exploration

December 9, 2023

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[ ]:
```
data=pd.read_csv('dataset2.csv')
```

[ ]:
```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2864056 entries, 0 to 2864055
Data columns (total 33 columns):
 #   Column                      Dtype
---  ------                      -----
 0   activityID                  object
 1   heart_rate                  float64
 2   hand temperature (°C)       float64
 3   hand acceleration X ±16g    float64
 4   hand acceleration Y ±16g    float64
 5   hand acceleration Z ±16g    float64
 6   hand gyroscope X            float64
 7   hand gyroscope Y            float64
 8   hand gyroscope Z            float64
 9   hand magnetometer X         float64
 10  hand magnetometer Y         float64
 11  hand magnetometer Z         float64
 12  chest temperature (°C)      float64
 13  chest acceleration X ±16g   float64
 14  chest acceleration Y ±16g   float64
 15  chest acceleration Z ±16g   float64
 16  chest gyroscope X           float64
 17  chest gyroscope Y           float64
 18  chest gyroscope Z           float64
 19  chest magnetometer X        float64
 20  chest magnetometer Y        float64
 21  chest magnetometer Z        float64
 22  ankle temperature (°C)      float64
 23  ankle acceleration X ±16g   float64
```

```
 24  ankle acceleration Y ±16g  float64
 25  ankle acceleration Z ±16g  float64
 26  ankle gyroscope X          float64
 27  ankle gyroscope Y          float64
 28  ankle gyroscope Z          float64
 29  ankle magnetometer X       float64
 30  ankle magnetometer Y       float64
 31  ankle magnetometer Z       float64
 32  PeopleId                   int64
dtypes: float64(31), int64(1), object(1)
memory usage: 721.1+ MB
None
```

[ ]:
```
print(data.isnull().sum())
data=data.drop(['PeopleId'],axis=1)
```

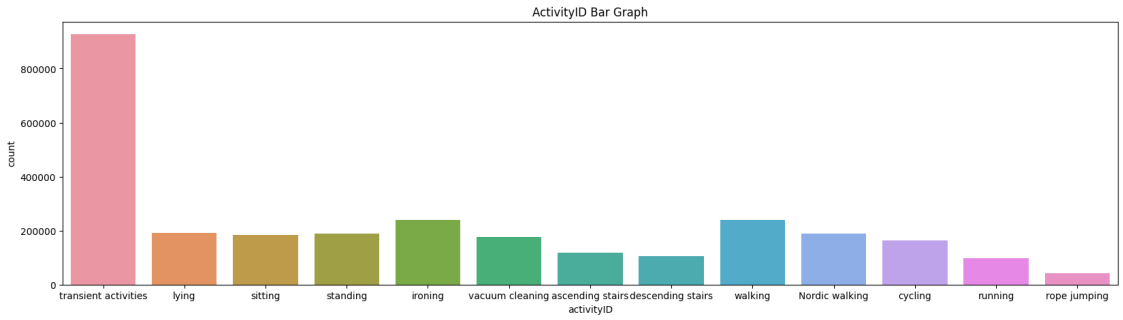
```
activityID                   0
heart_rate                  46
hand temperature (°C)        0
hand acceleration X ±16g     0
hand acceleration Y ±16g     0
hand acceleration Z ±16g     0
hand gyroscope X             0
hand gyroscope Y             0
hand gyroscope Z             0
hand magnetometer X          0
hand magnetometer Y          0
hand magnetometer Z          0
chest temperature (°C)       0
chest acceleration X ±16g    0
chest acceleration Y ±16g    0
chest acceleration Z ±16g    0
chest gyroscope X            0
chest gyroscope Y            0
chest gyroscope Z            0
chest magnetometer X         0
chest magnetometer Y         0
chest magnetometer Z         0
ankle temperature (°C)       0
ankle acceleration X ±16g    0
ankle acceleration Y ±16g    0
ankle acceleration Z ±16g    0
ankle gyroscope X            0
ankle gyroscope Y            0
ankle gyroscope Z            0
ankle magnetometer X         0
ankle magnetometer Y         0
ankle magnetometer Z         0
```

```
PeopleId                      0
dtype: int64
```

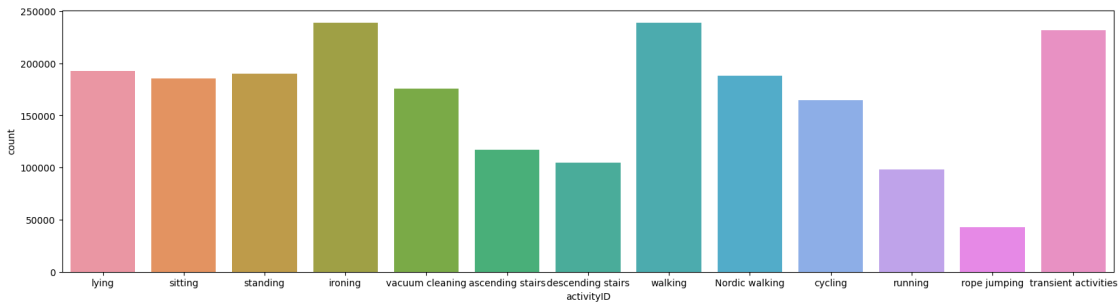
[ ]:
```
#plot activityID distribution
plt.figure(figsize=(20,5))
sns.countplot(x='activityID',data=data)
plt.title('ActivityID Bar Graph')
plt.show()

#filter data if activity id is transient activities
data_transient = data[data['activityID'] == 'transient activities']
data_transient = data_transient.sample(frac=0.25)
data = data[data['activityID'] != 'transient activities']
data = pd.concat([data,data_transient])

#plot new activityID distribution
plt.figure(figsize=(20,5))
sns.countplot(data['activityID'])
plt.show()
```




```
/Users/franklin/opt/anaconda3/lib/python3.9/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



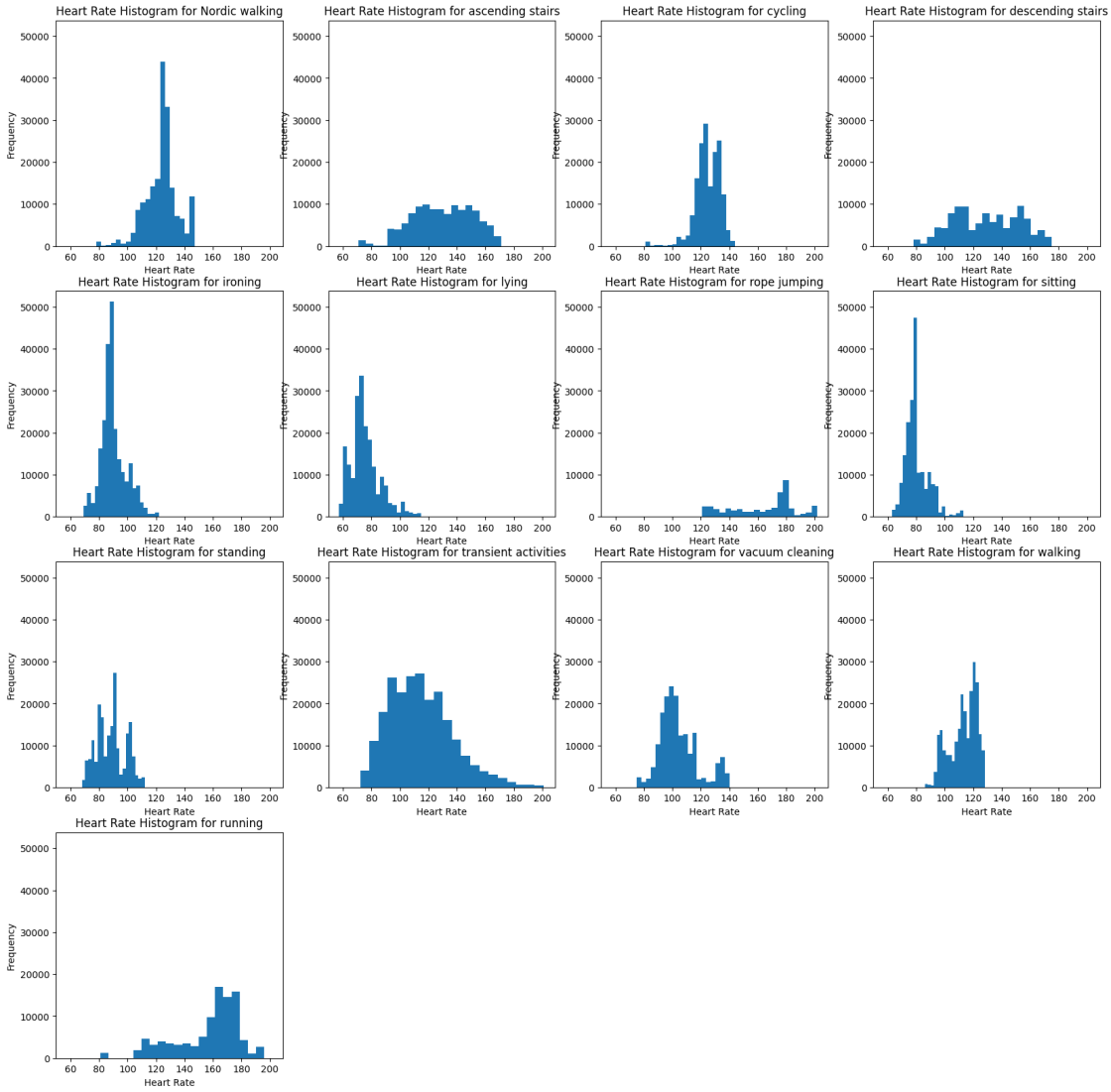
[ ]:
```
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

y=data["activityID"]
X=data.drop(["activityID"],axis=1)
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
 ↪3,random_state=42)
```

[ ]:
```
#plot subplot of histogram of heart rate for every activity
plt.figure(figsize=(20,20))
plt.subplot(4,4,1)
plt.hist(data[data["activityID"]=="Nordic walking"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for Nordic walking")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,2,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="ascending stairs"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for ascending stairs")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,3,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="cycling"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for cycling")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.xlabel("Heart Rate")
plt.subplot(4,4,4,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="descending stairs"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for descending stairs")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,5,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="ironing"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for ironing")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,6,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="lying"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for lying")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,7,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="rope jumping"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for rope jumping")
```

```
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,8,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="sitting"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for sitting")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,9,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="standing"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for standing")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,10,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="transient activities"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for transient activities")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,11,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="vacuum cleaning"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for vacuum cleaning")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,12,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="walking"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for walking")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
plt.subplot(4,4,13,sharey=plt.subplot(4,4,1),sharex=plt.subplot(4,4,1))
plt.hist(data[data["activityID"]=="running"]["heart_rate"],bins=20)
plt.title("Heart Rate Histogram for running")
plt.xlabel("Heart Rate")
plt.ylabel("Frequency")
```

[ ]:
```
Text(0, 0.5, 'Frequency')
```

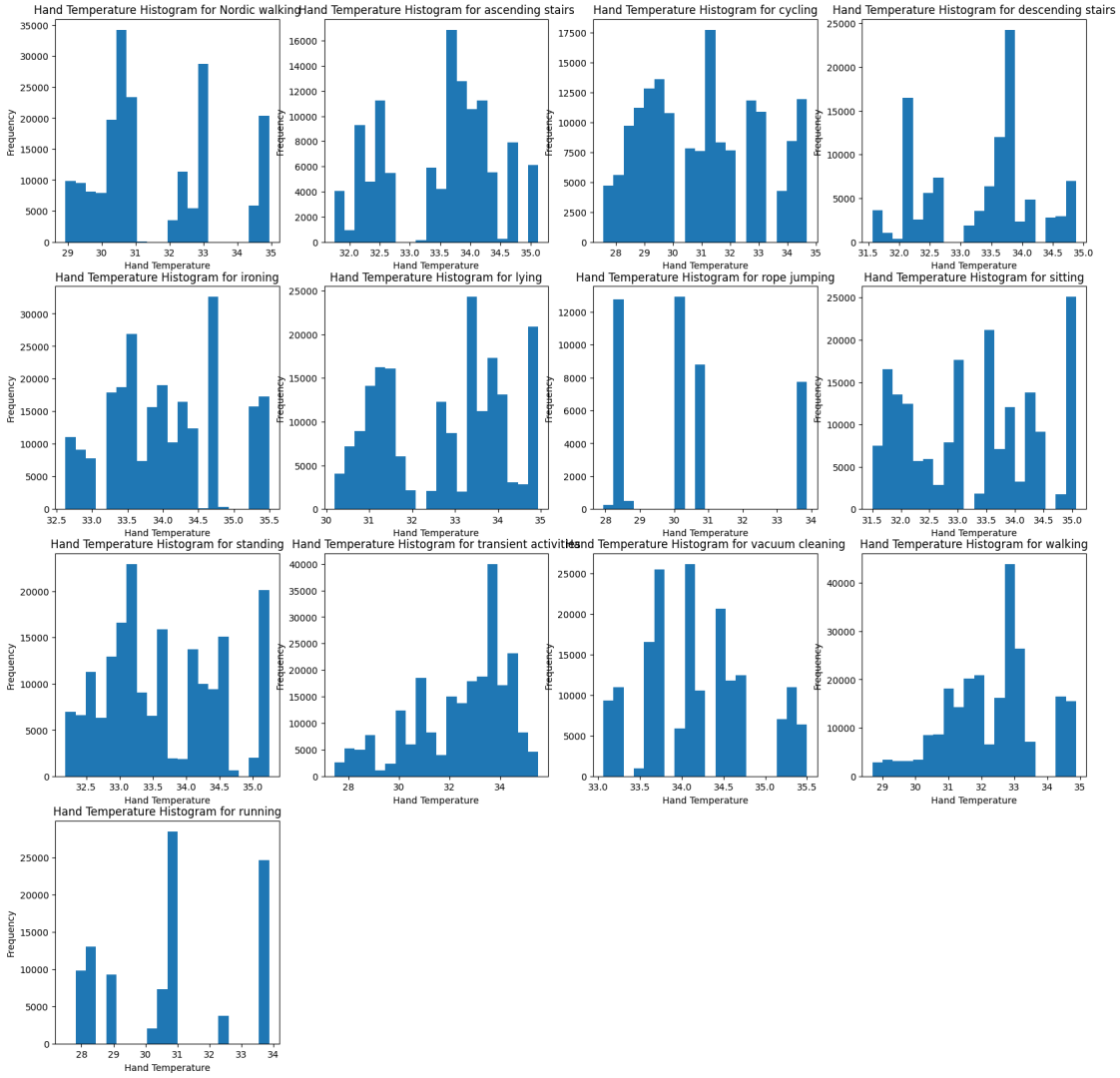


[ ]:
```
#do the same thing for hand temperature
plt.figure(figsize=(20,20))
plt.subplot(4,4,1)
plt.hist(data[data["activityID"]=="Nordic walking"]["hand temperature␣
  ↪(°C)"],bins=20)
plt.title("Hand Temperature Histogram for Nordic walking")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,2)
plt.hist(data[data["activityID"]=="ascending stairs"]["hand temperature␣
  ↪(°C)"],bins=20)
plt.title("Hand Temperature Histogram for ascending stairs")
plt.xlabel("Hand Temperature")
```

```
plt.ylabel("Frequency")
plt.subplot(4,4,3)
plt.hist(data[data["activityID"]=="cycling"]["hand temperature (°C)"],bins=20)
plt.title("Hand Temperature Histogram for cycling")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,4)
plt.hist(data[data["activityID"]=="descending stairs"]["hand temperature␣
 ↪(°C)"],bins=20)
plt.title("Hand Temperature Histogram for descending stairs")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,5)
plt.hist(data[data["activityID"]=="ironing"]["hand temperature (°C)"],bins=20)
plt.title("Hand Temperature Histogram for ironing")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,6)
plt.hist(data[data["activityID"]=="lying"]["hand temperature (°C)"],bins=20)
plt.title("Hand Temperature Histogram for lying")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,7)
plt.hist(data[data["activityID"]=="rope jumping"]["hand temperature␣
 ↪(°C)"],bins=20)
plt.title("Hand Temperature Histogram for rope jumping")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,8)
plt.hist(data[data["activityID"]=="sitting"]["hand temperature (°C)"],bins=20)
plt.title("Hand Temperature Histogram for sitting")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,9)
plt.hist(data[data["activityID"]=="standing"]["hand temperature (°C)"],bins=20)
plt.title("Hand Temperature Histogram for standing")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,10)
plt.hist(data[data["activityID"]=="transient activities"]["hand temperature␣
 ↪(°C)"],bins=20)
plt.title("Hand Temperature Histogram for transient activities")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,11)
plt.hist(data[data["activityID"]=="vacuum cleaning"]["hand temperature␣
 ↪(°C)"],bins=20)
```

```
plt.title("Hand Temperature Histogram for vacuum cleaning")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,12)
plt.hist(data[data["activityID"]=="walking"]["hand temperature (°C)"],bins=20)
plt.title("Hand Temperature Histogram for walking")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,13)
plt.hist(data[data["activityID"]=="running"]["hand temperature (°C)"],bins=20)
plt.title("Hand Temperature Histogram for running")
plt.xlabel("Hand Temperature")
plt.ylabel("Frequency")
```

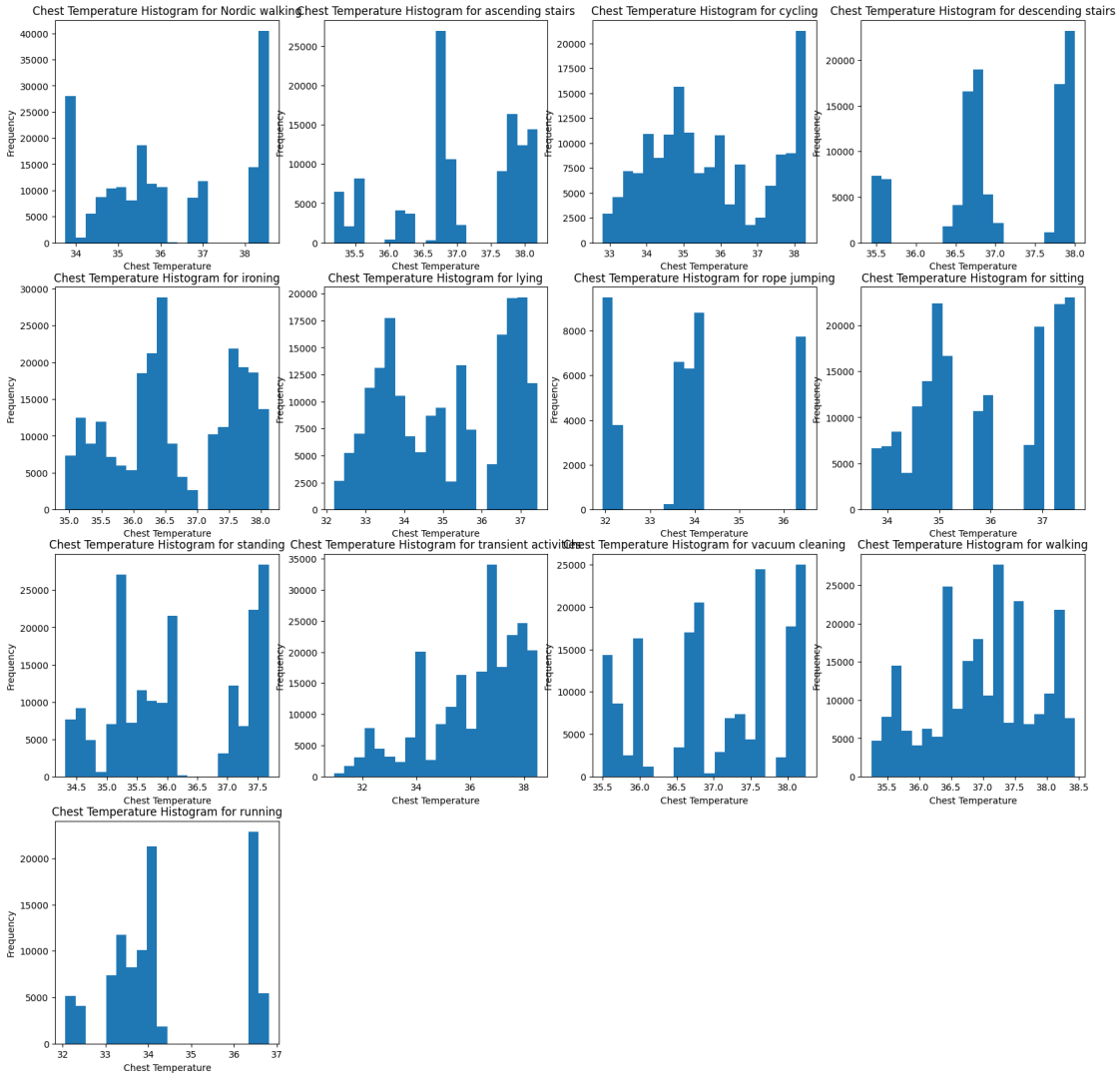
[ ]: Text(0, 0.5, 'Frequency')



[ ]:
```
#do the same thing for hand temperature
plt.figure(figsize=(20,20))
plt.subplot(4,4,1)
plt.hist(data[data["activityID"]=="Nordic walking"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for Nordic walking")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,2)
plt.hist(data[data["activityID"]=="ascending stairs"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for ascending stairs")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,3)
plt.hist(data[data["activityID"]=="cycling"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for cycling")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,4)
plt.hist(data[data["activityID"]=="descending stairs"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for descending stairs")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,5)
plt.hist(data[data["activityID"]=="ironing"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for ironing")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,6)
plt.hist(data[data["activityID"]=="lying"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for lying")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,7)
plt.hist(data[data["activityID"]=="rope jumping"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for rope jumping")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,8)
plt.hist(data[data["activityID"]=="sitting"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for sitting")
plt.xlabel("Chest Temperature")
```

```
plt.ylabel("Frequency")
plt.subplot(4,4,9)
plt.hist(data[data["activityID"]=="standing"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for standing")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,10)
plt.hist(data[data["activityID"]=="transient activities"]["chest temperature␣
↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for transient activities")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,11)
plt.hist(data[data["activityID"]=="vacuum cleaning"]["chest temperature␣
↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for vacuum cleaning")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,12)
plt.hist(data[data["activityID"]=="walking"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for walking")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,13)
plt.hist(data[data["activityID"]=="running"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for running")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
```

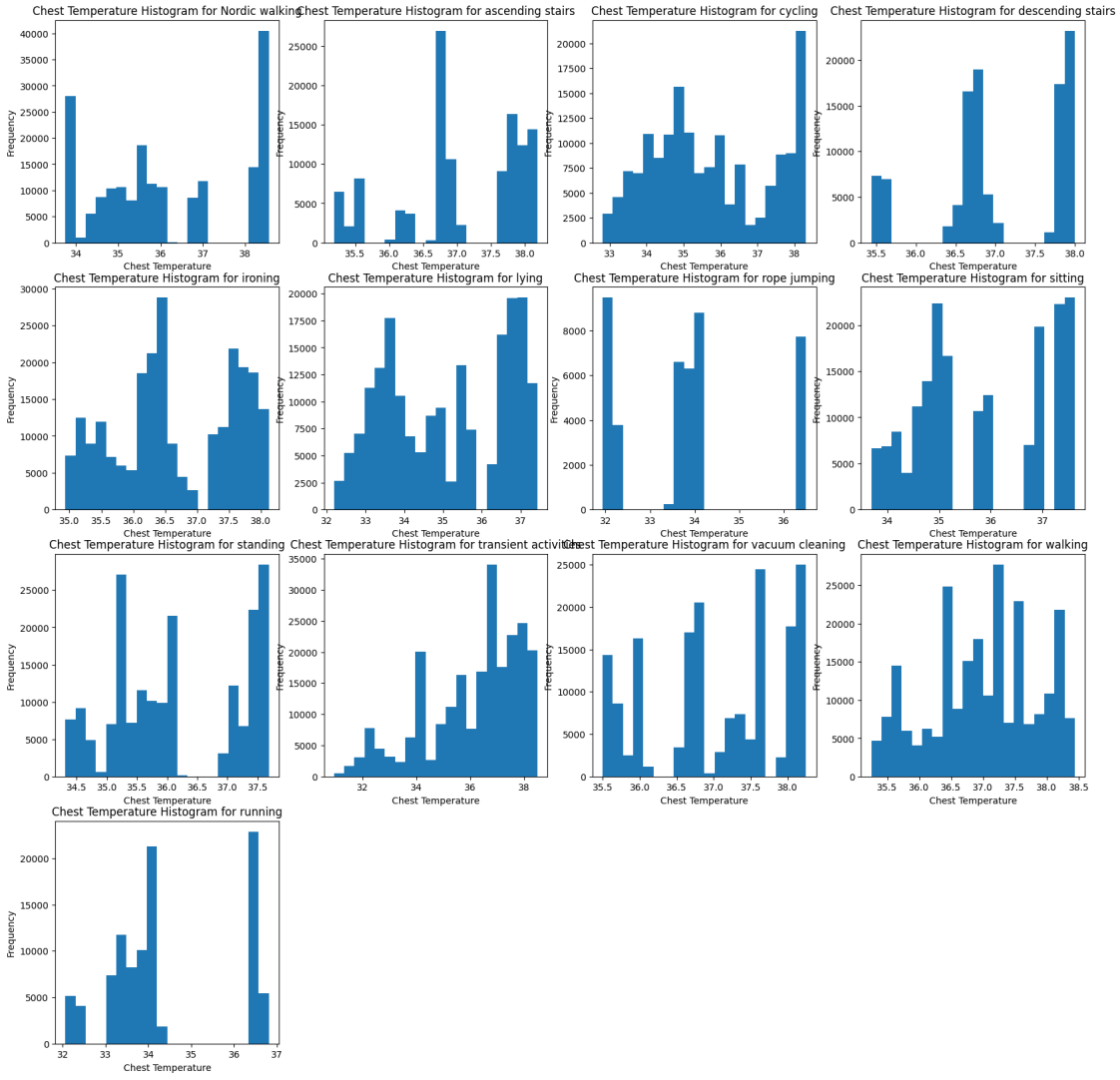
[ ]:
```
Text(0, 0.5, 'Frequency')
```



[ ]:
```
#Do the same thing for chest temperature
plt.figure(figsize=(20,20))
plt.subplot(4,4,1)
plt.hist(data[data["activityID"]=="Nordic walking"]["chest temperature␣
  ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for Nordic walking")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,2)
plt.hist(data[data["activityID"]=="ascending stairs"]["chest temperature␣
  ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for ascending stairs")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
```

```
plt.subplot(4,4,3)
plt.hist(data[data["activityID"]=="cycling"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for cycling")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,4)
plt.hist(data[data["activityID"]=="descending stairs"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for descending stairs")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,5)
plt.hist(data[data["activityID"]=="ironing"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for ironing")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,6)
plt.hist(data[data["activityID"]=="lying"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for lying")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,7)
plt.hist(data[data["activityID"]=="rope jumping"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for rope jumping")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,8)
plt.hist(data[data["activityID"]=="sitting"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for sitting")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,9)
plt.hist(data[data["activityID"]=="standing"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for standing")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,10)
plt.hist(data[data["activityID"]=="transient activities"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for transient activities")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,11)
plt.hist(data[data["activityID"]=="vacuum cleaning"]["chest temperature␣
 ↪(°C)"],bins=20)
plt.title("Chest Temperature Histogram for vacuum cleaning")
```

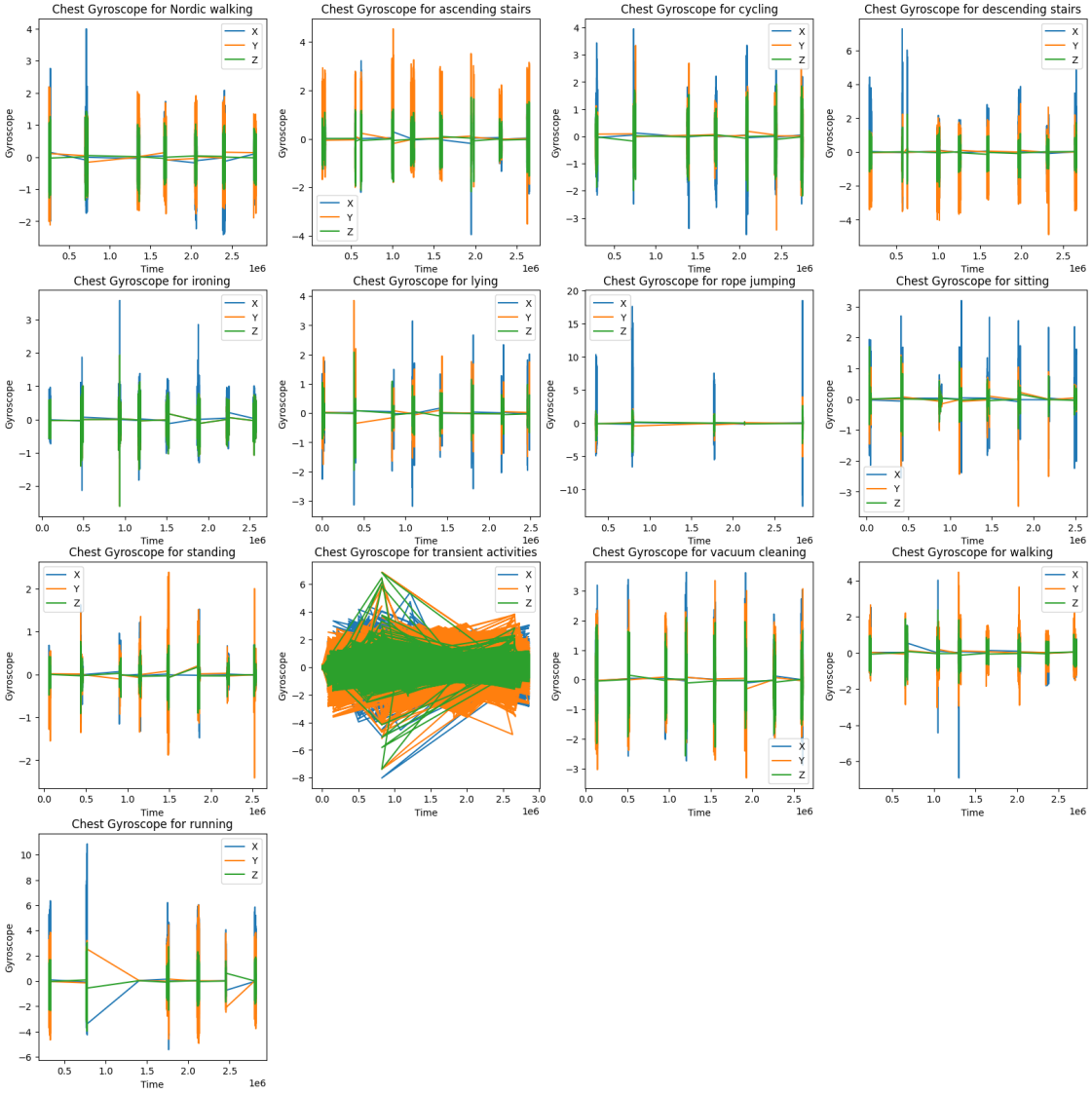
```
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,12)
plt.hist(data[data["activityID"]=="walking"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for walking")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.subplot(4,4,13)
plt.hist(data[data["activityID"]=="running"]["chest temperature (°C)"],bins=20)
plt.title("Chest Temperature Histogram for running")
plt.xlabel("Chest Temperature")
plt.ylabel("Frequency")
plt.show()
```



[ ]:
```
plt.figure(figsize=(20,20))
plt.subplot(4,4,1)
plt.plot(data[data["activityID"]=="Nordic walking"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="Nordic walking"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="Nordic walking"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for Nordic walking")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,2)
plt.plot(data[data["activityID"]=="ascending stairs"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="ascending stairs"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="ascending stairs"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for ascending stairs")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,3)
plt.plot(data[data["activityID"]=="cycling"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="cycling"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="cycling"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for cycling")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,4)
plt.plot(data[data["activityID"]=="descending stairs"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="descending stairs"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="descending stairs"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for descending stairs")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,5)
plt.plot(data[data["activityID"]=="ironing"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="ironing"]["chest gyroscope Z"])
plt.plot(data[data["activityID"]=="ironing"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for ironing")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,6)
plt.plot(data[data["activityID"]=="lying"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="lying"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="lying"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for lying")
plt.xlabel("Time")
```

```
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,7)
plt.plot(data[data["activityID"]=="rope jumping"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="rope jumping"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="rope jumping"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for rope jumping")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,8)
plt.plot(data[data["activityID"]=="sitting"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="sitting"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="sitting"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for sitting")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,9)
plt.plot(data[data["activityID"]=="standing"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="standing"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="standing"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for standing")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,10)
plt.plot(data[data["activityID"]=="transient activities"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="transient activities"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="transient activities"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for transient activities")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,11)
plt.plot(data[data["activityID"]=="vacuum cleaning"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="vacuum cleaning"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="vacuum cleaning"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for vacuum cleaning")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,12)
plt.plot(data[data["activityID"]=="walking"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="walking"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="walking"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for walking")
```

```
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.subplot(4,4,13)
plt.plot(data[data["activityID"]=="running"]["chest gyroscope X"])
plt.plot(data[data["activityID"]=="running"]["chest gyroscope Y"])
plt.plot(data[data["activityID"]=="running"]["chest gyroscope Z"])
plt.title("Chest Gyroscope for running")
plt.xlabel("Time")
plt.ylabel("Gyroscope")
plt.legend(["X","Y","Z"])
plt.show()
```



[ ]:
```
#check for correlation between features
plt.figure(figsize=(5,5))
sns.heatmap(data.corr(),annot=True)
plt.show()
```