In [1]:

```
import pandas as pd
```

In [2]:

```
pd.to_datetime('2018-01-15 3:45pm')
```

Out[2]:

```
Timestamp('2018-01-15 15:45:00')
```

In [3]:

```
pd.to_datetime('7/8/1952')
```

Out[3]:

```
Timestamp('1952-07-08 00:00:00')
```

In [4]:

```
pd.to_datetime("1952")
```

Out[4]:

```
Timestamp('1952-01-01 00:00:00')
```

In [5]:

```
pd.to_datetime(['2/25/10', '8/6/17', '12/15/12'], format='%m/%d/%y')
```

Out[5]:

```
DatetimeIndex(['2010-02-25', '2017-08-06', '2012-12-15'], dtype='datetime64[ns]',
freq=None)
```

In [16]:

```
df=pd.read_csv('C:\\Users\\Alexandros\\Downloads\\opsd_germany_daily.csv')
```

In [17]:

```
df
```

Out[17]:

	Date	Consumption	Wind	Solar	Wind+Solar
0	2006-01-01	1069.18400	NaN	NaN	NaN
1	2006-01-02	1380.52100	NaN	NaN	NaN
2	2006-01-03	1442.53300	NaN	NaN	NaN

	Date	Consumption	Wind	Solar	Wind+Solar
3	2006-01-04	1457.21700	NaN	NaN	NaN
4	2006-01-05	1477.13100	NaN	NaN	NaN
...	...	...	...	...	...
4378	2017-12-27	1263.94091	394.507	16.530	411.037
4379	2017-12-28	1299.86398	506.424	14.162	520.586
4380	2017-12-29	1295.08753	584.277	29.854	614.131
4381	2017-12-30	1215.44897	721.247	7.467	728.714
4382	2017-12-31	1107.11488	721.176	19.980	741.156

4383 rows × 5 columns

In [18]:

```
df.head()
```

Out[18]:

	Date	Consumption	Wind	Solar	Wind+Solar
0	2006-01-01	1069.184	NaN	NaN	NaN
1	2006-01-02	1380.521	NaN	NaN	NaN
2	2006-01-03	1442.533	NaN	NaN	NaN
3	2006-01-04	1457.217	NaN	NaN	NaN
4	2006-01-05	1477.131	NaN	NaN	NaN

In [19]:

```
df.dtypes
```

Out[19]:

```
Date            object
Consumption    float64
Wind           float64
Solar          float64
Wind+Solar     float64
dtype: object
```

In [ ]:

In [20]:

```
df["Date"]=pd.to_datetime(df["Date"]) #convert to date
```

In [21]:

```
df.dtypes
```

Out[21]:

```
Date           datetime64[ns]
Consumption           float64
Wind                  float64
Solar                 float64
Wind+Solar            float64
dtype: object
```

In [22]:

```
df.head()
```

Out[22]:

	Date	Consumption	Wind	Solar	Wind+Solar
0	2006-01-01	1069.184	NaN	NaN	NaN
1	2006-01-02	1380.521	NaN	NaN	NaN
2	2006-01-03	1442.533	NaN	NaN	NaN
3	2006-01-04	1457.217	NaN	NaN	NaN
4	2006-01-05	1477.131	NaN	NaN	NaN

In [24]:

```
df.index
```

Out[24]:

```
RangeIndex(start=0, stop=4383, step=1)
```

In [25]:

```
### set date as index
df=df.set_index("Date")
```

In [26]:

```
df.index
```

Out[26]:

```
DatetimeIndex(['2006-01-01', '2006-01-02', '2006-01-03', '2006-01-04',
```

```
               '2006-01-05', '2006-01-06', '2006-01-07', '2006-01-08',
               '2006-01-09', '2006-01-10',
               ...
               '2017-12-22', '2017-12-23', '2017-12-24', '2017-12-25',
               '2017-12-26', '2017-12-27', '2017-12-28', '2017-12-29',
               '2017-12-30', '2017-12-31'],
              dtype='datetime64[ns]', name='Date', length=4383, freq=None)
```

In [27]:

```
## ELSE from read.csv
#opsd_daily = pd.read_csv('opsd_germany_daily.csv', index_col=0, parse_dates=True)
```

In [33]:

```
#Add columns with year month
df["Year"]=df.index.year
df["Month"]=df.index.month
df["Weekday"]=df.index.weekday
```

In [ ]:

In [ ]:

In [34]:

```
df.head()
```

Out[34]:

	Consumption	Wind	Solar	Wind+Solar	Year	Month	Weekday
Date							
2006-01-01	1069.184	NaN	NaN	NaN	2006	1	6
2006-01-02	1380.521	NaN	NaN	NaN	2006	1	0
2006-01-03	1442.533	NaN	NaN	NaN	2006	1	1
2006-01-04	1457.217	NaN	NaN	NaN	2006	1	2
2006-01-05	1477.131	NaN	NaN	NaN	2006	1	3

In [35]:

```
#Time based indexing
df.loc['2017-08-10']
```

Out[35]:

```
Consumption    1351.491
Wind            100.274
Solar            71.160
Wind+Solar      171.434
Year           2017.000
Month             8.000
Weekday           3.000
Name: 2017-08-10 00:00:00, dtype: float64
```

In [36]:

```
df.loc['2014-01-20':'2014-01-22']
```

Out[36]:

	Consumption	Wind	Solar	Wind+Solar	Year	Month	Weekday
Date							
2014-01-20	1590.687	78.647	6.371	85.018	2014	1	0
2014-01-21	1624.806	15.643	5.835	21.478	2014	1	1
2014-01-22	1625.155	60.259	11.992	72.251	2014	1	2

In [37]:

```
## select all for february 2012
df.loc['2012-02']
```

Out[37]:

	Consumption	Wind	Solar	Wind+Solar	Year	Month	Weekday
Date							
2012-02-01	1511.866	199.607	43.502	243.109	2012	2	2
2012-02-02	1563.407	73.469	44.675	118.144	2012	2	3
2012-02-03	1563.631	36.352	46.510	82.862	2012	2	4
2012-02-04	1372.614	20.551	45.225	65.776	2012	2	5
2012-02-05	1279.432	55.522	54.572	110.094	2012	2	6
2012-02-06	1574.766	34.896	55.389	90.285	2012	2	0
2012-02-07	1615.078	100.312	19.867	120.179	2012	2	1
2012-02-08	1613.774	93.763	36.930	130.693	2012	2	2
2012-02-09	1591.532	132.219	19.042	151.261	2012	2	3
2012-02-10	1581.287	52.122	34.873	86.995	2012	2	4

	Consumption	Wind	Solar	Wind+Solar	Year	Month	Weekday
Date							
2012-02-11	1377.404	32.375	44.629	77.004	2012	2	5
2012-02-12	1264.254	62.659	45.176	107.835	2012	2	6
2012-02-13	1561.987	25.984	11.287	37.271	2012	2	0
2012-02-14	1550.366	146.495	9.610	156.105	2012	2	1
2012-02-15	1476.037	413.367	18.877	432.244	2012	2	2
2012-02-16	1504.119	130.247	38.176	168.423	2012	2	3
2012-02-17	1438.857	196.515	17.328	213.843	2012	2	4
2012-02-18	1236.069	237.889	26.248	264.137	2012	2	5
2012-02-19	1107.431	272.655	30.382	303.037	2012	2	6
2012-02-20	1401.873	160.315	53.794	214.109	2012	2	0
2012-02-21	1434.533	281.909	57.984	339.893	2012	2	1
2012-02-22	1453.507	287.635	74.904	362.539	2012	2	2
2012-02-23	1427.402	353.510	18.927	372.437	2012	2	3
2012-02-24	1373.800	382.777	29.281	412.058	2012	2	4
2012-02-25	1133.184	302.102	42.667	344.769	2012	2	5
2012-02-26	1086.743	95.234	37.214	132.448	2012	2	6
2012-02-27	1436.095	86.956	43.099	130.055	2012	2	0
2012-02-28	1408.211	231.923	16.190	248.113	2012	2	1
2012-02-29	1434.062	77.024	30.360	107.384	2012	2	2

In [ ]:

```
## visualizing timeseries data
```

In [38]:

```
import matplotlib.pyplot as plt
```

In [39]:

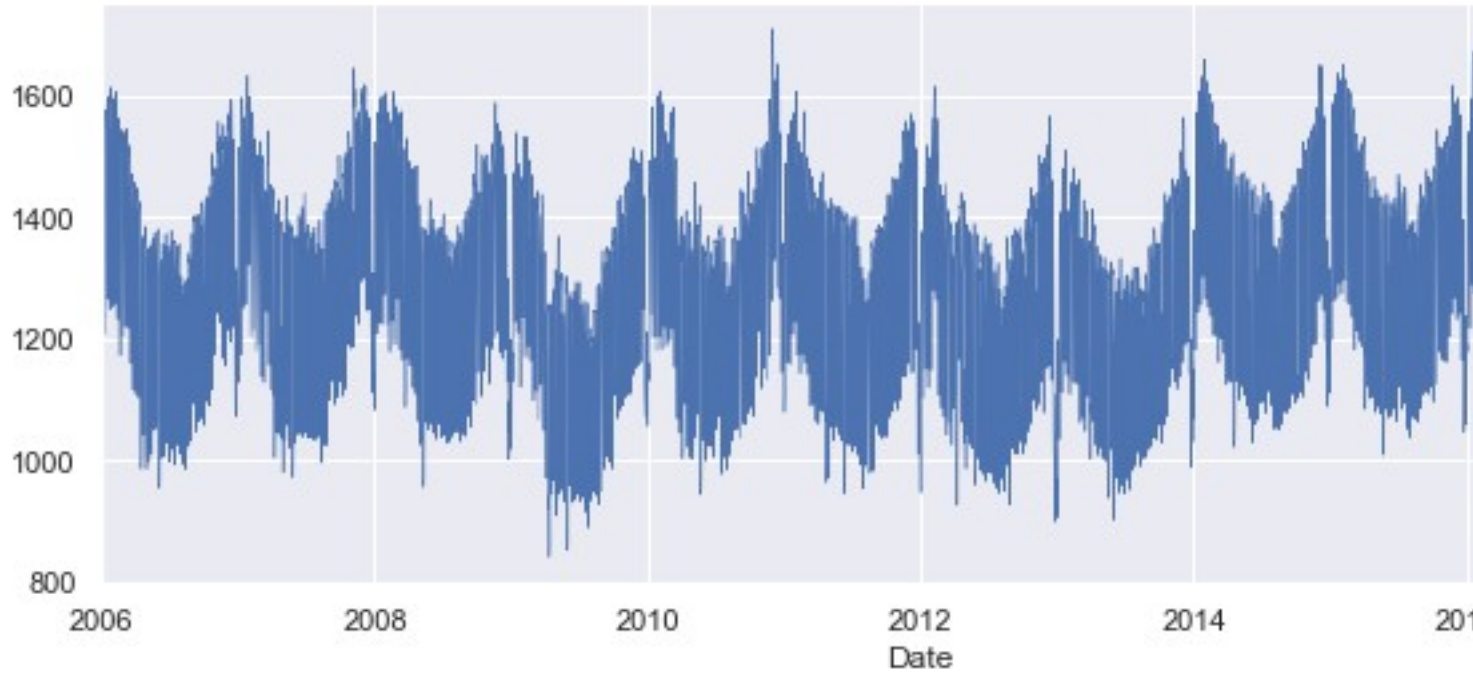
```
import seaborn as sns
# Use seaborn style defaults and set the default figure size
sns.set(rc={'figure.figsize':(11, 4)})
```

In [42]:

```
df["Consumption"].plot(linewidth=0.5)
```

Out[42]:

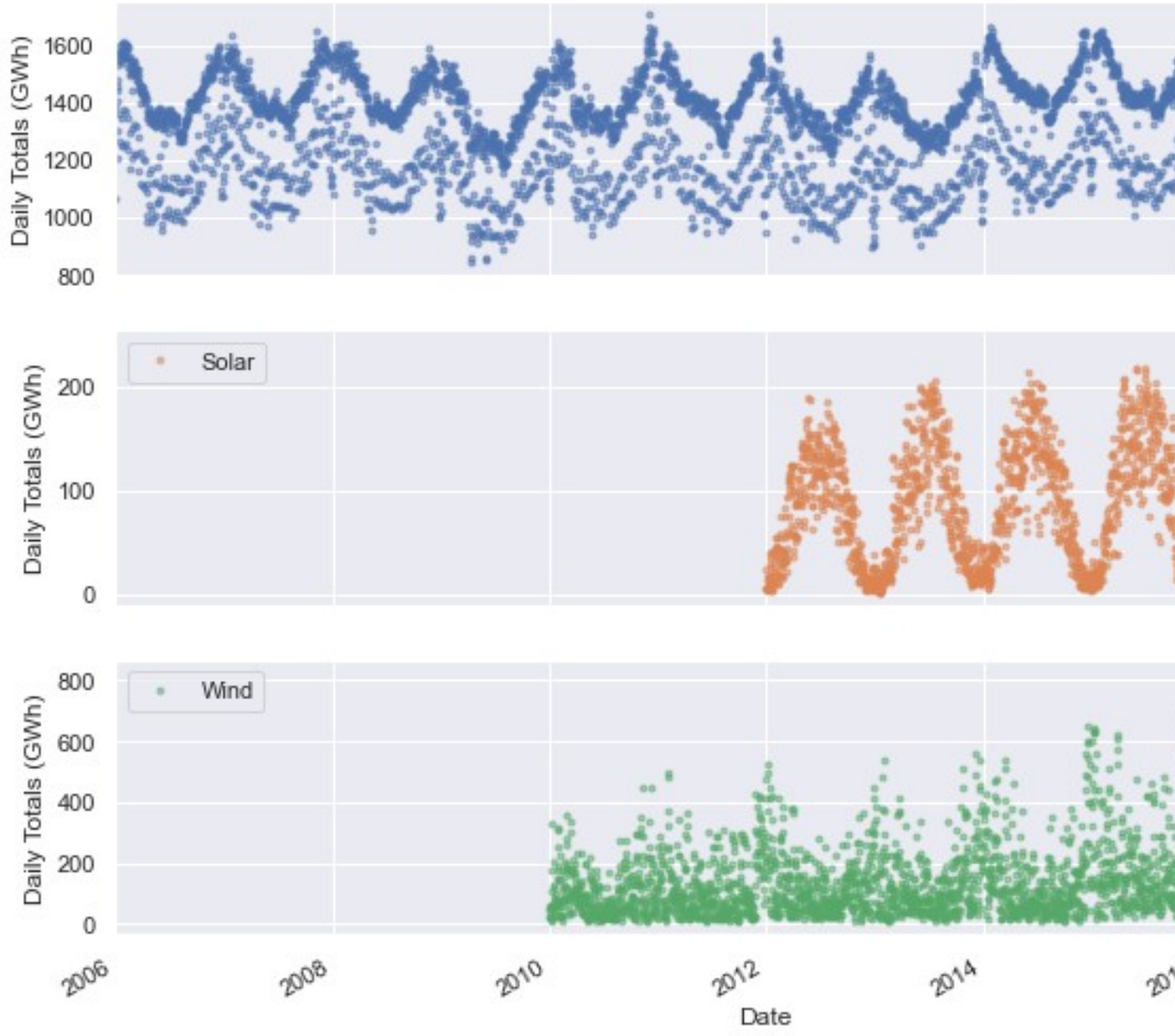
```
<AxesSubplot:xlabel='Date'>
```



In [43]:

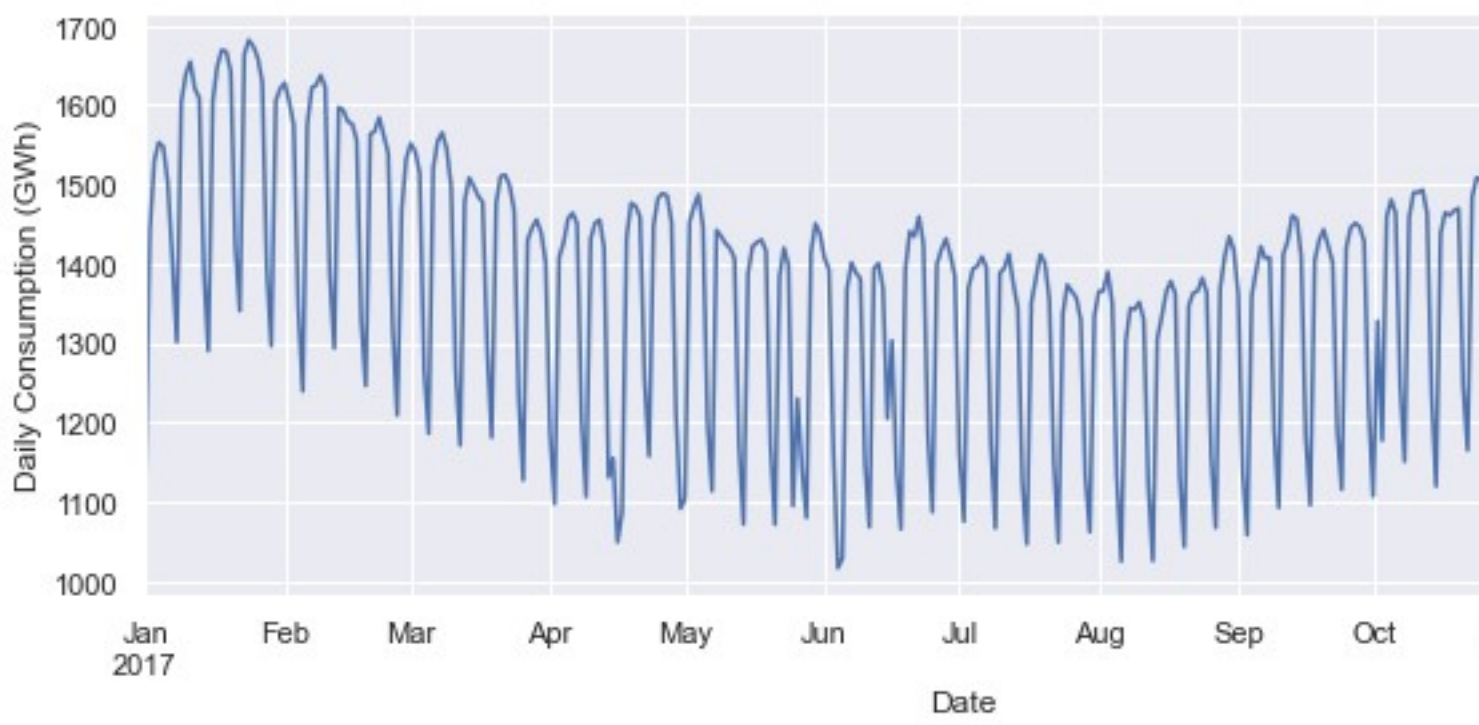
```
### plot many as dot plots

cols_plot = ['Consumption', 'Solar', 'Wind']
axes = df[cols_plot].plot(marker='.', alpha=0.5, linestyle='None', figsize=(11, 9),
subplots=True)
for ax in axes:
    ax.set_ylabel('Daily Totals (GWh)')
```



In [44]:

