In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('sleep_data.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	date	sleep_hours
0	2015-02-19	6.400000
1	2015-02-20	7.583333
2	2015-02-21	6.350000
3	2015-02-22	6.500000
4	2015-02-23	8.916667

In [4]:

```
df.tail()
```

Out[4]:

	date	sleep_hours
2349	2021-12-25	7.933333
2350	2021-12-26	3.850000
2351	2021-12-29	6.175000
2352	2021-12-30	5.158333
2353	2021-12-31	5.908333

In [5]:

```
df.shape
```

Out[5]:

```
(2354, 2)
```

In [6]:

```
df.columns
```

Out[6]:

```
Index(['date', 'sleep_hours'], dtype='object')
```

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
0
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
date           0
sleep_hours    0
dtype: int64
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   date         2354 non-null   object 
 1   sleep_hours  2354 non-null   float64
dtypes: float64(1), object(1)
memory usage: 36.9+ KB
```

In [10]:

```
df.describe()
```

Out[10]:

	sleep_hours
count	2354.000000
mean	7.356560
std	2.213308
min	1.266667
25%	6.235417
50%	6.816667
75%	7.483333
max	17.433333

In [11]:

```
df.nunique()
```

Out[11]:

```
date           2354
sleep_hours     604
dtype: int64
```

In [12]:

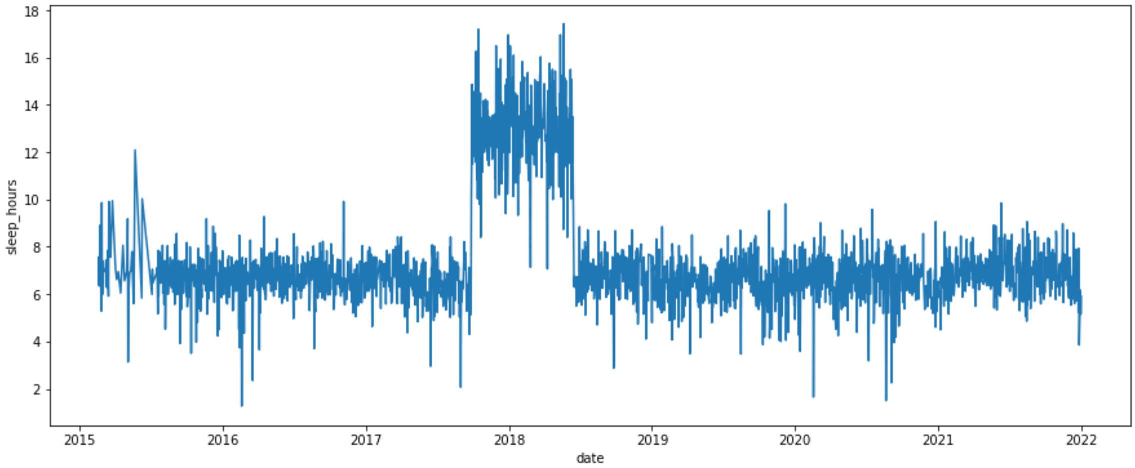
```
df["date"]= pd.to_datetime(df["date"])
```

In [13]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from IPython import get_ipython
import warnings
warnings.filterwarnings("ignore")
```

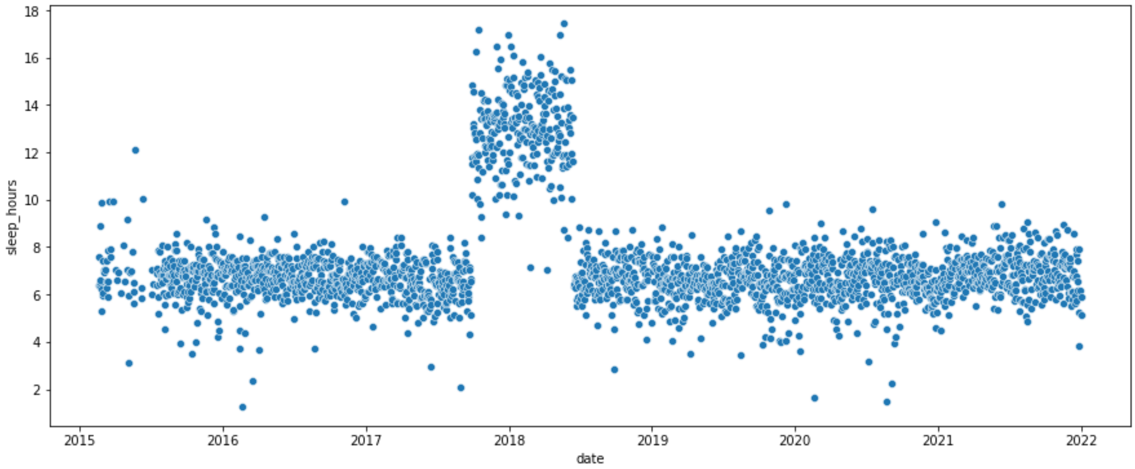
In [14]:

```
plt.figure(figsize=(15,6))
sns.lineplot(x = df['date'], y = df['sleep_hours'],
             data = df, palette = 'hls')
plt.show()
```



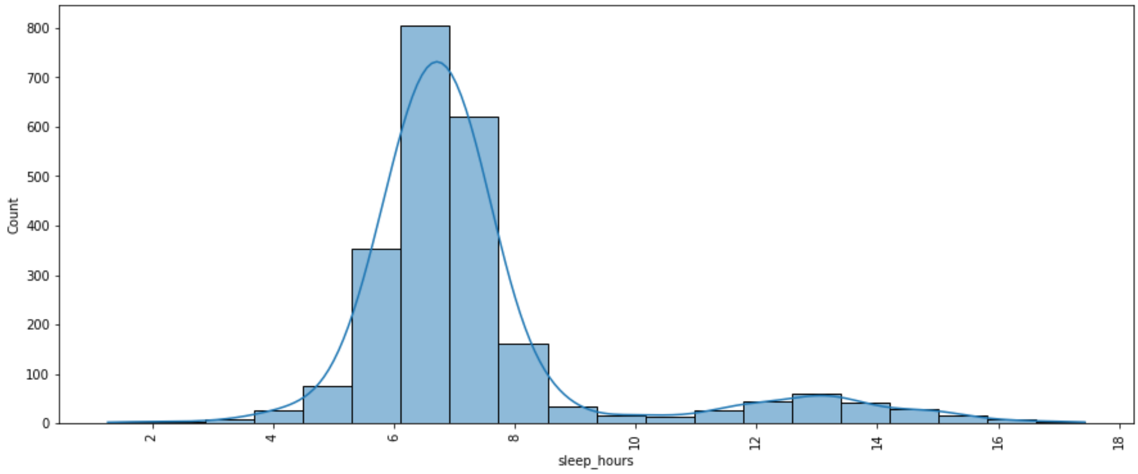
In [15]:

```
plt.figure(figsize=(15,6))
sns.scatterplot(x = df['date'], y = df['sleep_hours'],
             data = df, palette = 'hls')
plt.show()
```



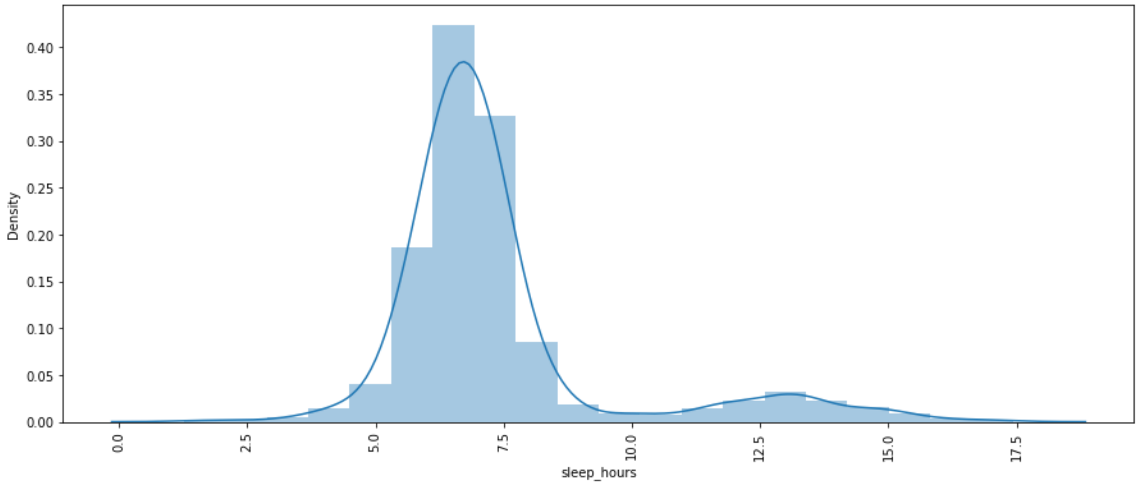
In [16]:

```
plt.figure(figsize=(15,6))
sns.histplot(df['sleep_hours'], bins = 20, kde = True, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



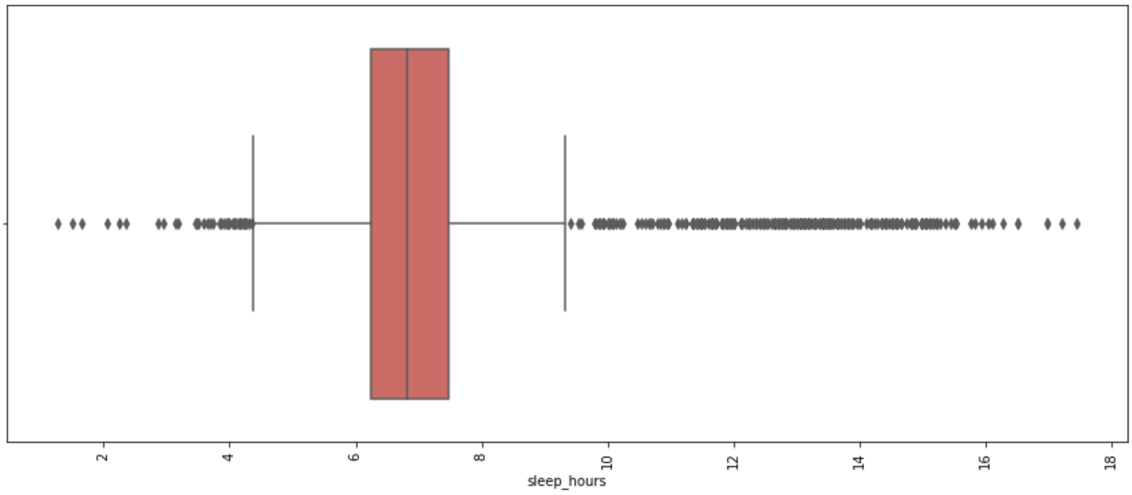
In [17]:

```
plt.figure(figsize=(15,6))
sns.distplot(df['sleep_hours'], bins = 20, kde = True)
plt.xticks(rotation = 90)
plt.show()
```



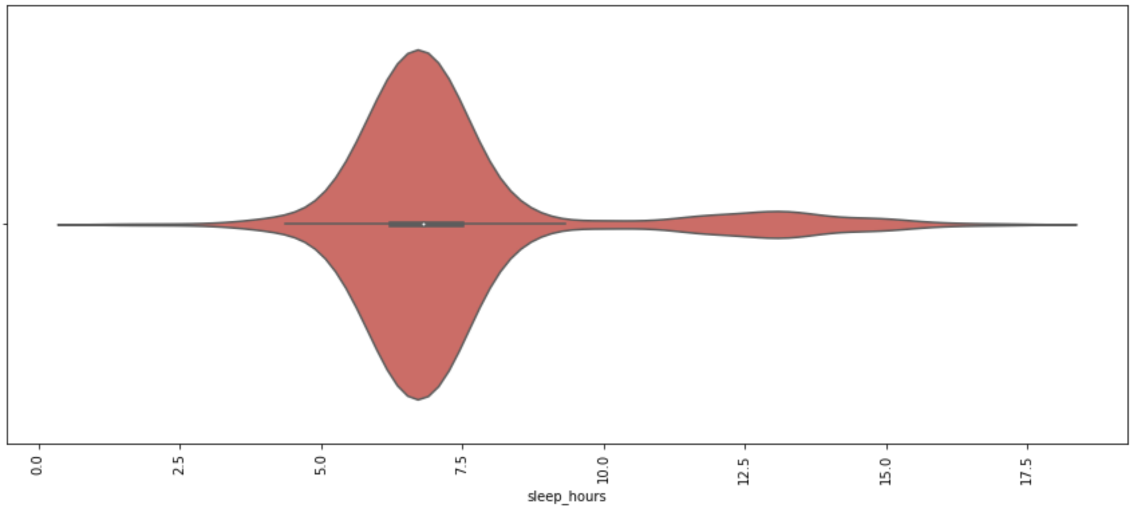
In [18]:

```
plt.figure(figsize=(15,6))
sns.boxplot(df['sleep_hours'], palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [19]:

```
plt.figure(figsize=(15,6))
sns.violinplot(df['sleep_hours'], palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [20]:

```
df = df.set_index('date')
```

In [21]:

```
df1 = pd.read_csv('sleep_data_new.csv')
```

In [22]:

```
df1.head()
```

Out[22]:

	type	sourceName	creationDate	startDate	endDate	
0	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	19-04-2015 09:42	20-02-2015 01:45	20-02-2015 08:09	HKCat
1	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	19-04-2015 09:42	20-02-2015 01:39	20-02-2015 08:46	
2	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	19-04-2015 09:42	21-02-2015 01:59	21-02-2015 09:34	HKCat
3	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	19-04-2015 09:42	21-02-2015 01:52	21-02-2015 09:57	
4	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	19-04-2015 09:42	22-02-2015 02:50	22-02-2015 09:11	HKCat

In [23]:

```
df1.tail()
```

Out[23]:

	type	sourceName	creationDate	startDate	endDate	
8029	HKCategoryTypeIdentifierSleepAnalysis	Rob’s Apple Watch	31-12-2021 09:04	31-12-2021 04:25	31-12-2021 05:28	HK
8030	HKCategoryTypeIdentifierSleepAnalysis	Rob’s Apple Watch	31-12-2021 09:04	31-12-2021 05:31	31-12-2021 05:54	HK
8031	HKCategoryTypeIdentifierSleepAnalysis	Rob’s Apple Watch	31-12-2021 09:04	31-12-2021 06:22	31-12-2021 06:30	HK
8032	HKCategoryTypeIdentifierSleepAnalysis	Rob’s iPhone	31-12-2021 09:04	31-12-2021 02:06	31-12-2021 09:04	
8033	HKCategoryTypeIdentifierSleepAnalysis	Rob’s Apple Watch	31-12-2021 09:04	31-12-2021 06:37	31-12-2021 09:05	HK

In [24]:

```
df1.shape
```

Out[24]:

```
(8034, 8)
```

In [25]:

```
df1.columns
```

Out[25]:

```
Index(['type', 'sourceName', 'creationDate', 'startDate', 'endDate', 'valu
e',
       'sourceVersion', 'device'],
      dtype='object')
```

In [26]:

```
df1.duplicated().sum()
```

Out[26]:

```
44
```

In [27]:

```
df1 = df1.drop_duplicates()
```

In [28]:

```
df1.isnull().sum()
```

Out[28]:

```
type                0
sourceName          0
creationDate        0
startDate           0
endDate             0
value               0
sourceVersion     272
device           7876
dtype: int64
```

In [29]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7990 entries, 0 to 8033
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   type           7990 non-null   object
 1   sourceName     7990 non-null   object
 2   creationDate   7990 non-null   object
 3   startDate      7990 non-null   object
 4   endDate        7990 non-null   object
 5   value          7990 non-null   object
 6   sourceVersion  7718 non-null   object
 7   device         114 non-null    object
dtypes: object(8)
memory usage: 561.8+ KB
```

In [30]:

```
df1.nunique()
```

Out[30]:

```
type                1
sourceName          4
creationDate      630
startDate        5653
endDate          7240
value               2
sourceVersion      29
device              5
dtype: int64
```

In [31]:

```
df1['sourceName'].unique()
```

Out[31]:

```
array(['Sync Solver', 'Clock', 'Rob’s Apple\xa0Watch', 'Rob’s iPhone'],
      dtype=object)
```

In [32]:

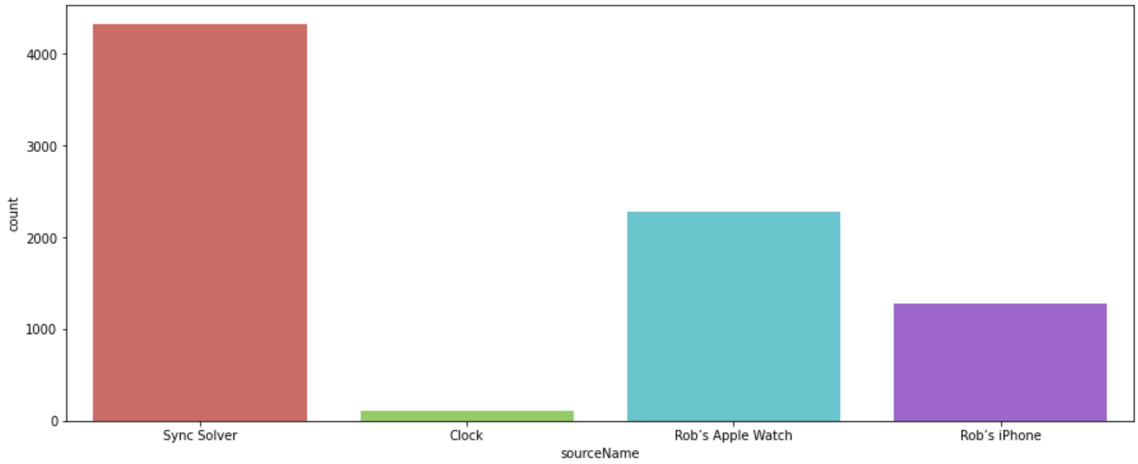
```
df1['sourceName'].value_counts()
```

Out[32]:

```
Sync Solver          4320
Rob’s Apple Watch    2282
Rob’s iPhone         1274
Clock                 114
Name: sourceName, dtype: int64
```

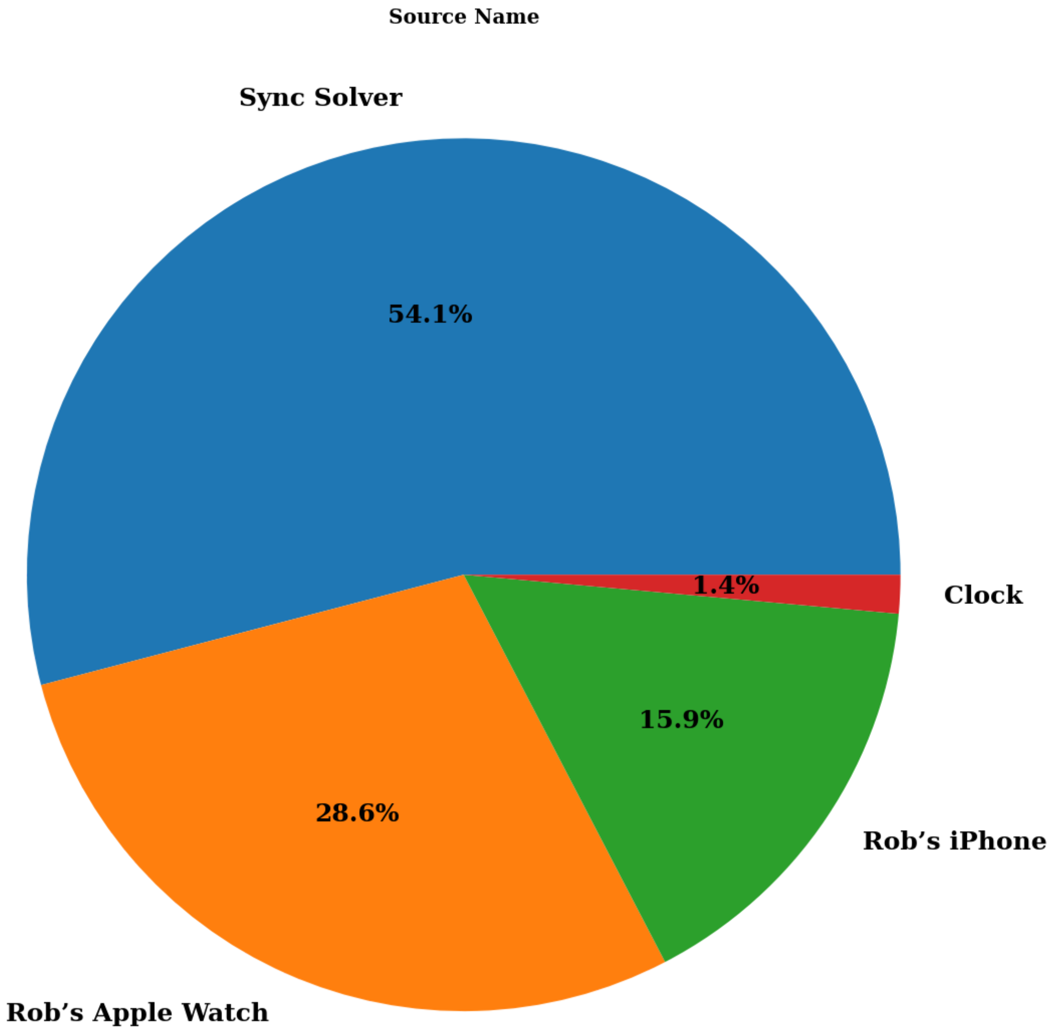
In [33]:

```
plt.figure(figsize=(15,6))
sns.countplot('sourceName', data = df1, palette = 'hls')
plt.show()
```



In [34]:

```
plt.figure(figsize=(30,20))
plt.pie(df1['sourceName'].value_counts(), labels=df1['sourceName'].value_counts().index,
        autopct='%1.1f%%', textprops={ 'fontsize': 25,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Source Name', size=20, **hfont)
plt.show()
```



In [35]:

```
df1['value'].unique()
```

Out[35]:

```
array(['HKCategoryValueSleepAnalysisAsleepUnspecified',
       'HKCategoryValueSleepAnalysisInBed'], dtype=object)
```

In [36]:

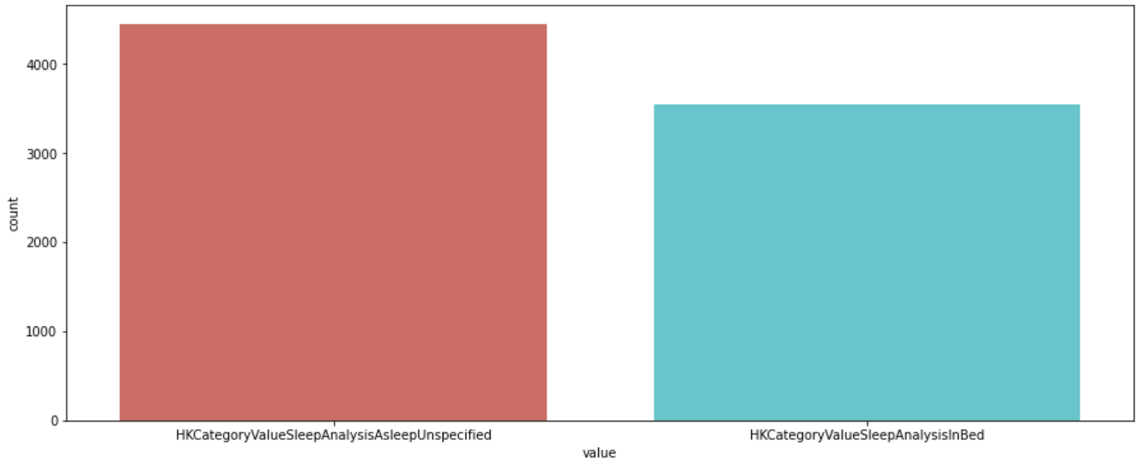
```
df1['value'].value_counts()
```

Out[36]:

```
HKCategoryValueSleepAnalysisAsleepUnspecified    4441
HKCategoryValueSleepAnalysisInBed                3549
Name: value, dtype: int64
```

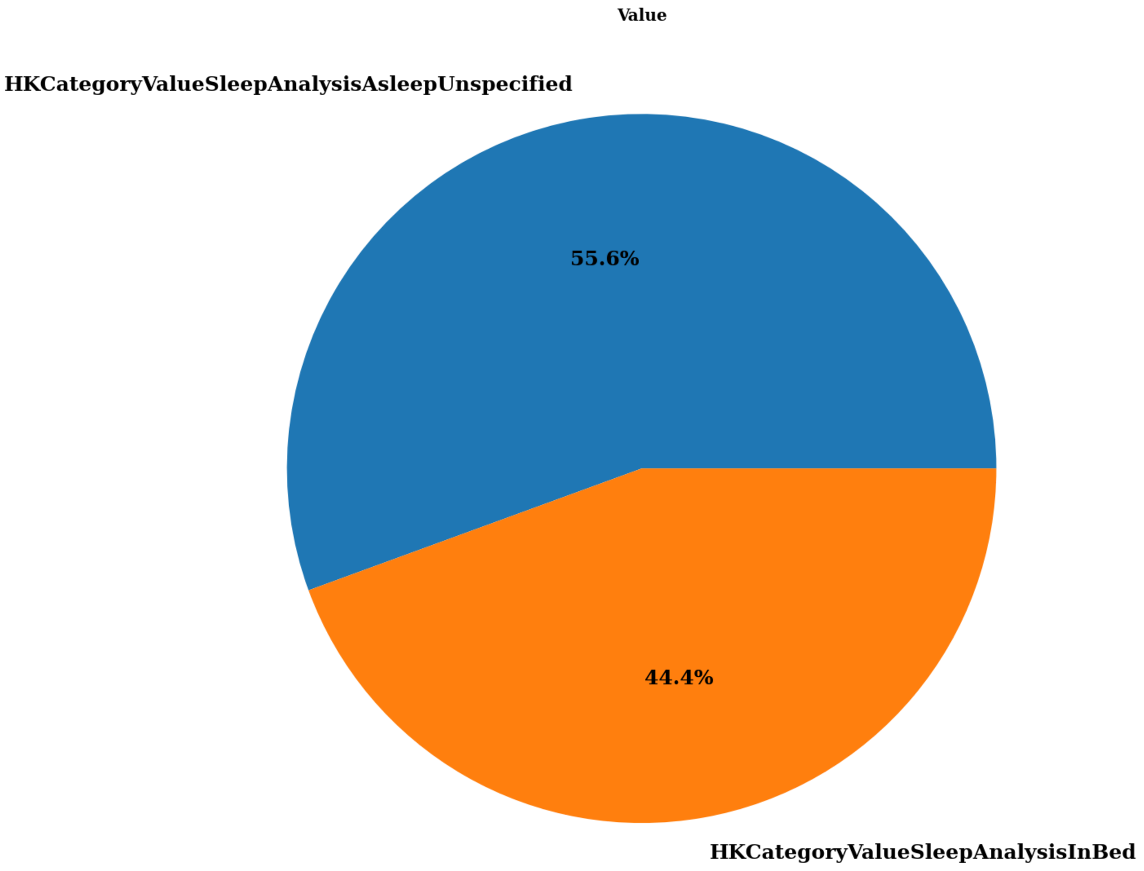
In [37]:

```
plt.figure(figsize=(15,6))
sns.countplot('value', data = df1, palette = 'hls')
plt.show()
```



In [38]:

```
plt.figure(figsize=(30,20))
plt.pie(df1['value'].value_counts(), labels=df1['value'].value_counts().index,
        autopct='%1.1f%%', textprops={ 'fontsize': 25,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Value', size=20, **hfont)
plt.show()
```



In [39]:

```
df1['device'].unique()
```

Out[39]:

```
array([nan,
       '<<HKDevice: 0x2807b0f50>, name:iPhone, manufacturer:Apple, model:i
Phone, hardware:iPhone9,1, software:12.0>',
       '<<HKDevice: 0x2807b0870>, name:iPhone, manufacturer:Apple, model:i
Phone, hardware:iPhone9,1, software:12.0.1>',
       '<<HKDevice: 0x2807ac1e0>, name:iPhone, manufacturer:Apple, model:i
Phone, hardware:iPhone9,1, software:12.1>',
       '<<HKDevice: 0x2807ac140>, name:iPhone, manufacturer:Apple, model:i
Phone, hardware:iPhone9,1, software:12.1.2>',
       '<<HKDevice: 0x2807ad900>, name:iPhone, manufacturer:Apple, model:i
Phone, hardware:iPhone9,1, software:12.1.2>'],
      dtype=object)
```

In [40]:

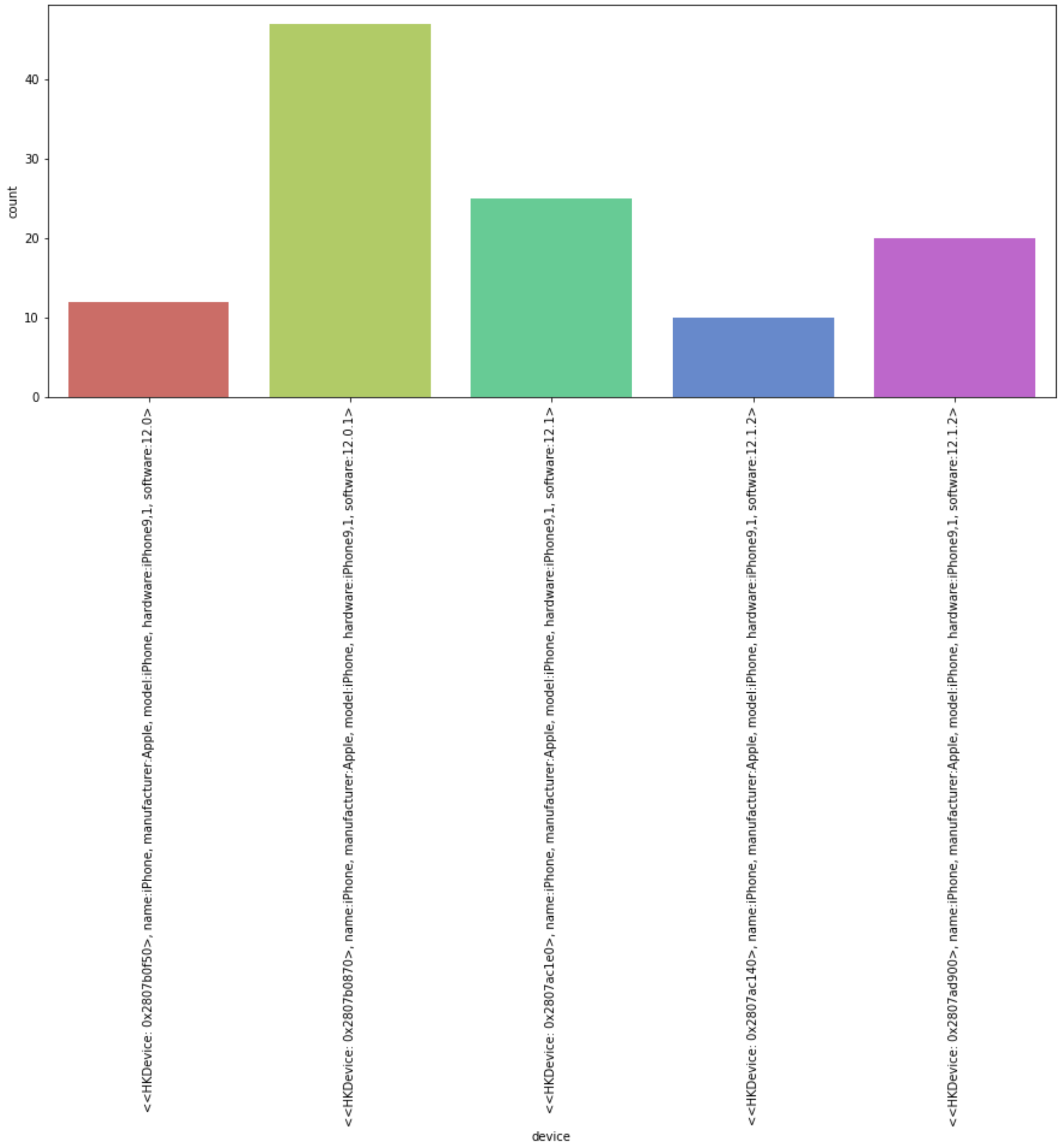
```
df1['device'].value_counts()
```

Out[40]:

```
<<HKDevice: 0x2807b0870>, name:iPhone, manufacturer:Apple, model:iPhone, h
ardware:iPhone9,1, software:12.0.1>    47
<<HKDevice: 0x2807ac1e0>, name:iPhone, manufacturer:Apple, model:iPhone, h
ardware:iPhone9,1, software:12.1>      25
<<HKDevice: 0x2807ad900>, name:iPhone, manufacturer:Apple, model:iPhone, h
ardware:iPhone9,1, software:12.1.2>    20
<<HKDevice: 0x2807b0f50>, name:iPhone, manufacturer:Apple, model:iPhone, h
ardware:iPhone9,1, software:12.0>      12
<<HKDevice: 0x2807ac140>, name:iPhone, manufacturer:Apple, model:iPhone, h
ardware:iPhone9,1, software:12.1.2>    10
Name: device, dtype: int64
```

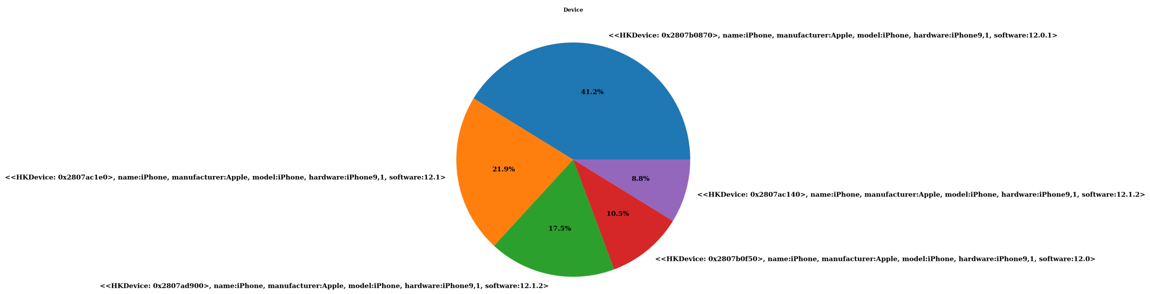
In [41]:

```
plt.figure(figsize=(15,6))
sns.countplot('device', data = df1, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [42]:

```
plt.figure(figsize=(30,20))
plt.pie(df1['device'].value_counts(), labels=df1['device'].value_counts().index,
        autopct='%1.1f%%', textprops={ 'fontsize': 25,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Device', size=20, **hfont)
plt.show()
```



In [43]:

```
df1["creationDate"]= pd.to_datetime(df1["creationDate"])
df1["startDate"]= pd.to_datetime(df1["startDate"])
df1["endDate"]= pd.to_datetime(df1["endDate"])
```

In [44]:

```
df1['Duration'] = df1["endDate"] - df1["startDate"]
```

In [45]:

```
df1
```

Out[45]:

	type	sourceName	creationDate	startDate	endDate	
0	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-20 01:45:00	2015-02-20 08:09:00	HK
1	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-20 01:39:00	2015-02-20 08:46:00	
2	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-21 01:59:00	2015-02-21 09:34:00	HK
3	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-21 01:52:00	2015-02-21 09:57:00	
4	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-22 02:50:00	2015-02-22 09:11:00	HK
...	...	...	...	...	...	
8029	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 04:25:00	2021-12-31 05:28:00	HK
8030	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 05:31:00	2021-12-31 05:54:00	HK
8031	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 06:22:00	2021-12-31 06:30:00	HK
8032	HKCategoryTypeIdentifierSleepAnalysis	Rob's iPhone	2021-12-31 09:04:00	2021-12-31 02:06:00	2021-12-31 09:04:00	
8033	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 06:37:00	2021-12-31 09:05:00	HK

7990 rows × 9 columns

In [46]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7990 entries, 0 to 8033
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   type           7990 non-null   object
 1   sourceName     7990 non-null   object
 2   creationDate   7990 non-null   datetime64[ns]
 3   startDate      7990 non-null   datetime64[ns]
 4   endDate        7990 non-null   datetime64[ns]
 5   value          7990 non-null   object
 6   sourceVersion  7718 non-null   object
 7   device         114 non-null    object
 8   Duration       7990 non-null   timedelta64[ns]
dtypes: datetime64[ns](3), object(5), timedelta64[ns](1)
memory usage: 624.2+ KB
```

In [47]:

```
to_hours = lambda x: x.total_seconds() / 3600
df1['hours'] = df1['Duration'].apply(to_hours)
```

In [48]:

```
df1
```

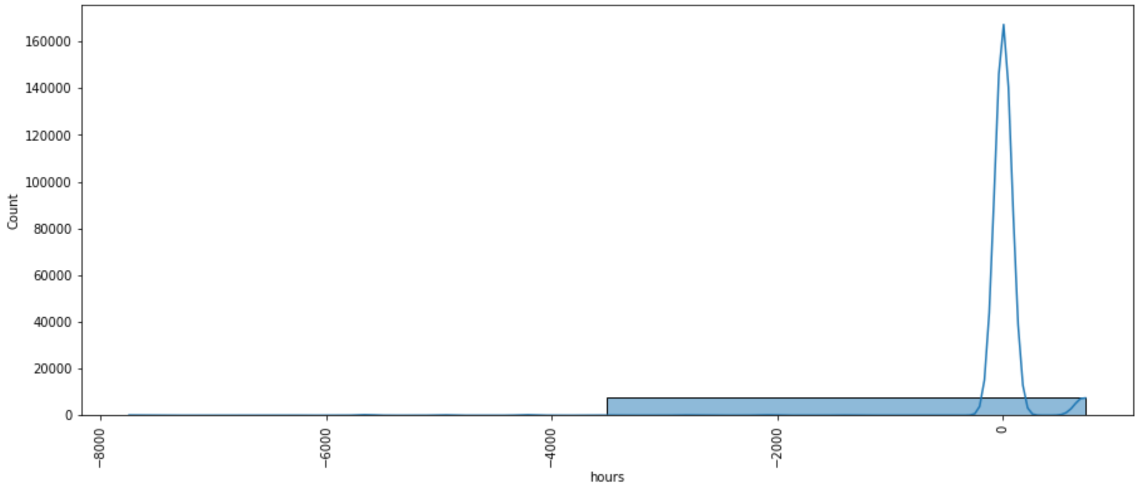
Out[48]:

	type	sourceName	creationDate	startDate	endDate	
0	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-20 01:45:00	2015-02-20 08:09:00	HK
1	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-20 01:39:00	2015-02-20 08:46:00	
2	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-21 01:59:00	2015-02-21 09:34:00	HK
3	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-21 01:52:00	2015-02-21 09:57:00	
4	HKCategoryTypeIdentifierSleepAnalysis	Sync Solver	2015-04-19 09:42:00	2015-02-22 02:50:00	2015-02-22 09:11:00	HK
...	...	...	...	...	...	
8029	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 04:25:00	2021-12-31 05:28:00	HK
8030	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 05:31:00	2021-12-31 05:54:00	HK
8031	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 06:22:00	2021-12-31 06:30:00	HK
8032	HKCategoryTypeIdentifierSleepAnalysis	Rob's iPhone	2021-12-31 09:04:00	2021-12-31 02:06:00	2021-12-31 09:04:00	
8033	HKCategoryTypeIdentifierSleepAnalysis	Rob's Apple Watch	2021-12-31 09:04:00	2021-12-31 06:37:00	2021-12-31 09:05:00	HK

7990 rows × 10 columns

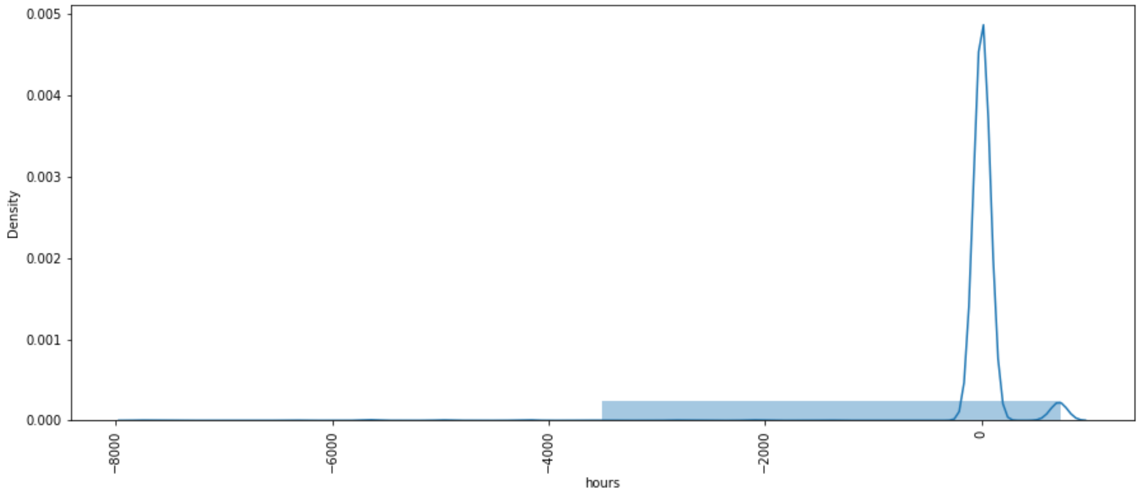
In [49]:

```
plt.figure(figsize=(15,6))
sns.histplot(df1['hours'], bins = 2, kde = True, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



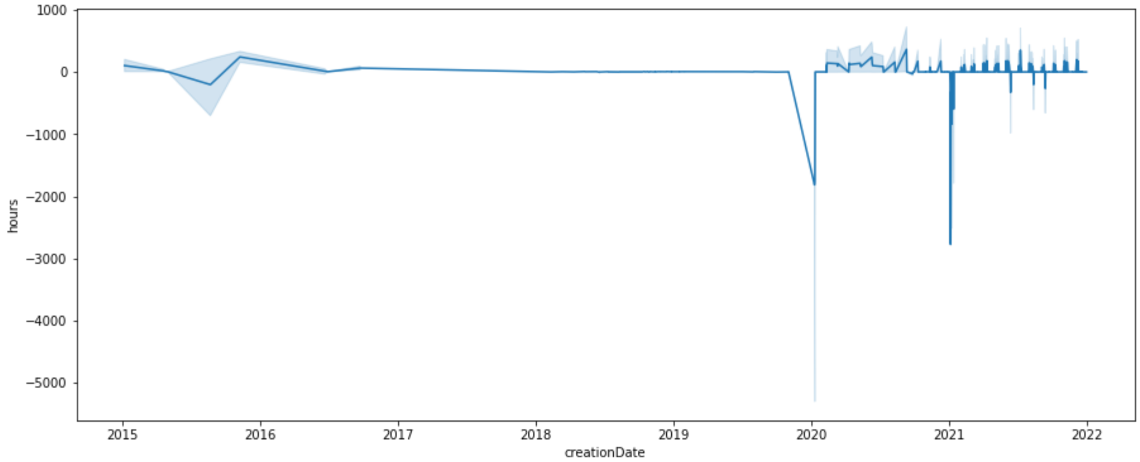
In [50]:

```
plt.figure(figsize=(15,6))
sns.distplot(df1['hours'], bins = 2, kde = True)
plt.xticks(rotation = 90)
plt.show()
```



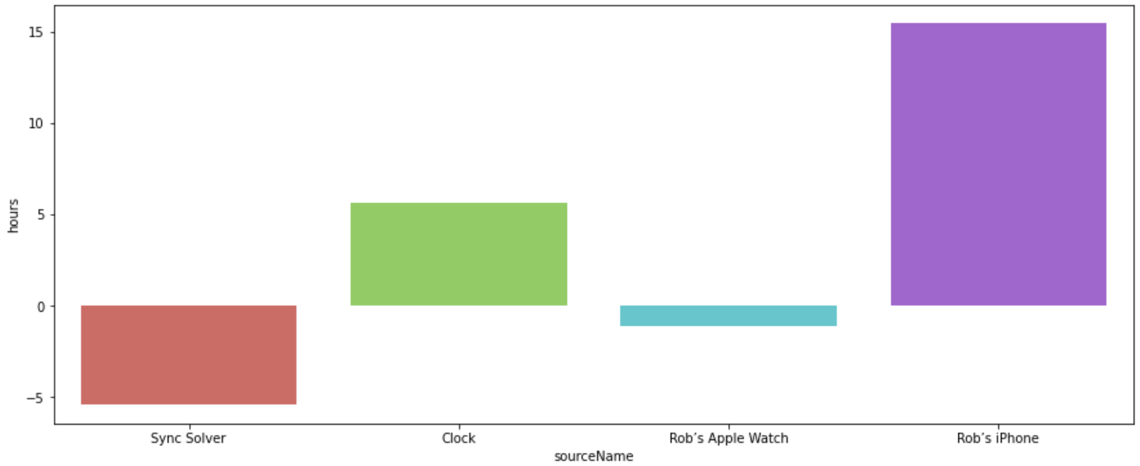
In [51]:

```
plt.figure(figsize=(15,6))
sns.lineplot(x = df1['creationDate'], y = df1['hours'],
             data = df1, palette = 'hls')
plt.show()
```



In [52]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = df1['sourceName'], y = df1['hours'],
             data = df1, ci = None, palette = 'hls')
plt.show()
```

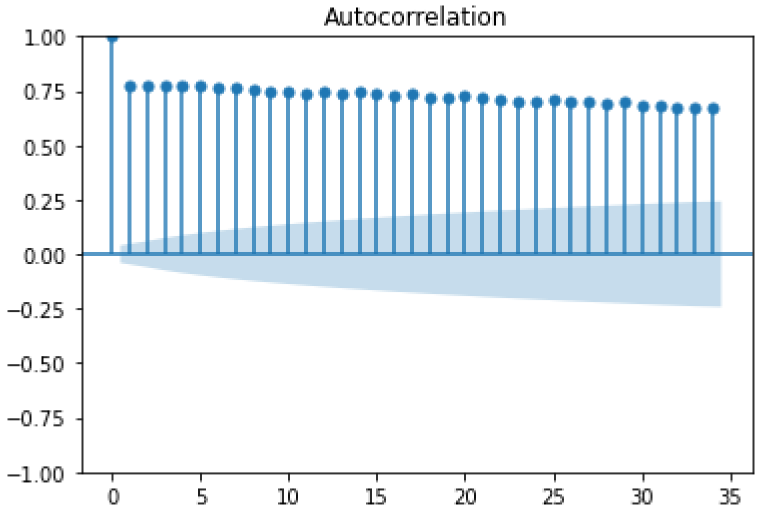


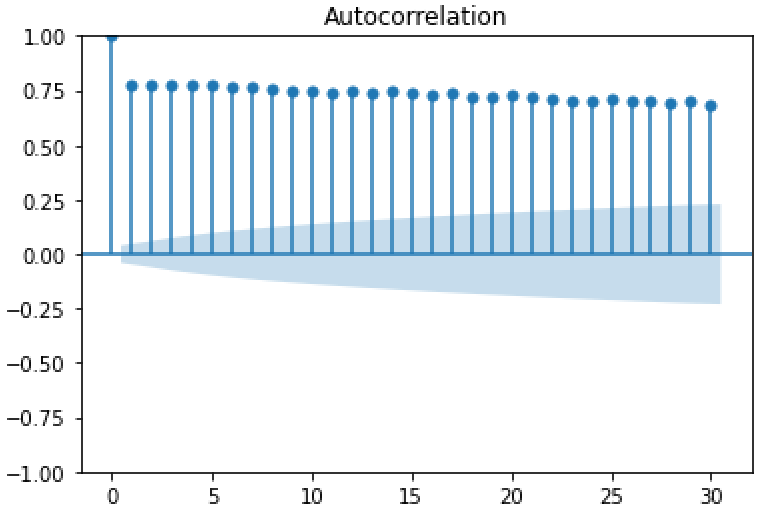
In [53]:

```
import statsmodels.api as sm
```

In [54]:

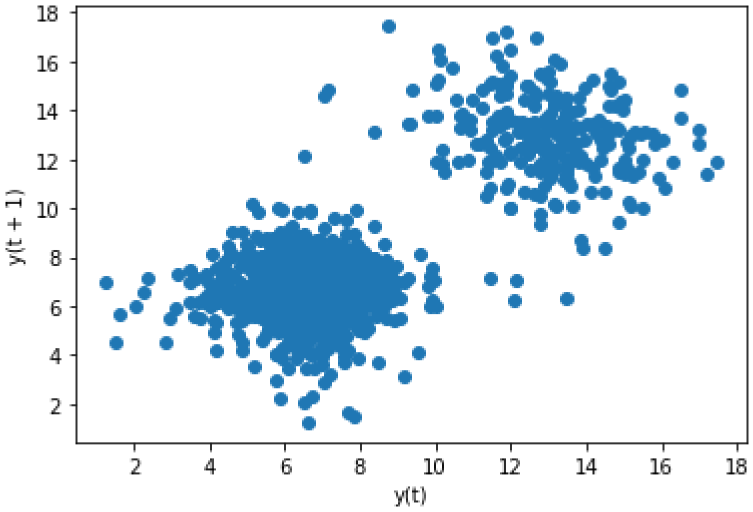
```
sm.graphics.tsa.plot_acf(df)
plt.show()
sm.graphics.tsa.plot_acf(df, lags=30)
plt.show()
```





In [55]:

```
from matplotlib import pyplot
from pandas.plotting import lag_plot
lag_plot(df)
pyplot.show()
```



In [56]:

```
from statsmodels.tsa.ar_model import AutoReg
from sklearn.metrics import mean_squared_error
```

In [57]:

```
model = AutoReg(df, lags=20)
model_fit = model.fit()
print('Coefficients: %s' % model_fit.params)
```

```
Coefficients: const              0.193272
sleep_hours.L1     0.114887
sleep_hours.L2     0.131713
sleep_hours.L3     0.135941
sleep_hours.L4     0.121907
sleep_hours.L5     0.109834
sleep_hours.L6     0.081247
sleep_hours.L7     0.064662
sleep_hours.L8     0.013882
sleep_hours.L9     0.005718
sleep_hours.L10    0.009981
sleep_hours.L11    0.001046
sleep_hours.L12    0.021532
sleep_hours.L13    0.022663
sleep_hours.L14    0.060778
sleep_hours.L15    0.033026
sleep_hours.L16    0.014353
sleep_hours.L17    0.027143
sleep_hours.L18   -0.030877
sleep_hours.L19   -0.021163
sleep_hours.L20    0.055161
dtype: float64
```

```
c:\pythonn\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueW
arning: A date index has been provided, but it has no associated frequency
information and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```