In [1]:
```
pip install -U scikit-learn
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: scikit-learn in c:\users\ana.abesamis\appdata\roaming
\python\python39\site-packages (1.1.3)
Requirement already satisfied: numpy>=1.17.3 in c:\programdata\anaconda3\lib\site-pac
kages (from scikit-learn) (1.21.5)
Requirement already satisfied: scipy>=1.3.2 in c:\programdata\anaconda3\lib\site-pack
ages (from scikit-learn) (1.7.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\s
ite-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: joblib>=1.0.0 in c:\programdata\anaconda3\lib\site-pac
kages (from scikit-learn) (1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [3]:
```
df = pd.read_csv('Pasion et al dataset.csv', sep=',', header=0)
df.head()
```

Out[3]:

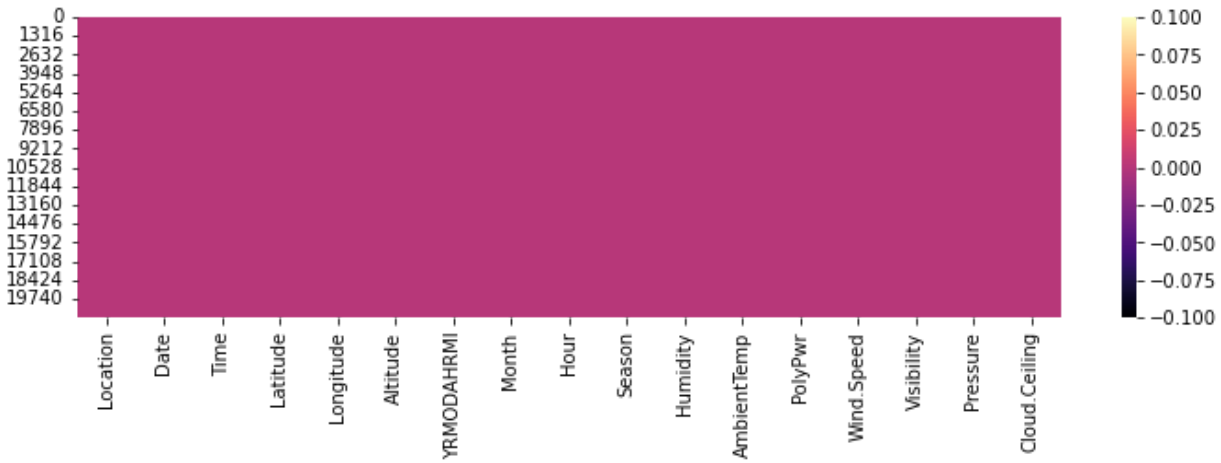
	Location	Date	Time	Latitude	Longitude	Altitude	YRMODAHRMI	Month	Hour	Season
0	Camp Murray	20171203	1145	47.11	-122.57	84	2.017120e+11	12	11	Winter
1	Camp Murray	20171203	1315	47.11	-122.57	84	2.017120e+11	12	13	Winter
2	Camp Murray	20171203	1330	47.11	-122.57	84	2.017120e+11	12	13	Winter
3	Camp Murray	20171204	1230	47.11	-122.57	84	2.017120e+11	12	12	Winter
4	Camp Murray	20171204	1415	47.11	-122.57	84	2.017120e+11	12	14	Winter

In [4]:
```
df.describe()
```

Out[6]:
```
Location         0
Date             0
Time             0
Latitude         0
Longitude        0
Altitude         0
YRMODAHRMI       0
Month            0
Hour             0
Season           0
Humidity         0
AmbientTemp      0
PolyPwr          0
Wind.Speed       0
Visibility       0
Pressure         0
Cloud.Ceiling    0
dtype: int64
```

In [7]:
```
# Missing Value Check - Visualized
plt.figure(figsize=(12,3))
sns.heatmap(df.isnull(), cmap="magma")
```

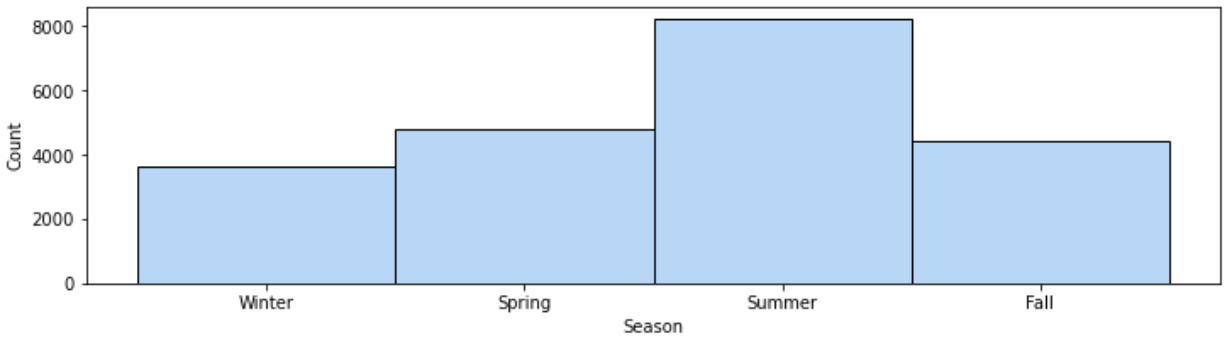
Out[7]:
```
<AxesSubplot:>
```



In [8]:
```
df['Date_Transform'] = pd.to_datetime(df['Date'], format='%Y%m%d')
```

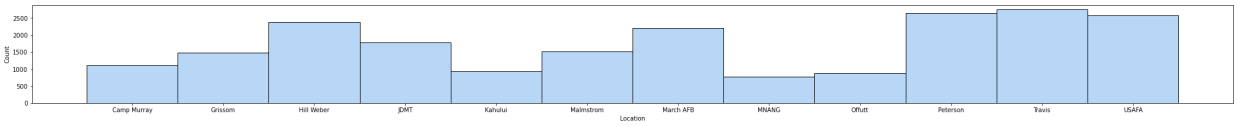
In [9]:
```
df.describe()

# Some insights to note:
# Observation times are only between 10 AM and 3:45 PM
# Dates (Month/Day/Year) are not considered as good input variables for the model
```



In [14]:
```
# Distribution of data across power generation sites
plt.figure(figsize=(36,3))
sns.histplot(x='Location', stat='count', data=df)
```

Out[14]:
```
<AxesSubplot:xlabel='Location', ylabel='Count'>
```



Feature Elimination of Variables

In [15]:
```
df = df.drop(columns=['Date', 'Time', 'YRMODAHRMI', 'Date_Transform'])
# These time variables are either redundant or unnecessary.
# Exact dates and exact times carry no meaning on their own, unlike other inputs such
```

In [16]:
```
df = df.drop('Location', axis=1)
# Site identification has no meaning to the prediction models, but its latitude and lo
```

In [17]:
```
df.head()
```

Out[17]:

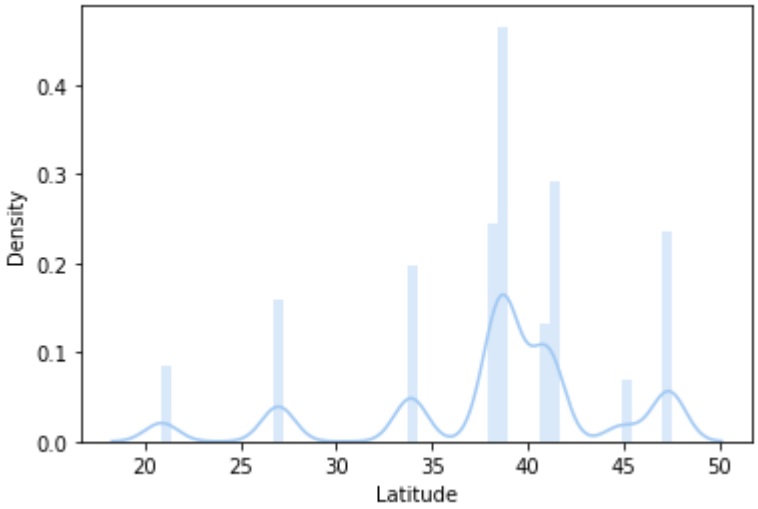
	Latitude	Longitude	Altitude	Month	Hour	Season	Humidity	AmbientTemp	PolyPwr	Wind.Sp
0	47.11	-122.57	84	12	11	Winter	81.71997	12.86919	2.42769	
1	47.11	-122.57	84	12	13	Winter	96.64917	9.66415	2.46273	
2	47.11	-122.57	84	12	13	Winter	93.61572	15.44983	4.46836	
3	47.11	-122.57	84	12	12	Winter	77.21558	10.36659	1.65364	
4	47.11	-122.57	84	12	14	Winter	54.80347	16.85471	6.57939	

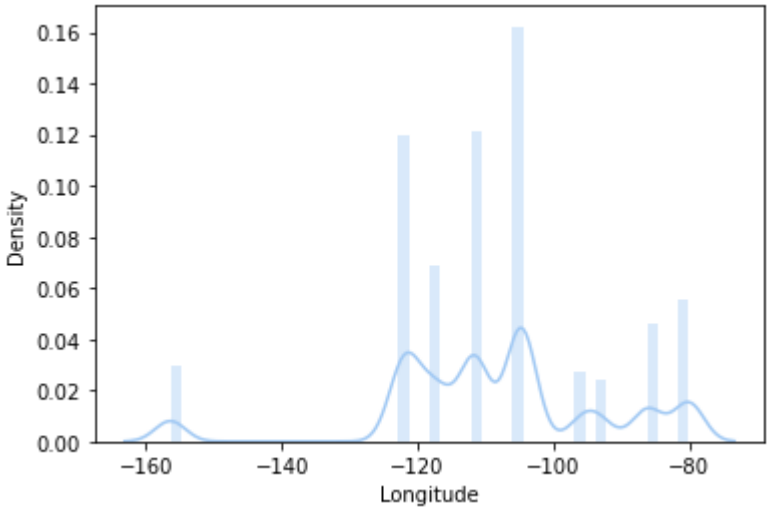
In [18]:
```
# Low Variance Check
df.var().apply(lambda x: '%.3f' % x)
# Only the categorical variables have extremely low variance, so we will not eliminate
```

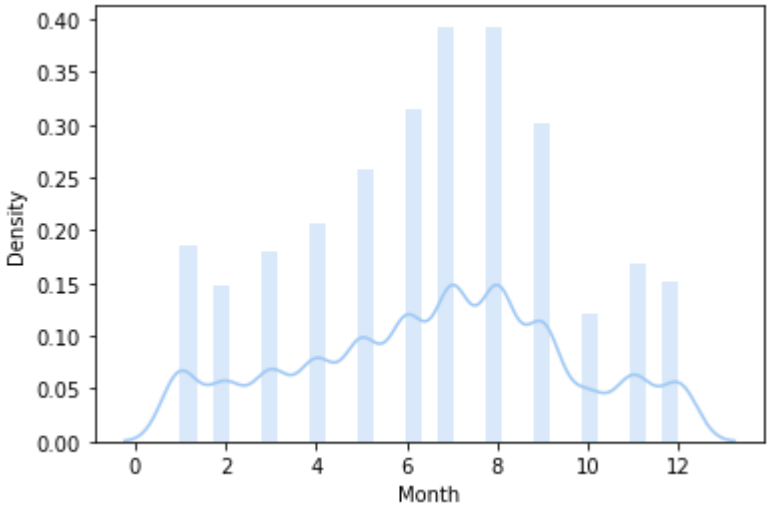
```
C:\Users\ana.abesamis\AppData\Local\Temp\ipykernel_16720\1778086645.py:2: FutureWar
ning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=Non
e') is deprecated; in a future version this will raise TypeError.  Select only vali
d columns before calling the reduction.
  df.var().apply(lambda x: '%.3f' % x)
```

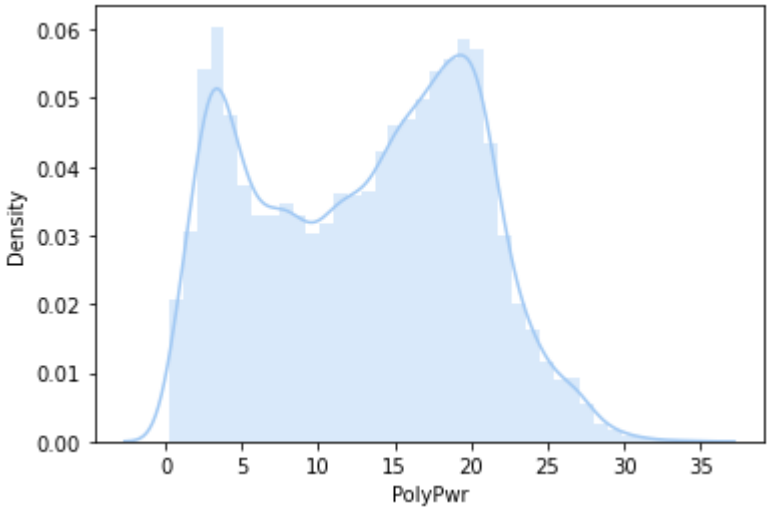
Normality Check

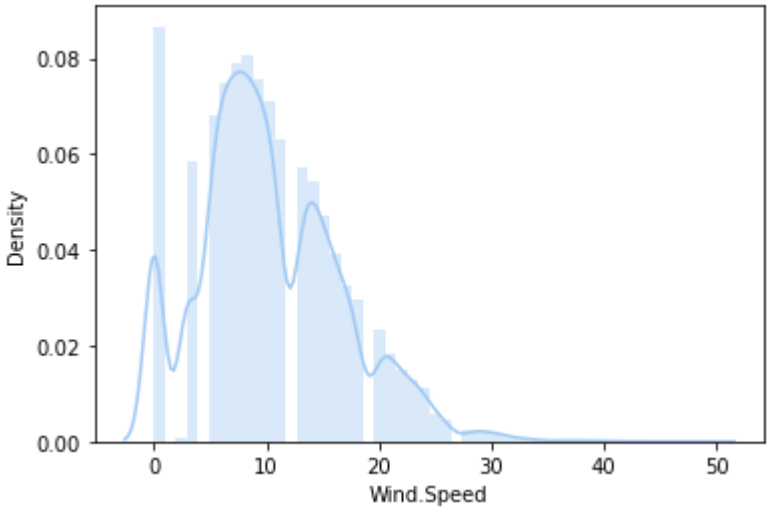
In [22]:
```
df_num = df.drop(df.loc[:, 'Fall':'Winter'].columns, axis=1)
for column in df_num.columns:
    plt.figure()
    sns.distplot(df_num[column])
