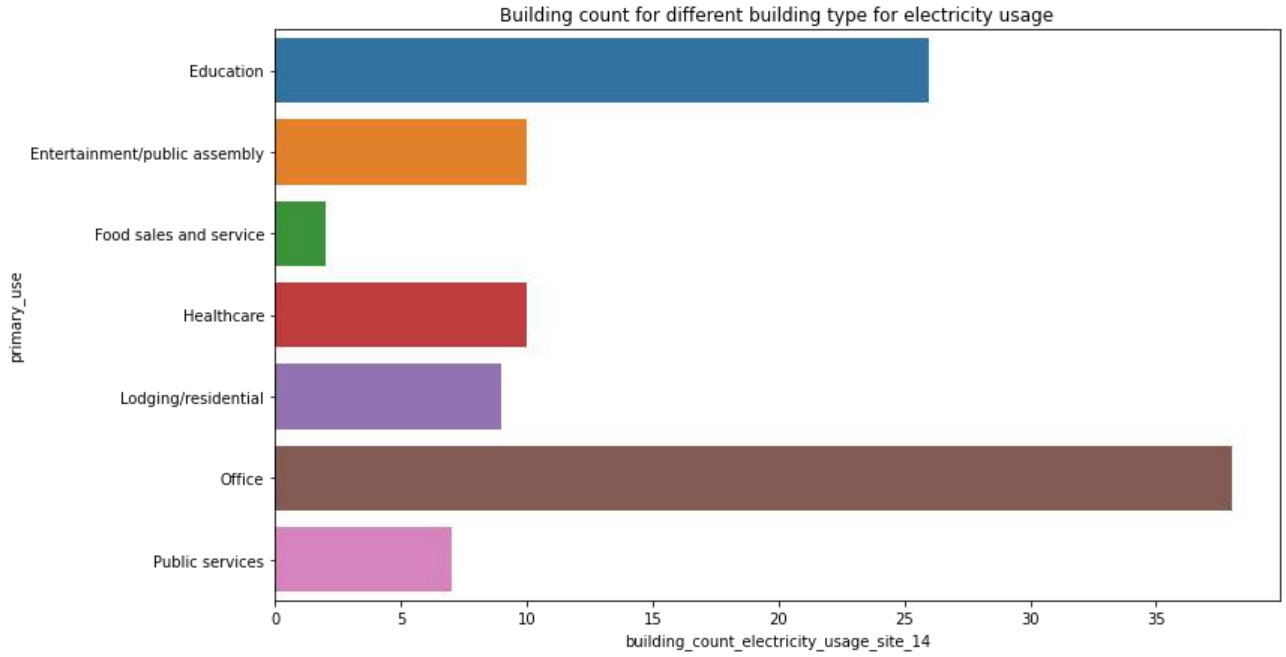
Important Observations

- Here the buildings are filled with constant meter readings and spikes which needs to be filtered out.




```
df_train_site_14_meter_0=df_train_site_14.loc[df_train_site_14['meter']=='electricity']
```

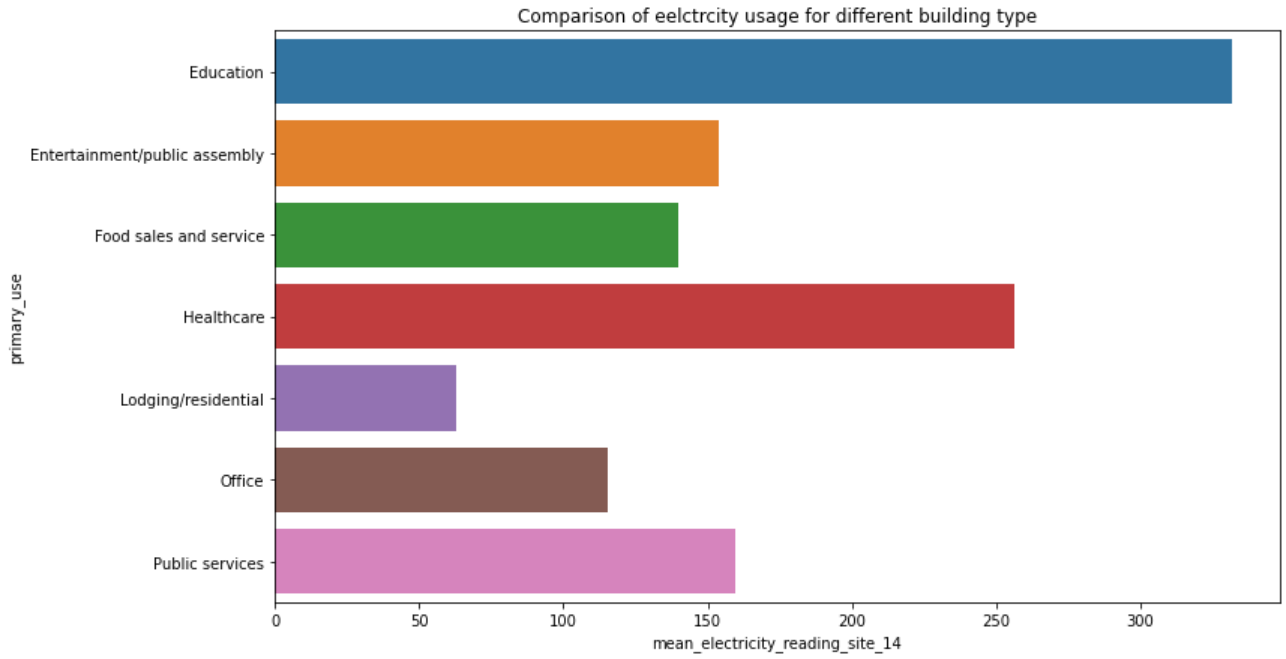
```
z=df_train_site_14_meter_0.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_electricity_usage_site_14')
plt.ylabel('primary_use')
plt.title('Building count for different building type for electricity usage')
plt.show()
```



The above plot reprersents the building count for diffrernt building type for electricity usage

```
z=df_train_site_14_meter_0.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean electricity reading site 14')
```

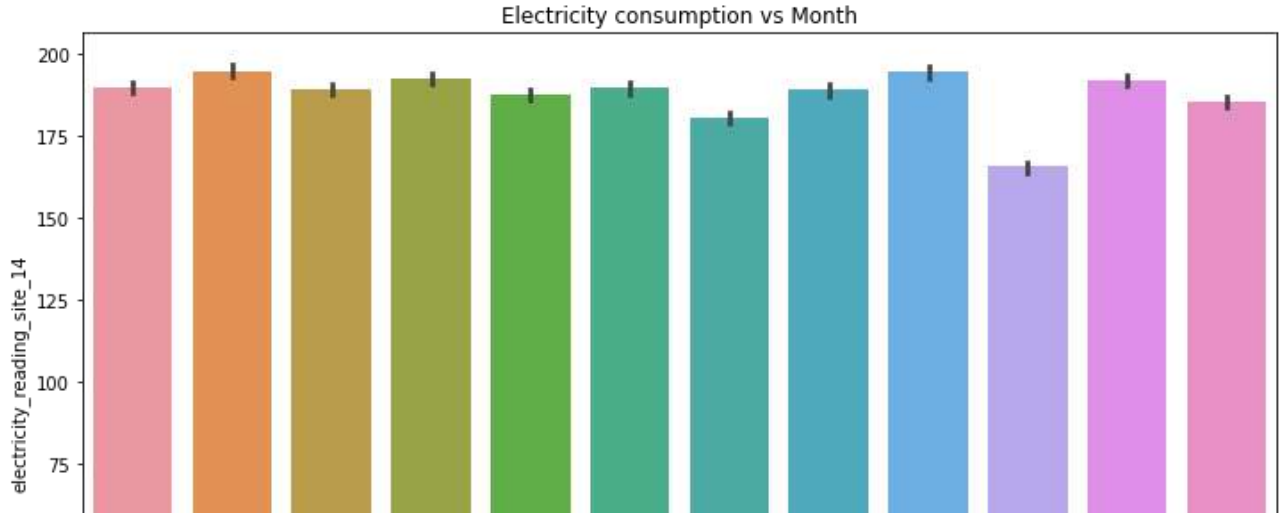
```
plt.ylabel('primary_use')
plt.title('Comparison of eelctrcity usage for different building type')
plt.show()
```



Here Education is having the highest electricity consumption

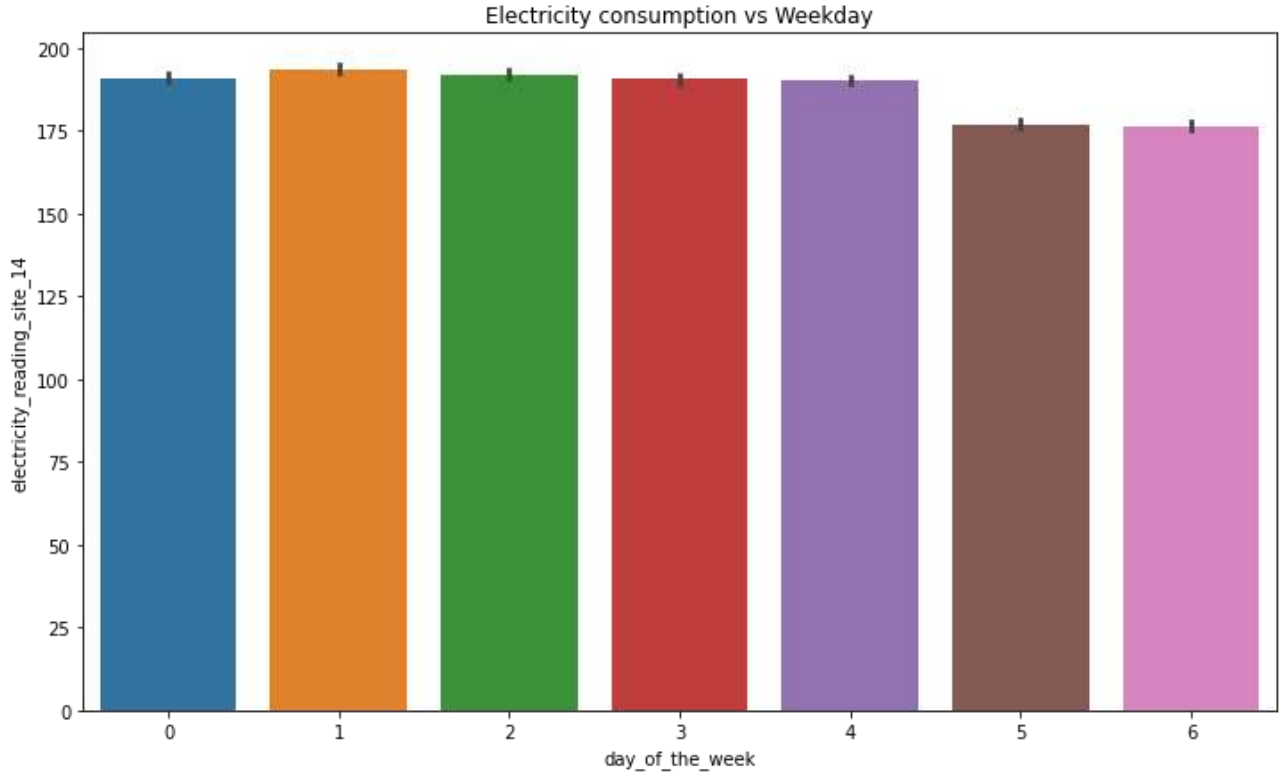
```
df_train_site_14_meter_0['month']=df_train_site_14_meter_0['timestamp'].dt.month
df_train_site_14_meter_0['weekday']=df_train_site_14_meter_0['timestamp'].dt.weekday
df_train_site_14_meter_0['hour']=df_train_site_14_meter_0['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_reading_site_14')
plt.title('Electricity consumption vs Month')
plt.show()
```



Here the elctricity consumption does not show specific pattern of usage overt he month

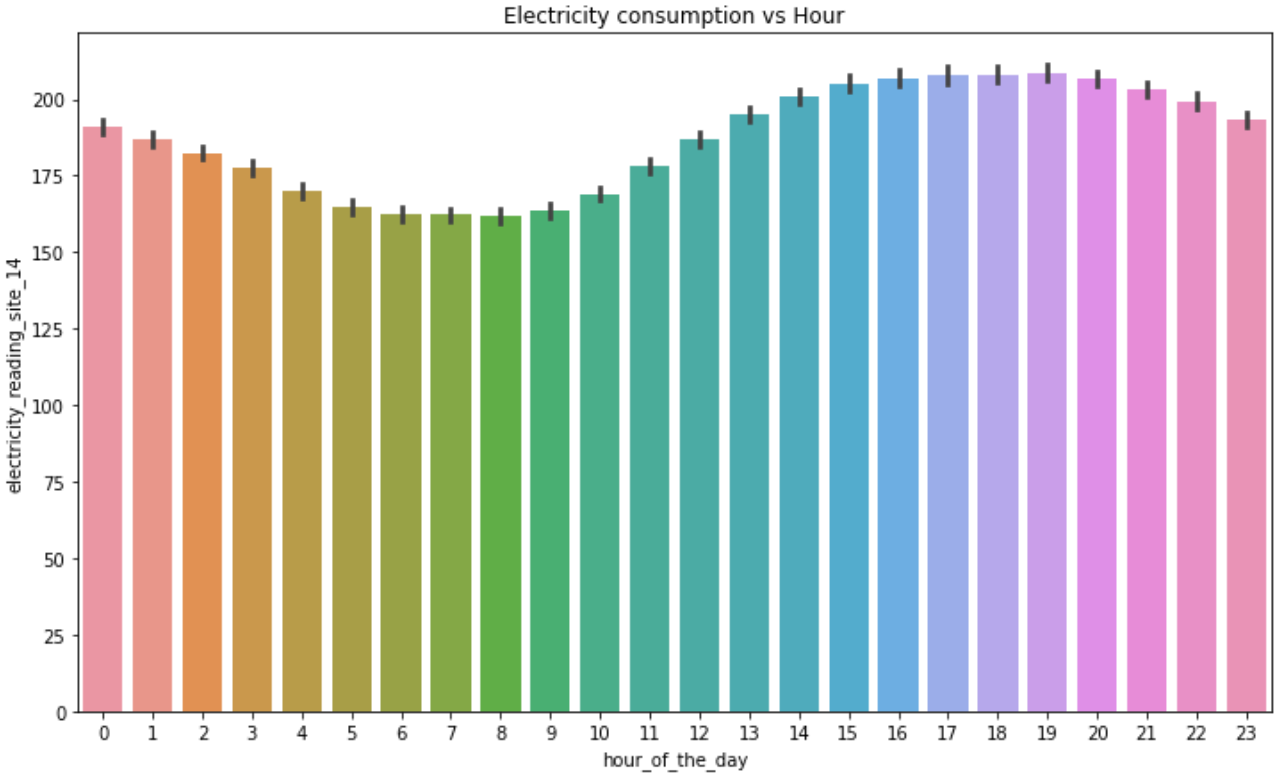
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_14')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



Here the elctricity consumption is lesser for the weekend as comparedd to the weekday

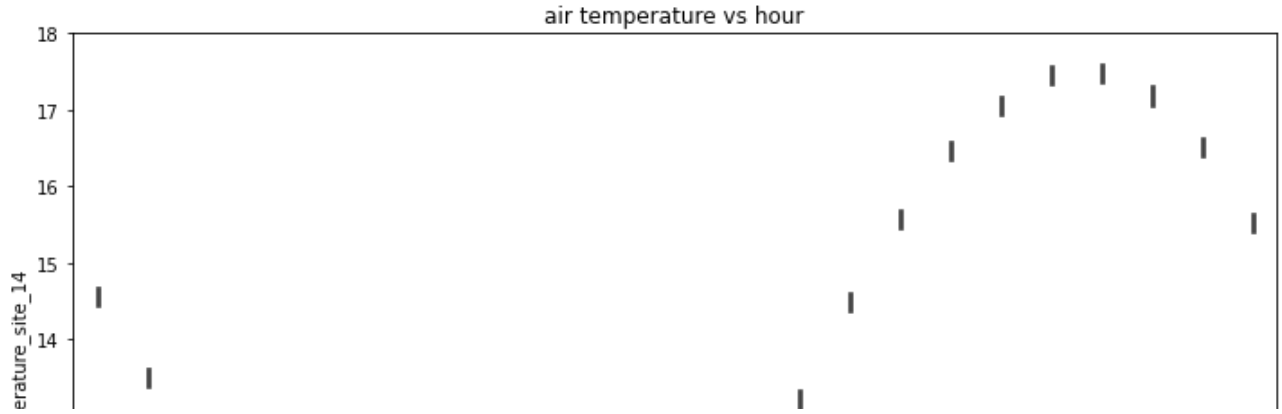
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_0
```