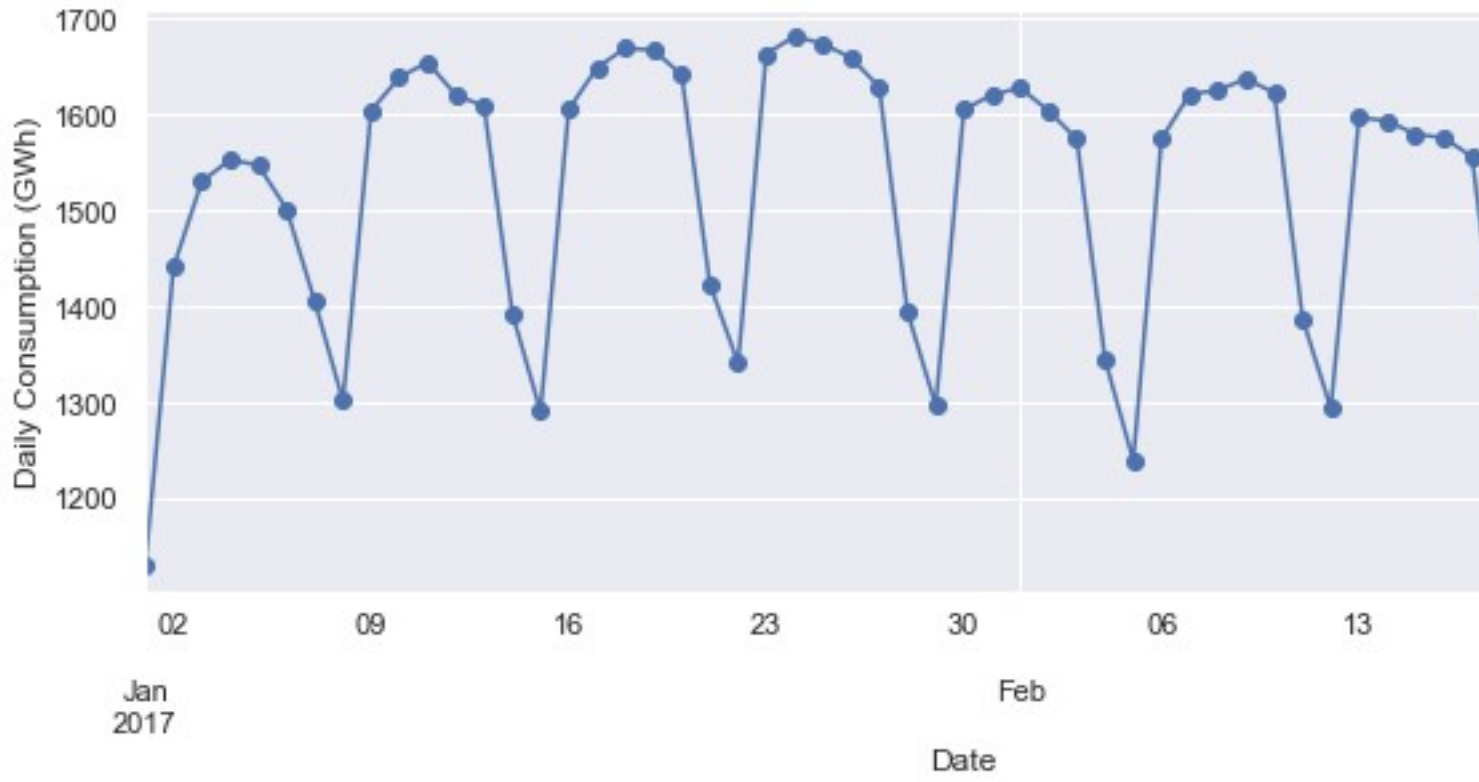
```
##plot for 2017
ax = df.loc['2017', 'Consumption'].plot()
ax.set_ylabel('Daily Consumption (GWh)');
```



In [45]:

```
### january and february

ax = df.loc['2017-01':'2017-02', 'Consumption'].plot(marker='o', linestyle='-')
ax.set_ylabel('Daily Consumption (GWh)');
```

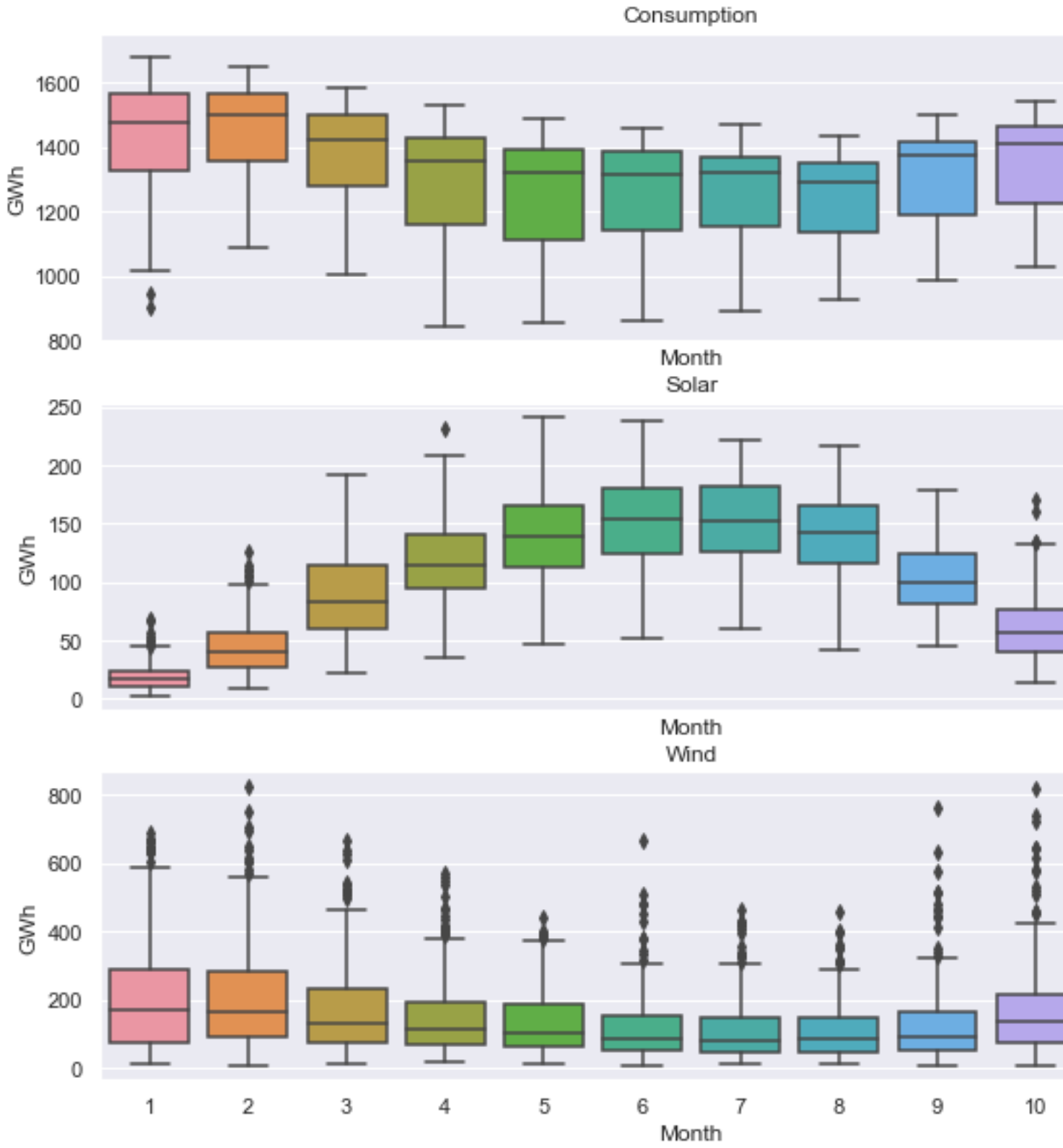


In [46]:

```
#seasonality with seaborn boxplot
```

In [48]:

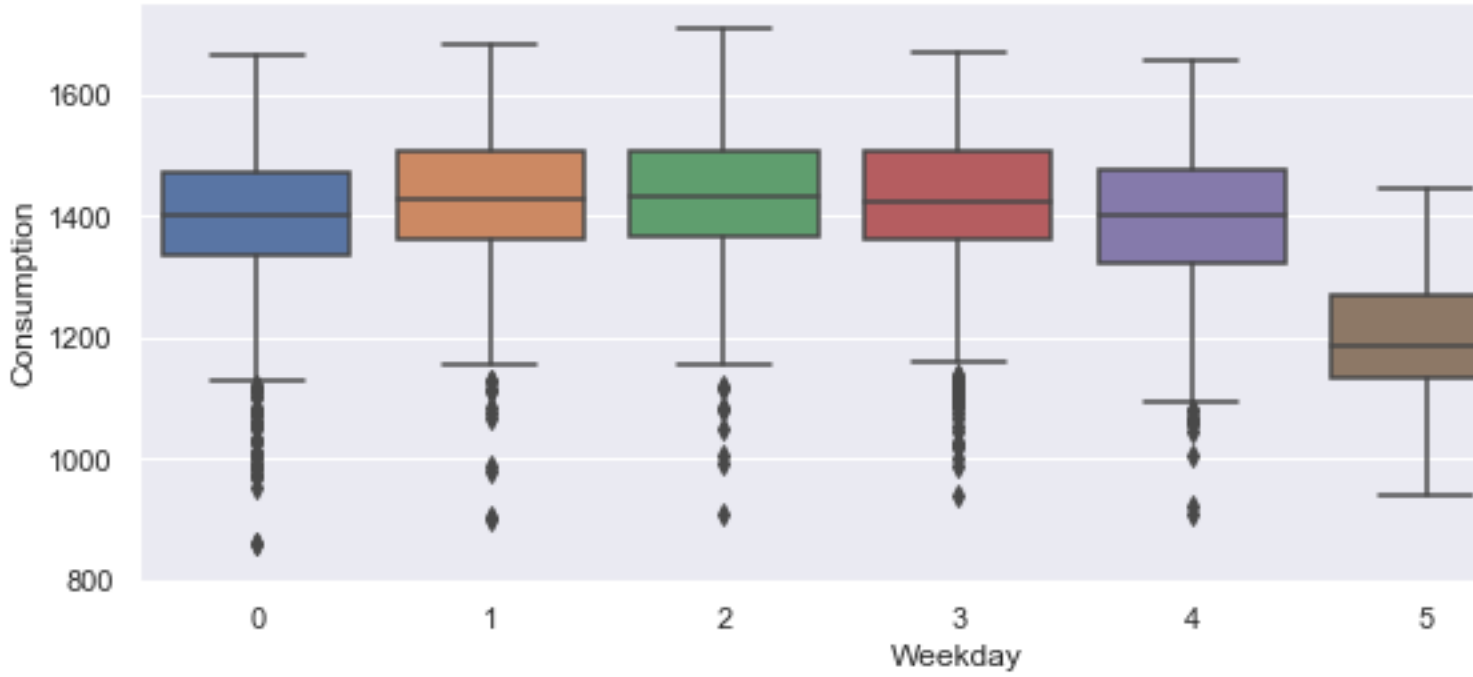
```
fig, axes = plt.subplots(3, 1, figsize=(11, 10), sharex=True)
for name, ax in zip(['Consumption', 'Solar', 'Wind'], axes):
    sns.boxplot(data=df, x='Month', y=name, ax=ax)
    ax.set_ylabel('GWh')
    ax.set_title(name)
# Remove the automatic x-axis label from all but the bottom subplot
if ax != axes[-1]:
    ax.set_xlabel('')
```



In [51]:

```
## by day

sns.boxplot(data=df, x='Weekday', y='Consumption');
```



In [53]:

```
# Compute the centered 7-day rolling mean
data_columns = ['Consumption', 'Wind', 'Solar', 'Wind+Solar']
df_7d = df[data_columns].rolling(7, center=True).mean()
df_7d.head(10)
```

Out[53]:

	Consumption	Wind	Solar	Wind+Solar
Date				
2006-01-01	NaN	NaN	NaN	NaN
2006-01-02	NaN	NaN	NaN	NaN
2006-01-03	NaN	NaN	NaN	NaN
2006-01-04	1361.471429	NaN	NaN	NaN
2006-01-05	1381.300143	NaN	NaN	NaN
2006-01-06	1402.557571	NaN	NaN	NaN
2006-01-07	1421.754429	NaN	NaN	NaN
2006-01-08	1438.891429	NaN	NaN	NaN
2006-01-09	1449.769857	NaN	NaN	NaN
2006-01-10	1469.994857	NaN	NaN	NaN

In [54]:

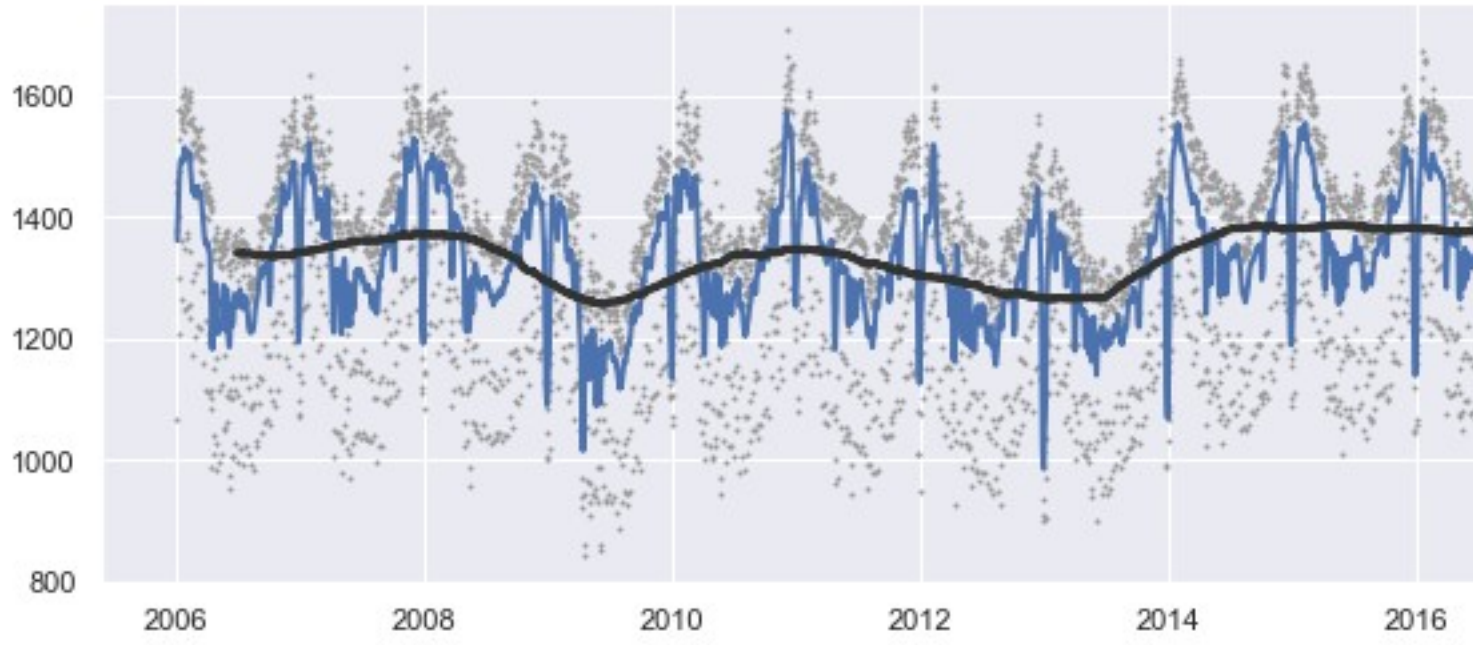
```
df_365d = df[data_columns].rolling(window=365, center=True, min_periods=360).mean()
```