```

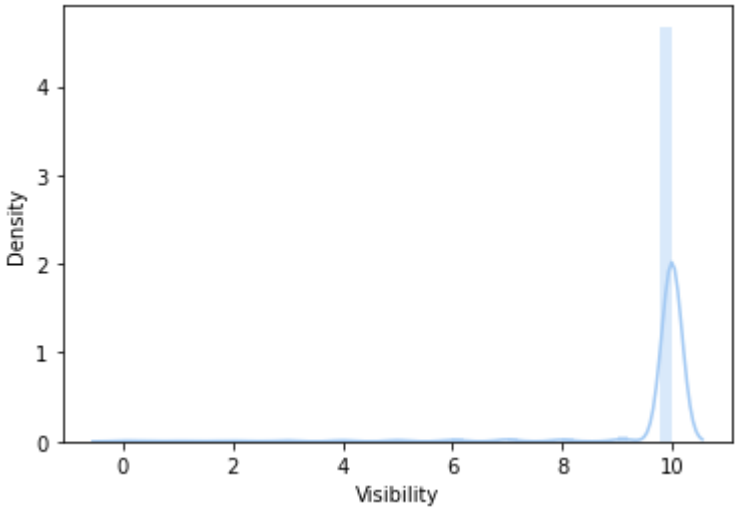

Density
0.0
0.1
0.2
0.3
0.4
20
25
30
35
40
45
50
Latitude

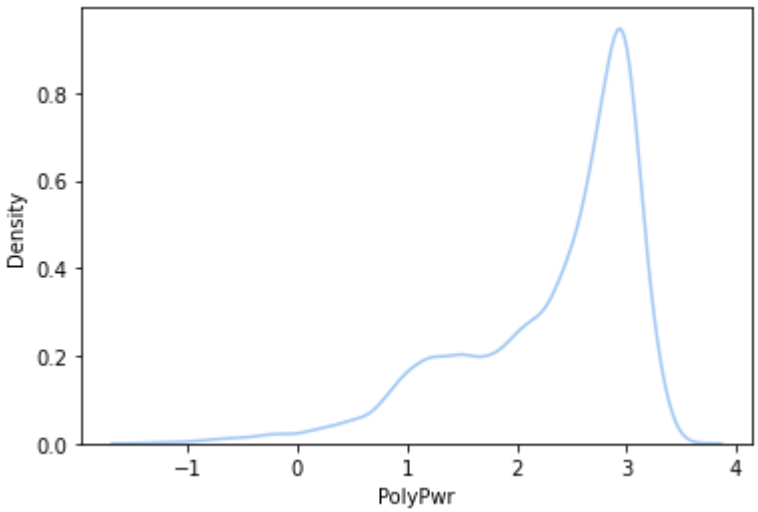

Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
−160
−140
−120
−100
−80
Longitude


Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0
2
4
6
8
10
12
Month


Density
PolyPwr


Density
Wind.Speed


Density
Visibility



Processed Data

In [24]:
```
# Download csv file
df.to_csv('clean_dataset.csv')
```

In [25]:
```
clean_df = df
```

In [26]:
```
clean_df.head()
```

Out[26]:

	Latitude	Longitude	Month	Hour	Humidity	AmbientTemp	PolyPwr	Wind.Speed	Visibility	Pre
0	47.11	-122.57	12	11	81.71997	12.86919	0.886940	5	10.0	
1	47.11	-122.57	12	13	96.64917	9.66415	0.901270	0	10.0	
2	47.11	-122.57	12	13	93.61572	15.44983	1.497021	5	10.0	
3	47.11	-122.57	12	12	77.21558	10.36659	0.502979	5	2.0	
4	47.11	-122.57	12	14	54.80347	16.85471	1.883942	3	3.0	

In [27]:
```
clean_df.dtypes
```

Out[27]:
```
Latitude         float64
Longitude        float64
Month              int64
Hour               int64
Humidity         float64
AmbientTemp      float64
PolyPwr          float64
Wind.Speed         int64
Visibility       float64
Pressure         float64
Cloud.Ceiling      int64
Fall               uint8
Spring             uint8
Summer             uint8
Winter             uint8
dtype: object
```