```
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_14')
plt.title('Electricity consumption vs Hour')
plt.show()
```



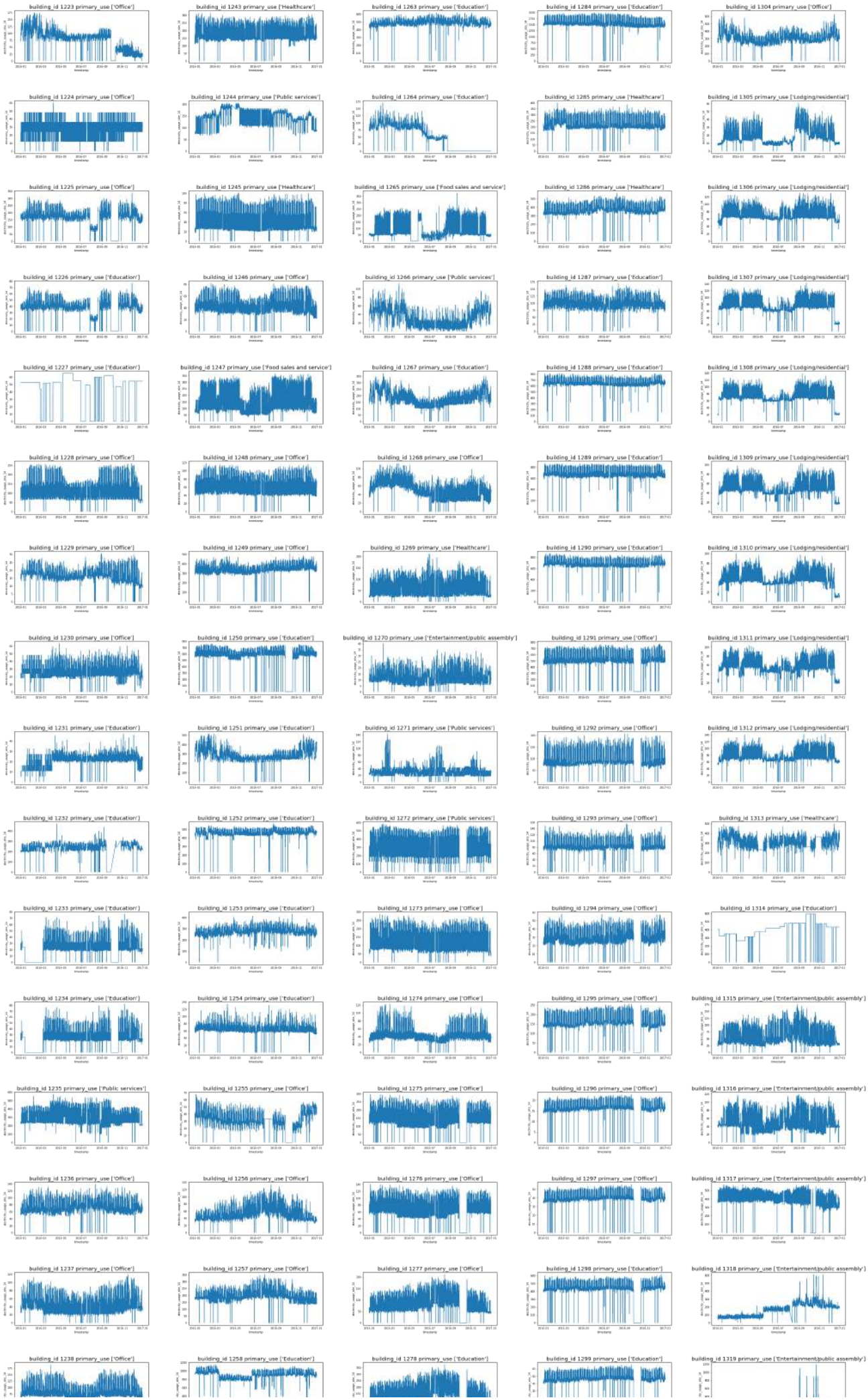
Here we can see that the consumption peaks around 19:00 pm which is unusual.

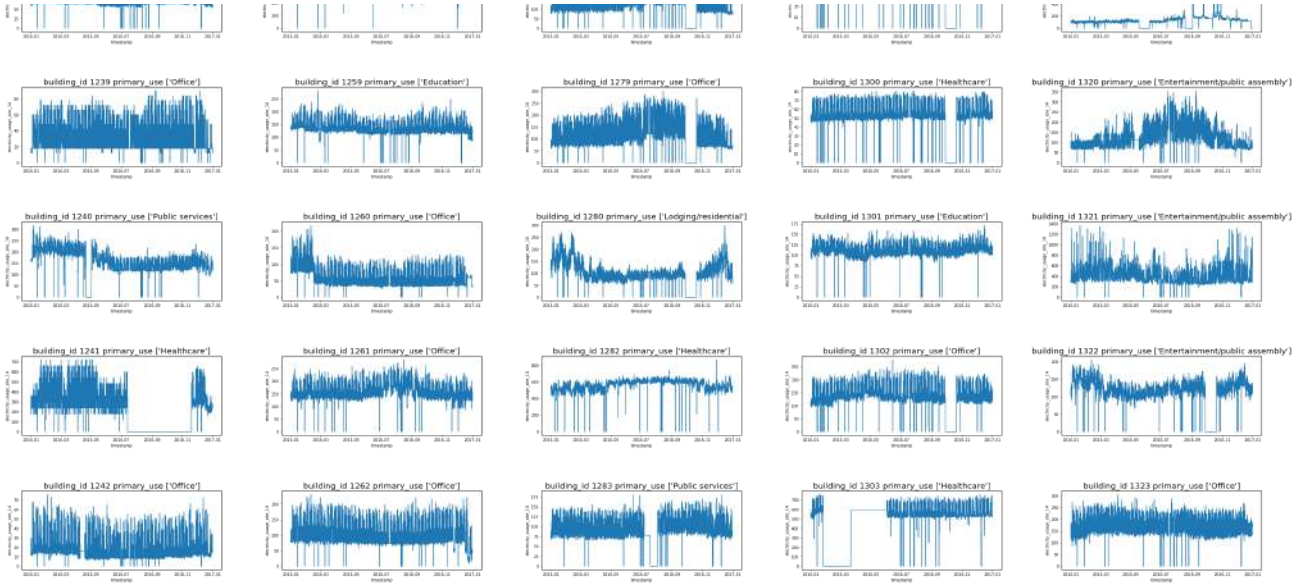
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_14_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_14')
plt.title('air temperature vs hour')
plt.show()
```



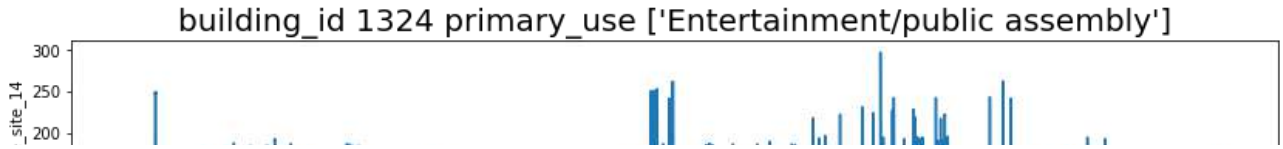
Here the weather timestamp might be in alignment with the local timestamp as both are showing peaks around 19:00 pm

```
fig,ax=plt.subplots(figsize=(55,120),nrows=20,ncols=5,squeeze=True)
for i in range(df_train_site_14_meter_0['building_id'].nunique()-2):
  g=df_train_site_14_meter_0['building_id'].unique()[i]
  z=df_train_site_14_meter_0.loc[df_train_site_14_meter_0['building_id']==g]
  axes=ax[i%20][i//20]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_usage_site_14')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.7,wspace=0.3)
```


building_id 1223 primary_use ['Office']
building_id 1243 primary_use ['Healthcare']
building_id 1263 primary_use ['Education']
building_id 1284 primary_use ['Education']
building_id 1304 primary_use ['Office']
building_id 1224 primary_use ['Office']
building_id 1244 primary_use ['Public services']
building_id 1264 primary_use ['Education']
building_id 1285 primary_use ['Healthcare']
building_id 1305 primary_use ['Lodging/residential']
building_id 1225 primary_use ['Office']
building_id 1245 primary_use ['Healthcare']
building_id 1265 primary_use ['Food sales and service']
building_id 1286 primary_use ['Healthcare']
building_id 1306 primary_use ['Lodging/residential']
building_id 1226 primary_use ['Education']
building_id 1246 primary_use ['Office']
building_id 1266 primary_use ['Public services']
building_id 1287 primary_use ['Education']
building_id 1307 primary_use ['Lodging/residential']
building_id 1227 primary_use ['Education']
building_id 1247 primary_use ['Food sales and service']
building_id 1267 primary_use ['Education']
building_id 1288 primary_use ['Education']
building_id 1308 primary_use ['Lodging/residential']
building_id 1228 primary_use ['Office']
building_id 1248 primary_use ['Office']
building_id 1268 primary_use ['Office']
building_id 1289 primary_use ['Education']
building_id 1309 primary_use ['Lodging/residential']
building_id 1229 primary_use ['Office']
building_id 1249 primary_use ['Office']
building_id 1269 primary_use ['Healthcare']
building_id 1290 primary_use ['Education']
building_id 1310 primary_use ['Lodging/residential']
building_id 1230 primary_use ['Office']
building_id 1250 primary_use ['Education']
building_id 1270 primary_use ['Entertainment/public assembly']
building_id 1291 primary_use ['Office']
building_id 1311 primary_use ['Lodging/residential']
building_id 1231 primary_use ['Education']
building_id 1251 primary_use ['Education']
building_id 1271 primary_use ['Public services']
building_id 1292 primary_use ['Office']
building_id 1312 primary_use ['Lodging/residential']
building_id 1232 primary_use ['Education']
building_id 1252 primary_use ['Education']
building_id 1272 primary_use ['Public services']
building_id 1293 primary_use ['Office']
building_id 1313 primary_use ['Healthcare']
building_id 1233 primary_use ['Education']
building_id 1253 primary_use ['Education']
building_id 1273 primary_use ['Office']
building_id 1294 primary_use ['Office']
building_id 1314 primary_use ['Education']
building_id 1234 primary_use ['Education']
building_id 1254 primary_use ['Education']
building_id 1274 primary_use ['Office']
building_id 1295 primary_use ['Office']
building_id 1315 primary_use ['Entertainment/public assembly']
building_id 1235 primary_use ['Public services']
building_id 1255 primary_use ['Office']
building_id 1275 primary_use ['Office']
building_id 1296 primary_use ['Office']
building_id 1316 primary_use ['Entertainment/public assembly']
building_id 1236 primary_use ['Office']
building_id 1256 primary_use ['Office']
building_id 1276 primary_use ['Office']
building_id 1297 primary_use ['Office']
building_id 1317 primary_use ['Entertainment/public assembly']
building_id 1237 primary_use ['Office']
building_id 1257 primary_use ['Office']
building_id 1277 primary_use ['Office']
building_id 1298 primary_use ['Education']
building_id 1318 primary_use ['Entertainment/public assembly']
building_id 1238 primary_use ['Office']
building_id 1258 primary_use ['Education']
building_id 1278 primary_use ['Education']
building_id 1299 primary_use ['Education']
building_id 1319 primary_use ['Entertainment/public assembly']



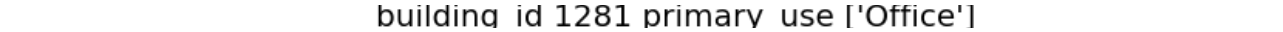

```
fig,ax=plt.subplots(figsize=(14,8),nrows=2,ncols=1,squeeze=False)
for i in range(df_train_site_14_meter_0['building_id'].nunique()-100):
  g=df_train_site_14_meter_0['building_id'].unique()[100:103][i]
  z=df_train_site_14_meter_0.loc[df_train_site_14_meter_0['building_id']==g]
  axes=ax[i%2][i//2]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_usage_site_14')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.5)
```



The total number of buildings which are consuming electrcicty at site 14 is 102.

Important Observations

- Buildings which are consuming electricity are filled with constant meter readings and spike which needs to be filtered out gradually




```
#Starting analysis for site 15
```

```
df_train_site_15=df_train_merge.loc[df_train_merge['site_id']==15]
```

```
df_train_site_15.isnull().sum()/df_train_site_15.shape[0]
```

```
building_id          0.00
meter                0.00
timestamp            0.00
meter_reading        0.00
site_id              0.00
primary_use          0.00
square_feet          0.00
year_built           0.08
floor_count          1.00
air_temperature      0.04
cloud_coverage       0.52
dew_temperature      0.04
precip_depth_1_hr    0.82
sea_level_pressure   0.10
wind_direction       0.07
wind_speed           0.04
dtype: float64
```

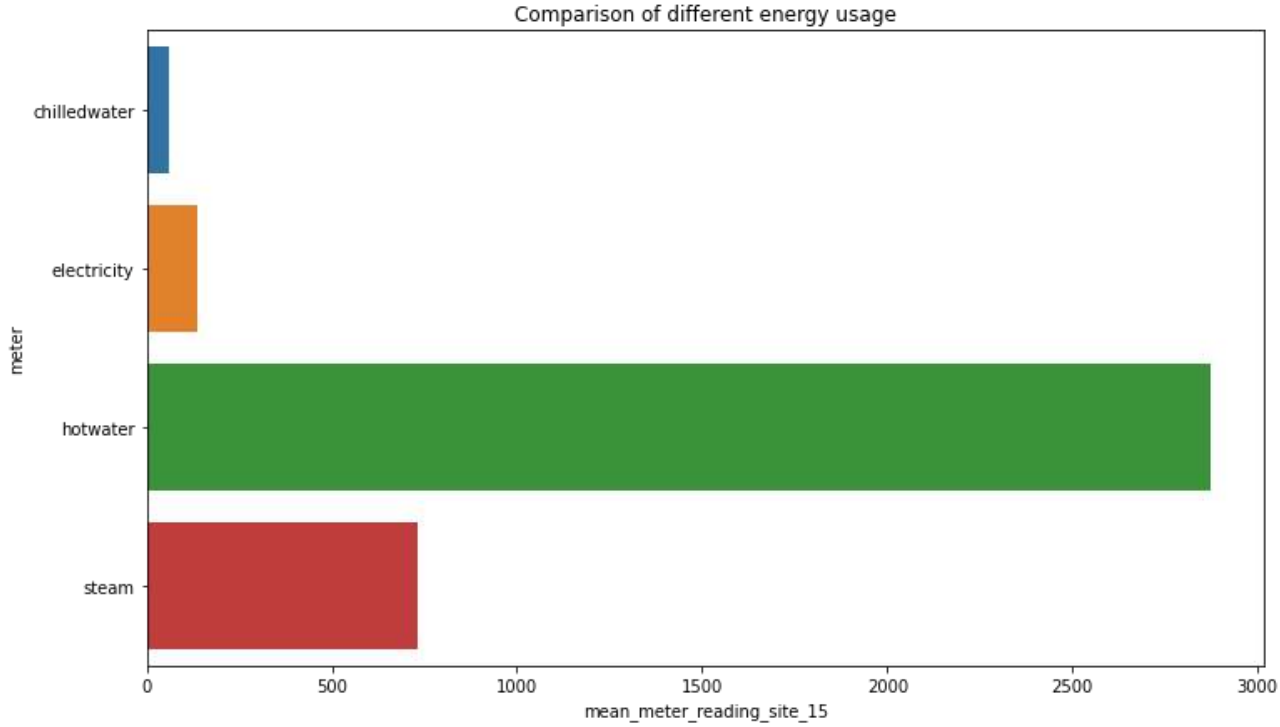
As we can see there are null values present at site 15 and we need to fill those missing values

```
df_corr_15=df_train_site_15.corr()
df_corr_15.style.background_gradient(cmap='hot_r').set_precision(2)
```

	building_id	meter_reading	site_id	square_feet	year_built	floor_count	air_ten
building_id	1.00	-0.08	nan	-0.21	0.21	nan	0.01
meter_reading	-0.08	1.00	nan	0.17	0.03	nan	-0.16
site_id	nan	nan	nan	nan	nan	nan	nan

As we can see from the correlation plot that meter reading was not correlated with any of the features

```
z=df_train_site_15.groupby(['meter'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='meter')
plt.xlabel('mean_meter_reading_site_15')
plt.ylabel('meter')
plt.title('Comparison of different energy usage')
plt.show()
```

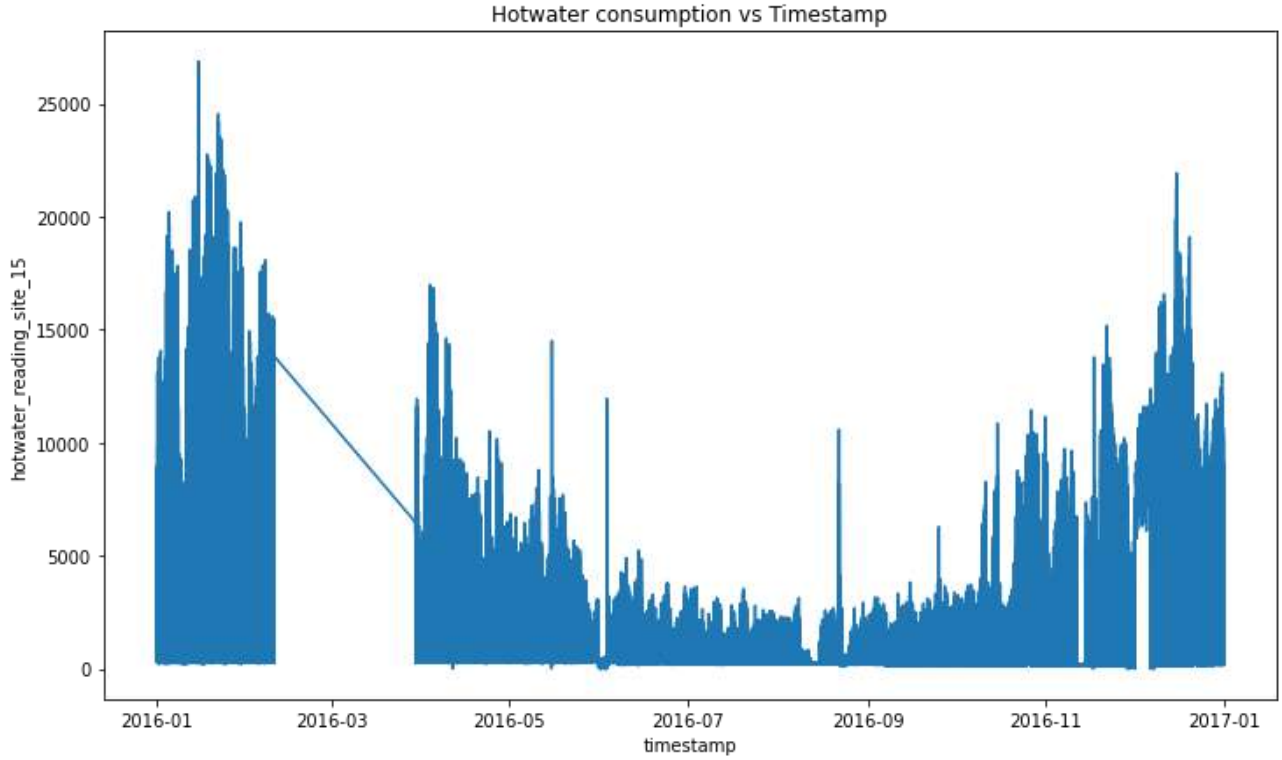


Here we can see that hotwater consumption is having the highest energy usage

```
df_train_site_15_meter_3=df_train_site_15.loc[df_train_site_15['meter']=='hotwater']
```

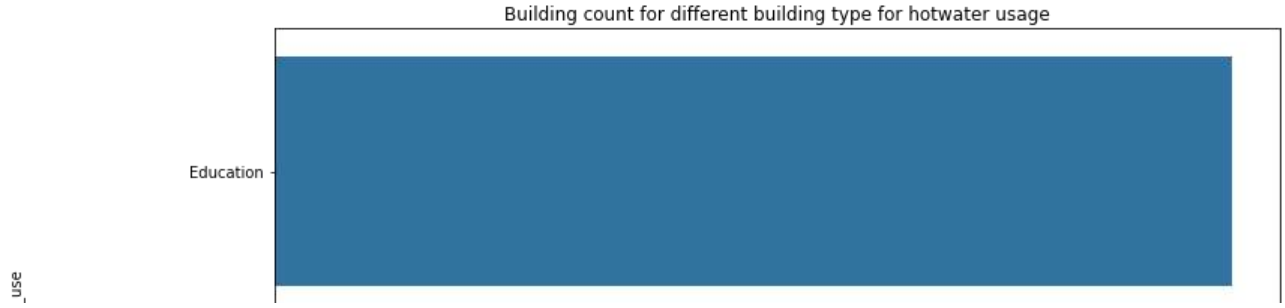
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_3
ax.plot(z['timestamp'],z['meter_reading'])
plt.xlabel('timestamp')
plt.ylabel('hotwater_reading_site_15')
plt.title('Hotwater consumption vs Timestamp')
```

```
plt.title('Hotwater consumption vs Timestamp')
plt.show()
```



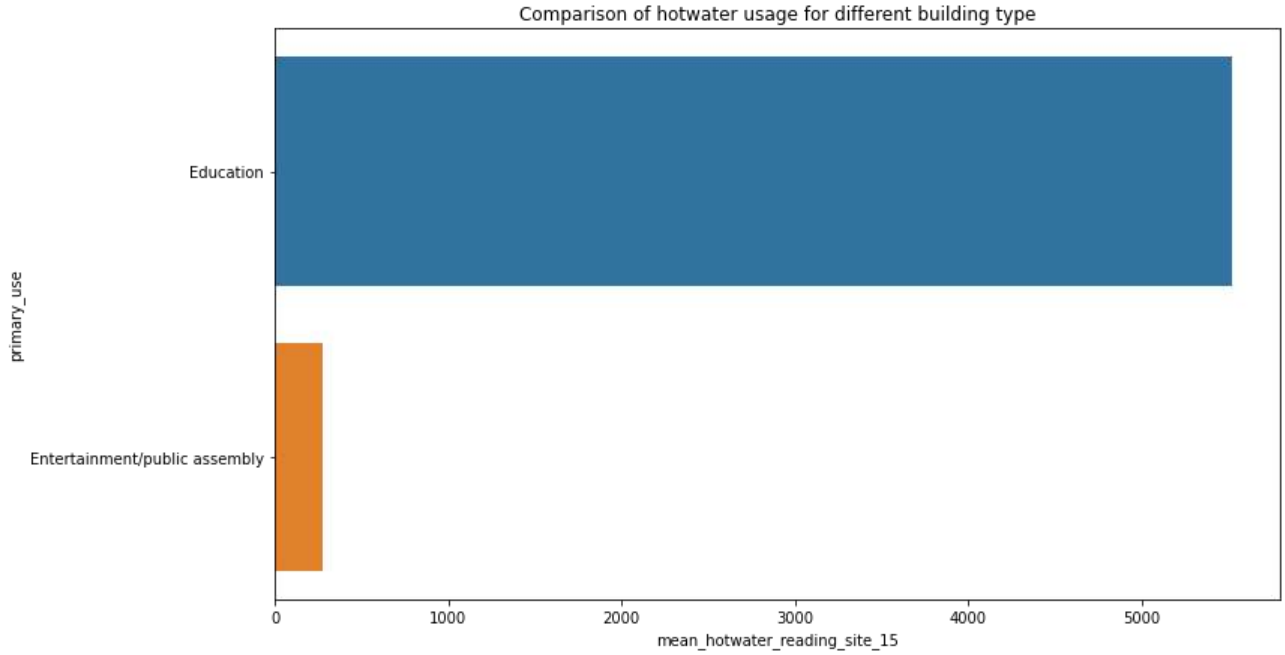
This plot shows the overall hotwater consumption for all the buildings

```
z=df_train_site_15_meter_3.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_hotwater_usage_site_15')
plt.ylabel('primary_use')
plt.title('Building count for different building type for hotwater usage')
plt.show()
```



The above plot shows the building count for hotwater usage for different building type

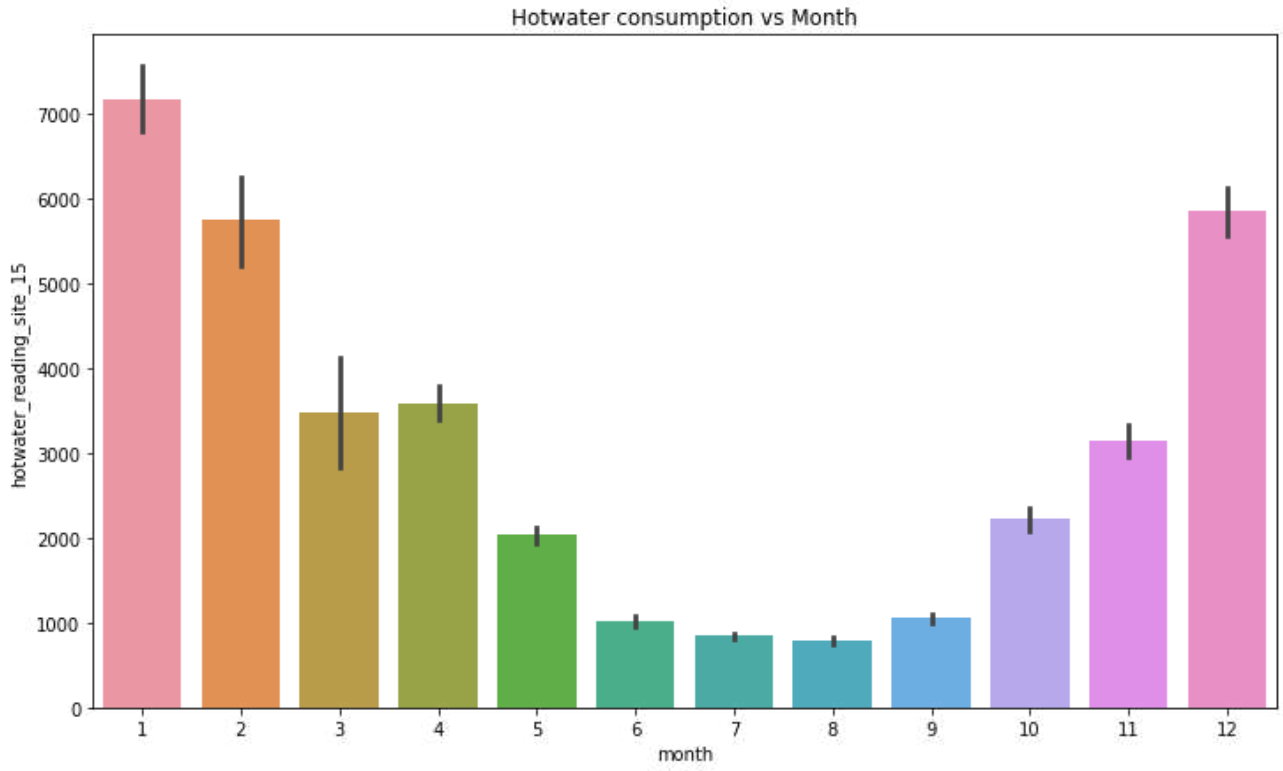
```
z=df_train_site_15_meter_3.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_hotwater_reading_site_15')
plt.ylabel('primary_use')
plt.title('Comparison of hotwater usage for different building type')
plt.show()
```



Here we can see that education is having very high consumption of hotwater as compared to the entertainment building

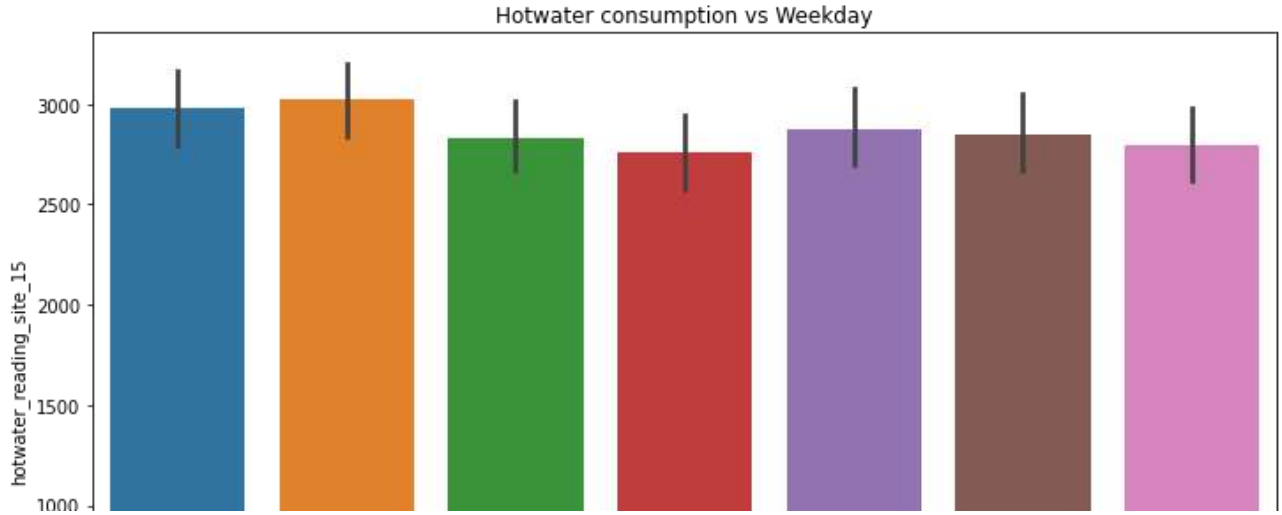
```
df_train_site_15_meter_3['month']=df_train_site_15_meter_3['timestamp'].dt.month
df_train_site_15_meter_3['weekday']=df_train_site_15_meter_3['timestamp'].dt.weekday
df_train_site_15_meter_3['hour']=df_train_site_15_meter_3['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_3
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('hotwater_reading_site_15')
plt.title('Hotwater consumption vs Month')
plt.show()
```



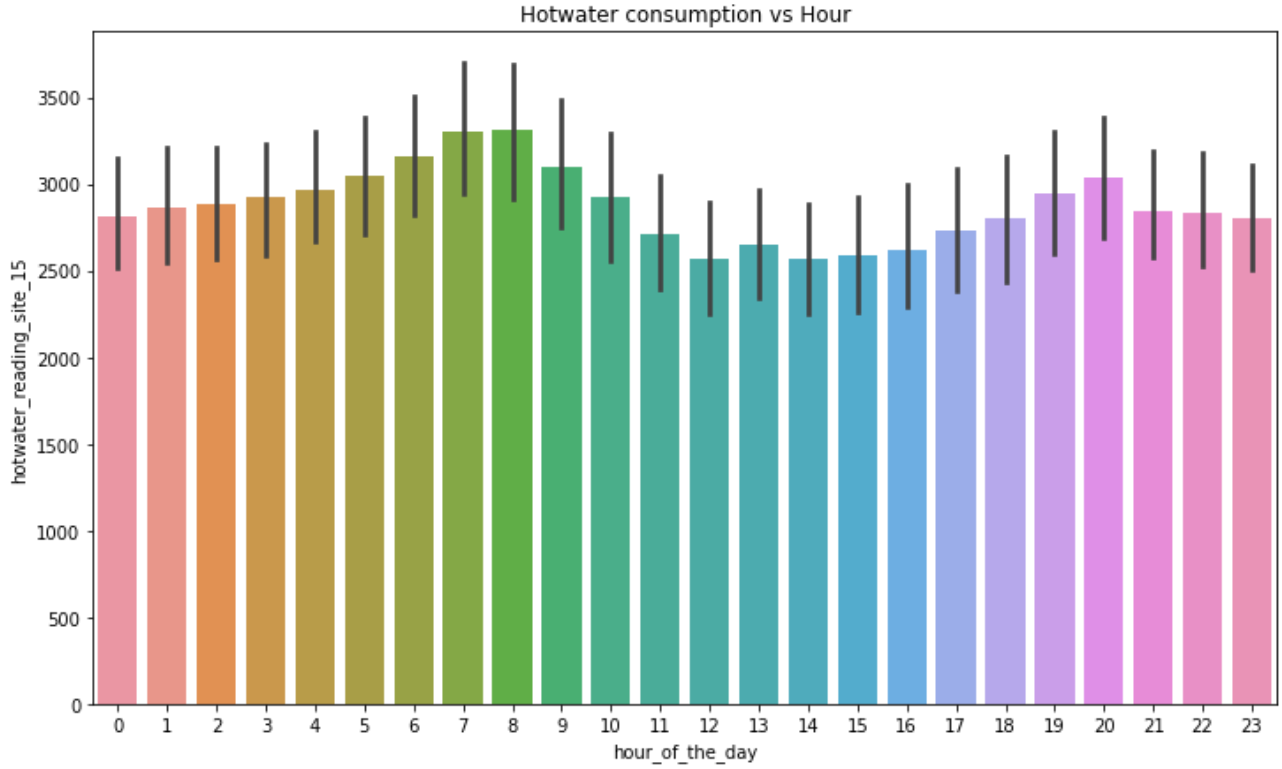
From the above plot we can see that hotwater consumption is higher for the winter months and it decreses gradually as we reach the summer month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_3
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('hotwater_reading_site_15')
plt.title('Hotwater consumption vs Weekday')
plt.show()
```



From the above plot we can see that hotwater does not show specific pattern of usage

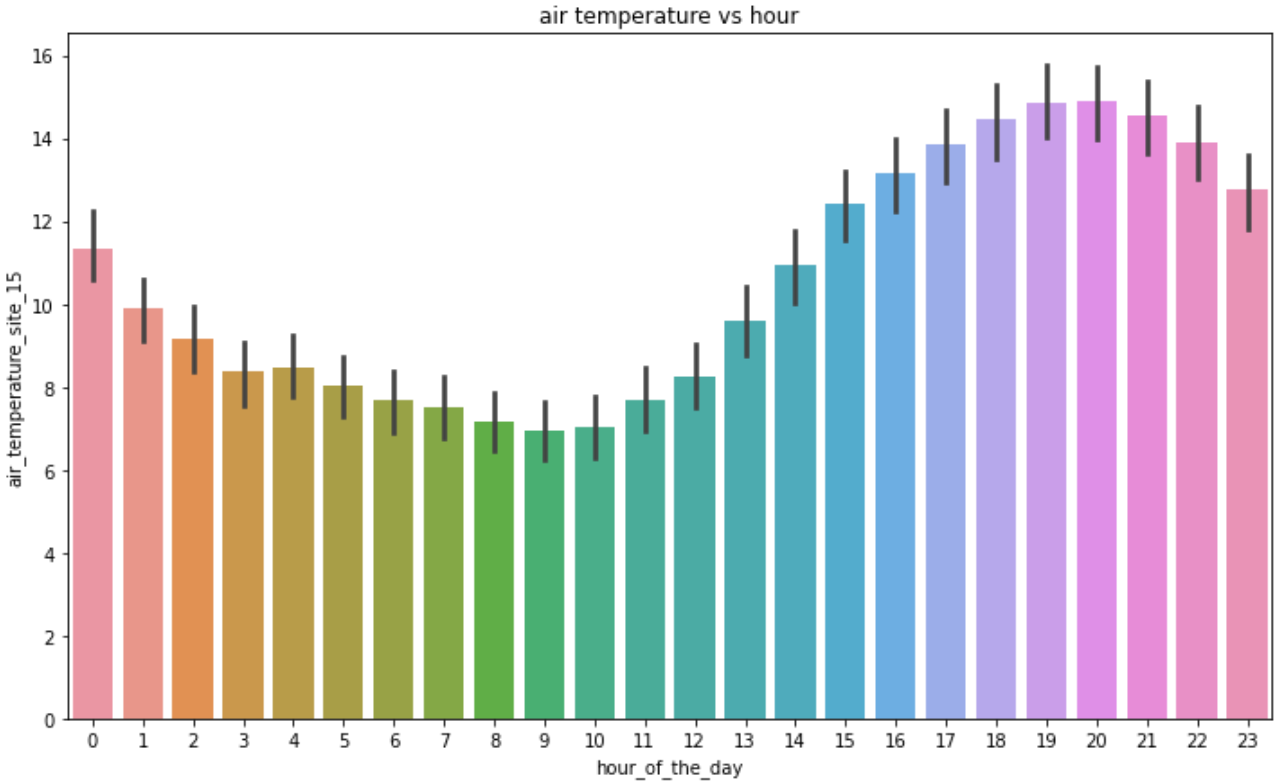
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('hotwater_reading_site_15')
plt.title('Hotwater consumption vs Hour')
plt.show()
```



From the above plot we can see that the hotwater consumption is maximum at the morning time and again shows a slight peak during 20:00 pm abd then again decreases

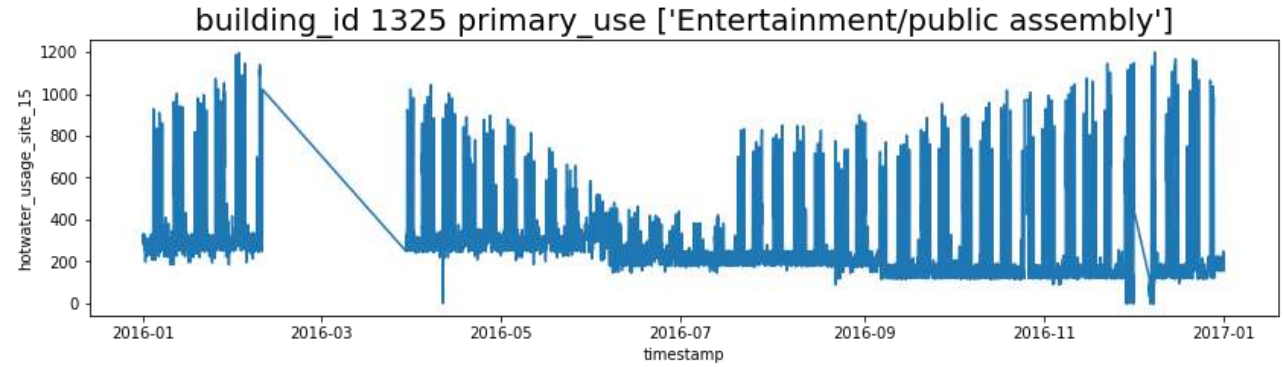
```
fig,ax=plt.subplots(figsize=(12,7))
```

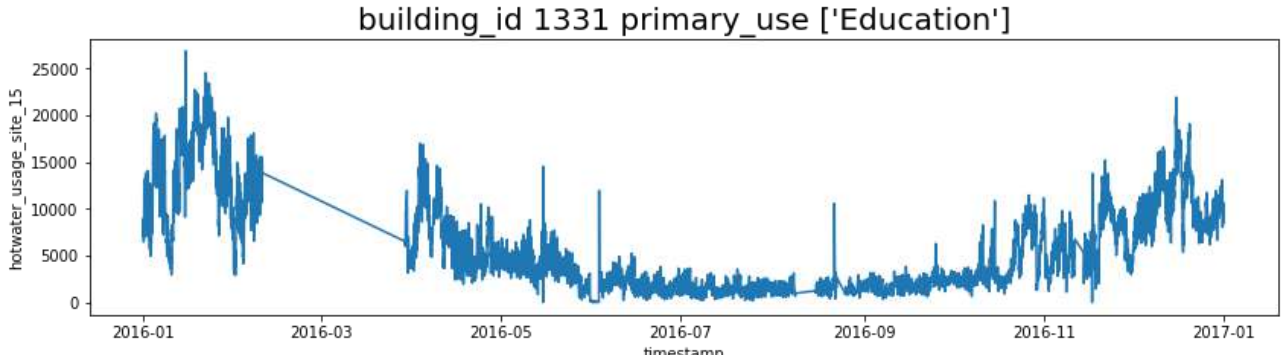
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_3
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_15')
plt.title('air temperature vs hour')
plt.show()
```



The weather timestamp is not in alignment with the local timestamp with the hourly meter readings.The temperature peaks around 20:00 pm

```
fig,ax=plt.subplots(figsize=(14,8),nrows=2,ncols=1,squeeze=False)
for i in range(df_train_site_15_meter_3['building_id'].nunique()):
  g=df_train_site_15_meter_3['building_id'].unique()[i]
  z=df_train_site_15_meter_3.loc[df_train_site_15_meter_3['building_id']==g]
  axes=ax[i%2][i//2]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('hotwater_usage_site_15')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4)
```

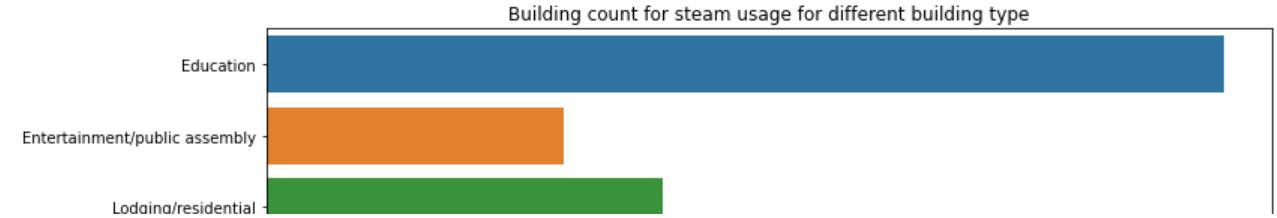




Important Observations

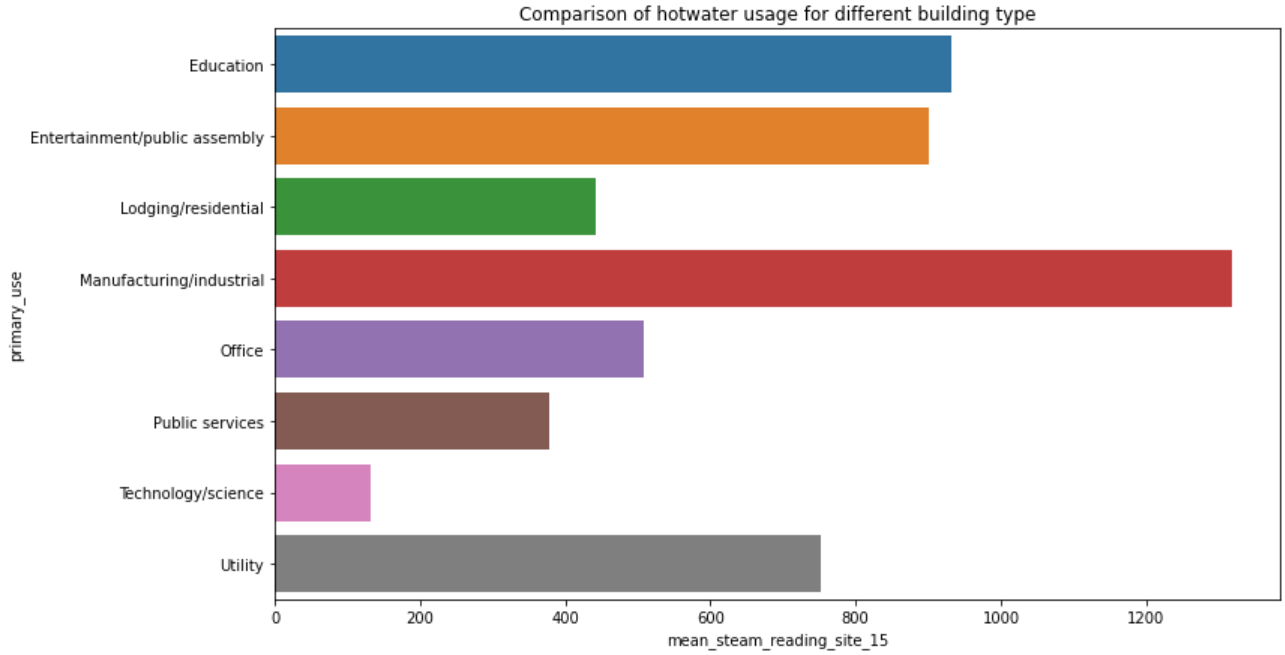
- Here we do not observe any anomaly for the buildings consuming hotwater

```
df_train_site_15_meter_2=df_train_site_15.loc[df_train_site_15['meter']=='steam']


z=df_train_site_15_meter_2.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_steam_usage_site_15')
plt.ylabel('primary_use')
plt.title('Building count for steam usage for different building type')
plt.show()
```



The above plot shows the building count for steam usage for different building type.Education is having the highest building count

```
z=df_train_site_15_meter_2.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_steam_reading_site_15')
plt.ylabel('primary_use')
plt.title('Comparison of hotwater usage for different building type')
plt.show()
```

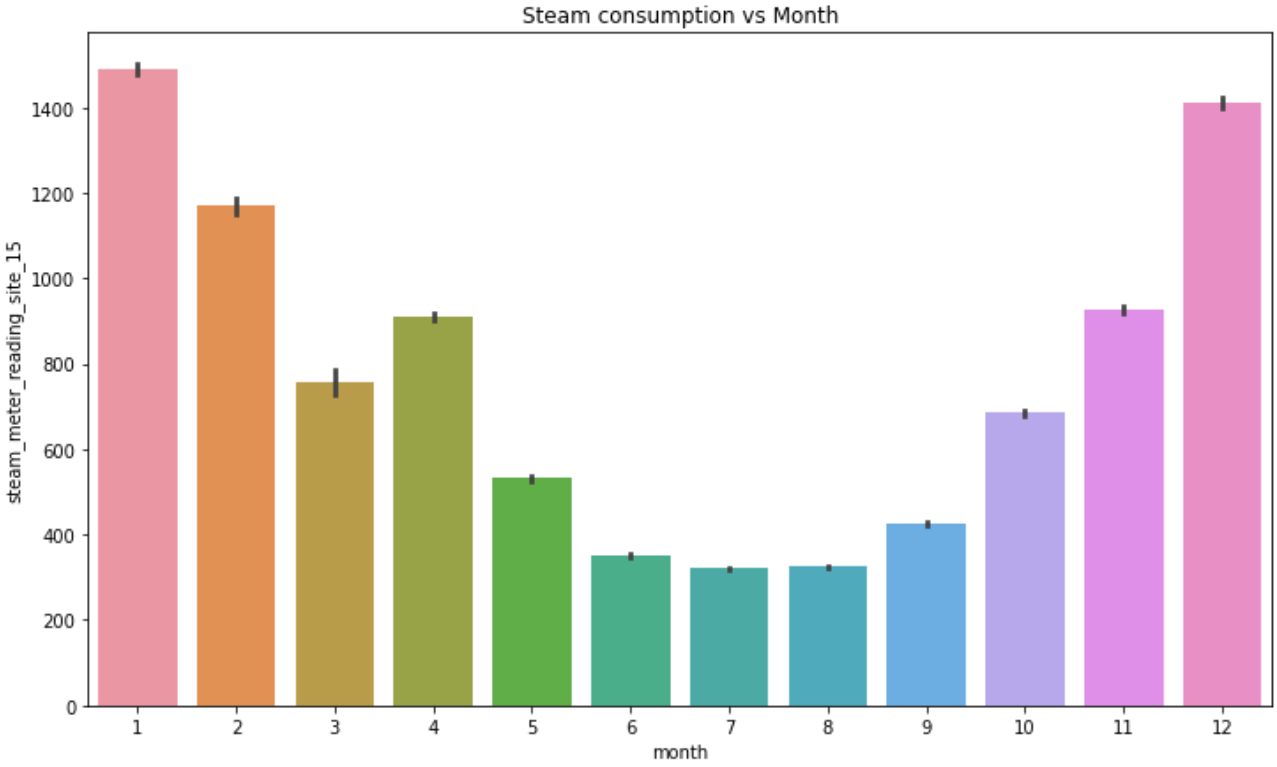


Here we can see that on an average Manufacturing buildings are having the highest consumption of steam

```
df_train_site_15_meter_2['month']=df_train_site_15_meter_2['timestamp'].dt.month
df_train_site_15_meter_2['weekday']=df_train_site_15_meter_2['timestamp'].dt.weekday
df_train_site_15_meter_2['hour']=df_train_site_15_meter_2['timestamp'].dt.hour
```

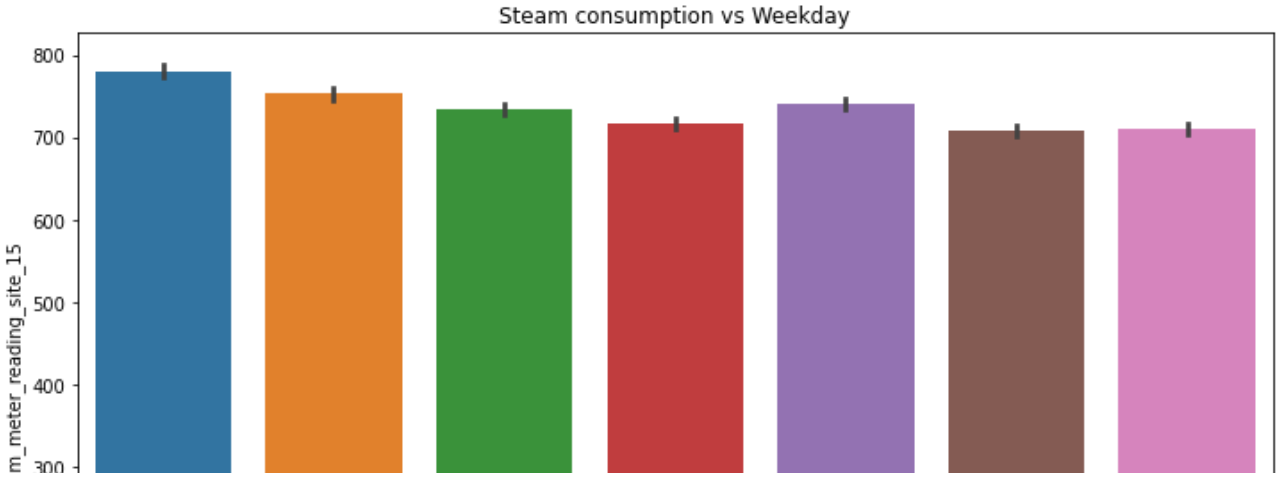
```
fig,ax=plt.subplots(figsize=(12,7))
```

```
z=df_train_site_15_meter_2
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('steam_meter_reading_site_15')
plt.title('Steam consumption vs Month')
plt.show()
```



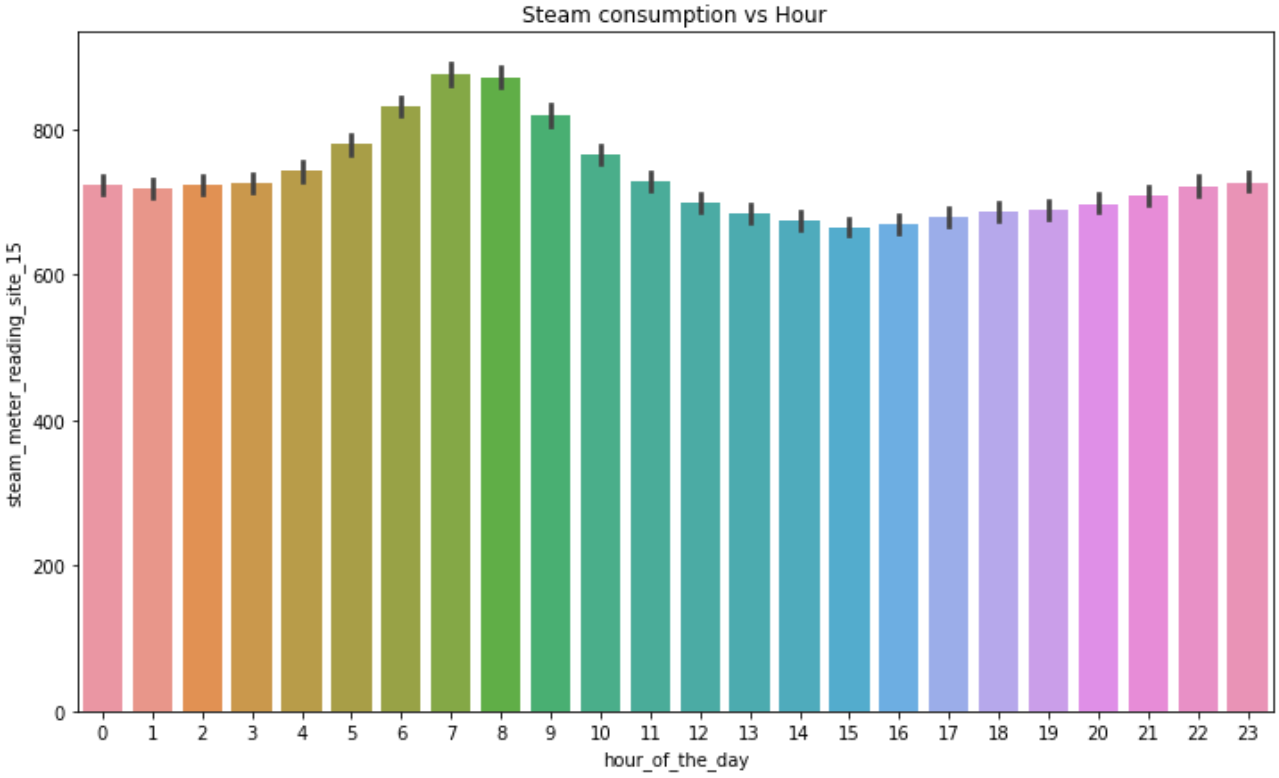
Here we can see that steam is having higher consumption for the winter month as compared to the summer month

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_2
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('steam_meter_reading_site_15')
plt.title('Steam consumption vs Weekday')
plt.show()
```



Here we can see that steam consumption is not showing a definite pattern over the week

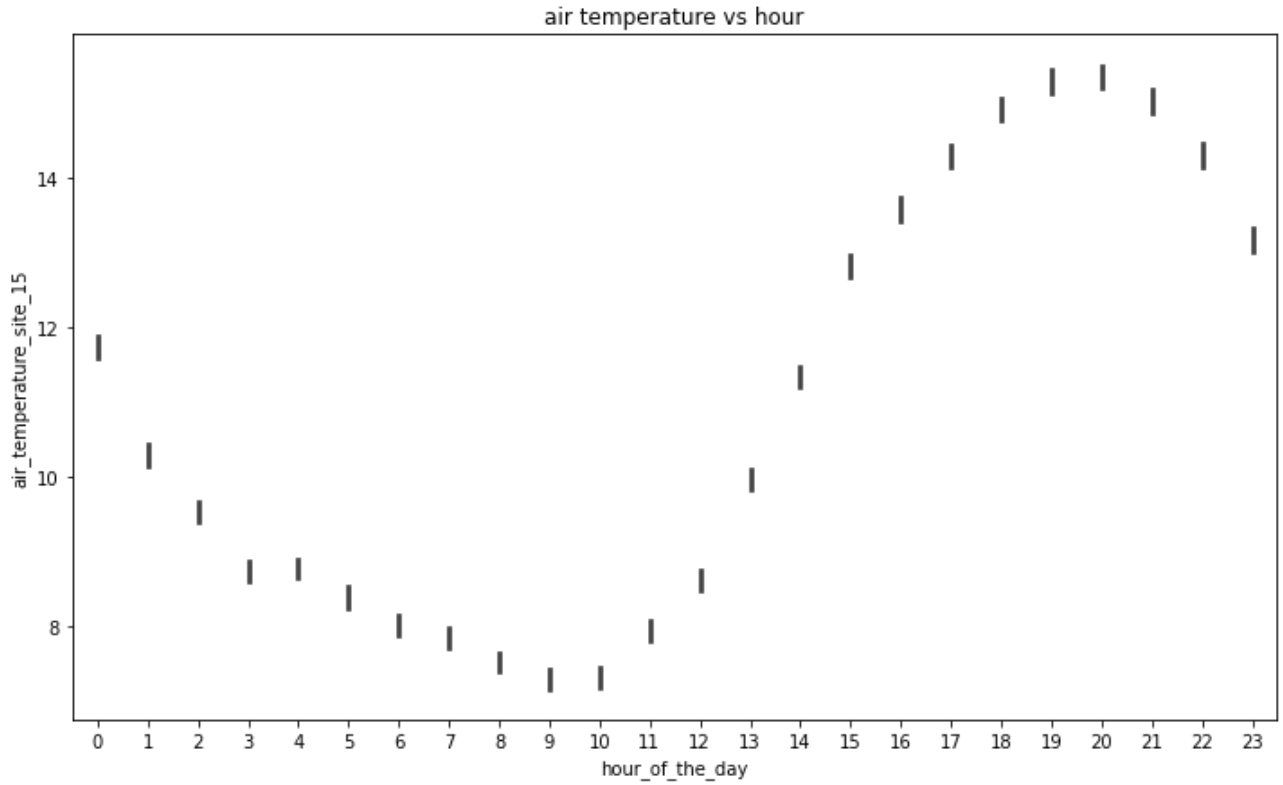
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('steam_meter_reading_site_15')
plt.title('Steam consumption vs Hour')
plt.show()
```



Steam consumption peaks around the morning hours decreases after that and again we see an increase after 16:00 pm

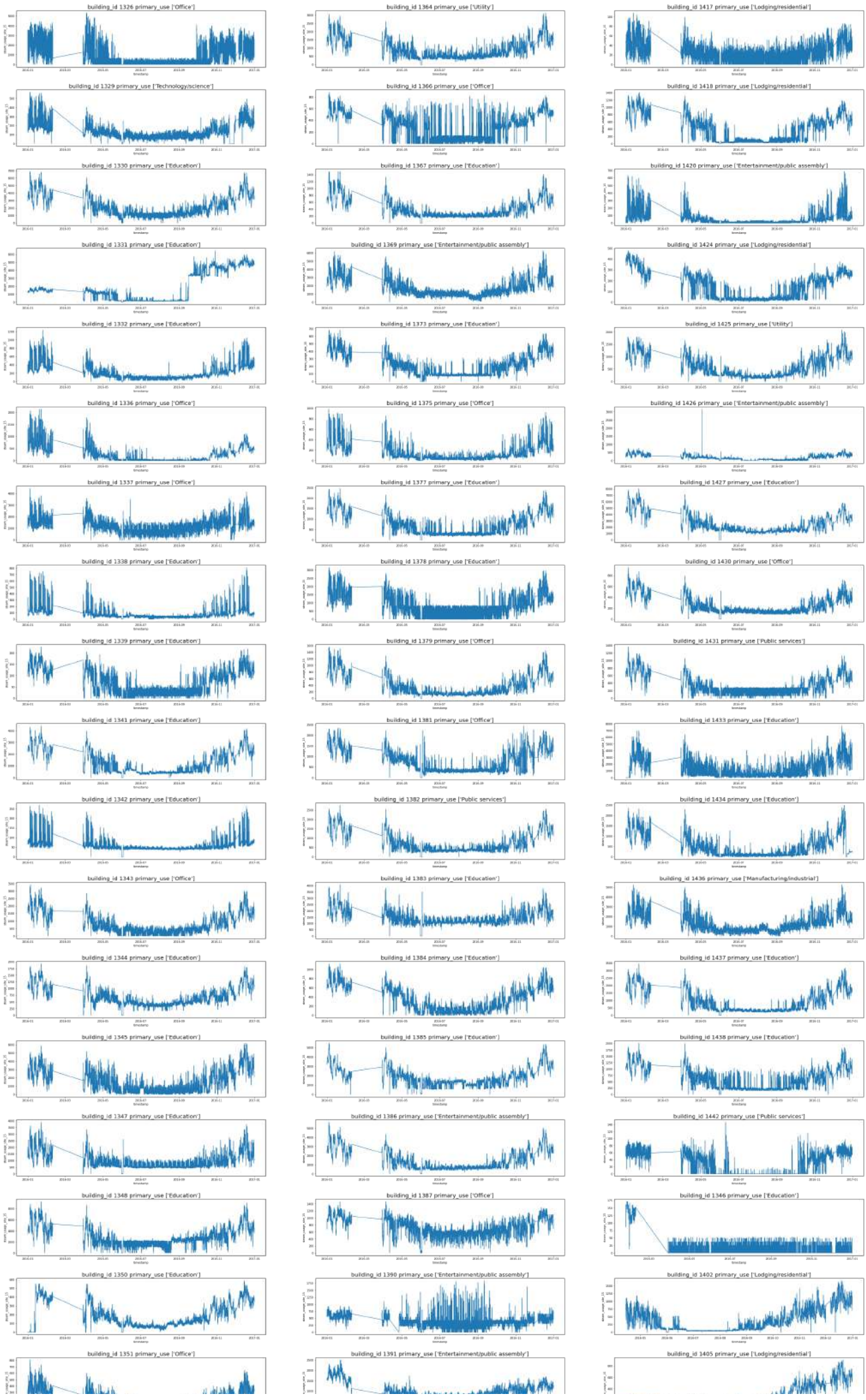
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_2
```

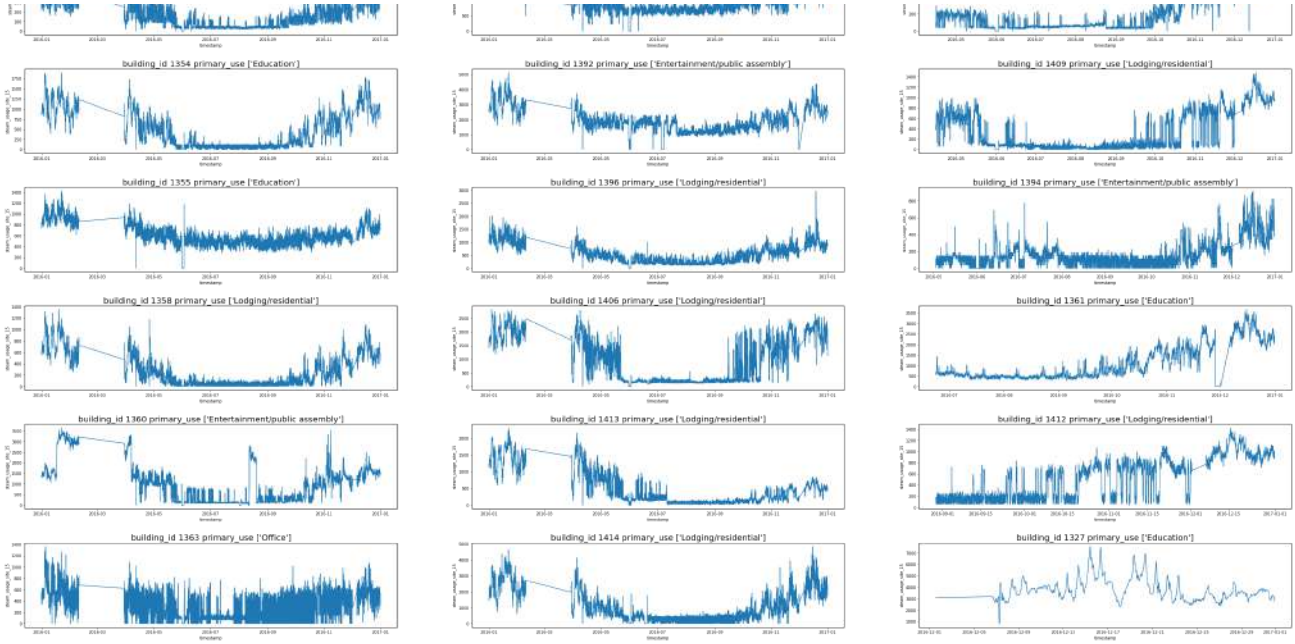
```
z=df_train_site_15_meter_2
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_15')
plt.title('air temperature vs hour')
plt.show()
```



From here we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter readings.The temperature peaks around 20:00 pm

```
fig,ax=plt.subplots(figsize=(55,120),nrows=23,ncols=3,squeeze=True)
for i in range(df_train_site_15_meter_2['building_id'].nunique()):
  g=df_train_site_15_meter_2['building_id'].unique()[i]
  z=df_train_site_15_meter_2.loc[df_train_site_15_meter_2['building_id']==g]
  axes=ax[i%23][i//23]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('steam_usage_site_15')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.4)
```


building_id 1326 primary_use ['Office']
building_id 1329 primary_use ['Technology/science']
building_id 1330 primary_use ['Education']
building_id 1331 primary_use ['Education']
building_id 1332 primary_use ['Education']
building_id 1336 primary_use ['Office']
building_id 1337 primary_use ['Office']
building_id 1338 primary_use ['Education']
building_id 1339 primary_use ['Education']
building_id 1341 primary_use ['Education']
building_id 1342 primary_use ['Education']
building_id 1343 primary_use ['Office']
building_id 1344 primary_use ['Education']
building_id 1345 primary_use ['Education']
building_id 1347 primary_use ['Education']
building_id 1348 primary_use ['Education']
building_id 1350 primary_use ['Education']
building_id 1351 primary_use ['Office']
building_id 1364 primary_use ['Utility']
building_id 1366 primary_use ['Office']
building_id 1367 primary_use ['Education']
building_id 1369 primary_use ['Entertainment/public assembly']
building_id 1373 primary_use ['Education']
building_id 1375 primary_use ['Office']
building_id 1377 primary_use ['Education']
building_id 1378 primary_use ['Education']
building_id 1379 primary_use ['Office']
building_id 1381 primary_use ['Office']
building_id 1382 primary_use ['Public services']
building_id 1383 primary_use ['Education']
building_id 1384 primary_use ['Education']
building_id 1385 primary_use ['Education']
building_id 1386 primary_use ['Entertainment/public assembly']
building_id 1387 primary_use ['Office']
building_id 1390 primary_use ['Entertainment/public assembly']
building_id 1391 primary_use ['Entertainment/public assembly']
building_id 1417 primary_use ['Lodging/residential']
building_id 1418 primary_use ['Lodging/residential']
building_id 1420 primary_use ['Entertainment/public assembly']
building_id 1424 primary_use ['Lodging/residential']
building_id 1425 primary_use ['Utility']
building_id 1426 primary_use ['Entertainment/public assembly']
building_id 1427 primary_use ['Education']
building_id 1430 primary_use ['Office']
building_id 1431 primary_use ['Public services']
building_id 1433 primary_use ['Education']
building_id 1434 primary_use ['Education']
building_id 1436 primary_use ['Manufacturing/industrial']
building_id 1437 primary_use ['Education']
building_id 1438 primary_use ['Education']
building_id 1442 primary_use ['Public services']
building_id 1346 primary_use ['Education']
building_id 1402 primary_use ['Lodging/residential']
building_id 1405 primary_use ['Lodging/residential']



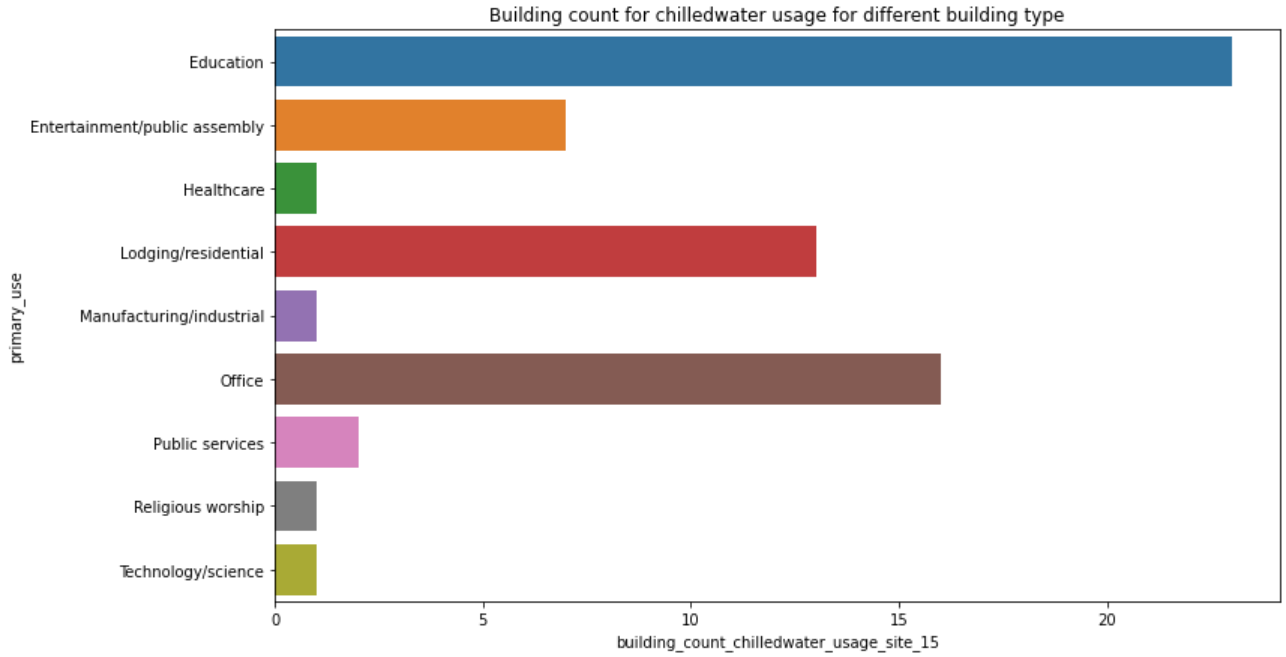
Important Observations

- Here we can observe that for most of the buildings we are seeing a constant slope of steam meter readings which does not needs to be filtered out.Here we need to remove the constant meter readings ans spikes.

```
df_train_site_15_meter_1=df_train_site_15.loc[df_train_site_15['meter']=='chilledwater']
```

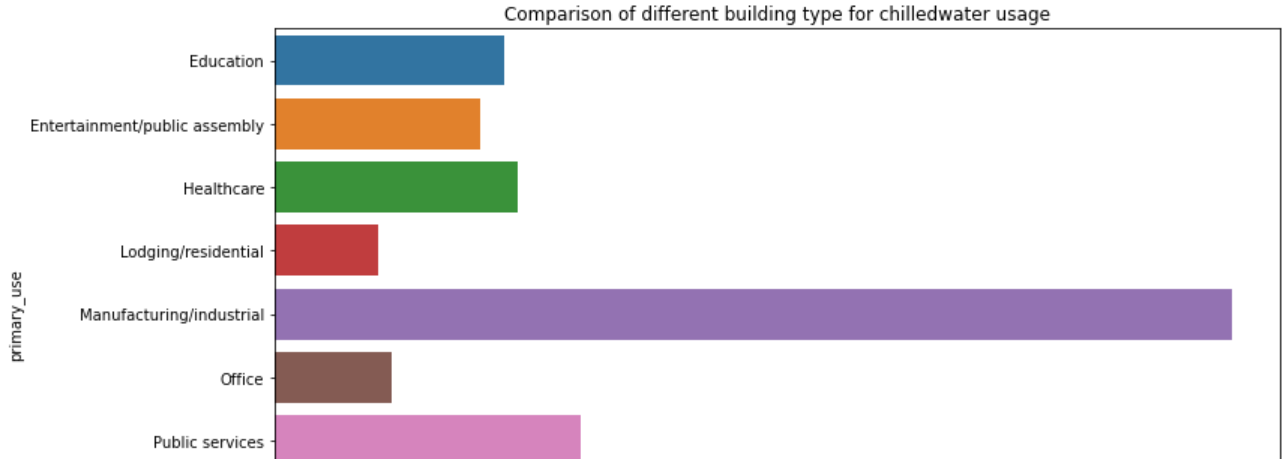
```
z=df_train_site_15_meter_1.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
```

```
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_chilledwater_usage_site_15')
plt.ylabel('primary_use')
plt.title('Building count for chilledwater usage for different building type')
plt.show()
```

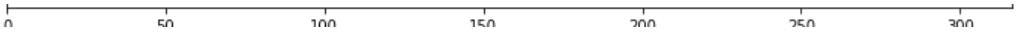


The above plot shows the building count for chilledwater usage for different building type

```
z=df_train_site_15_meter_1.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_chilledwater_reading_site_15')
plt.ylabel('primary_use')
plt.title('Comparison of different building type for chilledwater usage')
plt.show()
```

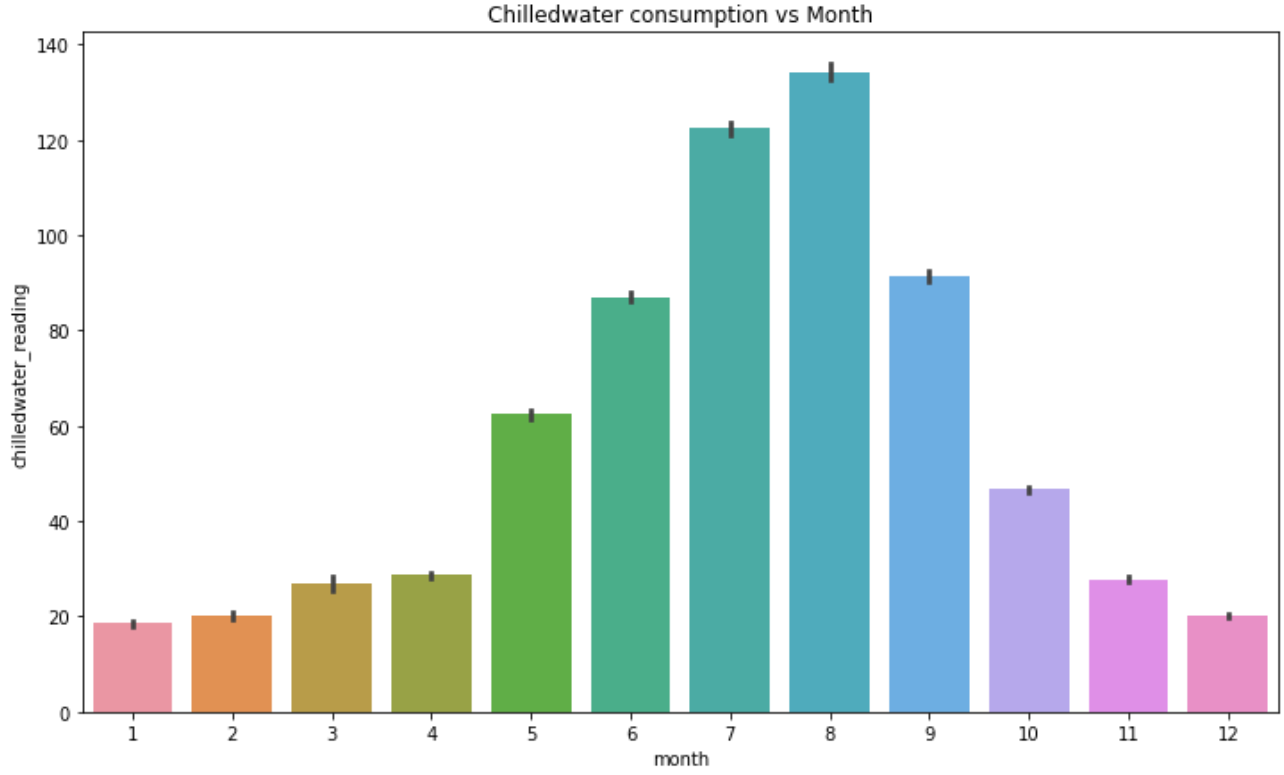


From the above plot we can see that manufacturing and science buildings are having higher chilledwater consumption as compared to the other building type



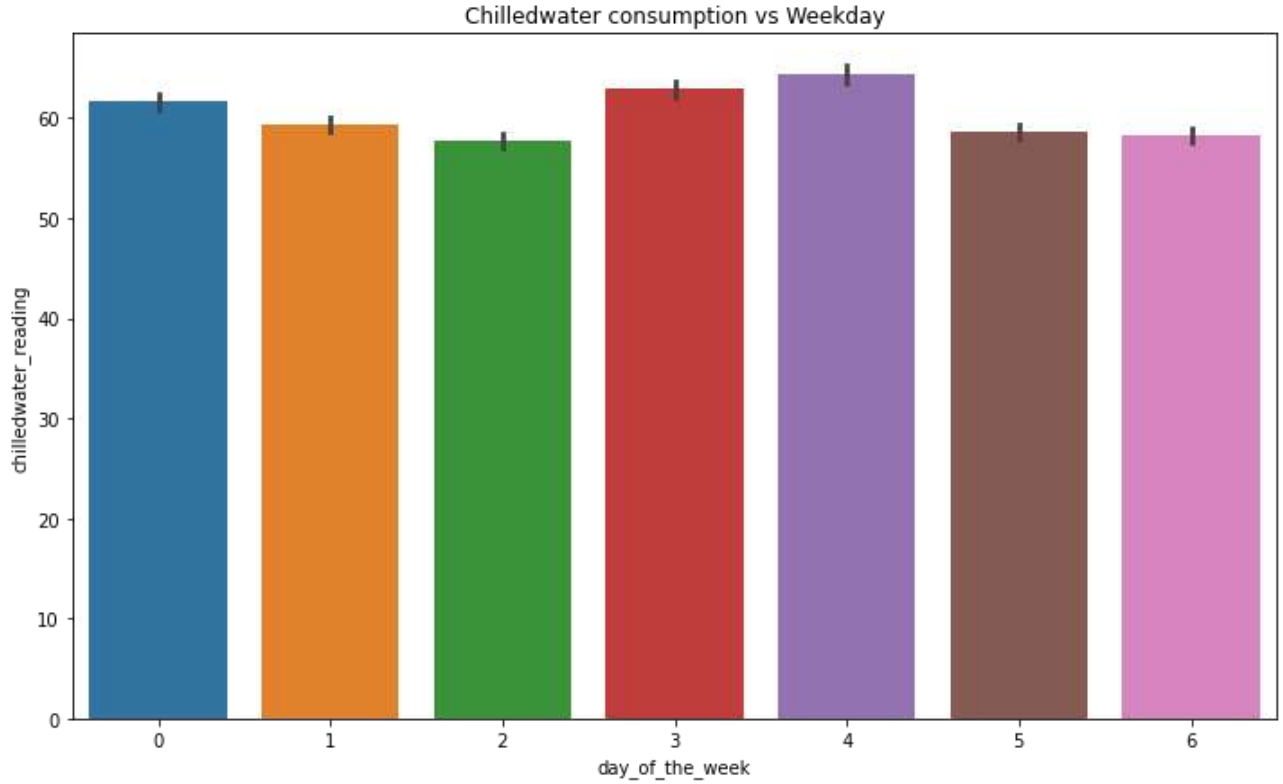

```
df_train_site_15_meter_1['month']=df_train_site_15_meter_1['timestamp'].dt.month
df_train_site_15_meter_1['weekday']=df_train_site_15_meter_1['timestamp'].dt.weekday
df_train_site_15_meter_1['hour']=df_train_site_15_meter_1['timestamp'].dt.hour
```

```
z=df_train_site_15_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('chilledwater_reading')
plt.title('Chilledwater consumption vs Month')
plt.show()
```



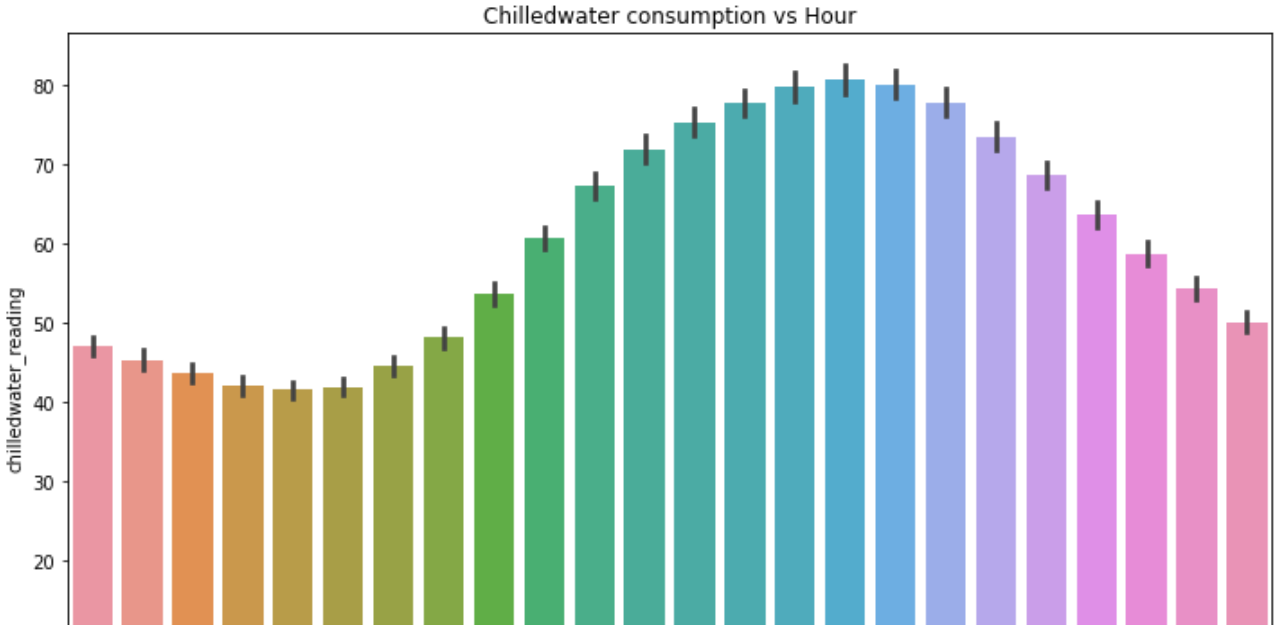
Chilledwater consumption is higher for the summer month as compared to the winter month

```
z=df_train_site_15_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('chilledwater_reading')
plt.title('Chilledwater consumption vs Weekday')
plt.show()
```



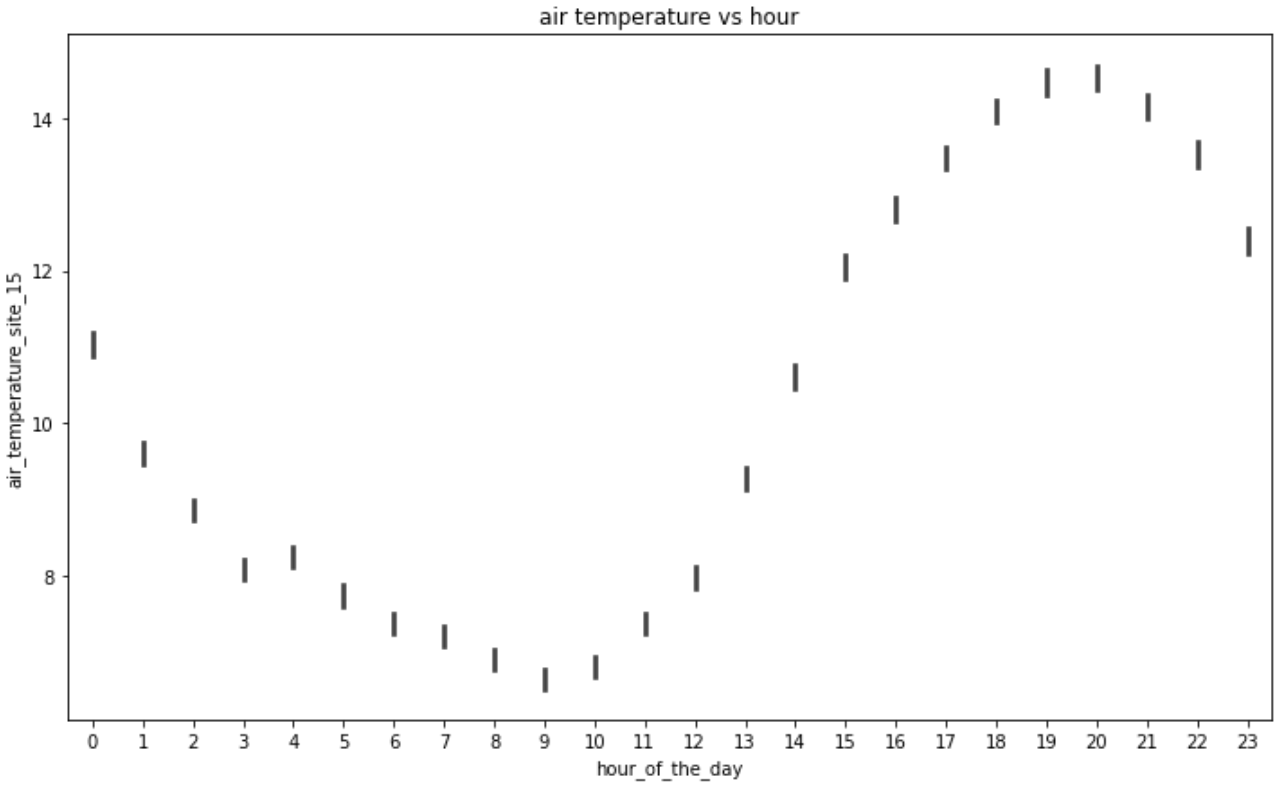
Here also chilledwater consumption is not showing specific pattern over the week

```
z=df_train_site_15_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('chilledwater_reading')
plt.title('Chilledwater consumption vs Hour')
plt.show()
```



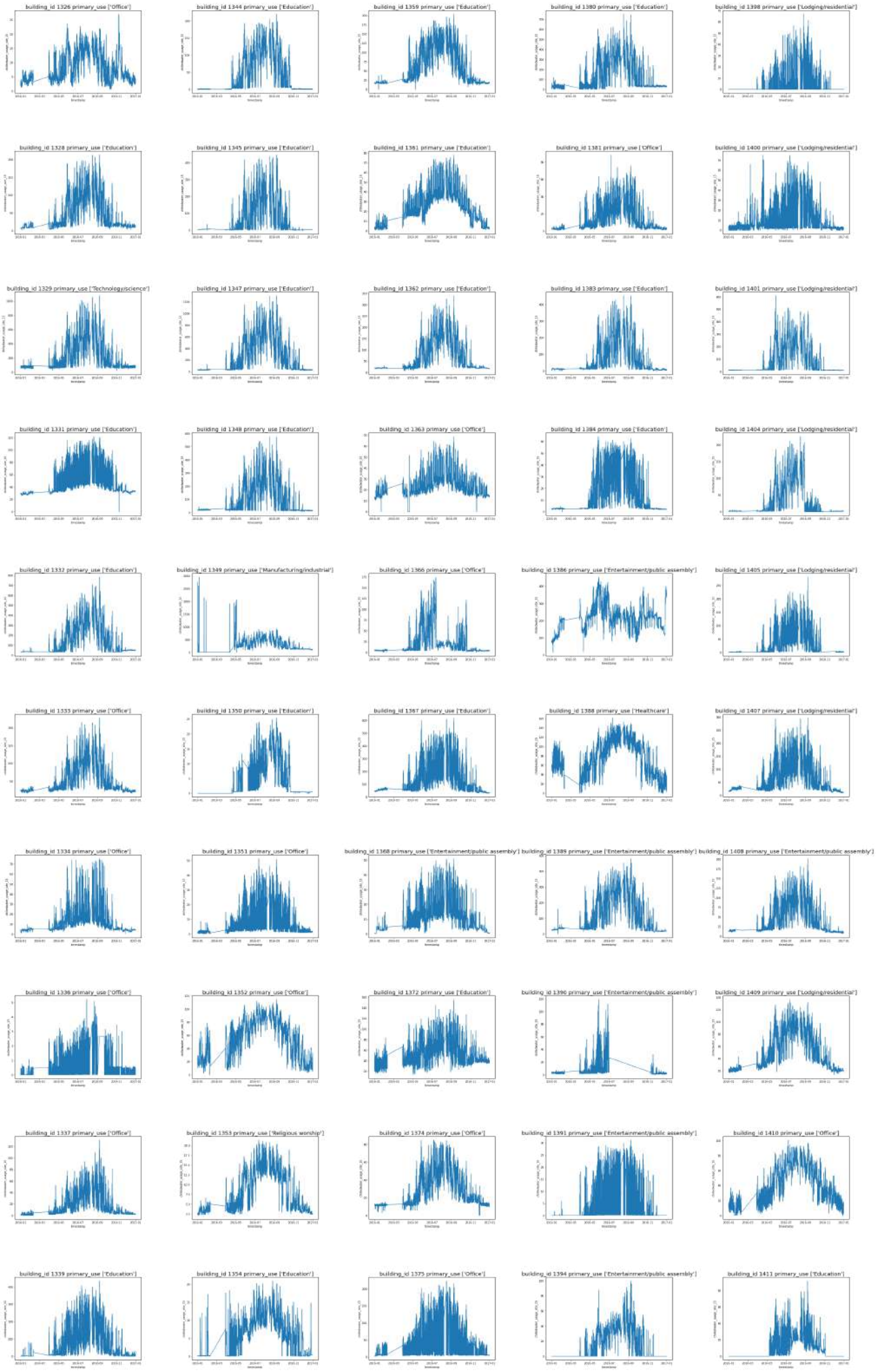
From the above plot we can see that chilledwater consumption starts rising from 06:00 am in the morning and it peaks around 15:00 pm and then again decreases gradually

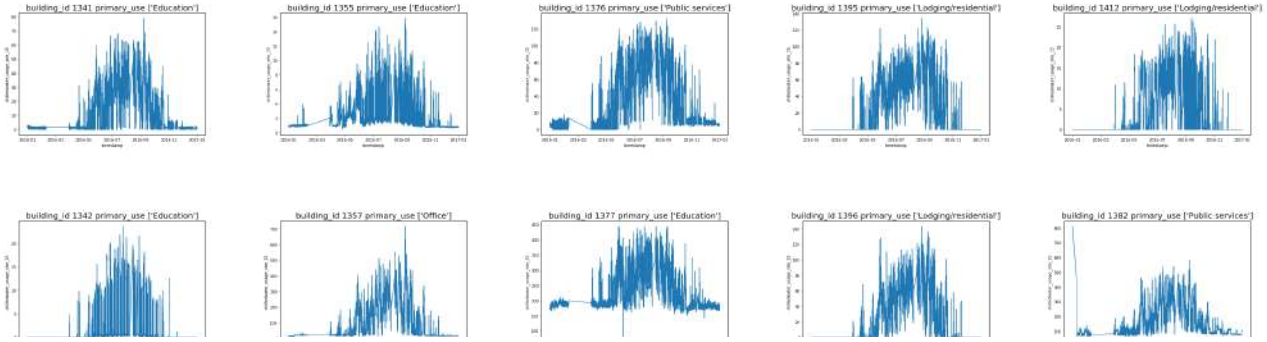
```
z=df_train_site_15_meter_1
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_15')
plt.title('air temperature vs hour')
plt.show()
```



Here we can see that the weather timestamp is not in alignment with the local timestamp with the hourly meter readind.The temperature peaks around 20:00 pm

```
fig,ax=plt.subplots(figsize=(55,120),nrows=13,ncols=5,squeeze=True)
for i in range(df_train_site_15_meter_1['building_id'].nunique()):
  g=df_train_site_15_meter_1['building_id'].unique()[i]
  z=df_train_site_15_meter_1.loc[df_train_site_15_meter_1['building_id']==g]
  axes=ax[i%13][i//13]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('chilledwater_usage_site_15')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.7,wspace=0.4)
```


building_id 1326 primary_use ['Office']
building_id 1344 primary_use ['Education']
building_id 1359 primary_use ['Education']
building_id 1380 primary_use ['Education']
building_id 1398 primary_use ['Lodging/residential']
building_id 1328 primary_use ['Education']
building_id 1345 primary_use ['Education']
building_id 1361 primary_use ['Education']
building_id 1381 primary_use ['Office']
building_id 1400 primary_use ['Lodging/residential']
building_id 1329 primary_use ['Technology/science']
building_id 1347 primary_use ['Education']
building_id 1362 primary_use ['Education']
building_id 1383 primary_use ['Education']
building_id 1401 primary_use ['Lodging/residential']
building_id 1331 primary_use ['Education']
building_id 1348 primary_use ['Education']
building_id 1363 primary_use ['Office']
building_id 1384 primary_use ['Education']
building_id 1404 primary_use ['Lodging/residential']
building_id 1332 primary_use ['Education']
building_id 1349 primary_use ['Manufacturing/industrial']
building_id 1366 primary_use ['Office']
building_id 1386 primary_use ['Entertainment/public assembly']
building_id 1405 primary_use ['Lodging/residential']
building_id 1333 primary_use ['Office']
building_id 1350 primary_use ['Education']
building_id 1367 primary_use ['Education']
building_id 1388 primary_use ['Healthcare']
building_id 1407 primary_use ['Lodging/residential']
building_id 1334 primary_use ['Office']
building_id 1351 primary_use ['Office']
building_id 1368 primary_use ['Entertainment/public assembly']
building_id 1389 primary_use ['Entertainment/public assembly']
building_id 1408 primary_use ['Entertainment/public assembly']
building_id 1336 primary_use ['Office']
building_id 1352 primary_use ['Office']
building_id 1372 primary_use ['Education']
building_id 1390 primary_use ['Entertainment/public assembly']
building_id 1409 primary_use ['Lodging/residential']
building_id 1337 primary_use ['Office']
building_id 1353 primary_use ['Religious worship']
building_id 1374 primary_use ['Office']
building_id 1391 primary_use ['Entertainment/public assembly']
building_id 1410 primary_use ['Office']
building_id 1339 primary_use ['Education']
building_id 1354 primary_use ['Education']
building_id 1375 primary_use ['Office']
building_id 1394 primary_use ['Entertainment/public assembly']
building_id 1411 primary_use ['Education']

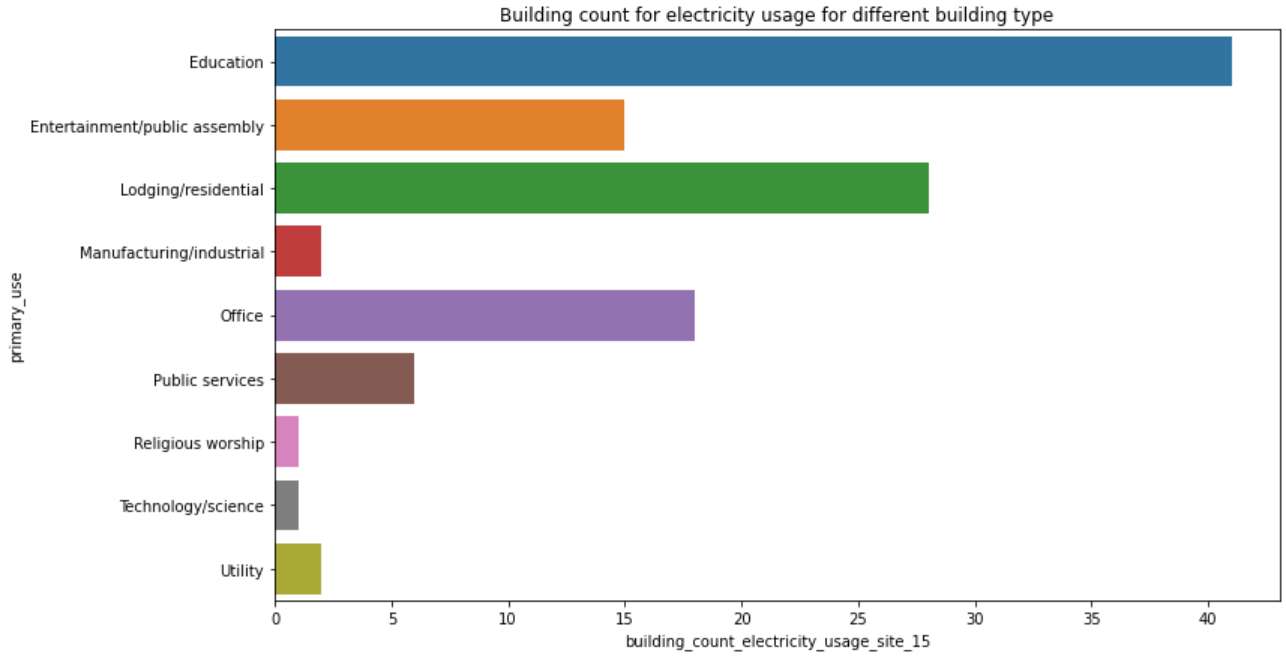


Important Observations

- The building 1363 1410 are havinf constant zero mete rreadings which needs to be filtered out.
- For buildings 1349 and 1382 we need to filter out the spikes.

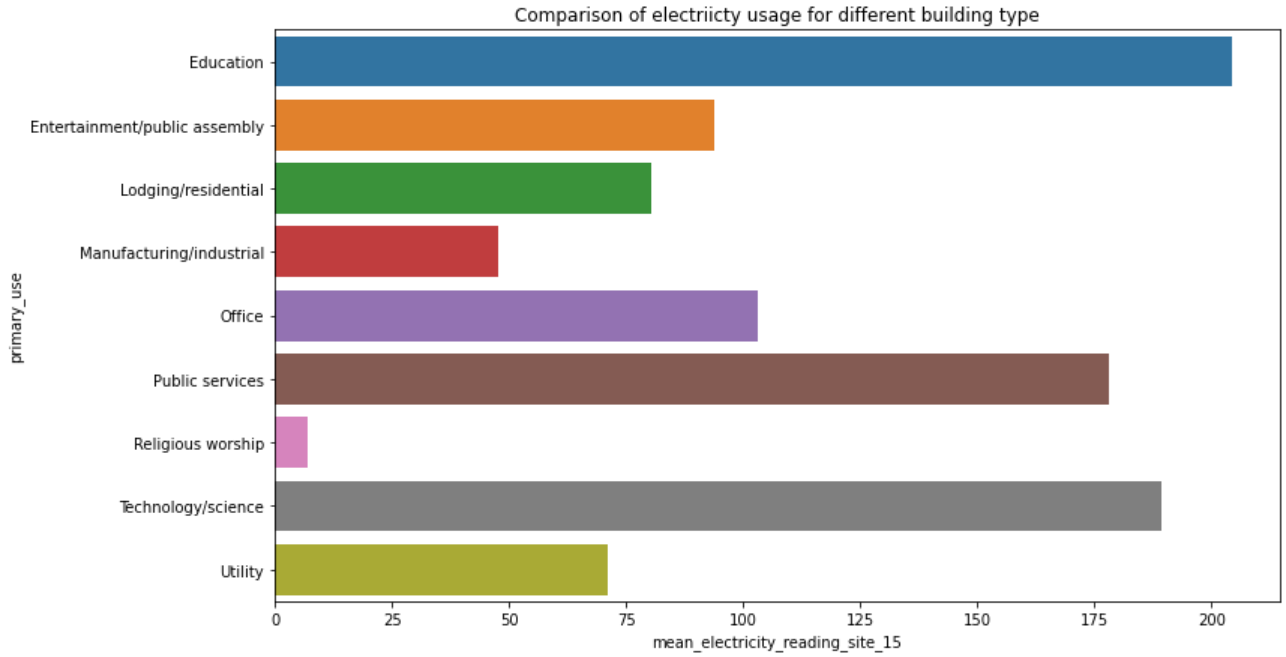
```
df_train_site_15_meter_0=df_train_site_15.loc[df_train_site_15['meter']=='electricity']
```

```
z=df_train_site_15_meter_0.groupby(['primary_use'])
z=z['building_id'].nunique().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='building_id',y='primary_use')
plt.xlabel('building_count_electricity_usage_site_15')
plt.ylabel('primary_use')
plt.title('Building count for electricity usage for different building type')
plt.show()
```



The above plot shows the building count for electricity usage for different building

```
z=df_train_site_15_meter_0.groupby(['primary_use'])
z=z['meter_reading'].mean().reset_index()
fig,ax=plt.subplots(figsize=(12,7))
sns.barplot(ax=ax,data=z,x='meter_reading',y='primary_use')
plt.xlabel('mean_electricity_reading_site_15')
plt.ylabel('primary_use')
plt.title('Comparison of electriicty usage for different building type')
plt.show()
```

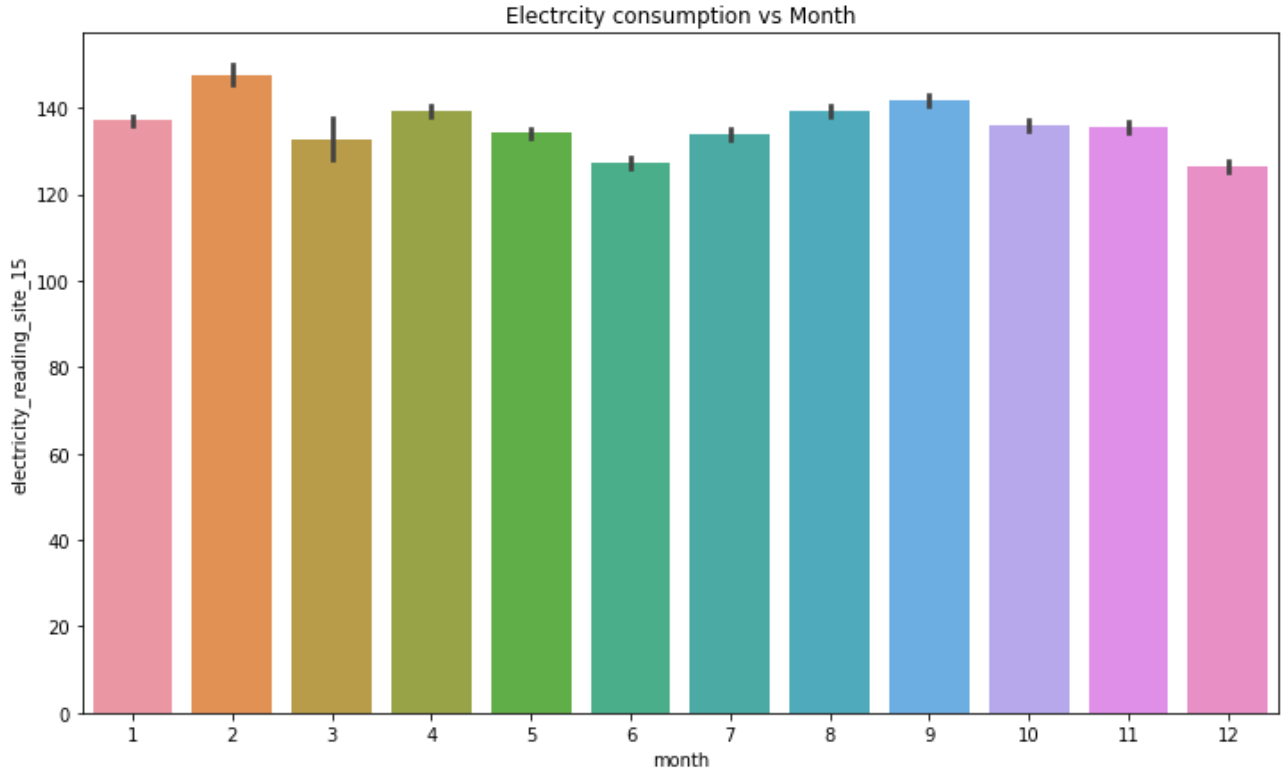


Important Observations

- Here we can see that educational buildings are having the highest eelctrical consumption

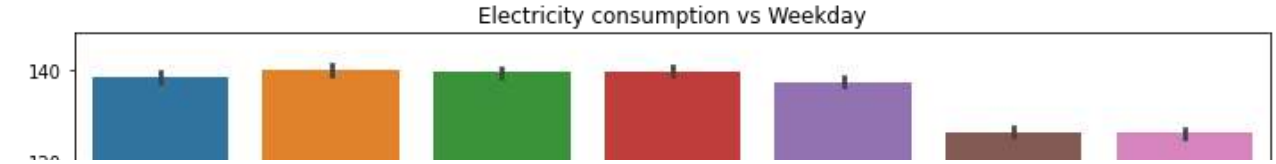
```
df_train_site_15_meter_0['month']=df_train_site_15_meter_0['timestamp'].dt.month
df_train_site_15_meter_0['weekday']=df_train_site_15_meter_0['timestamp'].dt.weekday
df_train_site_15_meter_0['hour']=df_train_site_15_meter_0['timestamp'].dt.hour
```

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_0
sns.barplot(ax=ax,data=z,x='month',y='meter_reading')
plt.xlabel('month')
plt.ylabel('electricity_reading_site_15')
plt.title('Electrcity consumption vs Month')
plt.show()
```



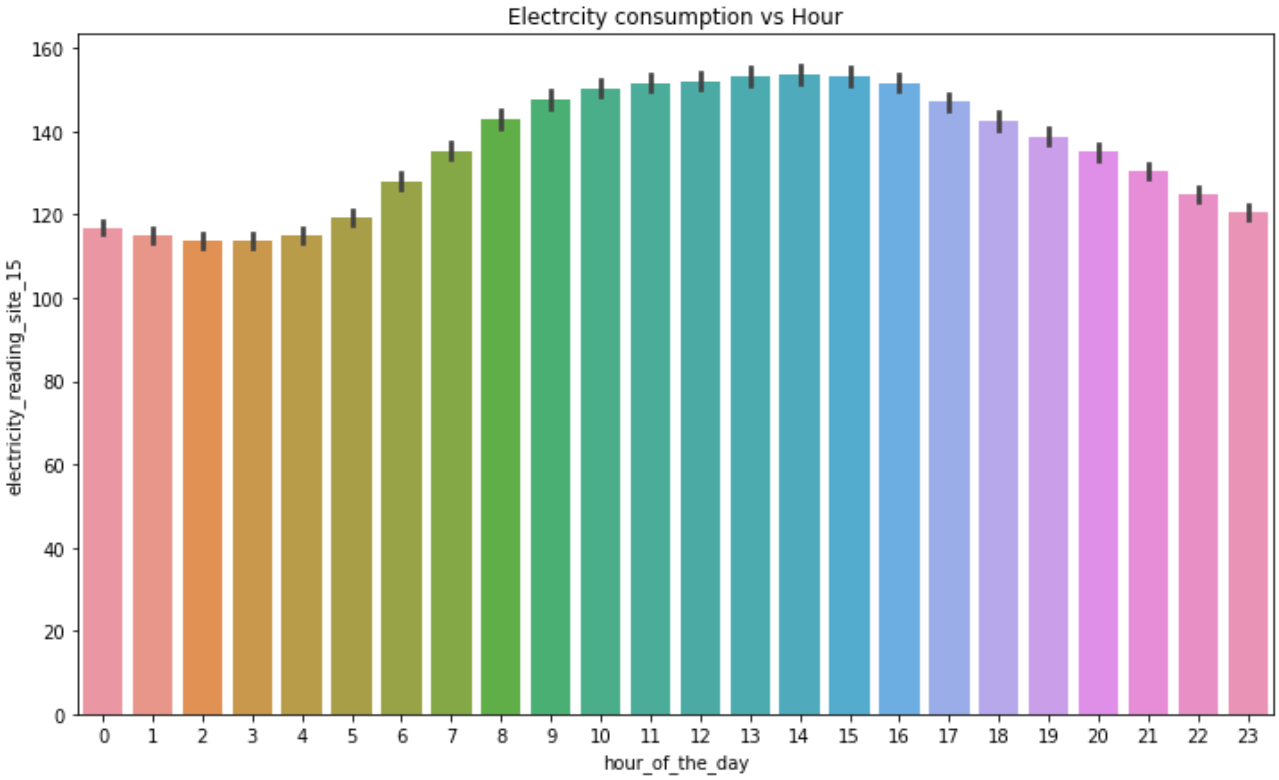
Here we can see that electricity consumption varies over the month but doea not follow a specific pattern

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_0
sns.barplot(ax=ax,data=z,x='weekday',y='meter_reading')
plt.xlabel('day_of_the_week')
plt.ylabel('electricity_reading_site_15')
plt.title('Electricity consumption vs Weekday')
plt.show()
```



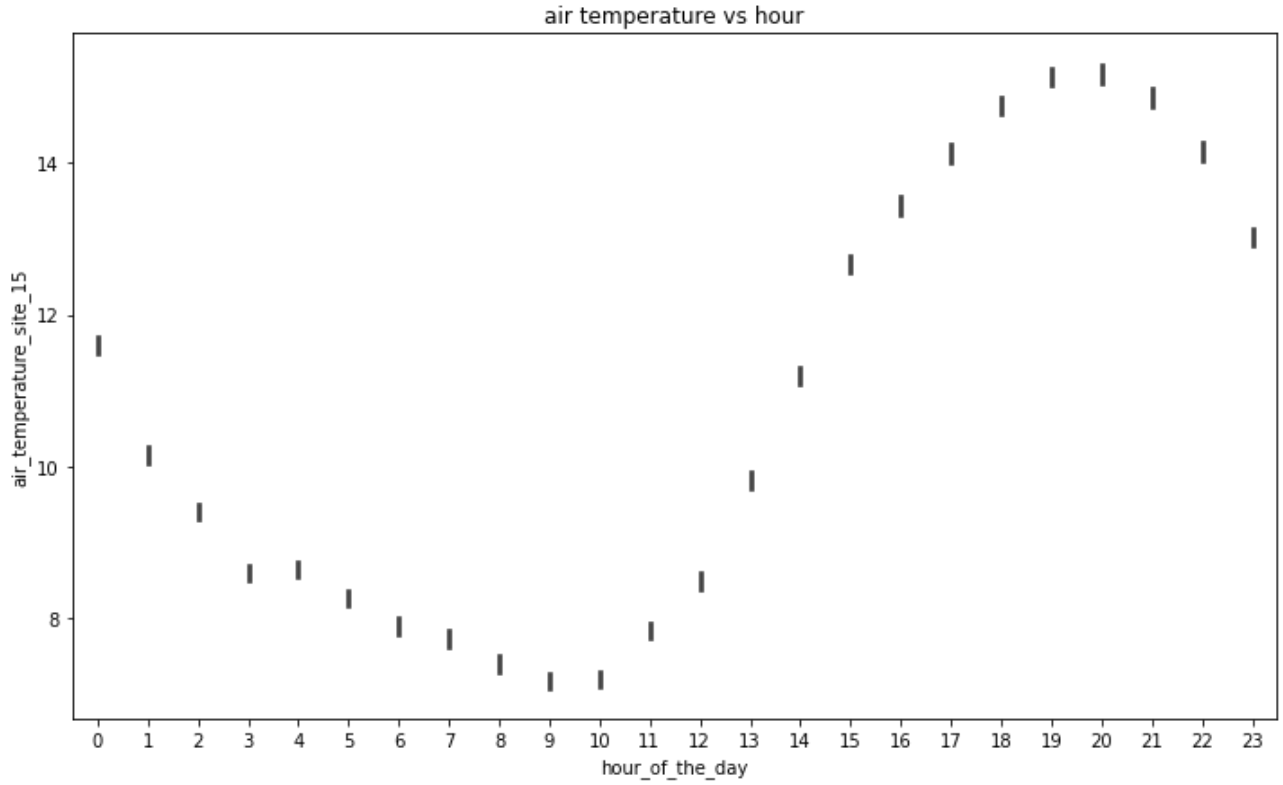
From the above plot we can see that electrcity consumption is less on the weekend as comapred to the weekday

```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='meter_reading')
plt.xlabel('hour_of_the_day')
plt.ylabel('electricity_reading_site_15')
plt.title('Electrcity consumption vs Hour')
plt.show()
```



From the above plot we can see that electricity consumption starts increasing from 6:00 am in the morning and peaks around 14:00 pm and then again starts decreasing gradually over time

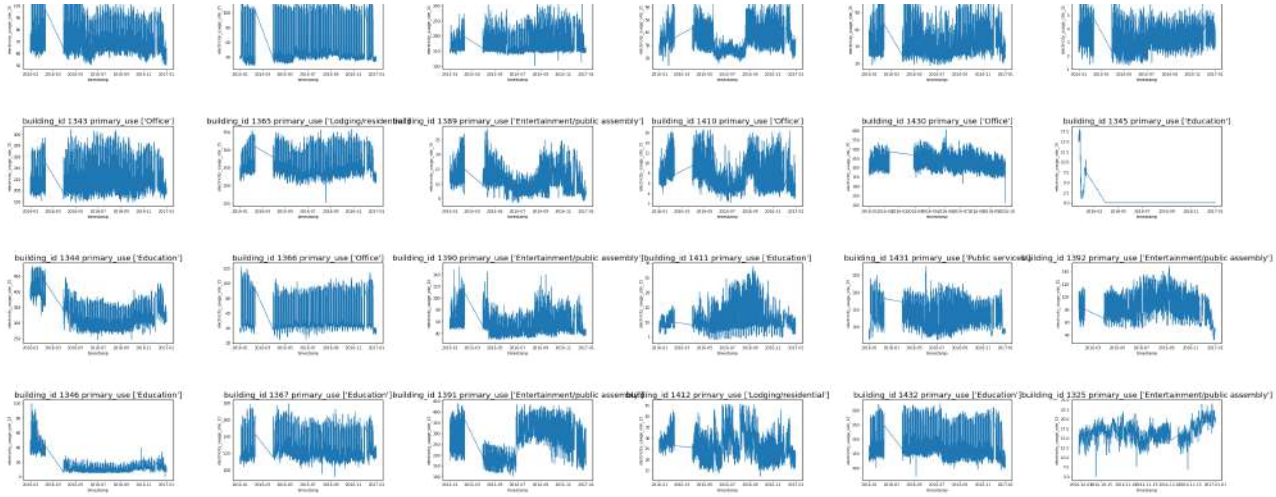
```
fig,ax=plt.subplots(figsize=(12,7))
z=df_train_site_15_meter_0
sns.barplot(ax=ax,data=z,x='hour',y='air_temperature')
plt.xlabel('hour_of_the_day')
plt.ylabel('air_temperature_site_15')
plt.title('air temperature vs hour')
plt.show()
```



FRom the above plot we can see that the weather timestamp is not in alignment with the local timestamp of the hourly meter reading.The temperature peaks around 20:00 pm

```
fig,ax=plt.subplots(figsize=(55,120),nrows=19,ncols=6,squeeze=True)
for i in range(df_train_site_15_meter_0['building_id'].nunique()):
  g=df_train_site_15_meter_0['building_id'].unique()[i]
  z=df_train_site_15_meter_0.loc[df_train_site_15_meter_0['building_id']==g]
  axes=ax[i%19][i//19]
  axes.plot(z['timestamp'],z['meter_reading'])
  axes.set_xlabel('timestamp')
  axes.set_ylabel('electricity_usage_site_15')
  axes.set_title('building_id {} primary_use {}'.format(g,z['primary_use'].unique()),fonts
  plt.subplots_adjust(hspace=0.7,wspace=0.4)
```


building_id 1326 primary_use ['Office']
building_id 1347 primary_use ['Education']
building_id 1368 primary_use ['Entertainment/public assembly']
building_id 1393 primary_use ['Entertainment/public assembly']
building_id 1413 primary_use ['Lodging/residential']
building_id 1433 primary_use ['Education']
building_id 1327 primary_use ['Education']
building_id 1348 primary_use ['Education']
building_id 1369 primary_use ['Entertainment/public assembly']
building_id 1394 primary_use ['Entertainment/public assembly']
building_id 1414 primary_use ['Lodging/residential']
building_id 1434 primary_use ['Education']
building_id 1328 primary_use ['Education']
building_id 1350 primary_use ['Education']
building_id 1370 primary_use ['Education']
building_id 1395 primary_use ['Lodging/residential']
building_id 1415 primary_use ['Lodging/residential']
building_id 1435 primary_use ['Education']
building_id 1329 primary_use ['Technology/science']
building_id 1351 primary_use ['Office']
building_id 1371 primary_use ['Public services']
building_id 1396 primary_use ['Lodging/residential']
building_id 1416 primary_use ['Lodging/residential']
building_id 1436 primary_use ['Manufacturing/industrial']
building_id 1331 primary_use ['Education']
building_id 1352 primary_use ['Office']
building_id 1375 primary_use ['Office']
building_id 1398 primary_use ['Lodging/residential']
building_id 1417 primary_use ['Lodging/residential']
building_id 1437 primary_use ['Education']
building_id 1332 primary_use ['Education']
building_id 1353 primary_use ['Religious worship']
building_id 1376 primary_use ['Public services']
building_id 1399 primary_use ['Lodging/residential']
building_id 1418 primary_use ['Lodging/residential']
building_id 1438 primary_use ['Education']
building_id 1333 primary_use ['Office']
building_id 1355 primary_use ['Education']
building_id 1377 primary_use ['Education']
building_id 1400 primary_use ['Lodging/residential']
building_id 1419 primary_use ['Lodging/residential']
building_id 1439 primary_use ['Education']
building_id 1334 primary_use ['Office']
building_id 1356 primary_use ['Education']
building_id 1378 primary_use ['Education']
building_id 1401 primary_use ['Lodging/residential']
building_id 1420 primary_use ['Entertainment/public assembly']
building_id 1440 primary_use ['Lodging/residential']
building_id 1335 primary_use ['Public services']
building_id 1357 primary_use ['Office']
building_id 1379 primary_use ['Office']
building_id 1402 primary_use ['Lodging/residential']
building_id 1421 primary_use ['Education']
building_id 1441 primary_use ['Education']
building_id 1336 primary_use ['Office']
building_id 1358 primary_use ['Lodging/residential']
building_id 1380 primary_use ['Education']
building_id 1403 primary_use ['Lodging/residential']
building_id 1422 primary_use ['Lodging/residential']
building_id 1442 primary_use ['Public services']
building_id 1337 primary_use ['Office']
building_id 1359 primary_use ['Education']
building_id 1381 primary_use ['Office']
building_id 1404 primary_use ['Lodging/residential']
building_id 1423 primary_use ['Lodging/residential']
building_id 1443 primary_use ['Education']
building_id 1338 primary_use ['Education']
building_id 1360 primary_use ['Entertainment/public assembly']
building_id 1382 primary_use ['Public services']
building_id 1405 primary_use ['Lodging/residential']
building_id 1424 primary_use ['Lodging/residential']
building_id 1444 primary_use ['Entertainment/public assembly']
building_id 1339 primary_use ['Education']
building_id 1361 primary_use ['Education']
building_id 1383 primary_use ['Education']
building_id 1406 primary_use ['Lodging/residential']
building_id 1425 primary_use ['Utility']
building_id 1445 primary_use ['Education']
building_id 1340 primary_use ['Manufacturing/industrial']
building_id 1362 primary_use ['Education']
building_id 1384 primary_use ['Education']
building_id 1407 primary_use ['Lodging/residential']
building_id 1427 primary_use ['Education']
building_id 1446 primary_use ['Entertainment/public assembly']
building_id 1341 primary_use ['Education']
building_id 1363 primary_use ['Office']
building_id 1386 primary_use ['Entertainment/public assembly']
building_id 1408 primary_use ['Entertainment/public assembly']
building_id 1428 primary_use ['Education']
building_id 1447 primary_use ['Lodging/residential']
building_id 1342 primary_use ['Education']
building_id 1364 primary_use ['Utility']
building_id 1387 primary_use ['Office']
building_id 1409 primary_use ['Lodging/residential']
building_id 1429 primary_use ['Education']
building_id 1448 primary_use ['Office']



Important Observations

- We have to remove the constant zero meter readins from building 1359 1345 1446.
- For removing spikes we have to consider building 1383 1401 1414 1423.

FINAL CONCLUSION

1) So now for doing the overall analysis I have divided the data into 15 sites whch are present at different geographical location of North America abd Europe.This is basically done as the environmental conditions are a driving factor for the energy consumption.

2) Now site 0 site 2 site 3 site 4 site 5 site 6 site 7 site 8 site 9 site 10 site 11 site 13 site14 and site 15 weather timestamp is not in alignment with the local timestamp of the hourly meter readings.The air temperature peaks around 18:00 pm to 23:00 pm for all of the sites.

3) We can also observe that almost each and every site is having some missing values which needs to be imputed by proper method as null values cannot be used in the training process.Paricualrly site 0 is having very high percentage of null values for some of the features.

4) At site 0 we can observe that the meter readings from electricity are mostly zero or having some spikes before May 2020 which needs to be removed definitely.It is also having zero information for the floor count which definitely can be dropped as we have no way to impute it.

5) Now here we can see that each site consumes electricity but electricity is not the highest consumption of energy.Site 0->Chilledwater Site 6->Chilledwater Site 9->Chilledwater Site 10->Hotwater Site 13->Steam Site 14-> Steam SIte 15->Hotwater

6) We can also observe strong seasonal patterns for the chilledwater and hotwater usage.EvenS team shows higher consumption during the winter month as compared to the other months.If we see for electricty consumption we cannot see a specific pattern during the months.For some sites it shows more energy cinsumption for the summer months for some it shows the same for the winter monthss and for some it does not show any pattern at all it is kind of going up and down.the elctrcity consumption can depend on environmental factors building metadata such as square feet floor count.It can also depend upon the occupancy of the building.

7) Now for each of the site I have tried to analyze each and every building for outlier removal.This method is not efficient and time consuming but all the 4 readings have different nature and we cannot apply the same technique for the all 4 types.For ex->Consider if the electricity meter reading is showing zero reading in summer time for some days definitely this is an anomaly for the elctricity reading but not for hotwater reading as it might be happening due to the seasonal variation.One more thing I would like to add that for each building the rage of values it is taking is quite different so for removing the faulty readings we have to consider each and every building.

8) The reason for which I am doing the analysis for each and every building for every site is that the preprocessing part is super important for this particular case study.A significant boost in score can just be acheived by doing the processing in a proper way.One more important thing I would like to mention that I do not want my model to overfit by considering each and every reading(Faulty reading and Constant meter readings).