Name : Nitish Adhikari

Contact : +91 9650740295 (nitishbuzzpro@gmail.com (mailto:nitishbuzzpro@gmail.com))

Linkedin : https://www.linkedin.com/in/nitish-adhikari-6b2350248 (https://www.linkedin.com/in/nitish-adhikari-6b2350248)

Github : https://github.com/nitishbuzzpro (https://github.com/nitishbuzzpro)

PROJECT : Analyise logs form Healthapp and build the Analytics solution ¶

In [1]:
```
import piplite
await piplite.install('numpy')
await piplite.install('pandas')
await piplite.install('matplotlib')
await piplite.install('seaborn')
```

In [2]:
```
#Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
#Create dataframe
df = pd.read_csv('HealthApp_2k.log_structured.csv')
df = df.set_index('LineId')
df
```

Out[3]:

LineId	Time	Component	Pid	Content	Ever
1	20171223-22:15:29:606	Step_LSC	30002312	onStandStepChanged 3579	E
2	20171223-22:15:29:615	Step_LSC	30002312	onExtend:1514038530000 14 0 4	E
3	20171223-22:15:29:633	Step_StandReportReceiver	30002312	onReceive action: android.intent.action.SCREEN_ON	E
4	20171223-22:15:29:635	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
5	20171223-22:15:29:635	Step_StandStepCounter	30002312	flush sensor data	E
...	...	...	...	...	
1996	20171224-0:58:53:985	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1997	20171224-0:59:7:581	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1998	20171224-1:0:0:794	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1999	20171224-1:1:0:935	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
2000	20171224-1:2:35:789	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E

2000 rows × 6 columns

In [4]:
```
#Check for Null values
df.isnull().sum()
```

Out[4]:
```
Time            0
Component       0
Pid             0
Content         0
EventId         0
EventTemplate   0
dtype: int64
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['Time', 'Component', 'Pid', 'Content', 'EventId', 'EventTemplate'],
dtype='object')
```

In [6]:

```
#template data frame
df_template = pd.read_csv('HealthApp_2k.log_templates.csv')
df_template
```

Out[6]:

	EventId	EventTemplate
0	E1	Alarm uploadStaticsToDB totalSteps=<*>:<*>:<*>...
1	E2	bulkSaveDetailHiHealthData() size = <*>,totalT...
2	E3	calculateAltitudeWithCache totalAltitude=<*>
3	E4	calculateCaloriesWithCache totalCalories=<*>
4	E5	checkCurrentDay a new day comes , reset basicS...
...	...	...
70	E71	tryToReloadTodayBasicSteps<*>|<*>|<*>|<*>
71	E72	upLoadOneMinuteDataToEngine time=<*>,<*>,<*>,<...
72	E73	uploadStaticsToDB failed message=true
73	E74	uploadStaticsToDB() onResult type = <*> obj=true
74	E75	writeDataToDB size <*>

75 rows × 2 columns

In [7]:

```
df['Time'] =  pd.to_datetime(df['Time'], format='%Y%m%d-%H:%M:%S:%f')
df
```

Out[7]:

LineId	Time	Component	Pid	Content	Ever
1	2017-12-23 22:15:29.606	Step_LSC	30002312	onStandStepChanged 3579	E
2	2017-12-23 22:15:29.615	Step_LSC	30002312	onExtend:1514038530000 14 0 4	E
3	2017-12-23 22:15:29.633	Step_StandReportReceiver	30002312	onReceive action: android.intent.action.SCREEN_ON	E
4	2017-12-23 22:15:29.635	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
5	2017-12-23 22:15:29.635	Step_StandStepCounter	30002312	flush sensor data	E
...	...	...	...	...	
1996	2017-12-24 00:58:53.985	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1997	2017-12-24 00:59:07.581	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1998	2017-12-24 01:00:00.794	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1999	2017-12-24 01:01:00.935	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
2000	2017-12-24 01:02:35.789	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E

2000 rows × 6 columns

In [8]:
```
#Total Log time
total_log_time = df['Time'].iloc[-1] - df['Time'].iloc[0]
total_log_time
```

Out[8]:
```
Timedelta('0 days 02:47:06.183000')
```

In [ ]:
```

```

STEPS ANLYSIS

In [9]:
```
def data_extracter(content):
    global max_count
    if content.split(' ')[0] ==  'onStandStepChanged':
        max_count = int(content.split(' ')[1])
        return max_count
    else:
        return None
```

In [10]:
```
df['Step_change_Count'] = df['Content'].apply(data_extracter)
df
```

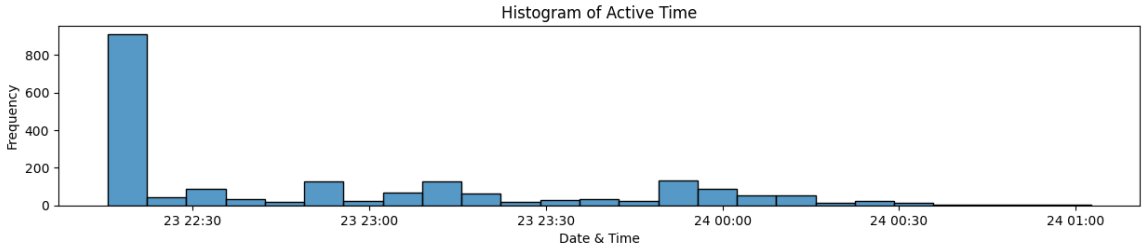
Out[10]:

LineId	Time	Component	Pid	Content	Ever
1	2017-12-23 22:15:29.606	Step_LSC	30002312	onStandStepChanged 3579	E
2	2017-12-23 22:15:29.615	Step_LSC	30002312	onExtend:1514038530000 14 0 4	E
3	2017-12-23 22:15:29.633	Step_StandReportReceiver	30002312	onReceive action: android.intent.action.SCREEN_ON	E
4	2017-12-23 22:15:29.635	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
5	2017-12-23 22:15:29.635	Step_StandStepCounter	30002312	flush sensor data	E
...	...	...	...	...	
1996	2017-12-24 00:58:53.985	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1997	2017-12-24 00:59:07.581	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1998	2017-12-24 01:00:00.794	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1999	2017-12-24 01:01:00.935	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
2000	2017-12-24 01:02:35.789	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E

2000 rows × 7 columns

In [35]:
```
#Histogram of Time when the person is most active
plt.figure(figsize=(15,2.5))
histogram = sns.histplot(df[(df['Step_change_Count']!=None)]['Time'], b
plt.xlabel('Date & Time')
plt.ylabel('Frequency')
plt.title('Histogram of Active Time')
```

Out[35]: Text(0.5, 1.0, 'Histogram of Active Time')



In [12]:
```
def timestamp_to_datetime(timestamp):
    from datetime import datetime, timedelta
    epoch = datetime(1970, 1, 1)
    datetime = epoch + timedelta(days=timestamp)

    return datetime.strftime("%Y-%m-%d %H:%M:%S")
```

In [13]:
```
#Determine the time range when person is most active

bars = [rect.get_height() for rect in histogram.patches]
bins = [rect.get_x() for rect in histogram.patches]

index_max_height = bars.index(max(bars))

range_of_bar = (bins[index_max_height],bins[index_max_height+1])

date_time_range = []

for i in range_of_bar:
    date_time_range.append(timestamp_to_datetime(i))

date_time_range
```

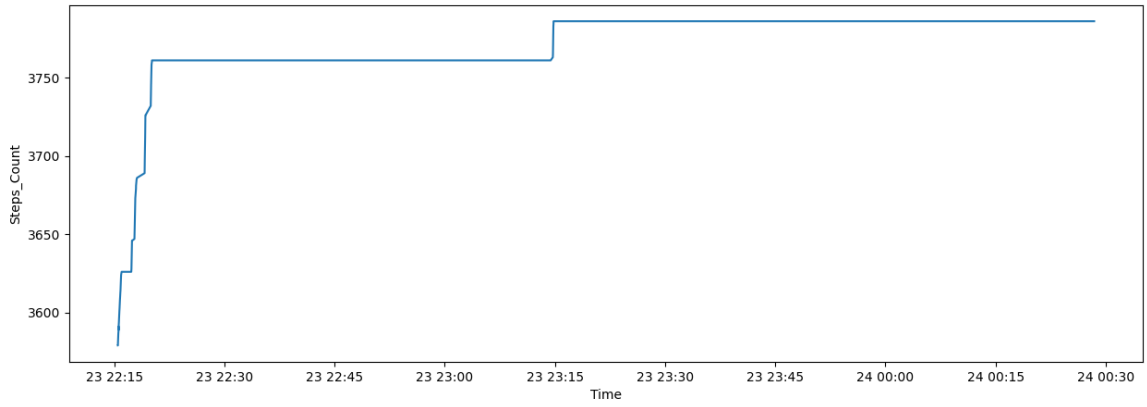
Out[13]: ['2017-12-23 22:15:29', '2017-12-23 22:22:10']

In [14]:
```
#Total number of steps taken
total_Steps_taken = df['Step_change_Count'].max() - df['Step_change_Cou
total_Steps_taken
```

Out[14]: 207.0

In [15]:
```
plt.figure(figsize=(15,5))
sns.lineplot(x=df['Time'] , y = df[df['Step_change_Count']!=None]['Step
plt.xlabel('Time')
plt.ylabel('Steps_Count')
```

Out[15]: Text(0, 0.5, 'Steps_Count')



In [16]:
```
#Time after When a person is idle(not moving)
time_idle = df[['Time']][df['Step_change_Count']==df['Step_change_Count
time_idle
```

Out[16]:
```
Time   2017-12-23 23:14:48.606
Name: 1425, dtype: datetime64[ns]
```

CALORIES ANALYSIS

In [17]:
```
def calorie_data_extracter(content):
    if content.split('=')[0] ==  'calculateCaloriesWithCache totalCalor
        max_calorie_count = int(content.split('=')[1])
        return max_calorie_count
    else:
        return None
```

In [18]:
```
df['Calorie_change_count'] = df['Content'].apply(calorie_data_extracter
df[df['EventTemplate']== 'calculateCaloriesWithCache totalCalories=<*>'
df
```

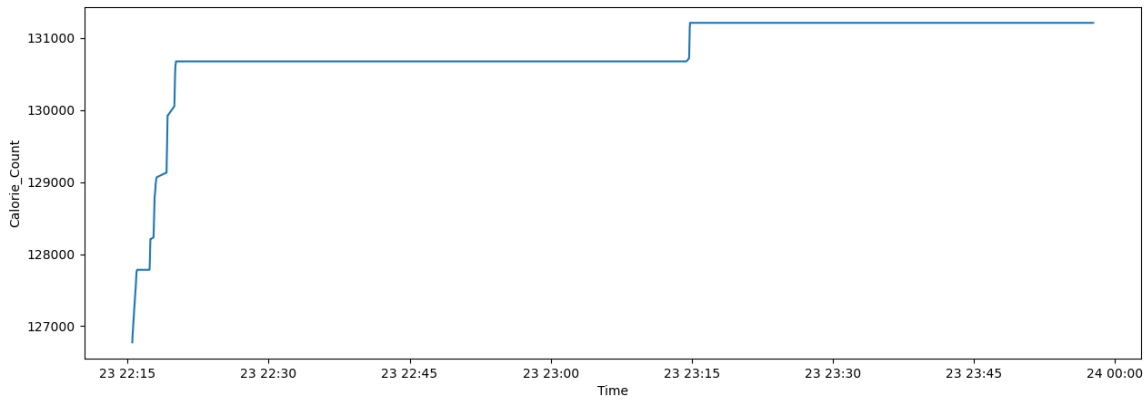
Out[18]:

LineId	Time	Component	Pid	Content	Ever
1	2017-12-23 22:15:29.606	Step_LSC	30002312	onStandStepChanged 3579	E
2	2017-12-23 22:15:29.615	Step_LSC	30002312	onExtend:1514038530000 14 0 4	E
3	2017-12-23 22:15:29.633	Step_StandReportReceiver	30002312	onReceive action: android.intent.action.SCREEN_ON	E
4	2017-12-23 22:15:29.635	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
5	2017-12-23 22:15:29.635	Step_StandStepCounter	30002312	flush sensor data	E
...	...	...	...	...	
1996	2017-12-24 00:58:53.985	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1997	2017-12-24 00:59:07.581	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1998	2017-12-24 01:00:00.794	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
1999	2017-12-24 01:01:00.935	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E
2000	2017-12-24 01:02:35.789	Step_LSC	30002312	processHandleBroadcastAction action:android.in...	E

2000 rows × 8 columns

In [19]:
```
plt.figure(figsize=(15,5))
sns.lineplot(x=df['Time'] , y=df[(df['Calorie_change_count']!=0) & (df[
plt.xlabel('Time')
plt.ylabel('Calorie_Count')
```

Out[19]:
```
Text(0, 0.5, 'Calorie_Count')
```

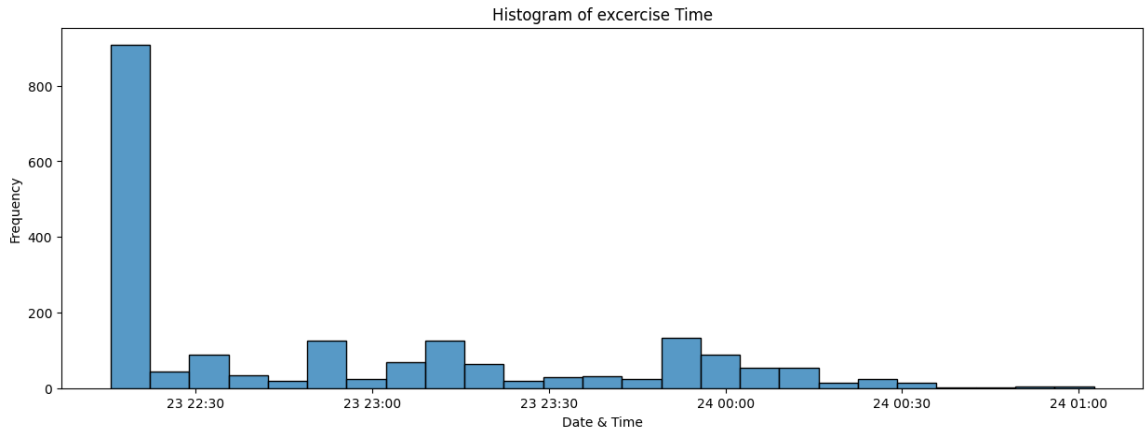


In [20]:
```
#Total calories burnt
max_calorie_count = df[(df['EventTemplate']== 'calculateCaloriesWithCac
min_calorie_count = df[(df['EventTemplate']== 'calculateCaloriesWithCac
Total_calories_burnt = max_calorie_count - min_calorie_count
Total_calories_burnt
```

Out[20]: 4433.0

In [21]:
```
#Histogram of Time when the person is excercing
plt.figure(figsize=(15,5))
histogram2 = sns.histplot(df[(df['Calorie_change_count']!=None)]['Time'
plt.xlabel('Date & Time')
plt.ylabel('Frequency')
plt.title('Histogram of excercise Time')
```

Out[21]: Text(0.5, 1.0, 'Histogram of excercise Time')



In [22]:
```
#Determine the time range when person is most active

bars = [rect.get_height() for rect in histogram2.patches]
bins = [rect.get_x() for rect in histogram2.patches]

index_max_height = bars.index(max(bars))

range_of_bar = (bins[index_max_height],bins[index_max_height+1])

date_time_range2 = []

for i in range_of_bar:
    date_time_range2.append(timestamp_to_datetime(i))

date_time_range2
```

Out[22]: ['2017-12-23 22:15:29', '2017-12-23 22:22:10']

In [23]:
```
#Time after the person sleeps
sleep_time = df[df['Calorie_change_count']==max_calorie_count]['Time'].
sleep_time
```

Out[23]: Timestamp('2017-12-23 23:14:48.922000')

Anlytical Solutions

1. Total recorded time period in the Log

In [24]:
```
1 total_log_time
```

Out[24]:
```
Timedelta('0 days 02:47:06.183000')
```

2. The time range when the person is most active

In [25]:
```
1 date_time_range
```

Out[25]:
```
['2017-12-23 22:15:29', '2017-12-23 22:22:10']
```

3. Total number of steps taken by the person

In [26]:
```
1 total_Steps_taken
```

Out[26]:
```
207.0
```

4. Time after which the person not walking

In [27]:
```
1 time_idle
```

Out[27]:
```
Time   2017-12-23 23:14:48.606
Name: 1425, dtype: datetime64[ns]
```

5. Total Calories Burnt

In [28]:
```
1 Total_calories_burnt
```

Out[28]:
```
4433.0
```

6. Time range when the maximum calories are burnt/ exercising

In [29]:
```
1 date_time_range2
```

Out[29]:
```
['2017-12-23 22:15:29', '2017-12-23 22:22:10']
```

7. Time after the person sleeps

In [30]:
```
1 sleep_time
```

Out[30]:
```
Timestamp('2017-12-23 23:14:48.922000')
```