In [57]:

```
# Plot daily, 7-day rolling mean, and 365-day rolling mean time series
fig, ax = plt.subplots()
ax.plot(df['Consumption'], marker='.', markersize=2, color='0.6',
linestyle='None', label='Daily')
ax.plot(df_7d['Consumption'], linewidth=2, label='7-d Rolling Mean')
ax.plot(df_365d['Consumption'], color='0.2', linewidth=3,
label='Trend (365-d Rolling Mean)')
```

Out[57]:

```
[<matplotlib.lines.Line2D at 0x145132c8>]
```

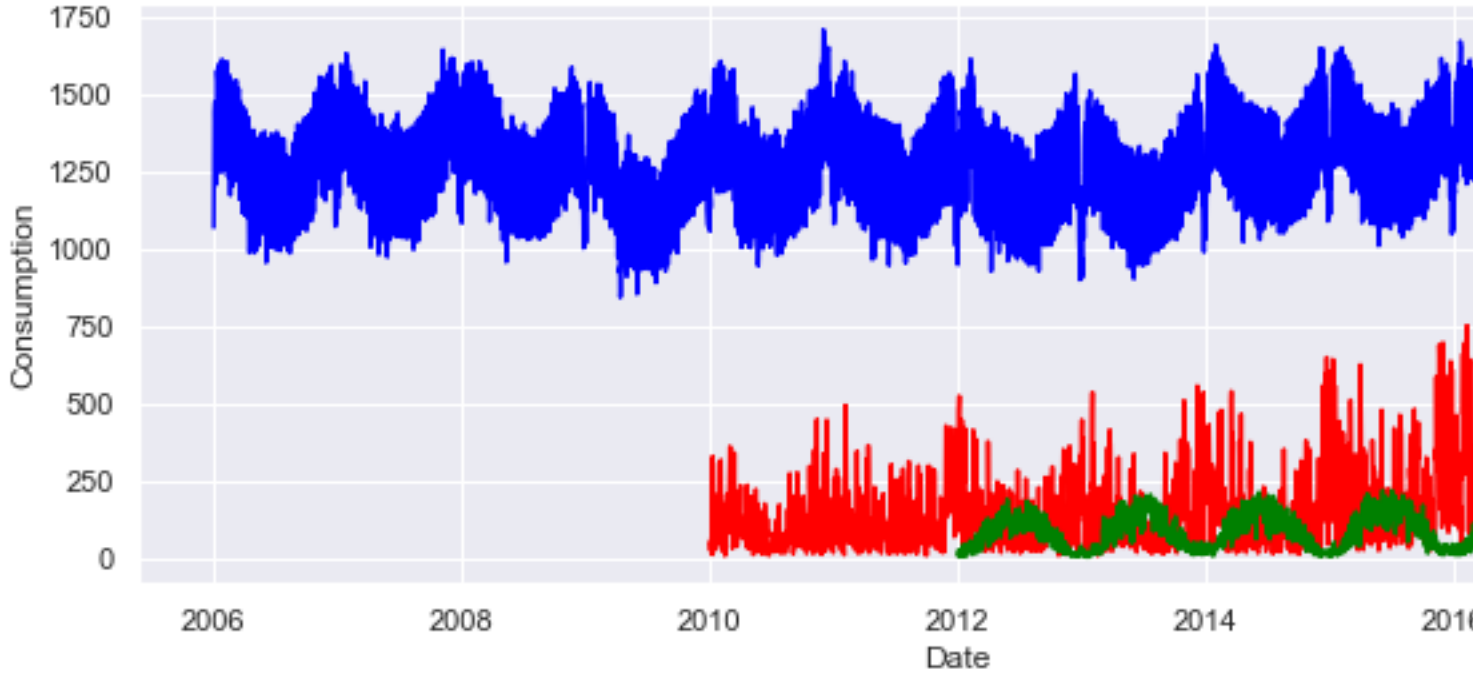


In [63]:

```
sns.lineplot(x="Date", y="Consumption",
             data=df,color="blue")
sns.lineplot(x="Date", y="Wind",
             data=df,color="red")
sns.lineplot(x="Date", y="Solar",
             data=df,color="Green")
```

Out[63]:

```
<AxesSubplot:xlabel='Date', ylabel='Consumption'>
```



In [ ]:

In [ ]:

In [61]:

```
help(snNew)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-61-7aa63a951731> in <module>
----> 1 help(snNew)

NameError: name 'snNew' is not defined
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: