sleep_EDA

January 18, 2024

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

[ ]:
```
sleep = pd.read_csv('sleep.csv')
sleep.head()
```

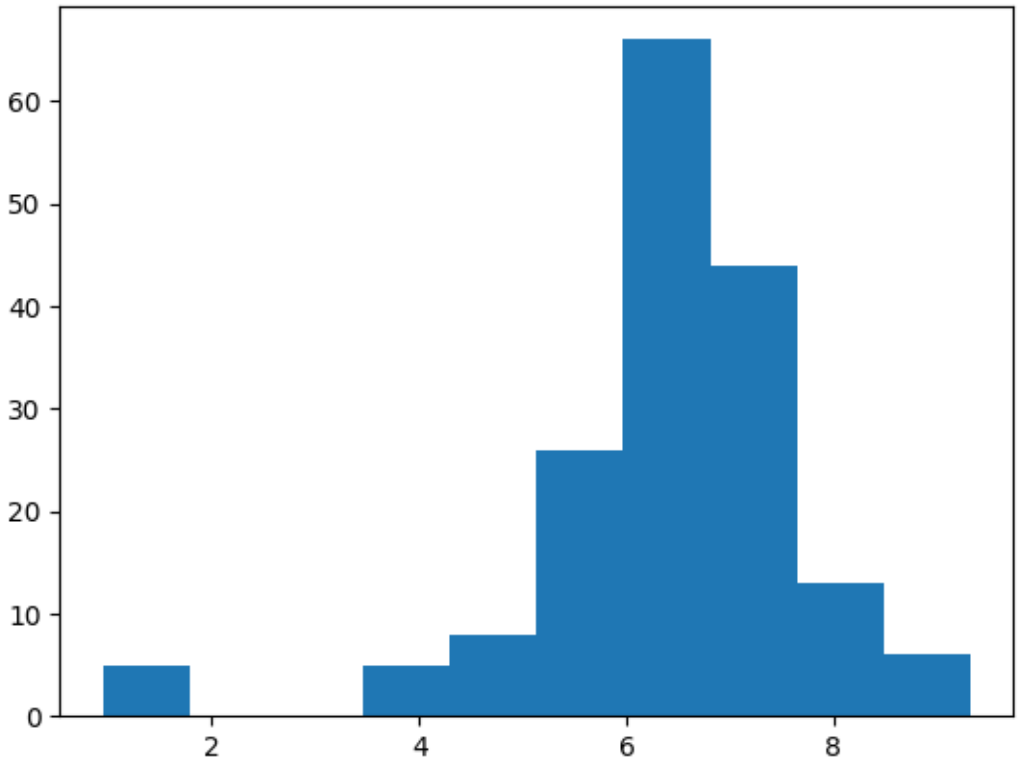
[ ]:

	Unnamed: 0	date	sleepStart	sleepEnd	asleep	awake	timeInBed	sleepEfficiency
0	0	2023-06-28	00:04:00	07:55:30	7.083333	0.766667	7.850000	95
1	1	2023-06-29	23:32:00	07:26:30	6.316667	1.583333	7.900000	91
2	2	2023-06-30	01:39:00	08:00:30	5.166667	1.183333	6.350000	91
3	3	2023-07-01	01:21:00	09:29:30	7.200000	0.933333	8.133333	94
4	4	2023-07-02	01:24:00	11:34:30	8.683333	1.483333	10.166667	93

1 Distribution

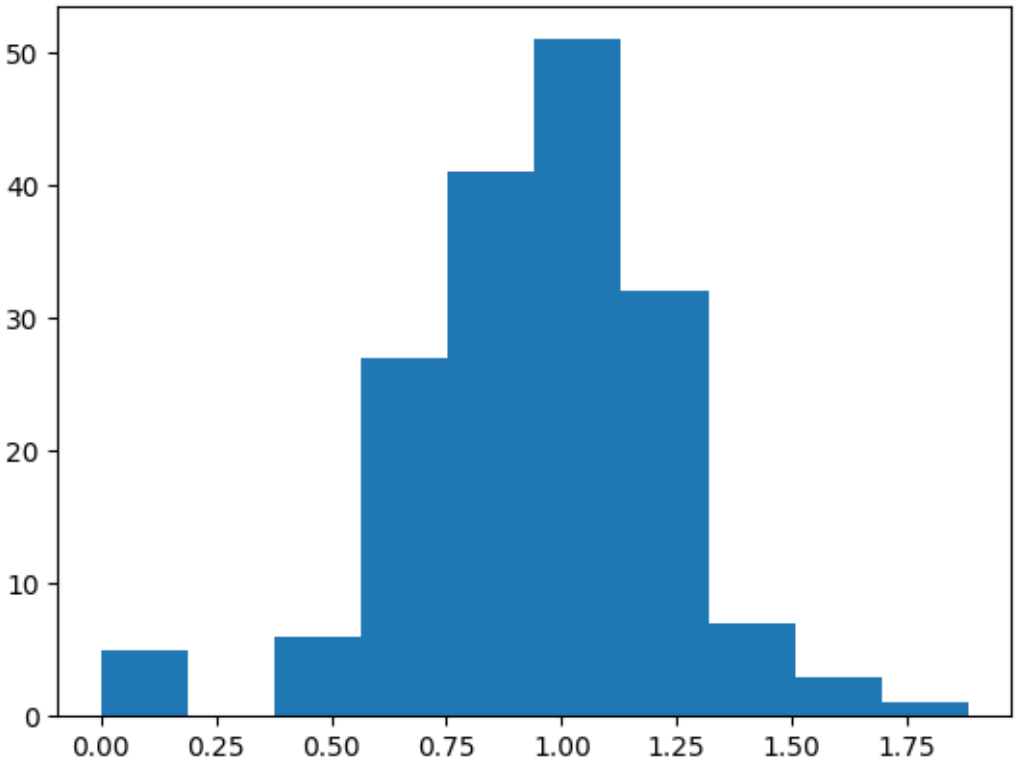
1.0.1 asleep

[ ]:
```
plt.hist(x=sleep['asleep'])
plt.show()
```



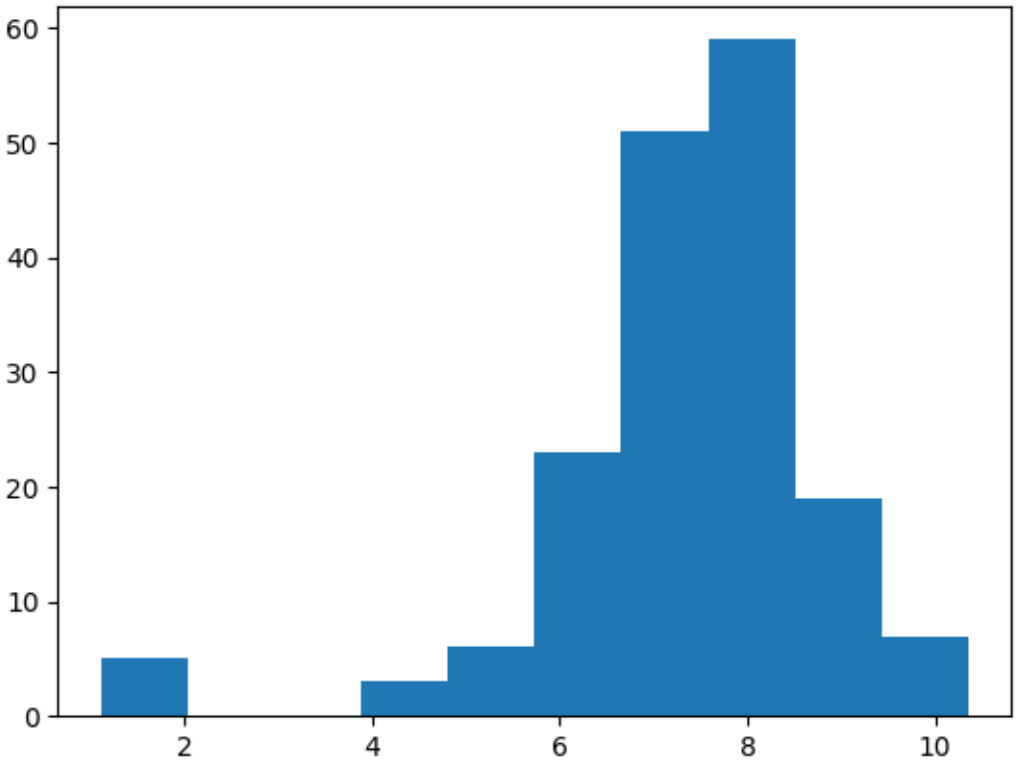
1.0.2 awake

[ ]:
```
plt.hist(x=sleep['awake'])
plt.show()
```



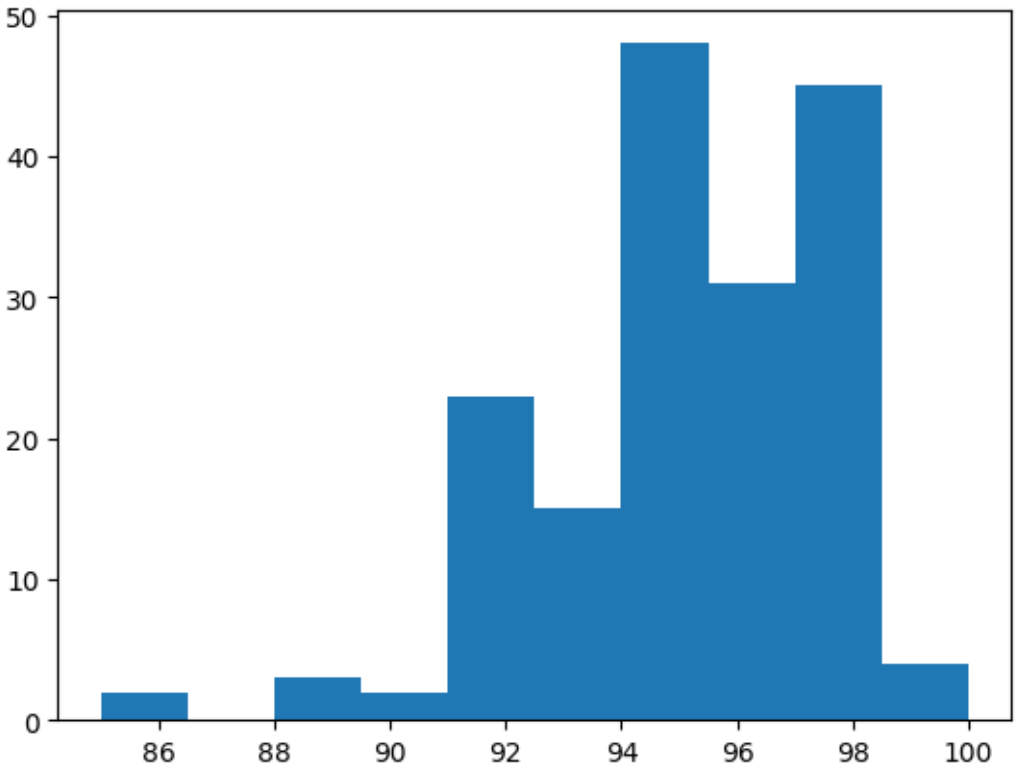
1.0.3 timeInBed

[ ]:
```
plt.hist(x=sleep['timeInBed'])
plt.show()
```



1.0.4 sleepEfficiency

[ ]:
```
plt.hist(x=sleep['sleepEfficiency'])
plt.show()
```

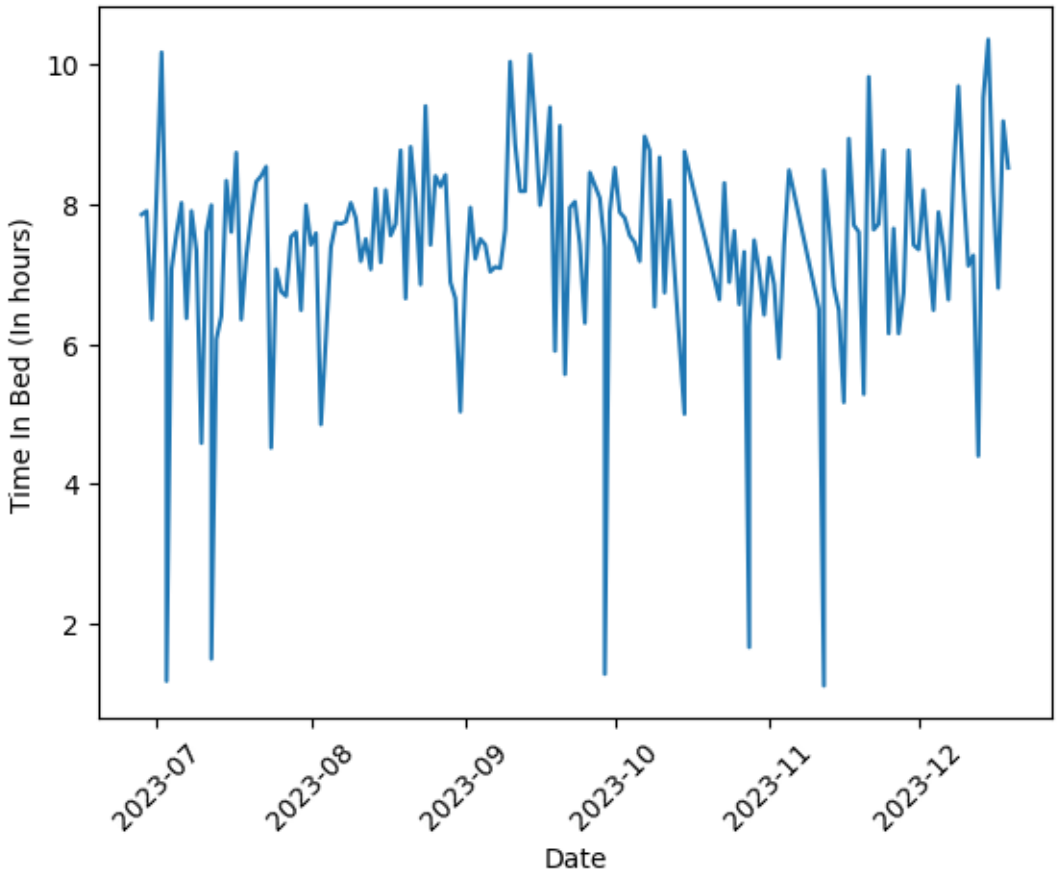


2 Multivariate

2.1 timeInBed

2.1.1 Time Series Plot (whole dataset from June to December)

[ ]:
```
fig, ax = plt.subplots()
ax.plot(sleep['date'], sleep['timeInBed'])
ax.set_xlabel('Date')
ax.set_ylabel('Time In Bed (In hours)')
plt.xticks(rotation=45)
plt.show()
```



2.1.2 Time Series Plot (Monthly)

[ ]:
```
sleep['date'] = pd.to_datetime(sleep['date'])
```

June

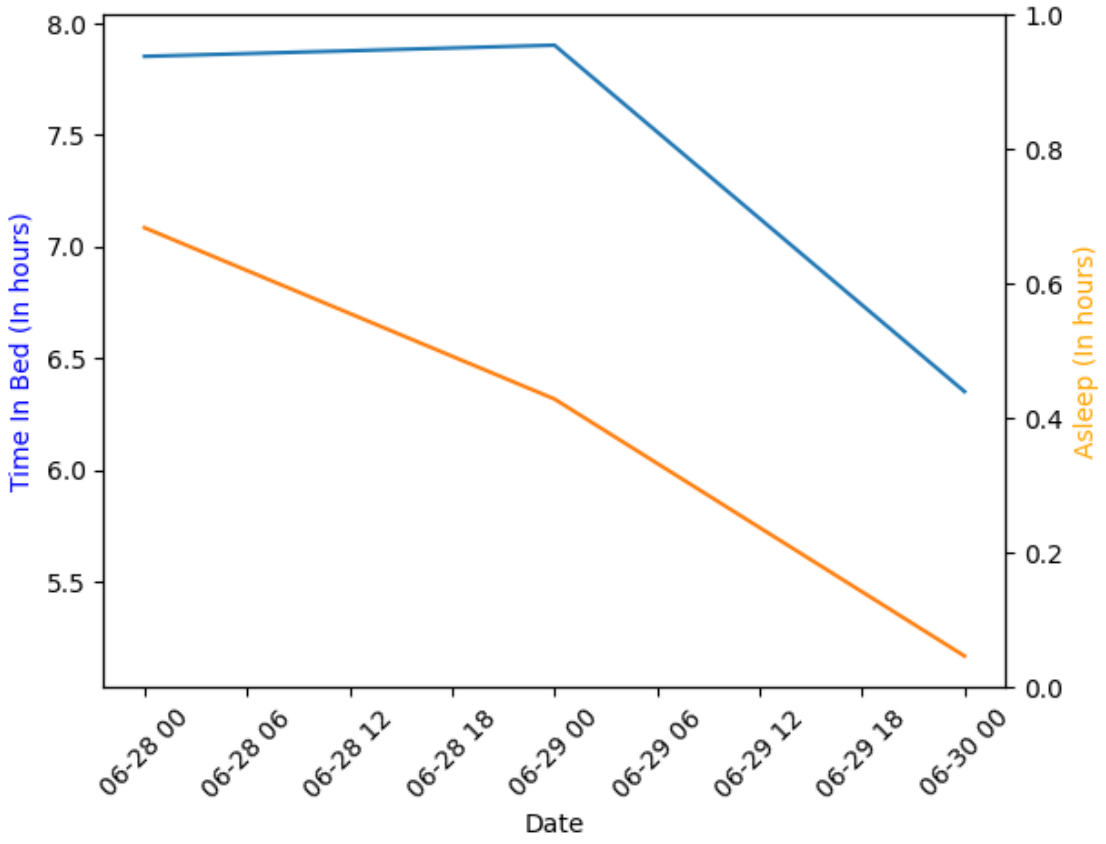
[ ]:
```
june = sleep[sleep['date'].dt.month == 6]

fig, ax = plt.subplots()
ax.plot(june['date'], june['timeInBed'])
ax.plot(june['date'], june['asleep'])
ax.set_xlabel('Date')
ax.set_ylabel('Time In Bed (In hours)', color='blue')

ax2 = ax.twinx()
ax2.set_ylabel('Asleep (In hours)', color='orange')

ax.tick_params(axis='x', rotation=45)
ax2.tick_params(axis='x', rotation=45)
```

```
plt.show()
```



July

[ ]:
```
july = sleep[sleep['date'].dt.month == 7]

fig, ax = plt.subplots()
ax.plot(july['date'], july['timeInBed'])
ax.plot(july['date'], july['asleep'])
ax.set_xlabel('Date')
ax.set_ylabel('Time In Bed (In hours)', color='blue')
ax2 = ax.twinx()
ax2.set_ylabel('Asleep (In hours)', color='orange')

ax.tick_params(axis='x', rotation=45)
ax2.tick_params(axis='x', rotation=45)
plt.show()
```

Define a plot_timeseries function to get monthly times series plots more efficiently!

[ ]:
```
def plot_timeseries(ax, x, y, xlabel, ylabel, color):
    ax.plot(x, y, label=ylabel, color=color)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel, color=color)
    ax.tick_params(axis='y', labelcolor=color)
    ax.tick_params(axis='x', rotation=45)
```

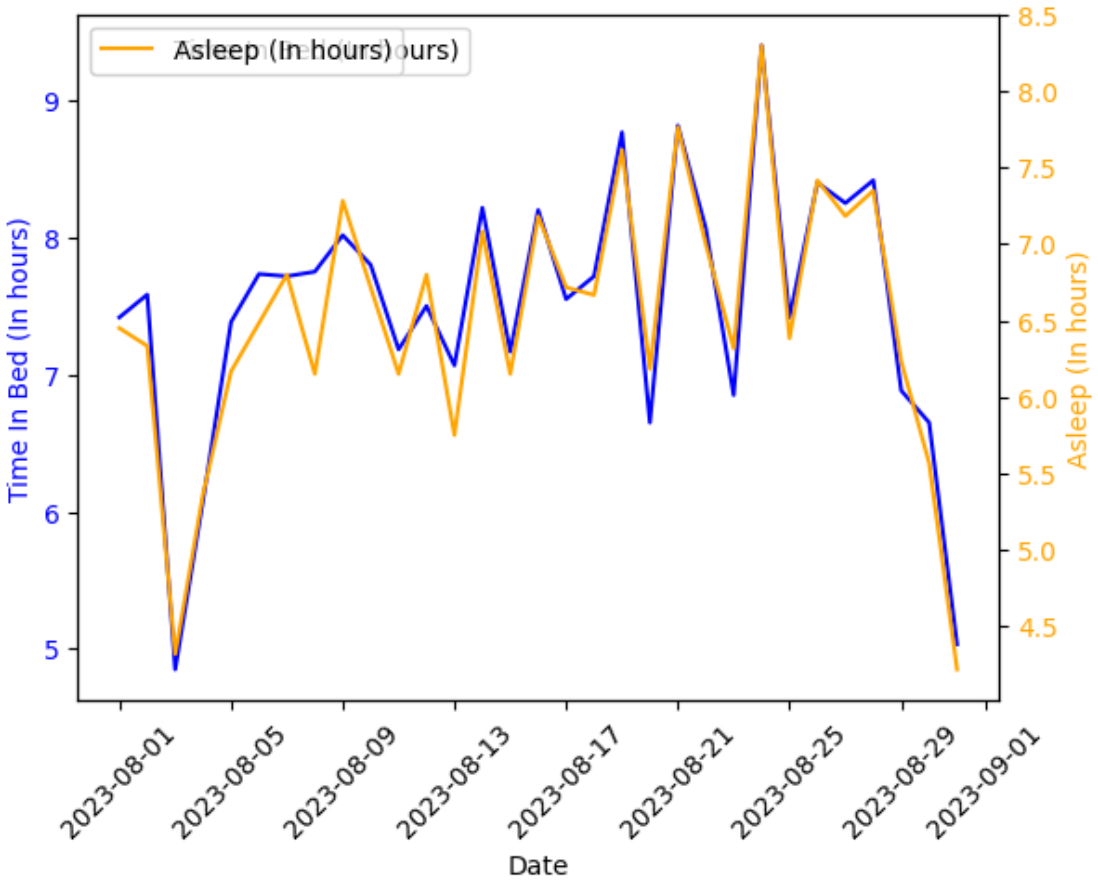
August

[ ]:
```
august = sleep[sleep['date'].dt.month == 8]

fig, ax = plt.subplots()
plot_timeseries(ax, august['date'], august['timeInBed'], 'Date', 'Time In Bed␣
↪(In hours)', color='blue')

ax2 = ax.twinx()
plot_timeseries(ax2, august['date'], august['asleep'], 'Date', 'Asleep (In␣
↪hours)', color='orange')
```

```
plt.show()
```

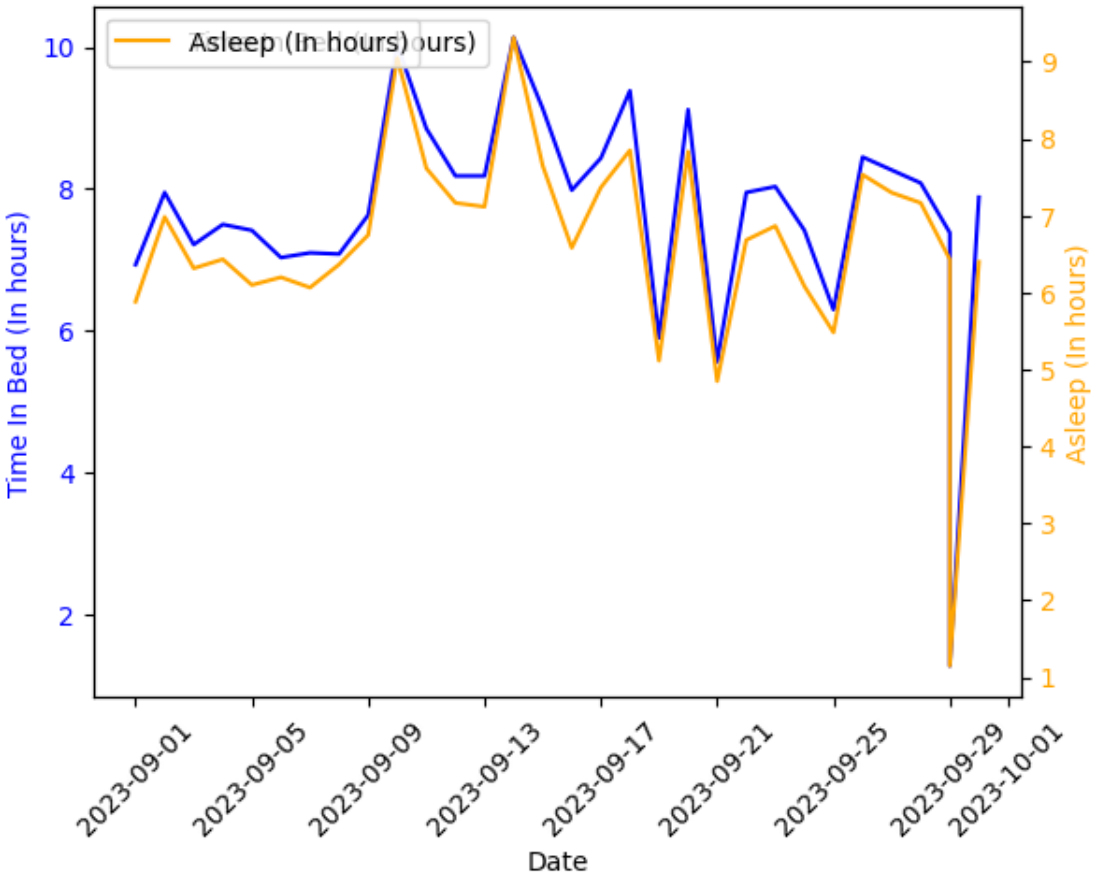


September

[ ]:
```
sep = sleep[sleep['date'].dt.month == 9]
```

[ ]:
```
fig, ax = plt.subplots()
plot_timeseries(ax, sep['date'], sep['timeInBed'], xlabel='Date', ylabel='Time␣
↪In Bed (In hours)', color='blue')

ax2 = plt.twinx()
plot_timeseries(ax2, sep['date'], sep['asleep'], xlabel='Date', ylabel='Asleep␣
↪(In hours)', color='orange')
plt.show()
```


Asleep (In hours)
Time In Bed (In hours)
Asleep (In hours)
Date
2023-09-01
2023-09-05
2023-09-09
2023-09-13
2023-09-17
2023-09-21
2023-09-25
2023-09-29
2023-10-01