In [1]:
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
plt.style.use('ggplot')

#!pip install plotly
#!pip install chart_studio
#!pip install cufflinks
import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import iplot, init_notebook_mode
import cufflinks
cufflinks.go_offline(connected = True)
init_notebook_mode(connected = True)
```

In [2]:
```
files = ('0418.csv','0518.csv','0618.csv','0718.csv','0818.csv','0918.csv',
df = pd.DataFrame()
for file in files:
    df_month = pd.read_csv(file)
    df = pd.concat([df,df_month])
```

In [3]:
```
df = df[['Start Time','Activity','Duration (min)','Quantity','Caregiver']]
df['timestamp'] = pd.to_datetime(df['Start Time'])
cols = df.columns.tolist()
cols = cols[-1:] + cols [1:5]
print(cols)
df = df[cols]
df.rename(columns = {'Duration (min)' : 'minutes', 'Caregiver' : 'Room'}, i
df.set_index('timestamp', inplace = True)
df.sort_index(inplace = True)
print (df.shape)
df.head()
```

```
['timestamp', 'Activity', 'Duration (min)', 'Quantity', 'Caregiver']
(5802, 4)
```

Out[3]:

timestamp	Activity	minutes	Quantity	Room
2018-04-04 09:00:00	Sign In	NaN	NaN	Infant E
2018-04-04 09:05:00	Diaper	NaN	NaN	Infant E
2018-04-04 09:20:00	Bottle	NaN	4.0	Infant E
2018-04-04 09:38:00	Photo	NaN	NaN	Infant E
2018-04-04 09:39:00	Sleep	22.0	NaN	Infant E

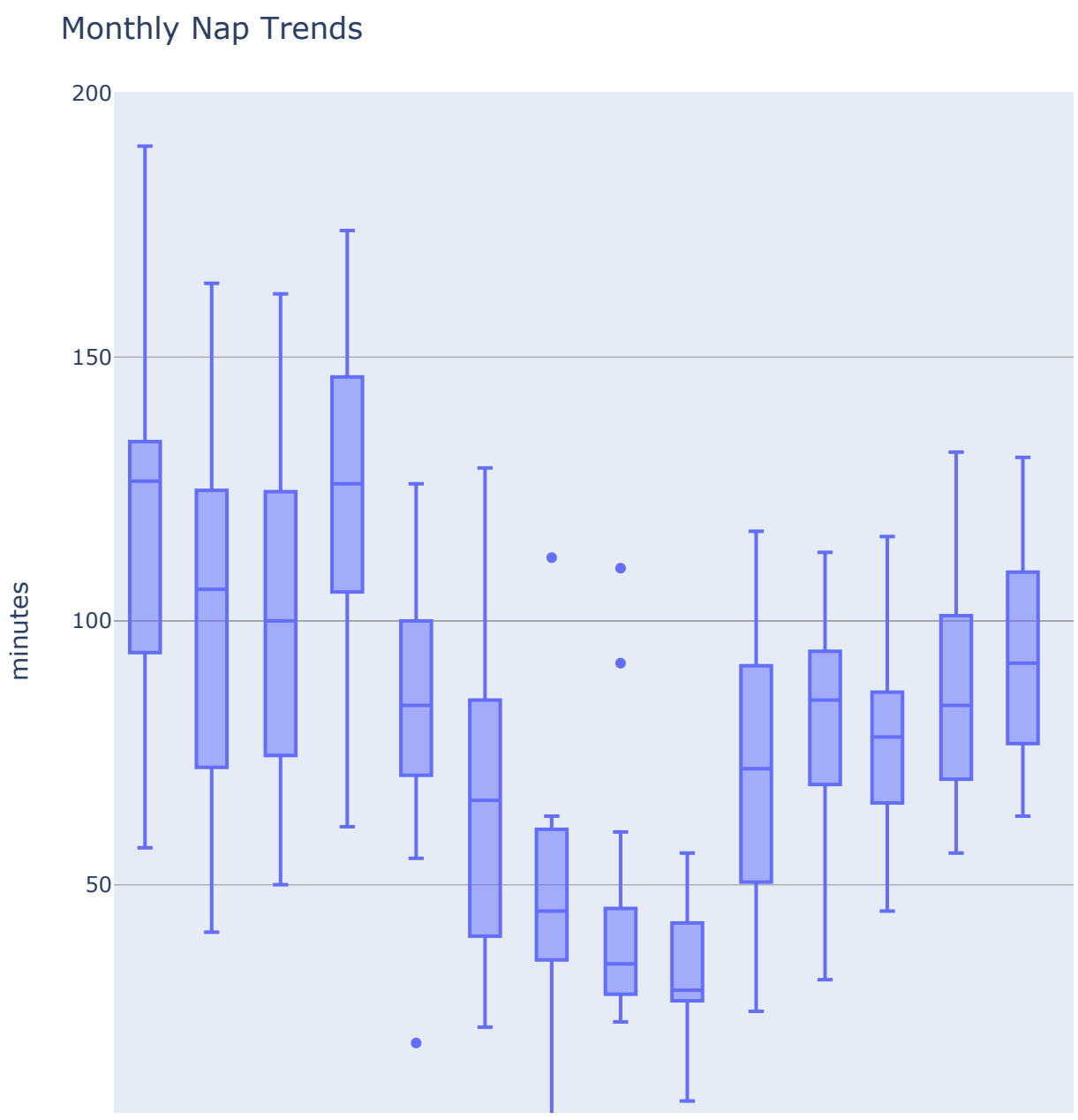
In [4]:
```
df_sleep = df[df['Activity'] == 'Sleep']
df_sleep = df_sleep[df_sleep['minutes'] < 200]
#df_sleep.head(10)
```

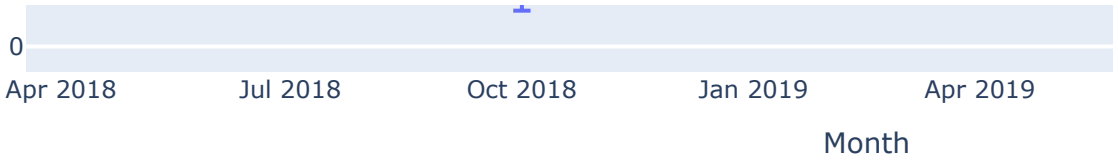
In [5]:
```
#import plotly express
import plotly.express as px

#daily total and remove incorrect entries and days with no sleep (no school
df_sleep_daily = df_sleep.resample('d', how = np.sum)
df_sleep_daily = df_sleep_daily[(df_sleep_daily['minutes'] > 0)]
#assign each row month of the year
df_sleep_daily['Month'] = df_sleep_daily.index.to_period ('M')
df_sleep_daily['Month'] = df_sleep_daily['Month'].astype(str)
fig = px.box(df_sleep_daily, x="Month", y="minutes", width = 1000, height =
fig.update_layout(title = 'Monthly Nap Trends')
fig.show()
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:5: FutureWa
rning:

how in .resample() is deprecated
the new syntax is .resample(...)..apply(<func>)
```





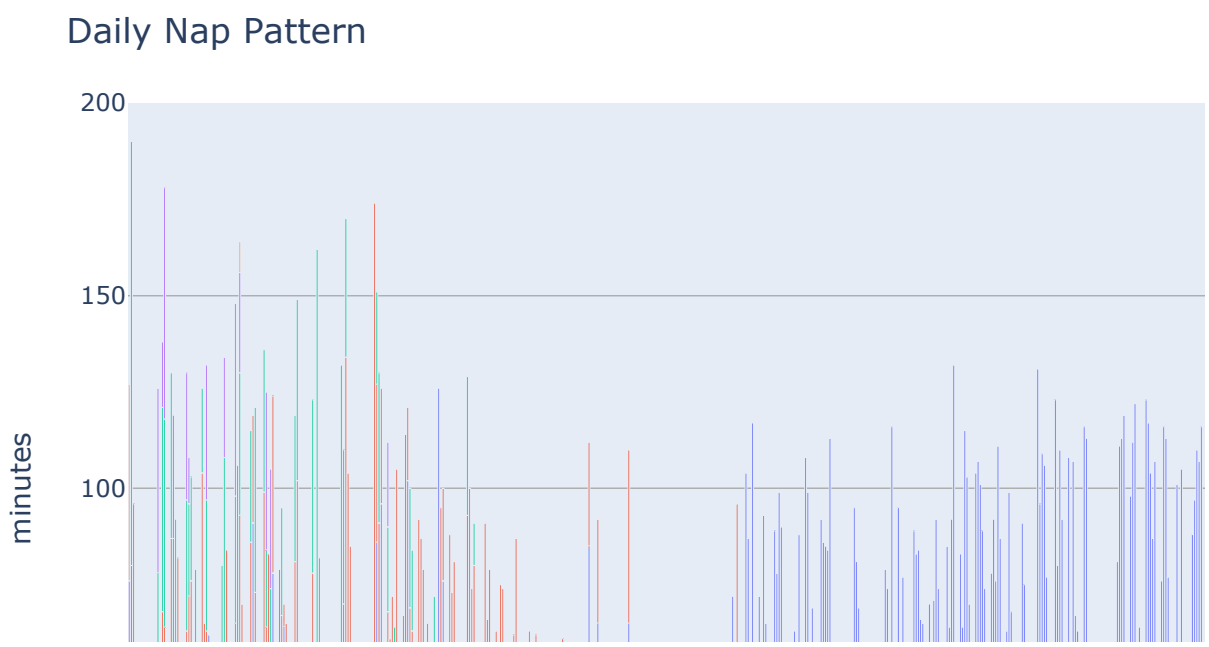
How do daily sleep patterns look? A stacked bar chart showing total naps for each day will allow for visualization of both total nap time and pattern.

In [6]:
```
#cumulative count of naps per day
c = df_sleep.groupby(["Activity",df_sleep.index.date]).cumcount() + 1
c = c.replace(0, '').astype(str)
c.head(10)
df_sleep["Activity"] += c
df_sleep.head()
```

Out[6]:

timestamp	Activity	minutes	Quantity	Room
2018-04-04 09:39:00	Sleep1	22.0	NaN	Infant E
2018-04-04 10:13:00	Sleep2	19.0	NaN	Infant E
2018-04-04 11:07:00	Sleep3	13.0	NaN	Infant E
2018-04-04 12:38:00	Sleep4	22.0	NaN	Infant E
2018-04-04 14:46:00	Sleep5	51.0	NaN	Infant E

In [7]:
```
fig = px.bar(df_sleep, x=df_sleep.index.date, y="minutes", color='Activity'
fig.update_layout(title="Daily Nap Pattern", xaxis_title="Date")
fig.show()
```



If we add start and end time, that will allow for sleep schedule visualization, along with previously visualized time and pattern. A 3D bar chart (x = date, y = duration, z = start time) will facilitate this.

In [8]:
```
#set date and time as spearate columns
df_sleep['Date'] = df_sleep.index.date
df_sleep['Start'] = df_sleep.index.time
#df_sleep.reset_index(inplace = True)
#df_sleep.drop(['timestamp'], axis = 1, inplace = True)
df_sleep = df_sleep[['Date','Start','minutes','Activity']]
df_sleep.head()

#save index to insert later
index = df_sleep.index
```

In [9]:
```
#number unique dates to set position as x-axis
unique_dates = pd.DataFrame(df_sleep['Date'].unique())
unique_dates.reset_index(inplace = True)
unique_dates.rename(columns = {'index': 'x position', 0: 'Date'}, inplace =

df_sleep = pd.merge(df_sleep, unique_dates, on = 'Date')
```

In [10]:
```
df_sleep.head()
```

Out[10]:

	Date	Start	minutes	Activity	x position
0	2018-04-04	09:39:00	22.0	Sleep1	0
1	2018-04-04	10:13:00	19.0	Sleep2	0
2	2018-04-04	11:07:00	13.0	Sleep3	0
3	2018-04-04	12:38:00	22.0	Sleep4	0
4	2018-04-04	14:46:00	51.0	Sleep5	0

In [11]:
```
unique_times = pd.DataFrame(df_sleep['Start'].unique())
unique_times.sort_values(by = 0, inplace = True)
unique_times.reset_index(drop = True, inplace = True)
unique_times.reset_index(inplace = True)
unique_times.rename(columns = {'index': 'y position', 0: 'Start'}, inplace
```

In [12]:
```
df_sleep.sort_values(by = "Start", inplace = True)
df_sleep = pd.merge(df_sleep, unique_times, on = 'Start')

df_sleep.sort_values(['x position', 'y position'], inplace = True)
#df_sleep.set_index(index, inplace = True)

#remove sleep form activity column for multiple colored bars
df_sleep['Activity'] = df_sleep['Activity'].str.replace('Sleep', '')
df_sleep['Activity'] = df_sleep['Activity'].astype(int)
df_sleep.head()
```

Out[12]:

	Date	Start	minutes	Activity	x position	y position
27	2018-04-04	09:39:00	22.0	1	0	23
54	2018-04-04	10:13:00	19.0	2	0	40
104	2018-04-04	11:07:00	13.0	3	0	73
373	2018-04-04	12:38:00	22.0	4	0	134
489	2018-04-04	14:46:00	51.0	5	0	206

In [13]:
```
#Here, I am setting the date and time columns as a string so they can be sw
#later on in the 3D
df_sleep['Date'] = df_sleep['Date'].astype(str)
df_sleep['Start'] = df_sleep['Start'].astype(str)
df_sleep.dtypes
```

Out[13]:
```
Date           object
Start          object
minutes       float64
Activity        int64
x position      int64
y position      int64
dtype: object
```

In [14]:
```
#define colors
colors = ['k','royalblue','orangered','mediumspringgreen','blueviolet','ora

#store colors
clrs = []
for n in df_sleep['Activity']:
    c = colors[n]
    clrs.append(c)
```

In [15]:
```
#import 3D plotting, this step is still in development
from mpl_toolkits.mplot3d import Axes3D

#interactive plot
%matplotlib qt

#set bar positions
x = df_sleep ['x position']
y = df_sleep ['y position']
z = np.zeros (529)

#set bar depths
dx = 2*(np.ones (529))
dy = np.ones (529)
dz = df_sleep['minutes']

#initiate figure
fig = plt.figure(figsize = (12,8))
ax1 = fig.add_subplot(111, projection='3d')
ax1.bar3d(x,y,z,dx,dy,dz,alpha = 0.25, color = clrs)

ax1.set_xticklabels(['','April 18','July 18','October 18',' January 19','Ap
ax1.set_xlabel('')

ax1.set_yticklabels(['6 AM','8 AM','10 AM','12 PM','2 PM','4 PM'])
ax1.set_ylabel('Start Time')

ax1.set_zlabel('Duaration (minutes)')

plt.show()
```

```
No handles with labels found to put in legend.
```

In [ ]:

In [ ]:

In [ ]: