### In [1]:

```
# Importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

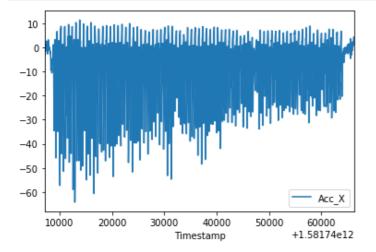
### In [2]:

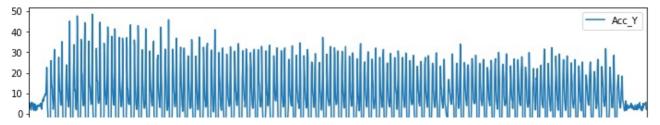
```
# Reading the CSV files
acc_jog = pd.read_csv("Acc_JOGGING.csv")
acc_run = pd.read_csv("Acc_RUNNING.csv")
acc_sit = pd.read_csv("Acc_SITTING.csv")
acc_sup = pd.read_csv("Acc_STAIRCLIMB.csv")
acc_sdn = pd.read_csv("Acc_STAIRDOWN.csv")
acc_wlk = pd.read_csv("Acc_WALKING.csv")

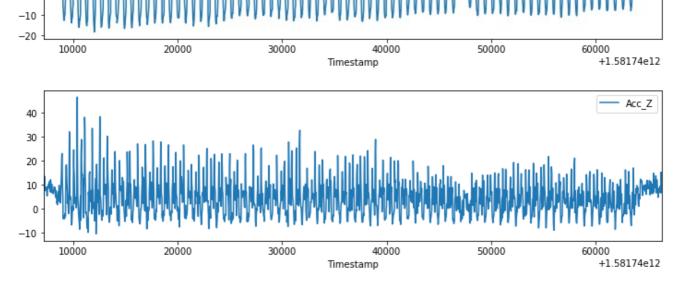
gyr_jog = pd.read_csv("Gyr_JOGGING.csv")
gyr_run = pd.read_csv("Gyr_RUNNING.csv")
gyr_sit = pd.read_csv("Gyr_SITTING.csv")
gyr_sup = pd.read_csv("Gyr_STAIRCLIMB.csv")
gyr_sdn = pd.read_csv("Gyr_STAIRDOWN.csv")
gyr_sdn = pd.read_csv("Gyr_STAIRDOWN.csv")
gyr_wlk = pd.read_csv("Gyr_WALKING.csv")
```

## In [3]:

```
# Plotting the time series data
acc_jog.plot(x='Timestamp',y='Acc_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_jog.plot(x='Timestamp',y='Acc_Y')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_jog.plot(x='Timestamp',y='Acc_Z')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
```

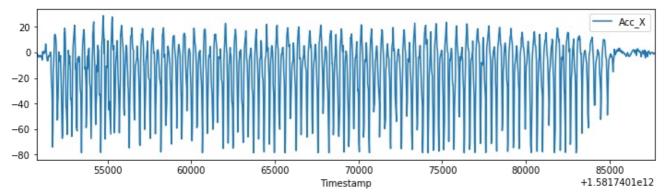


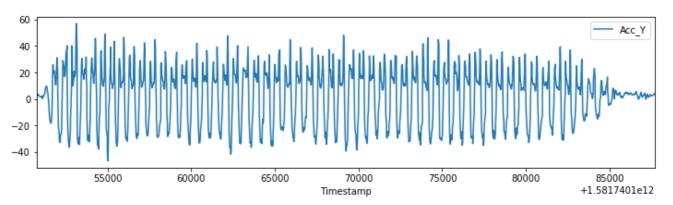


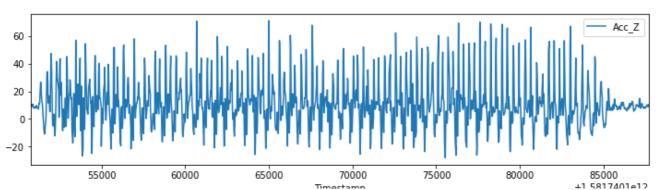


### In [4]:

```
# Plotting the time series data
acc_run.plot(x='Timestamp', y='Acc_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_run.plot(x='Timestamp', y='Acc_Y')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_run.plot(x='Timestamp', y='Acc_Z')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
```







11.501/701012 пппсэкаппр

### In [5]:

-7.5

540000

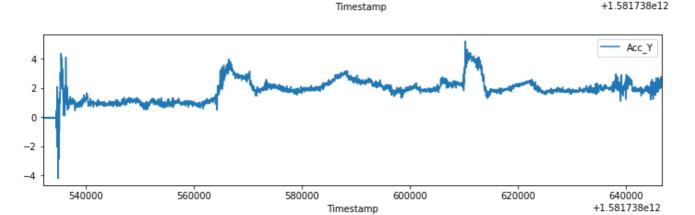
560000

```
# Plotting the time series data
acc_sit.plot(x='Timestamp',y='Acc_X')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()
acc_sit.plot(x='Timestamp',y='Acc_Y')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()
acc sit.plot(x='Timestamp',y='Acc Z')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()
                                                                                  Acc_X
 5.0
 2.5
 0.0
-2.5
-5.0
```

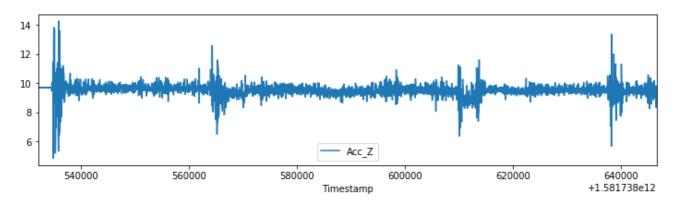
600,000

620000

640000 +1.581738e12



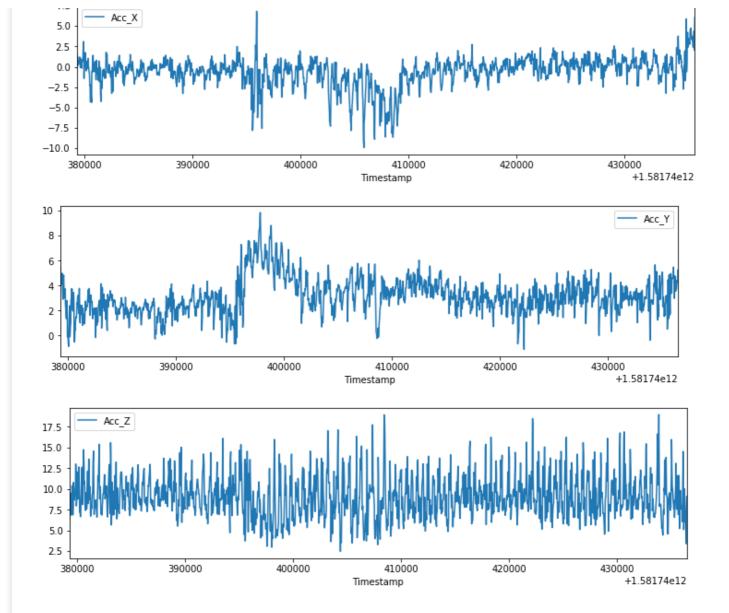
580000



## In [6]:

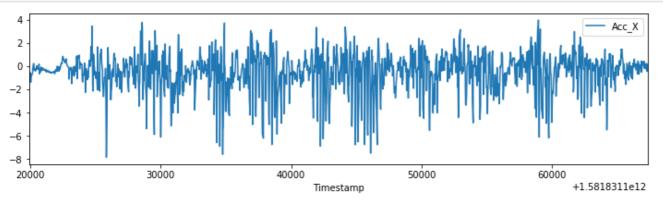
```
# Plotting the time series data
acc sup.plot(x='Timestamp',y='Acc X')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()
acc sup.plot(x='Timestamp',y='Acc Y')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()
acc_sup.plot(x='Timestamp',y='Acc_Z')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()
```

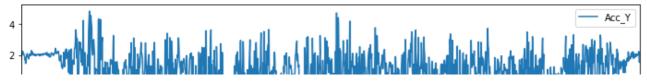
75 -

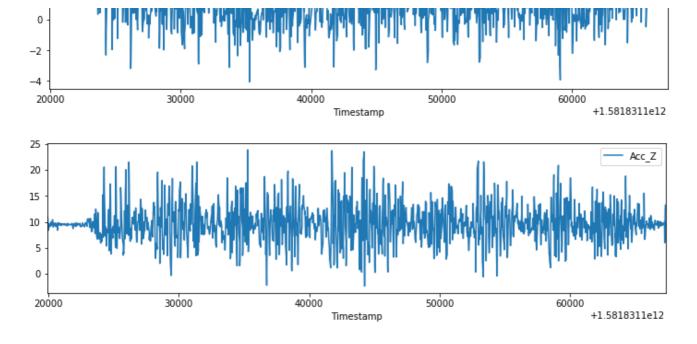


# In [7]:

```
# Plotting the time series data
acc_sdn.plot(x='Timestamp', y='Acc_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_sdn.plot(x='Timestamp', y='Acc_Y')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_sdn.plot(x='Timestamp', y='Acc_Z')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
```

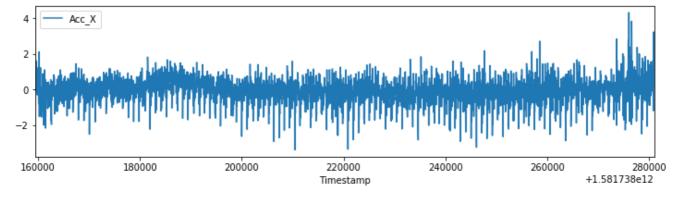


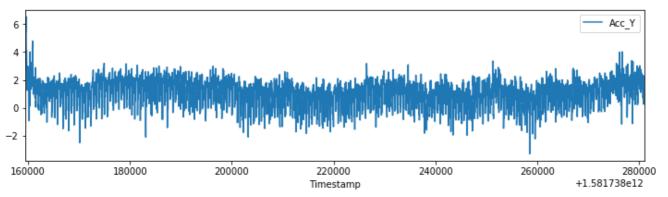


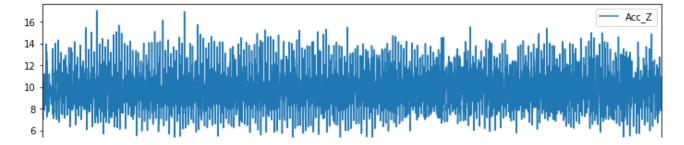


# In [8]:

```
# Plotting the time series data
acc_wlk.plot(x='Timestamp',y='Acc_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_wlk.plot(x='Timestamp',y='Acc_Y')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
acc_wlk.plot(x='Timestamp',y='Acc_Z')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
```







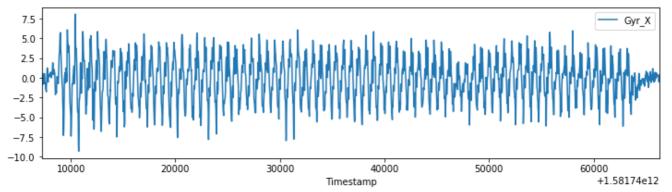
```
4 160000 180000 200000 220000 240000 260000 280000 Timestamp +1.581738e12
```

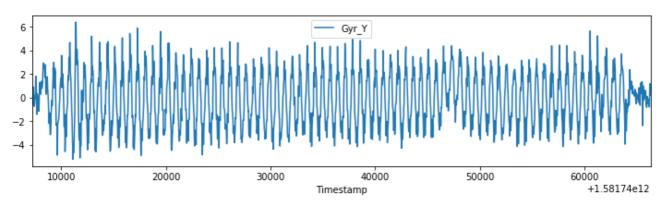
## In [9]:

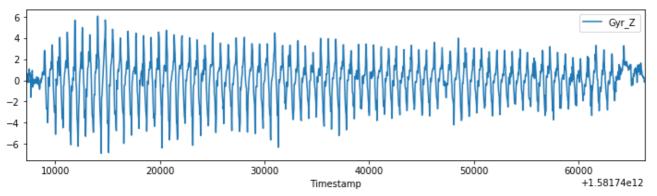
```
# Plotting the time series data
gyr_jog.plot(x='Timestamp', y='Gyr_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()

gyr_jog.plot(x='Timestamp', y='Gyr_Y')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()

gyr_jog.plot(x='Timestamp', y='Gyr_Z')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
```





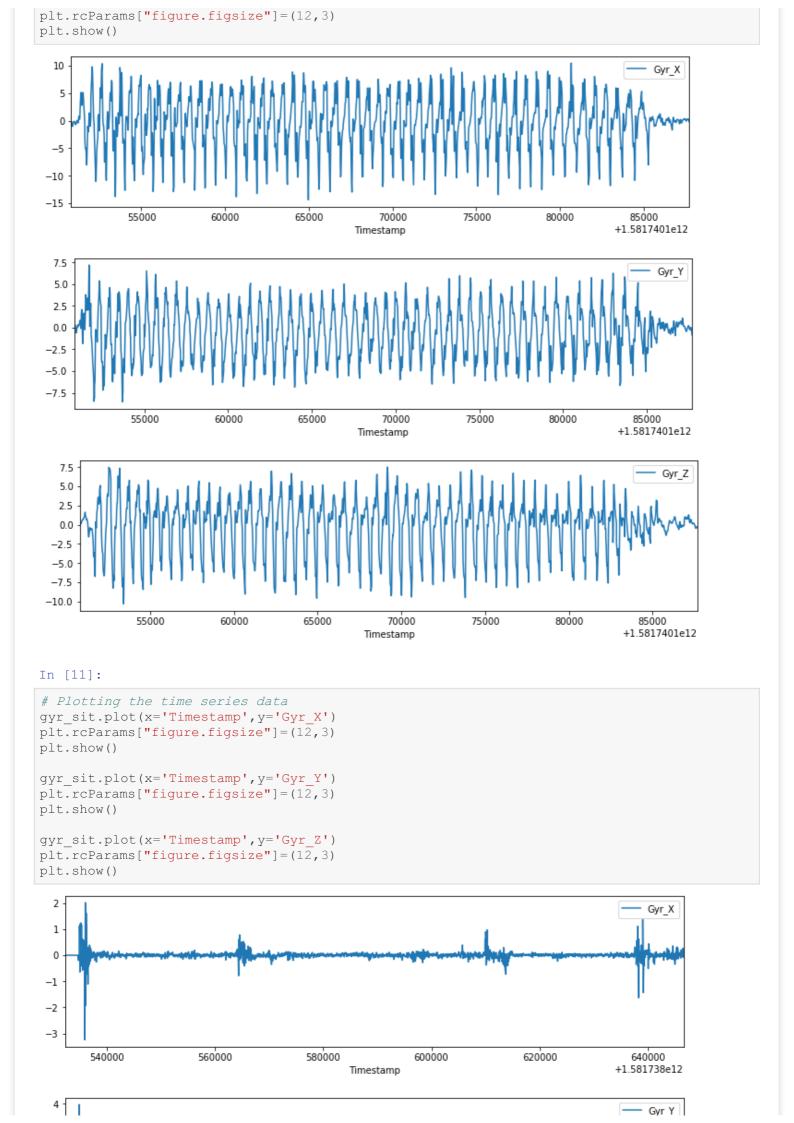


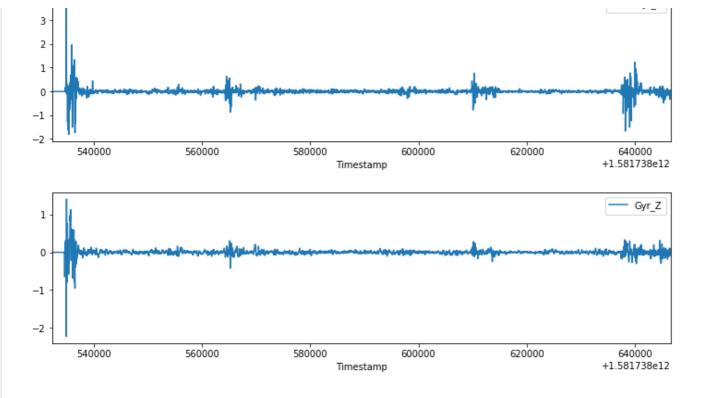
# In [10]:

```
# Plotting the time series data
gyr_run.plot(x='Timestamp', y='Gyr_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()

gyr_run.plot(x='Timestamp', y='Gyr_Y')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()

gyr_run.plot(x='Timestamp', y='Gyr_Z')
```



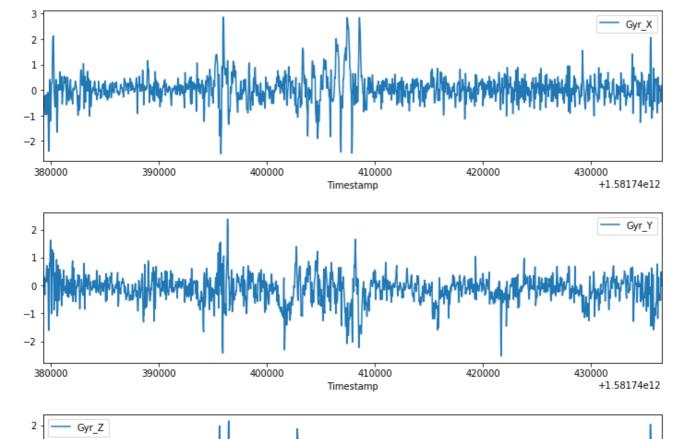


# In [12]:

```
# Plotting the time series data
gyr_sup.plot(x='Timestamp', y='Gyr_X')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()

gyr_sup.plot(x='Timestamp', y='Gyr_Y')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()

gyr_sup.plot(x='Timestamp', y='Gyr_Z')
plt.rcParams["figure.figsize"] = (12,3)
plt.show()
```



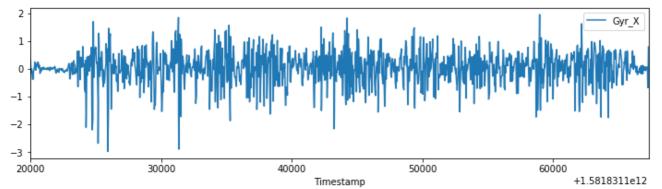
```
-1 -2 -380000 390000 400000 410000 420000 430000 +1.58174e12
```

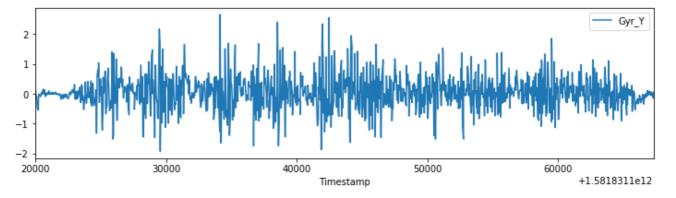
## In [13]:

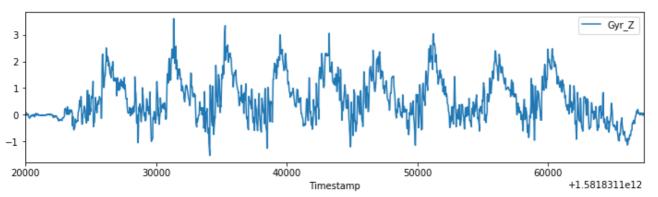
```
# Plotting the time series data
gyr_sdn.plot(x='Timestamp', y='Gyr_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()

gyr_sdn.plot(x='Timestamp', y='Gyr_Y')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()

gyr_sdn.plot(x='Timestamp', y='Gyr_Z')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
```







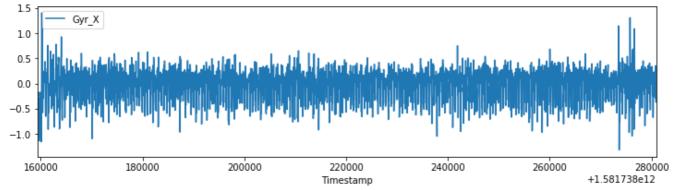
## In [14]:

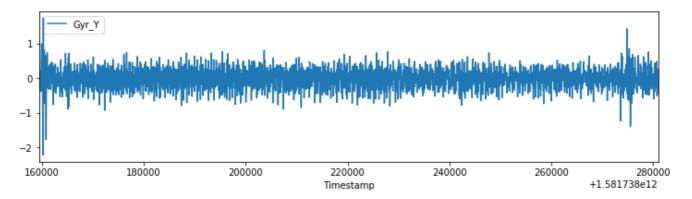
```
# Plotting the time series data
gyr_wlk.plot(x='Timestamp', y='Gyr_X')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()

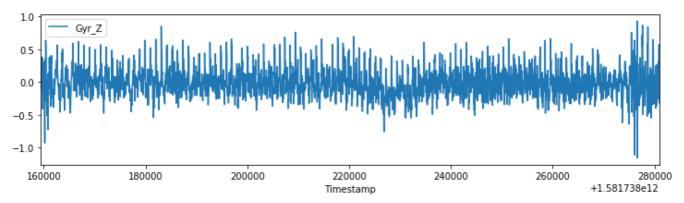
gyr_wlk.plot(x='Timestamp', y='Gyr_Y')
plt.rcParams["figure.figsize"]=(12,3)
```

```
plt.show()

gyr_wlk.plot(x='Timestamp', y='Gyr_Z')
plt.rcParams["figure.figsize"]=(12,3)
plt.show()
```







## In [15]:

```
# Removing certain samples from start and end of sensor readings, to avoid the error asso
ciated

# Jogging
acc_jog[(acc_jog.Acc_Y > 9.98)].head(1)
acc_jog = acc_jog.iloc[81:]
gyr_jog = gyr_jog.iloc[81:]
acc_jog[(acc_jog.Acc_Y > 9.98)].tail(1)
acc_jog = acc_jog.iloc[:-111]
gyr_jog = gyr_jog.iloc[:-111]
acc_jog.count()
```

#### Out[15]:

```
Timestamp 2789
Acc_X 2789
Acc_Y 2789
Acc_Z 2789
Activity 2789
dtype: int64
```

## In [16]:

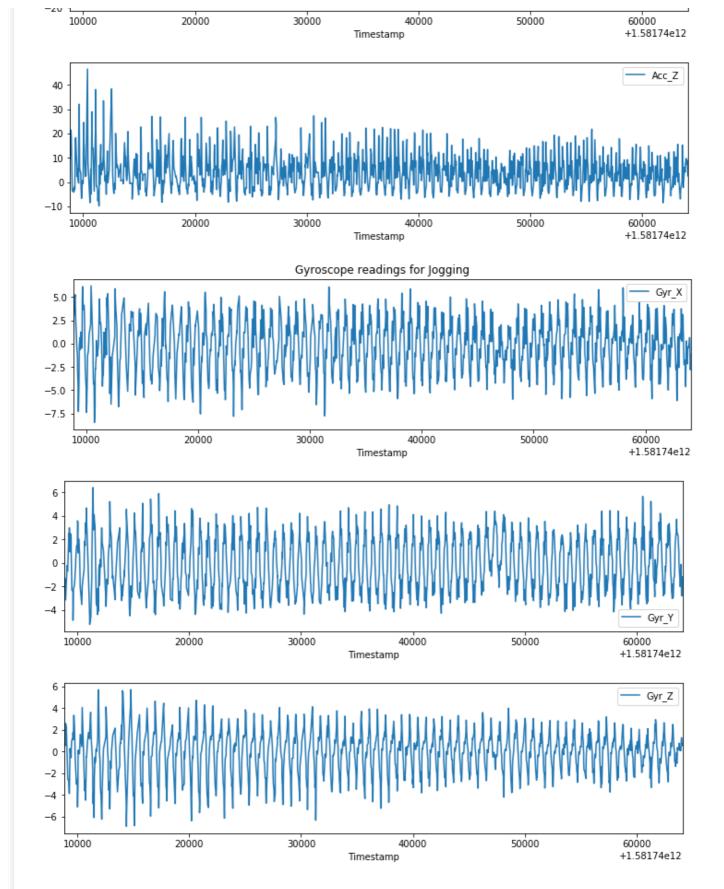
```
# Running
acc run[(acc run.Acc X < -20)].head(1)
acc run = acc run.iloc[44:]
gyr_run = gyr_run.iloc[44:]
acc run[(acc run.Acc X < -20)].tail(1)
acc run = acc run.iloc[:-120]
gyr run = gyr run.iloc[:-120]
acc run.count()
Out[16]:
Timestamp
            1697
Acc X
            1697
Acc Y
            1697
\operatorname{Acc}\ Z
             1697
          1697
Activity
dtype: int64
In [17]:
# Sitting
acc_sit[(acc_sit.Acc_X > 5)].head(1)
acc_sit = acc_sit.iloc[240:]
gyr sit = gyr sit.iloc[240:]
acc sit[(acc sit.Acc X > 2.5)].tail(1)
acc sit = acc sit.iloc[:-134]
gyr_sit = gyr_sit.iloc[:-134]
acc sit.count()
Out[17]:
           5400
Timestamp
             5400
Acc X
Acc_Y
             5400
Acc Z
            5400
           5400
Activity
dtype: int64
In [18]:
# Stair down
acc_sdn[(acc_sdn.Acc_X > 1)].head(1)
acc_sdn = acc_sdn.iloc[247:]
gyr_sdn = gyr_sdn.iloc[247:]
acc sdn[(acc sdn.Acc X > 1)].tail(1)
acc sdn = acc sdn.iloc[:-150]
gyr sdn = gyr sdn.iloc[:-150]
acc_sdn.count()
Out[18]:
Timestamp
             2000
Acc X
             2000
Acc Y
             2000
Acc Z
             2000
Activity
             2000
dtype: int64
```

# In [19]:

```
# Walking
acc_wlk = acc_wlk.iloc[200:]
gyr_wlk = gyr_wlk.iloc[200:]
acc_wlk = acc_wlk.iloc[:-327]
gyr_wlk = gyr_wlk.iloc[:-327]
```

```
acc wlk.count()
Out[19]:
Timestamp
              5600
              5600
Acc X
              5600
Acc Y
Acc Z
              5600
Activity
              5600
dtype: int64
In [20]:
# Merging the two frames consisting of Accelerometer and Gyroscope readings
sen_jog = pd.merge(acc_jog,gyr_jog, on=['Timestamp','Activity'])
sen run = pd.merge(acc run,gyr run, on=['Timestamp','Activity'])
sen_wlk = pd.merge(acc_wlk,gyr_wlk, on=['Timestamp','Activity'])
sen sup = pd.merge(acc sup,gyr sup, on=['Timestamp','Activity'])
sen sdn = pd.merge(acc sdn, gyr sdn, on=['Timestamp','Activity'])
sen sit = pd.merge(acc sit, gyr sit, on=['Timestamp','Activity'])
In [21]:
# Before forming the features, declaring an empty data frame to append features
features = ['meanAccX', 'meanAccY', 'meanAccZ', 'meanGyrX', 'meanGyrY', 'meanGyrZ',
             'maxAccX', 'maxAccY', 'maxAccZ', 'maxGyrX', 'maxGyrY', 'maxGyrZ',
             'minAccX', 'minAccY', 'minAccZ', 'minGyrX', 'minGyrY', 'minGyrZ',
             'sdAccX', 'sdAccY', 'sdAccZ', 'sdGyrX', 'sdGyrY', 'sdGyrZ', 'madAccX', 'madAccY', 'maxAccZ', 'madGyrX', 'madGyrY', 'maxGyrZ',
             'Activity']
dataset = pd.DataFrame(columns=features)
In [22]:
sen jog.plot(x='Timestamp',y='Acc X', title='Accelerometer readings for Jogging')
plt.show()
sen jog.plot(x='Timestamp',y='Acc Y')
plt.show()
sen jog.plot(x='Timestamp',y='Acc Z')
sen jog.plot(x='Timestamp',y='Gyr X', title='Gyroscope readings for Jogging')
plt.show()
sen jog.plot(x='Timestamp',y='Gyr Y')
plt.show()
sen jog.plot(x='Timestamp',y='Gyr Z')
plt.show()
                                 Accelerometer readings for Jogging
 -20
 -60
                                                                                    Acc X
    10000
                   20000
                                   30000
                                                  40000
                                                                 50000
                                                                                 60000
                                                                                +1.58174e12
                                           Timestamp
  50
                                                                                    Acc Y
  40
  30
  20
  10
```

-10

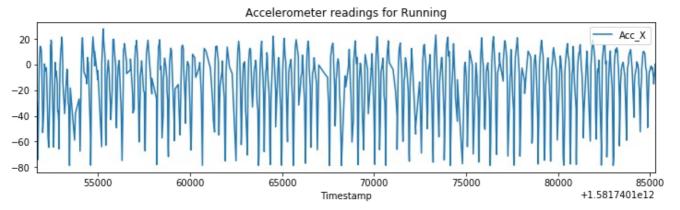


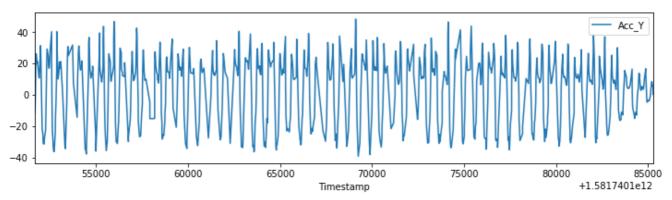
# In [23]:

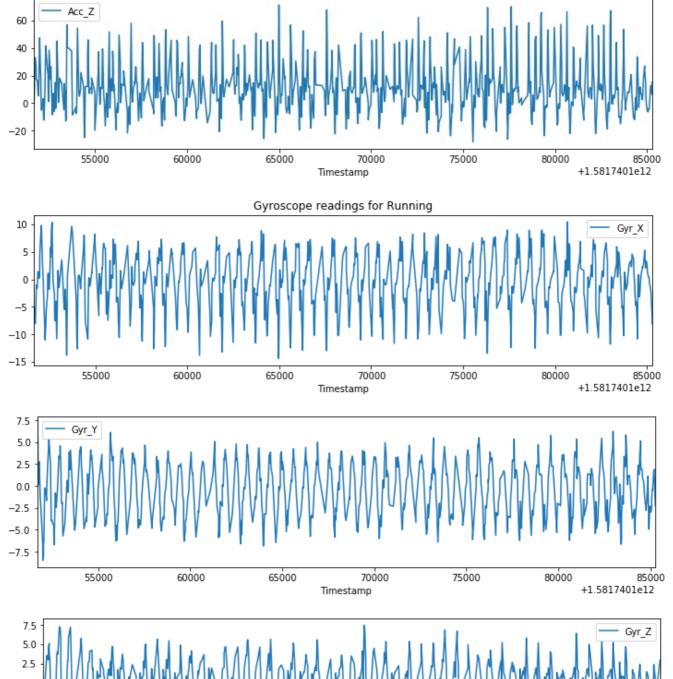
```
'meanGyrX':temp.Gyr_X.mean(),
      'meanGyrY':temp.Gyr_Y.mean(),
      'meanGyrZ':temp.Gyr Z.mean(),
      'maxAccX' :temp.Acc_X.max(),
      'maxAccY' :temp.Acc Y.max(),
      'maxAccZ' :temp.Acc Z.max(),
      'maxGyrX' :temp.Gyr X.max(),
      'maxGyrY' :temp.Gyr Y.max(),
      'maxGyrZ' :temp.Gyr Z.max(),
      'minAccX' :temp.Acc X.min(),
      'minAccY' :temp.Acc Y.min(),
      'minAccZ' :temp.Acc Z.min(),
      'minGyrX' :temp.Gyr X.min(),
      'minGyrY' :temp.Gyr Y.min(),
      'minGyrZ' :temp.Gyr_Z.min(),
'sdAccX' :temp.Acc_X.std(),
     'sdAccY' :temp.Acc_Y.std(),
     'sdAccZ' :temp.Acc_Z.std(),
     'sdGyrX' :temp.Gyr_X.std(),
'sdGyrY' :temp.Gyr_Y.std(),
'sdGyrZ' :temp.Gyr_Z.std(),
      'madAccX' :temp.Acc_X.mad(),
      'madAccY' :temp.Acc_Y.mad(),
      'maxAccZ' :temp.Acc Z.mad(),
      'madGyrX' :temp.Gyr X.mad(),
      'madGyrY' :temp.Gyr Y.mad(),
      'maxGyrZ' :temp.Gyr Z.mad(),
      'Activity':temp.Activity[i]}
dataset = dataset.append(f,ignore index=True)
```

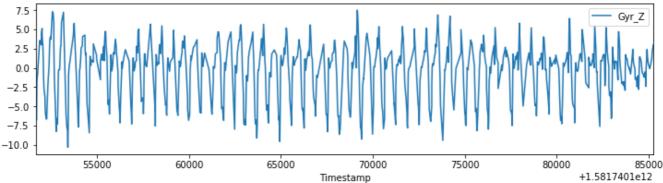
## In [24]:

```
sen_run.plot(x='Timestamp', y='Acc_X', title='Accelerometer readings for Running')
plt.show()
sen_run.plot(x='Timestamp', y='Acc_Y')
plt.show()
sen_run.plot(x='Timestamp', y='Acc_Z')
plt.show()
sen_run.plot(x='Timestamp', y='Gyr_X', title='Gyroscope readings for Running')
plt.show()
sen_run.plot(x='Timestamp', y='Gyr_Y')
plt.show()
sen_run.plot(x='Timestamp', y='Gyr_Z')
plt.show()
```









## In [25]:

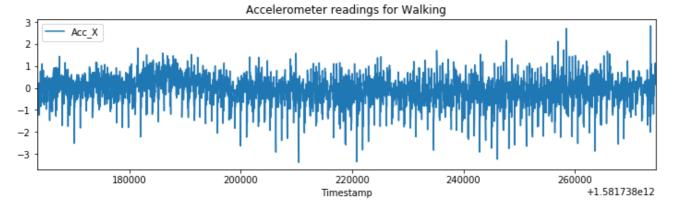
```
# For running, it is observed that, a cycle of activity repeats at around 24 samples
# Window of 24 is cut and parameters are computed.
sen_run = sen_run.iloc[:-11]
sen_run.count()

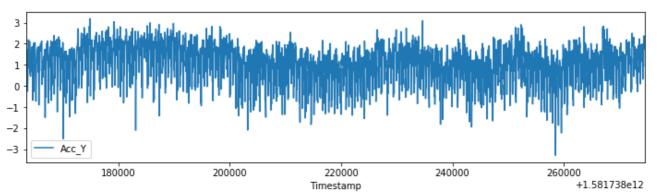
for i in range(0, sen_run.Activity.count()-1,24):
    temp = sen_run.iloc[i:i+24]
    f = {'meanAccX':temp.Acc_X.mean(),
        'meanAccY':temp.Acc_Y.mean(),
        'meanAccZ':temp.Acc_Z.mean(),
        'meanGyrX':temp.Gyr_X.mean(),
        'meanGyrY':temp.Gyr_Y.mean(),
        'meanGyrY':temp.Gyr_Y.mean(),
        'maxAccX':temp.Acc_X.max(),
        'maxAccX':temp.Acc_X.max(),
```

```
'maxAccZ' :temp.Acc_Z.max(),
     'maxGyrX' :temp.Gyr_X.max(),
     'maxGyrY' :temp.Gyr_Y.max(),
     'maxGyrZ' :temp.Gyr_Z.max(),
     'minAccX' :temp.Acc X.min(),
     'minAccY' :temp.Acc Y.min(),
     'minAccZ' :temp.Acc Z.min(),
     'minGyrX' :temp.Gyr X.min(),
     'minGyrY' :temp.Gyr Y.min(),
     'minGyrZ' :temp.Gyr Z.min(),
     'sdAccX' :temp.Acc X.std(),
     'sdAccY' :temp.Acc_Y.std(),
     'sdAccZ' :temp.Acc_Z.std(),
     'sdGyrX' :temp.Gyr_X.std(),
     'sdGyrY' :temp.Gyr_Y.std(),
     'sdGyrZ' :temp.Gyr_Z.std(),
     'madAccX' :temp.Acc_X.mad(),
     'madAccY' :temp.Acc_Y.mad(),
     'maxAccZ' :temp.Acc Z.mad(),
     'madGyrX' :temp.Gyr_X.mad(),
     'madGyrY' :temp.Gyr_Y.mad(),
     'maxGyrZ' :temp.Gyr_Z.mad(),
     'Activity':temp.Activity[i]}
dataset = dataset.append(f,ignore index=True)
```

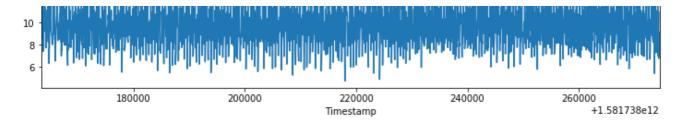
#### In [26]:

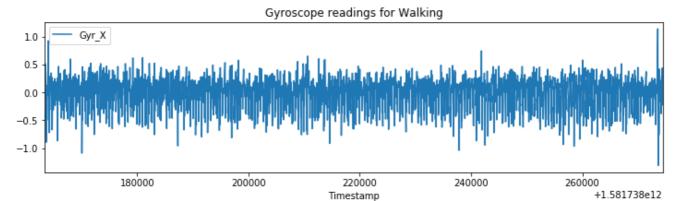
```
sen_wlk.plot(x='Timestamp',y='Acc_X', title='Accelerometer readings for Walking')
plt.show()
sen_wlk.plot(x='Timestamp',y='Acc_Z')
plt.show()
sen_wlk.plot(x='Timestamp',y='Gyr_X', title='Gyroscope readings for Walking')
plt.show()
sen_wlk.plot(x='Timestamp',y='Gyr_Y')
plt.show()
sen_wlk.plot(x='Timestamp',y='Gyr_Y')
plt.show()
sen_wlk.plot(x='Timestamp',y='Gyr_Z')
plt.show()
```

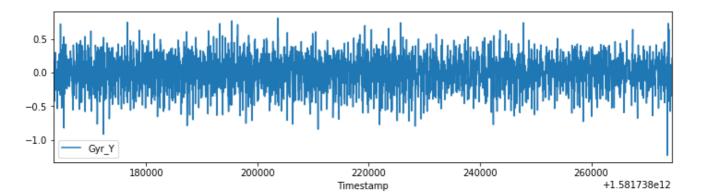


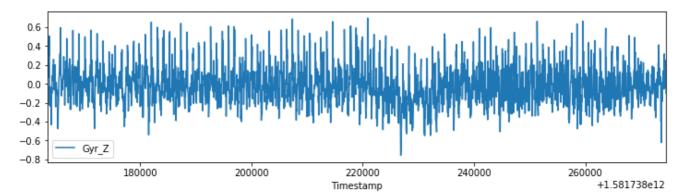












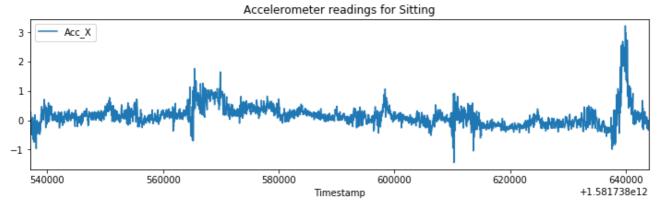
## In [27]:

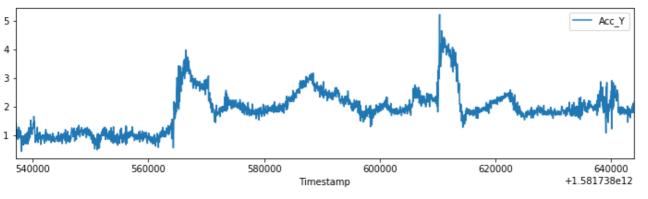
```
# For walking, it is observed that, a cycle of activity repeats at around 37 samples
# Window of 37 is cut and parameters are computed.
sen wlk = sen wlk.iloc[:-35]
sen_wlk.count()
for i in range(0, sen wlk.Activity.count()-1,37):
    temp = sen wlk.iloc[i:i+37]
    f = { 'meanAccX':temp.Acc X.mean(),
         'meanAccY':temp.Acc Y.mean(),
         'meanGyrZ':temp.Gyr Z.mean(),
         'maxAccX' :temp.Acc_X.max(),
         'maxAccY' :temp.Acc_Y.max(),
         'maxAccZ'
                  :temp.Acc_Z.max(),
         'maxGyrX'
                   :temp.Gyr X.max(),
         'maxGyrY' :temp.Gyr Y.max(),
         'maxGyrZ'
                   :temp.Gyr_Z.max(),
         'minAccX' :temp.Acc X.min(),
```

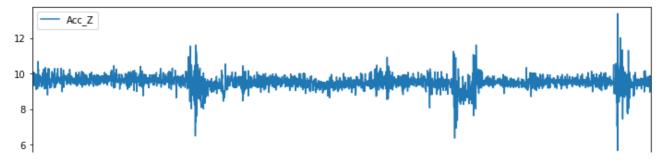
```
'minAccY' :temp.Acc_Y.min(),
     'minAccZ' :temp.Acc_Z.min(),
     'minGyrX' :temp.Gyr_X.min(),
     'minGyrY' :temp.Gyr_Y.min(),
     'minGyrZ' :temp.Gyr Z.min(),
     'sdAccX' :temp.Acc X.std(),
     'sdAccY' :temp.Acc Y.std(),
     'sdAccZ' :temp.Acc Z.std(),
     'sdGyrX' :temp.Gyr X.std(),
     'sdGyrY' :temp.Gyr Y.std(),
     'sdGyrZ' :temp.Gyr Z.std(),
     'madAccX' :temp.Acc X.mad(),
     'madAccY' :temp.Acc_Y.mad(),
     'maxAccZ' :temp.Acc Z.mad(),
     'madGyrX' :temp.Gyr_X.mad(),
     'madGyrY' :temp.Gyr_Y.mad(),
     'maxGyrZ' :temp.Gyr_Z.mad(),
     'Activity':temp.Activity[i]}
dataset = dataset.append(f,ignore index=True)
```

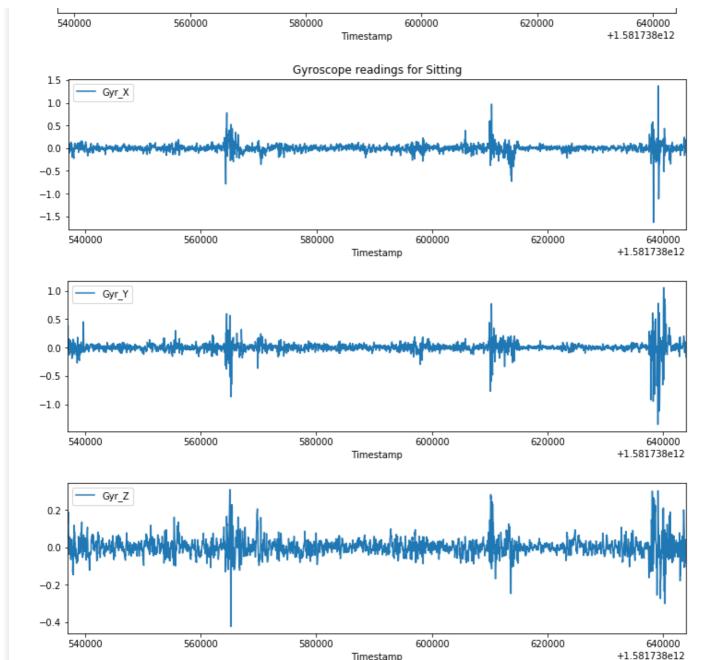
#### In [28]:

```
sen_sit.plot(x='Timestamp', y='Acc_X', title='Accelerometer readings for Sitting')
plt.show()
sen_sit.plot(x='Timestamp', y='Acc_Z')
plt.show()
sen_sit.plot(x='Timestamp', y='Acc_Z')
plt.show()
sen_sit.plot(x='Timestamp', y='Gyr_X', title='Gyroscope readings for Sitting')
plt.show()
sen_sit.plot(x='Timestamp', y='Gyr_Y')
plt.show()
sen_sit.plot(x='Timestamp', y='Gyr_Z')
plt.show()
```









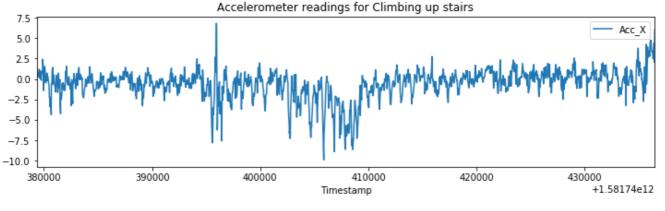
# In [29]:

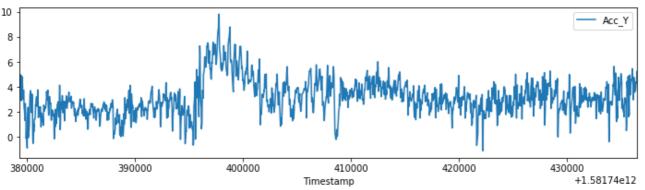
```
# For sitting, it is observed that, a cycle of activity repeats at around 30 samples
# Window of 30 is cut and parameters are computed.
sen sit = sen sit.iloc[:-26]
sen sit.count()
for i in range(0, sen sit.Activity.count()-1,30):
    temp = sen sit.iloc[i:i+30]
    f = { 'meanAccX':temp.Acc_X.mean(),
         'meanAccY':temp.Acc_Y.mean(),
         'meanAccZ':temp.Acc_Z.mean(),
         'meanGyrX':temp.Gyr_X.mean(),
         'meanGyrY':temp.Gyr Y.mean(),
         'meanGyrZ':temp.Gyr Z.mean(),
         'maxAccX' :temp.Acc X.max(),
         'maxAccY' :temp.Acc Y.max(),
         'maxAccZ' :temp.Acc Z.max(),
         'maxGyrX' :temp.Gyr X.max(),
         'maxGyrY' :temp.Gyr Y.max(),
         'maxGyrZ' :temp.Gyr_Z.max(),
         'minAccX' :temp.Acc X.min(),
         'minAccY' :temp.Acc_Y.min(),
         'minAccZ' :temp.Acc_Z.min(),
         'minGyrX' :temp.Gyr X.min(),
         'minGyrY' :temp.Gyr_Y.min(),
         'minGyrZ' :temp.Gyr Z.min(),
```

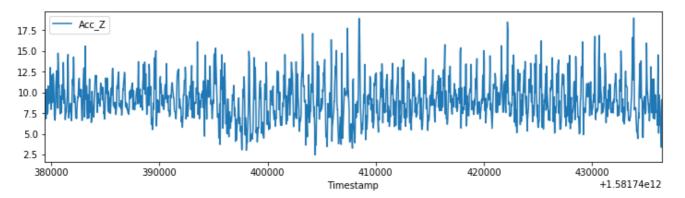
```
'sdAccX'
               :temp.Acc_X.std(),
               :temp.Acc_Y.std(),
     'sdAccY'
               :temp.Acc Z.std(),
     'sdGyrX'
               :temp.Gyr X.std(),
               :temp.Gyr Y.std(),
               :temp.Gyr_Z.std(),
     'madAccX' :temp.Acc X.mad(),
     'madAccY' :temp.Acc Y.mad(),
     'maxAccZ' :temp.Acc Z.mad(),
     'madGyrX' :temp.Gyr X.mad(),
     'madGyrY' :temp.Gyr Y.mad(),
     'maxGyrZ' :temp.Gyr Z.mad(),
     'Activity':temp.Activity[i]}
dataset = dataset.append(f,ignore index=True)
```

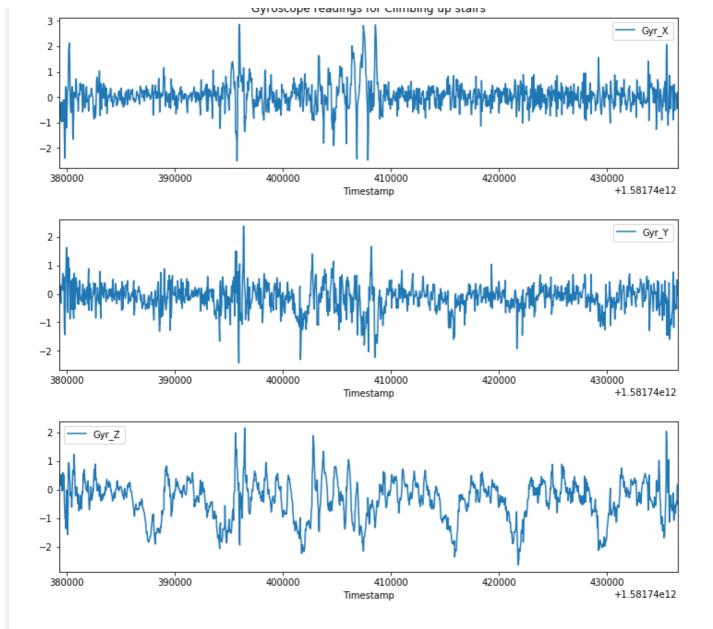
## In [30]:

```
sen_sup.plot(x='Timestamp',y='Acc_X', title='Accelerometer readings for Climbing up stair
s')
plt.show()
sen_sup.plot(x='Timestamp',y='Acc_Y')
plt.show()
sen_sup.plot(x='Timestamp',y='Acc_Z')
plt.show()
sen_sup.plot(x='Timestamp',y='Gyr_X', title='Gyroscope readings for Climbing up stairs')
plt.show()
sen_sup.plot(x='Timestamp',y='Gyr_Y')
plt.show()
sen_sup.plot(x='Timestamp',y='Gyr_Z')
plt.show()
```









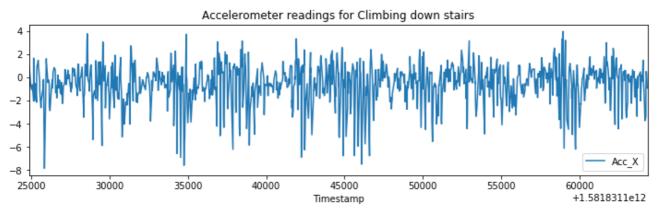
# In [31]:

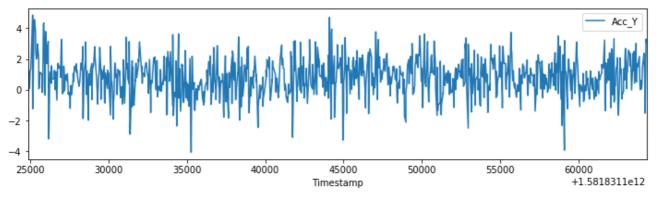
```
# For climbing up stairs, it is observed that, a cycle of activity repeats at around 45 s
amples
# Window of 45 is cut and parameters are computed.
sen sup = sen sup.iloc[:-24]
sen sup.count()
for i in range(0, sen sup.Activity.count()-1,45):
    temp = sen sup.iloc[i:i+45]
    f = { 'meanAccX':temp.Acc_X.mean(),
         'meanAccY':temp.Acc_Y.mean(),
         'meanAccZ':temp.Acc Z.mean(),
         'meanGyrX':temp.Gyr_X.mean(),
         'meanGyrY':temp.Gyr Y.mean(),
         'meanGyrZ':temp.Gyr Z.mean(),
         'maxAccX' :temp.Acc_X.max(),
         'maxAccY' :temp.Acc Y.max(),
         'maxAccZ' :temp.Acc Z.max(),
         'maxGyrX' :temp.Gyr X.max(),
         'maxGyrY' :temp.Gyr Y.max(),
         'maxGyrZ' :temp.Gyr Z.max(),
         'minAccX' :temp.Acc X.min(),
         'minAccY' :temp.Acc Y.min(),
         'minAccZ' :temp.Acc Z.min(),
         'minGyrX' :temp.Gyr X.min(),
                   :temp.Gyr Y.min(),
         'minGyrY'
                   :temp.Gyr Z.min(),
         'minGyrZ'
                   :temp.Acc X.std(),
         'sdAccX'
         'sdAccY'
                   :temp.Acc_Y.std(),
         'sdAccZ'
                   :temp.Acc Z.std(),
```

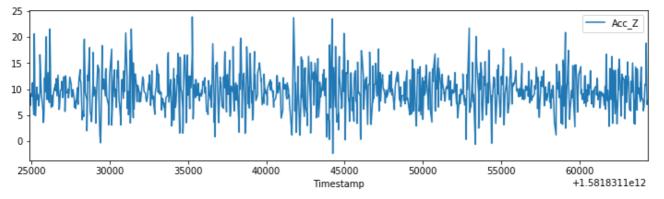
```
'sdGyrX' :temp.Gyr_X.std(),
'sdGyrZ' :temp.Gyr_Z.std(),
'madAccX' :temp.Acc_X.mad(),
'madAccY' :temp.Acc_Y.mad(),
'maxAccZ' :temp.Acc_Z.mad(),
'madGyrX' :temp.Gyr_X.mad(),
'madGyrY' :temp.Gyr_Y.mad(),
'macGyrZ' :temp.Gyr_Z.mad(),
'macGyrZ' :temp.Activity[i]}
dataset = dataset.append(f,ignore_index=True)
```

## In [32]:

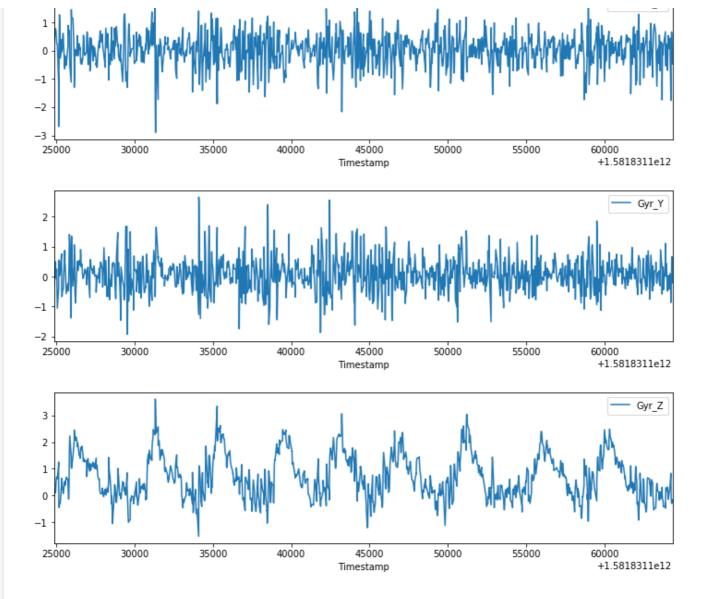
```
sen_sdn.plot(x='Timestamp',y='Acc_X', title='Accelerometer readings for Climbing down sta
irs')
plt.show()
sen_sdn.plot(x='Timestamp',y='Acc_Y')
plt.show()
sen_sdn.plot(x='Timestamp',y='Acc_Z')
plt.show()
sen_sdn.plot(x='Timestamp',y='Gyr_X', title='Gyroscope readings for Climbing down stairs')
plt.show()
sen_sdn.plot(x='Timestamp',y='Gyr_Y')
plt.show()
sen_sdn.plot(x='Timestamp',y='Gyr_Z')
plt.show()
```







Gyroscope readings for Climbing down stairs



### In [33]:

```
# For climbing down stairs, it is observed that, a cycle of activity repeats at around 45
samples
# Window of 45 is cut and parameters are computed.
sen_sdn = sen_sdn.iloc[:-13]
sen sdn.count()
for i in range(0, sen sdn.Activity.count()-1,45):
    temp = sen sdn.iloc[i:i+45]
    f = { 'meanAccX':temp.Acc_X.mean(),
         'meanAccY':temp.Acc Y.mean(),
         'meanAccZ':temp.Acc Z.mean(),
         'meanGyrX':temp.Gyr X.mean(),
         'meanGyrY':temp.Gyr Y.mean(),
         'meanGyrZ':temp.Gyr Z.mean(),
         'maxAccX' :temp.Acc X.max(),
         'maxAccY' :temp.Acc Y.max(),
         'maxAccZ' :temp.Acc Z.max(),
         'maxGyrX' :temp.Gyr X.max(),
         'maxGyrY' :temp.Gyr_Y.max(),
         'maxGyrZ' :temp.Gyr Z.max(),
         'minAccX' :temp.Acc X.min(),
         'minAccY' :temp.Acc Y.min(),
         'minAccZ' :temp.Acc Z.min(),
         'minGyrX'
                   :temp.Gyr X.min(),
         'minGyrY'
                   :temp.Gyr_Y.min(),
         'minGyrZ'
                   :temp.Gyr Z.min(),
         'sdAccX'
                   :temp.Acc X.std(),
                   :temp.Acc_Y.std(),
         'sdAccY'
         'sdAccZ'
                   :temp.Acc_Z.std(),
         'sdGyrX'
                   :temp.Gyr_X.std(),
         'sdGyrY'
                   :temp.Gyr Y.std(),
```

```
'sdGyrZ' :temp.Gyr_Z.std(),
    'madAccX' :temp.Acc_X.mad(),
    'madAccY' :temp.Acc_Y.mad(),
    'maxAccZ' :temp.Acc_Z.mad(),
    'madGyrX' :temp.Gyr_X.mad(),
    'madGyrY' :temp.Gyr_Y.mad(),
    'maxGyrZ' :temp.Gyr_Z.mad(),
    'Activity':temp.Activity[i]}
dataset = dataset.append(f,ignore_index=True)
```

#### In [34]:

```
plt.figure(figsize=(12,12))
ax = sns.heatmap(
    dataset.corr(),
    annot=True,
    cbar=False,
    cmap='Blues'
)
```

```
meanAccX - 1 -0.420.17 0.350.0870.32 0.810.87 -0.9 -0.9-0.860.95 0.930.880.86 0.930.880.91 -0.940.910.890.95 -0.9-0.940.950.92-0.9-0.950.890.95
meanAccY +0.42 1 -0.380.110.040.03 0.380.590.28 0.43 0.490.37 -0.4-0.250.37-0.380.470.38 0.42 0.43 0.27 0.410.53 0.38 0.43 0.4 0.28 0.410.540.37
meanAccZ -0.17-0.38 1 0.070.069.09-0.0650.450.0460.220.440.23 0.20.0870.29 0.160.280.22-0.17-0.30.03-0.210.370.230.160.280.0460.220.380.23
meanGyrX -0.35-0.110.07 1 -0.760.67 0.180.180.270.0950.3-0.310.27 0.2 0.22 0.390.110.33-0.270.190.260.260.180.270.27-0.2-0.270.260.170.31
meanGyry -0.0870.040.0690.76 1 -0.420.270.220.160.310.02-0.150.170.230.170.0470.330.12-0.180.230.17-0.17-0.2-0.180.170.220.160.170.210.15
meanGyrZ -0.320.036.0910.67-0.42 1 -0.140.180.230.130.210.290.250.180.190.310.150.42-0.24-0.2-0.230.240.170.260.240.210.230.240.160.29
 maxAccX +0.810.380.0650.180.270.14 1 0.84 0.9 0.890.790.87 0.920.93 -0.9-0.880.890.87 0.940.910.910.910.860.89 0.93 0.9 0.9 0.9 0.850.87
 maxAccY +0.870.59-0.450.180.220.180.84 1 0.830.910.92 0.9 +0.920.86 -0.9-0.860.920.86 0.920.960.83 0.910.960.91 0.910.950.830.91 0.96 0.9
 maxAccZ -0.9 0.280.0460.270.160.23 0.9 0.83 1 0.920.84 0.9 0.950.940.940.94-0.9-0.88 0.950.91 1 0.950.88 0.9 0.950.92 1 0.940.87 0.9
 maxGyrX -0.9 0.43 0.220.09 0.31 0.13 0.89 0.91 0.92 1 0.87 0.91 0.94 0.92 0.93 -0.9 -0.96 0.88 0.95 0.95 0.92 0.96 0.94 0.92 0.95 0.95 0.92 0.96 0.93 0.91
 maxGyrY +0.860.49-0.44-0.30.02-0.210.790.92-0.840.87 1 0.88-0.890.83-0.9-0.880.890.86 0.89 0.9 0.83 0.9 0.960.89 0.880.890.84 0.9 0.950.88
 maxGyrZ +0.950.37 0.230.310.150.250.87 0.9 0.9 0.910.88 1 +0.950.92 -0.9-0.930.910.960.940.940.89 0.950.92 1 0.940.94 0.9 0.96 0.91 1
 minAccX -0.93 -0.4 0.2 0.270.170.25 0.920.920.950.940.890.95 1 0.950.94 0.960.930.94 0.990.970.950.980.940.960.980.970.950.970.950.930.95
 minAccY -0.88 0.25 0.87 0.2 0.23 0.18 0.930.860.940.920.830.92 0.95 1 0.93 0.92 0.91 0.9 -0.950.960.950.950.890.930.950.960.940.940.880.92
 minAccZ -0.86-0.370.29 0.220.170.19 -0.9 -0.9-0.940.93-0.9 -0.9 0.940.93 1 0.910.930.88-0.930.950.940.940.940.940.910.920.950.940.940.92-0.9
  minGyrX -0.93 0.380.16 0.390 0470.31 0.880.860.94-0.9-0.880.93 0.960.92 0.91 1 0.9 0.93 0.960.910.940.97-0.9-0.930.960.920.940.970.890.93
  minGyrY -0.88-0.47 0.28 0.11 0.33 0.15 -0.89 0.92 -0.9-0.96 0.89 0.91 0.93 0.91 0.93 0.9 1 0.88 -0.94 0.95 -0.9 -0.94 0.97 0.92 0.93 0.94 -0.9 -0.94 0.96 0.91
  minGyrZ -0.91-0.380.22 0.330.12 0.42-0.870.880.880.880.860.96 0.94 0.9 0.88 0.930.88 1 -0.930.920.880.94-0.9-0.970.930.920.880.930.880.930.880.96
   sdAccX -0.940.42-0.170.270.180.240.940.920.95 0.950.890.94 0.990.950.930.960.940.93 1 0.970.95 0.980.940.95 1 0.960.95 0.980.930.94
   sdAccY -0.910.43 -0.3-0.190.23-0.2 0.910.960.91 0.95 0.9 0.940.970.960.950.910.950.92 0.97 1 0.910.960.960.95 0.96 1 0.910.960.95 0.94
   sdAccZ -0.890.270.0350.260.170.250.910.83 1 0.920.830.89 0.950.950.940.94-0.9-0.880.950.91 1 0.950.88 0.9 0.950.92 1 0.940.870.89
   sdGyrX -0.950.41-0.210.260.17-0.240.910.910.95 0.96 0.9 0.95-0.980.950.940.970.940.94 0.980.960.95 1 0.950.96 0.980.960.95 1 0.940.95
   sdGyrY --0.9 0.53-0.370.18-0.2-0.170.860.960.880.940.960.92-0.940.890.94-0.9-0.97-0.9 0.940.960.88 0.95 1 0.93 0.940.950.880.95 1 0.92
   sdGyrZ -0.940.38-0.230.270.180.260.890.91 0.9 0.920.89 1 0.960.930.910.930.920.970.950.95 0.9 0.960.93 1 0.950.95 0.9 0.960.92 1
 madAccX +0.950.43-0.160.270.170.240.930.910.950.950.880.94-0.980.950.920.960.930.93 1 0.960.950.980.940.95 1 0.960.950.970.930.94
 madAccY +0.92 0.4 +0.28-0.2-0.220.21 0.9 0.950.92 0.950.89 0.94 0.970.960.950.92 0.94 0.92 0.96 1 0.92 0.960.950.95 0.96 1 0.92 0.960.950.95
 maxAccZ -0.9 0.280.0460.270.160.23 0.9 0.83 1 0.920.84 0.9 0.950.940.940.940.9-0.88 0.950.91 1 0.950.88 0.9 0.950.92 1 0.940.87 0.9
 madGyrX +0.950.41-0.220.260.170.24 0.9 0.910.94 0.96 0.9 0.96-0.970.940.940.970.940.93 0.980.960.94 1 0.950.96 0.970.960.94 1 0.940.96
 madGyrY +0.890.54-0.380.170.210.160.850.960.870.930.950.91-0.930.880.920.890.960.8800.930.950.870.94 1 0.920.930.940.870.94 1 0.91
 maxGyrZ +0.950.37-0.230.310.150.250.87 0.9 0.9 0.910.88 1 +0.950.92-0.9-0.930.910.960.940.940.89 0.950.92 1 0.940.94 0.9 0.960.91 1
                                                                   minAccY
                                                      maxGyrY
                                                           naxGyrZ
                                                               minAccX
                                                                        minAccZ
                                                                                             sdAccY
```

## In [35]:

```
X = dataset.loc[:, dataset.columns != 'Activity']
Y = dataset.loc[:, dataset.columns == 'Activity']
```

```
# Split the data in 70:30 ratio for train and test
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42
In [37]:
clf = LogisticRegression(random state=0, multi class='multinomial', solver='newton-cg')
model = clf.fit(x train, y train.Activity)
y predicted = model.predict(x test)
In [38]:
model.score(x test,y test)
Out[38]:
0.9828571428571429
In [39]:
print(confusion matrix(y test, y predicted))
[[34 1 0 0 0 0]
[ 0 13 0 0 0 0]
 [ 0 0 49 1 0 0]
 [ 0 0 0 14 0 0]
 [ 0 0 0 0 19 1]
 [ 0 0 0 0 0 43]]
```

## In [40]:

	precision	recall	f1-score	support
JOGGING	1.00	0.97	0.99	35
RUNNING	0.93	1.00	0.96	13
SITTING	1.00	0.98	0.99	50
STAIRCLIMB	0.93	1.00	0.97	14
STAIRDOWN	1.00	0.95	0.97	20
WALKING	0.98	1.00	0.99	43
accuracy			0.98	175
macro avg	0.97	0.98	0.98	175
weighted avg	0.98	0.98	0.98	175