

## COMPUTER ENGINEERING CSE4062 2020 SPRING PROJECT ANALYSIS OF ELECTRIC PRODUCTION OF A WIND TURBINE

**Group: 8**

Name ID	Number	Mail Addresses
Furkan Can Ercan	150316044	furkanercan98@gmail.com
Muhammed Avci	150414007	muhammedavci96@gmail.com
Yasin Gök	150115058	ygok.96@gmail.com

In [2]:

```
import pandas as pd
import seaborn as sns
```

In [12]:

```
data=pd.read_excel("C:/Users/Furkan/Desktop/Final Data.xlsx") #Reading Data From Excel File
```

In [13]:

```
df=data.copy() # Copying data for just in case
```

In [67]:

```
df.Month=pd.Categorical(df.Month)
df["Day/Night"]=pd.Categorical(df["Day/Night"])
df.Date=pd.Categorical(df.Date)
df.Time=pd.Categorical(df.Time)
```

In [58]:

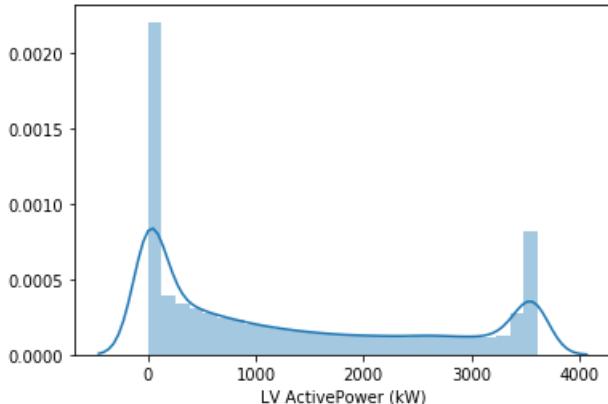
```
#Plots for Our Target Variable
```

In [59]:

```
sns.distplot(df["LV ActivePower (kW)"])
```

Out[59]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c424ceeac8>
```

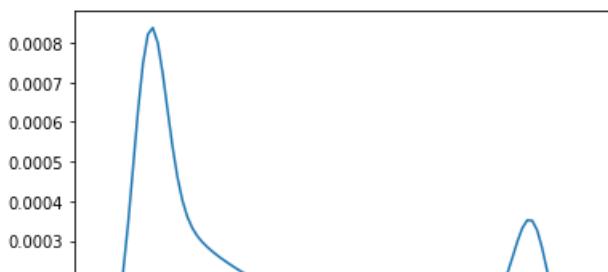


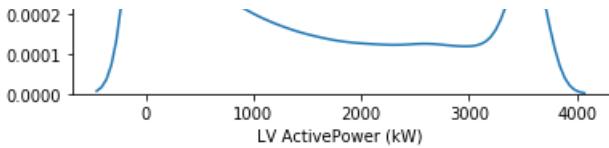
In [82]:

```
sns.distplot(df["LV ActivePower (kW)"],hist=False)
```

Out[82]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c4304dd3c8>
```



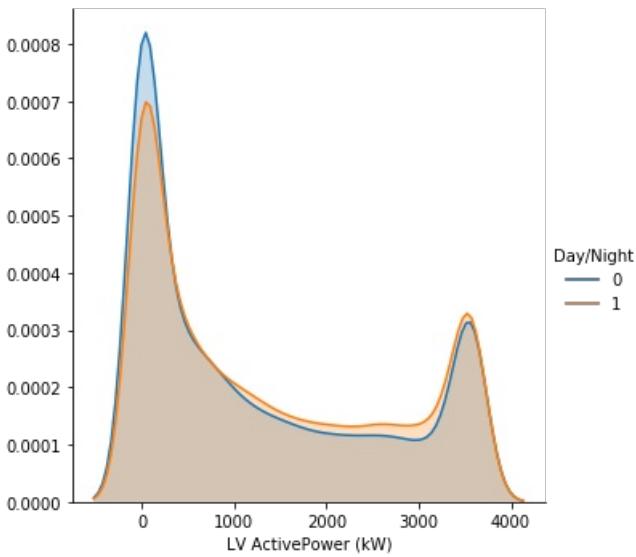


In [71]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"LV ActivePower (kW)",shade=True).add_legend()
#Distribution of Electric Production by Day-Night
```

Out[71]:

<seaborn.axisgrid.FacetGrid at 0x1c42f7846c8>

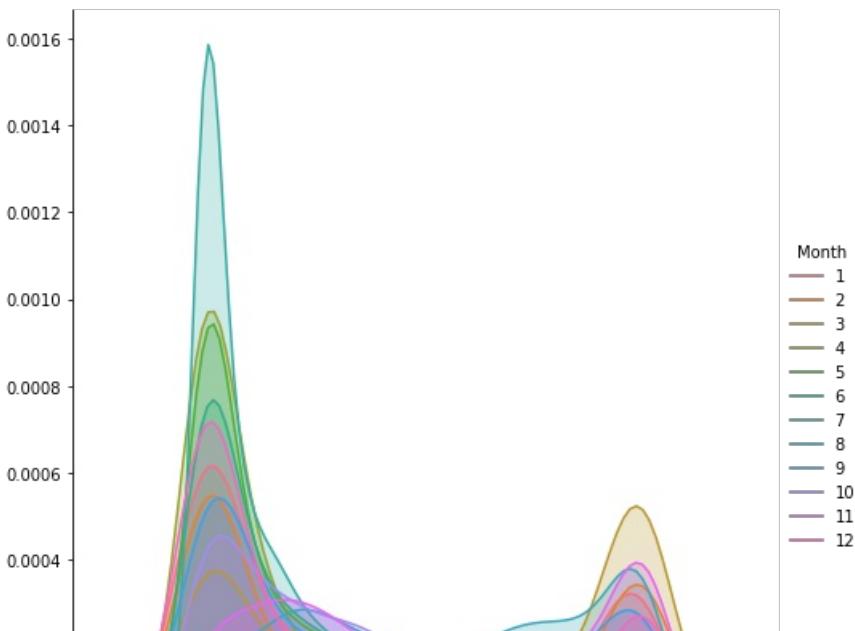


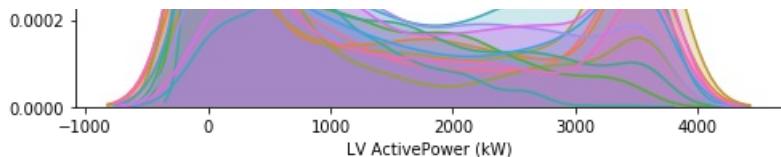
In [81]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"LV ActivePower (kW)",shade=True).add_legend()
#Distribution of Electric Production, According to Months
#It seems that 9th and 11th months does not fit the general distribution of data.
#The months 9 and 11 does not have 2 peaks like the rest of the data
```

Out[81]:

<seaborn.axisgrid.FacetGrid at 0x1c42e6fa448>



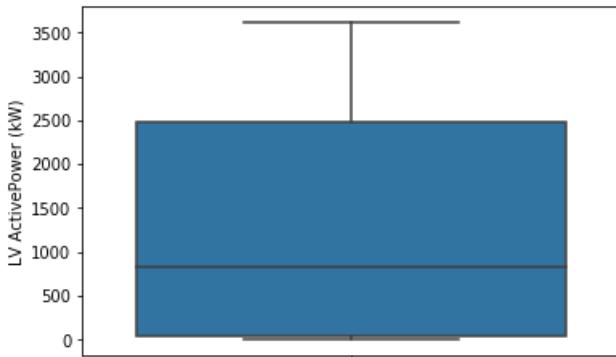


In [60]:

```
sns.boxplot(df["LV ActivePower (kW)"], orient="v") # It seems that other than 0 and max points,
# data seems nearly equally distributed.
```

Out[60]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c425b65f48>
```



In [61]:

```
df["LV ActivePower (kW)"].describe() #Our boxplot shows that our data is distributed generally between 25%-75% range. There
# very less amount of values in the 0-25% range. There seem to
be no outliers in the
#boxplot. Our median or 50% value -825- and mean value -1307-
seems highly different.
#For generality purposes we can rely much on this case since it
an is highly effected
#by large values.
```

Out[61]:

count	50530.000000
mean	1307.684332
std	1312.459242
min	-2.471405
25%	50.677890
50%	825.838074
75%	2482.507569
max	3618.732910

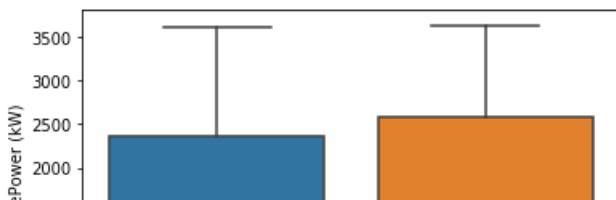
Name: LV ActivePower (kW), dtype: float64

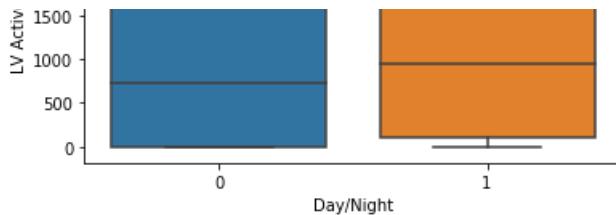
In [86]:

```
sns.boxplot(y=df["LV ActivePower (kW)"], orient="v", x="Day/Night", data=df)
#The difference between day and night productions
```

Out[86]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c420e94688>
```



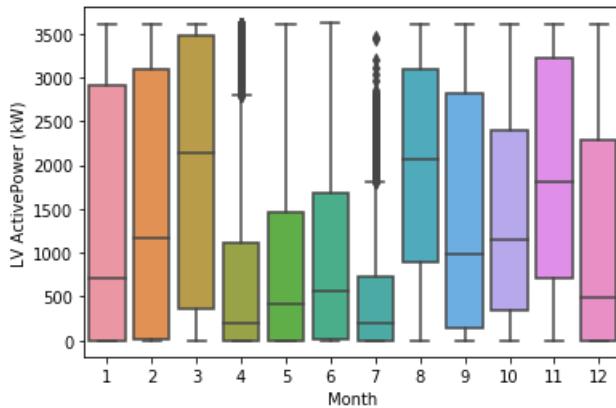


In [91]:

```
sns.boxplot(y=df["LV ActivePower (kW)"],orient="v",x="Month",data=df)
#In months spectrum we can see that months 4 and 7 has lots and lots of
#Outliers.
```

Out[91]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c430283848>
```

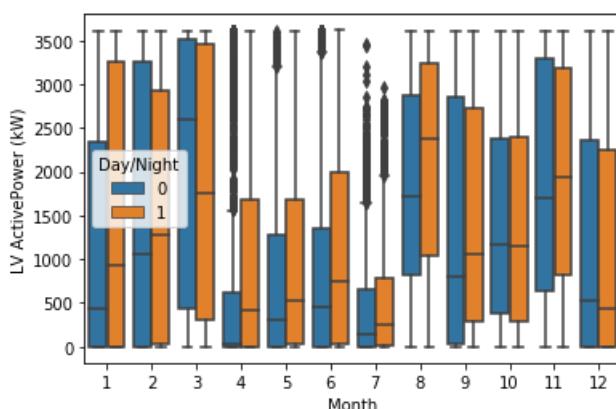


In [92]:

```
sns.boxplot(y=df["LV ActivePower (kW)"],orient="v",x="Month",data=df,hue="Day/Night")
# We can also see here that throughout the months, day and night productions are
#different within the same month. Generally Day productions(1) seem to have higher
#variance than the night productions. And we can see that in months 5 and 6
#additional to the month 4-7 have lots of outliers but only visible in night productions.
```

Out[92]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c429da9888>
```

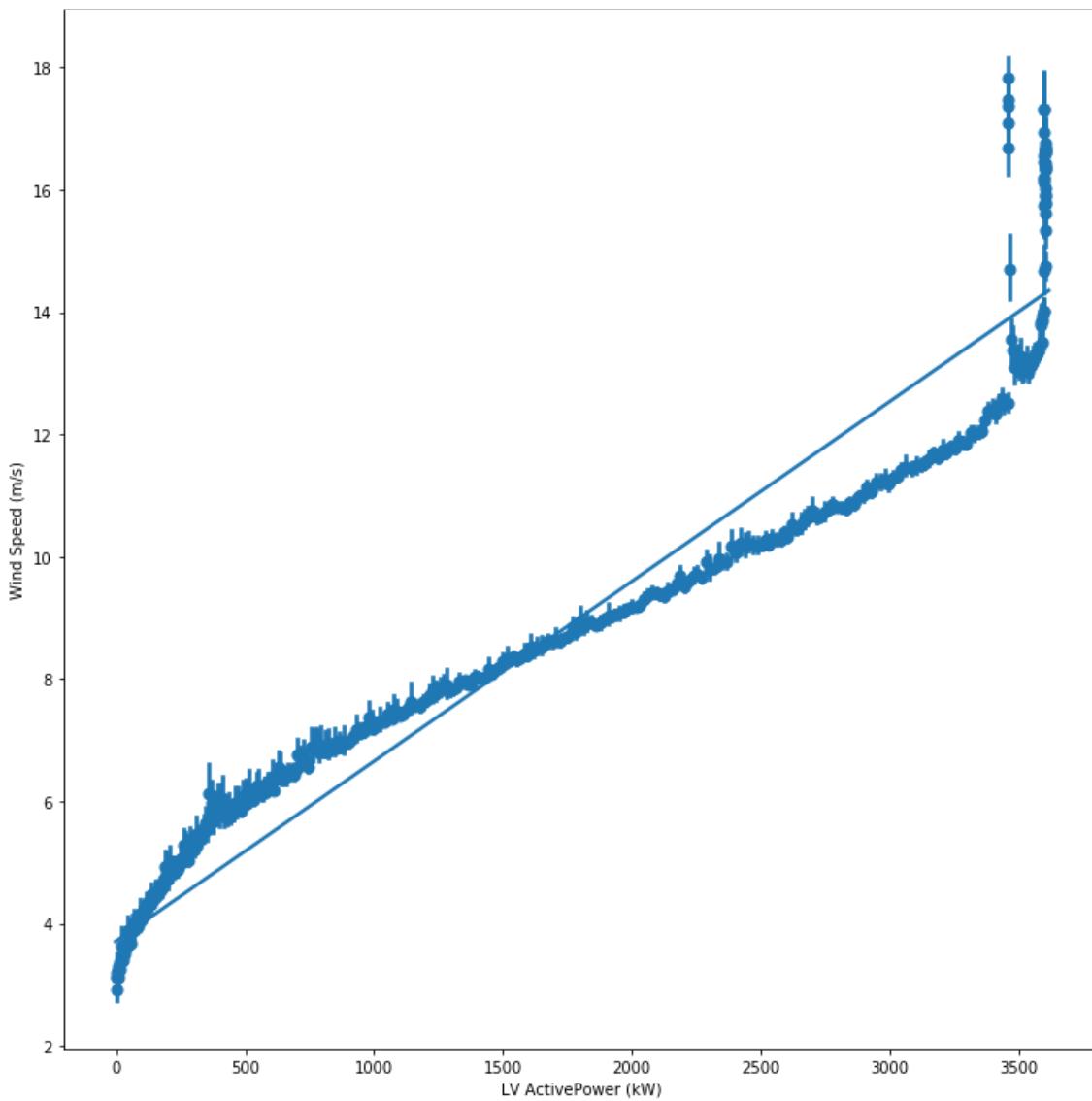


In [103]:

```
sns.lmplot(x="LV ActivePower (kW)",y="Wind Speed (m/s)",data=df,height=10,x_bins=500)
#Linear Model plot of electric production and wind speed. Discretized the xvalues
#Because 50k data is not demonstrated well in a graph. To see it clearly discretized
#it in to 500 bins. Production and Wind speed seem to have linear relationship
```

Out[103]:

```
<seaborn.axisgrid.FacetGrid at 0x1c43cf70b08>
```

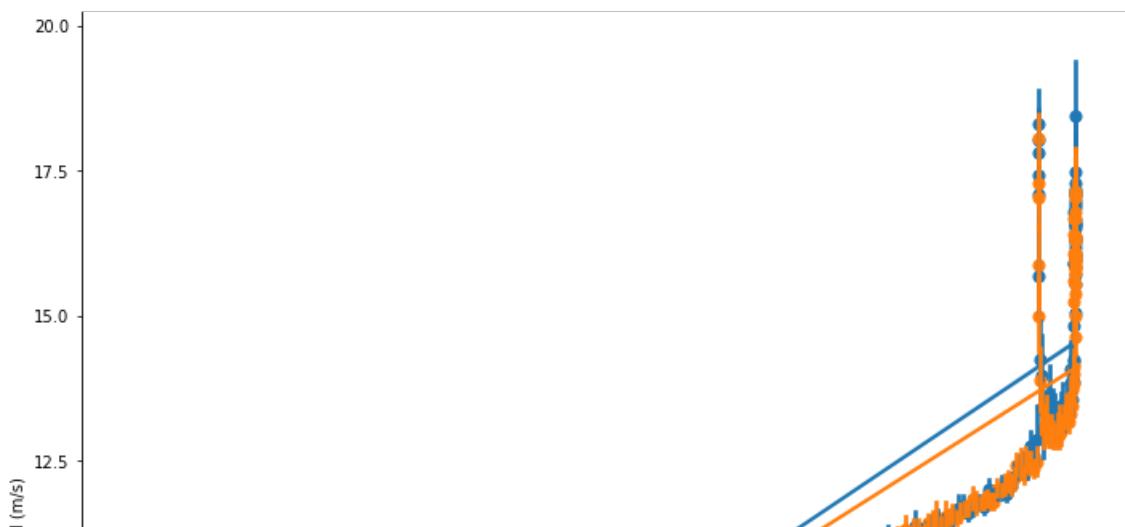


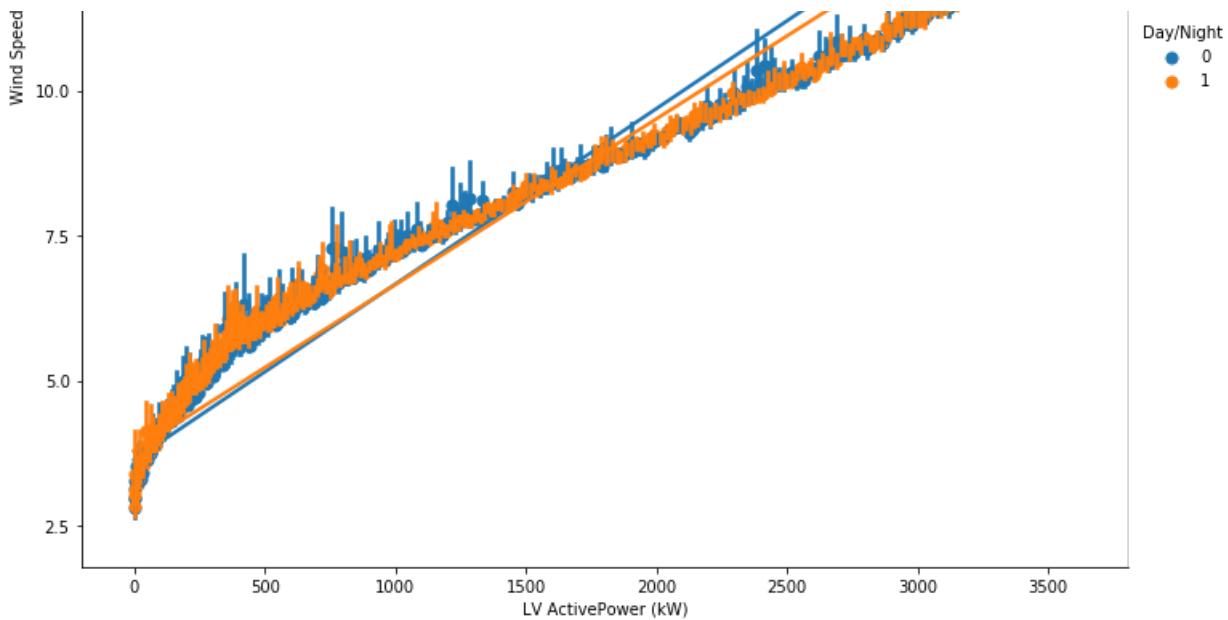
In [105]:

```
sns.lmplot(x="LV ActivePower (kW)",y="Wind Speed (m/s)",data=df,height=10,x_bins=500,hue="Day/Night")
#Day-Night comparison shows that they do not much differ in the relationship with the winds
```

Out [105]:

```
<seaborn.axisgrid.FacetGrid at 0x1c43d7f01c8>
```



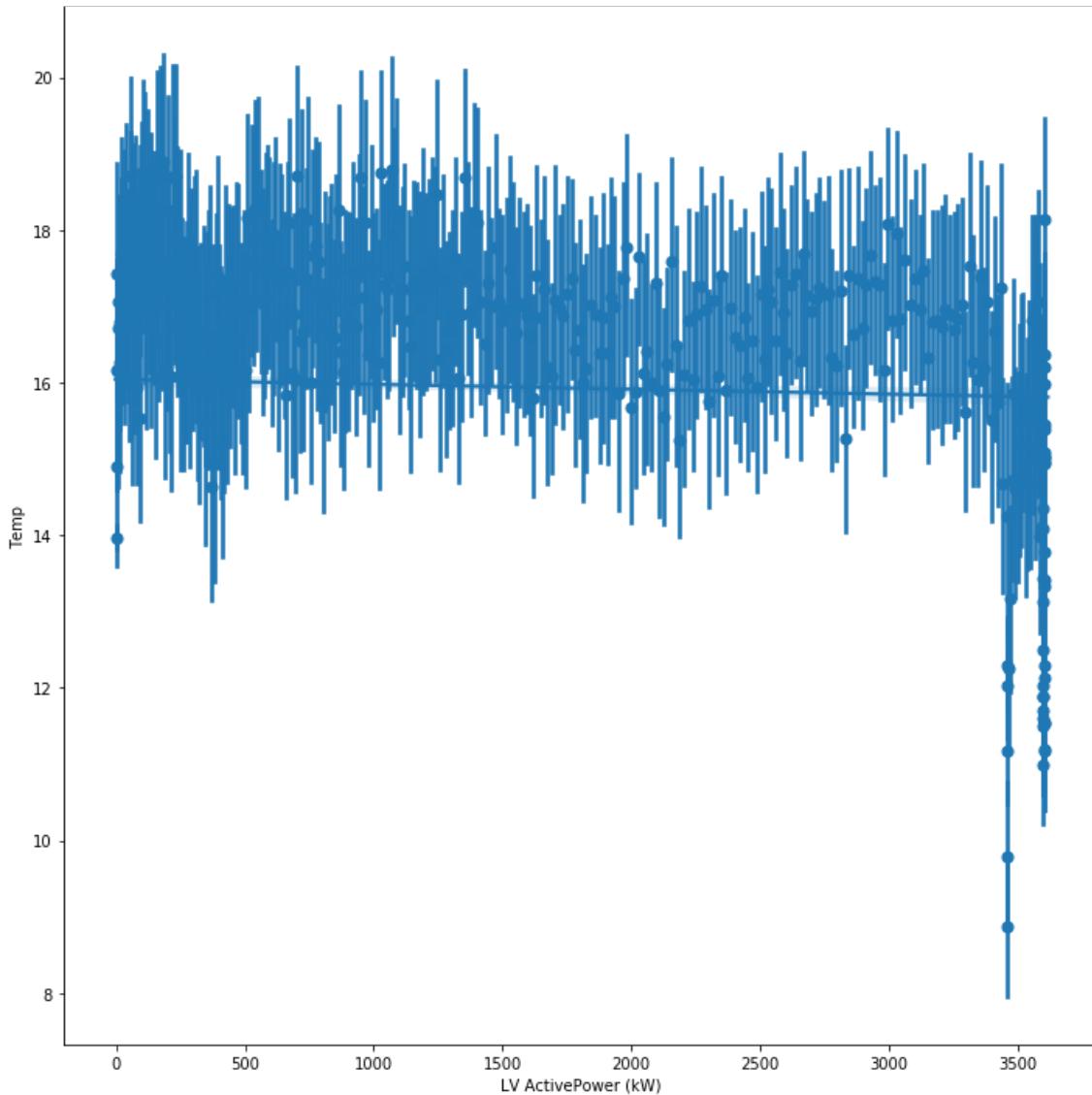


In [104]:

```
sns.lmplot(x="LV ActivePower (kW)",y="Temp",data=df,height=10,x_bins=500)# Temperature and Electric production does not  
#Seem to have a linear relationship
```

Out[104]:

```
<seaborn.axisgrid.FacetGrid at 0x1c43d38aec8>
```

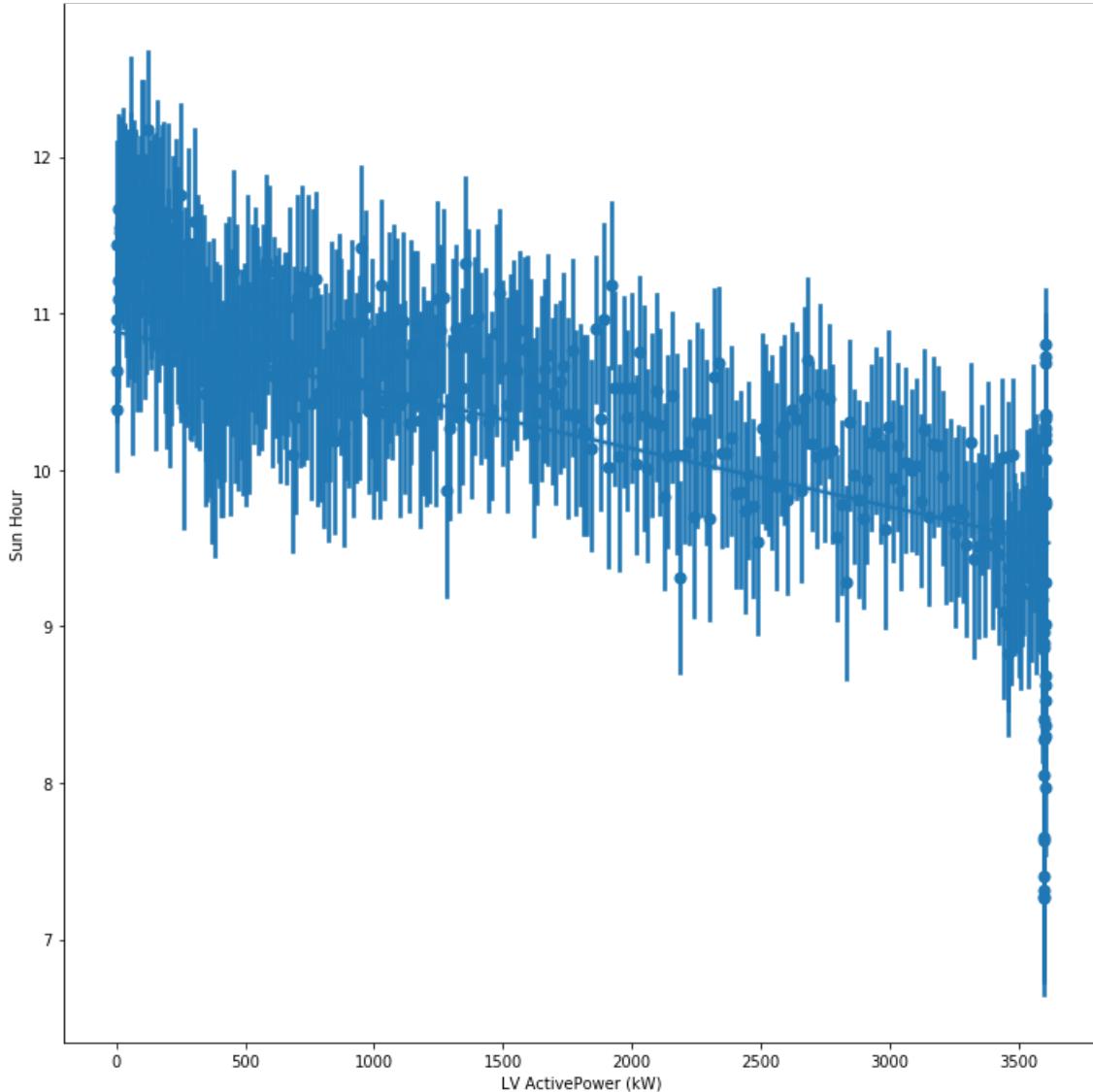


In [146]:

```
sns.lmplot(x="LV ActivePower (kW)", y="Sun Hour", data=df, height=10, x_bins=500)
#Sun Hour and Electric Production seem to have a strong negative linear relationship
```

Out[146]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d9770448>
```

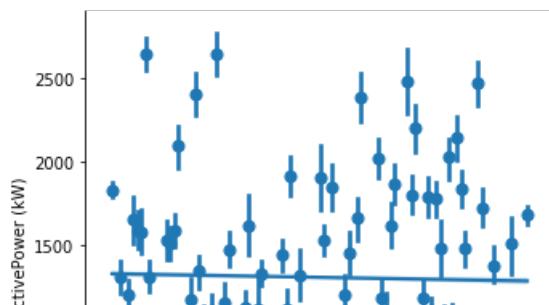


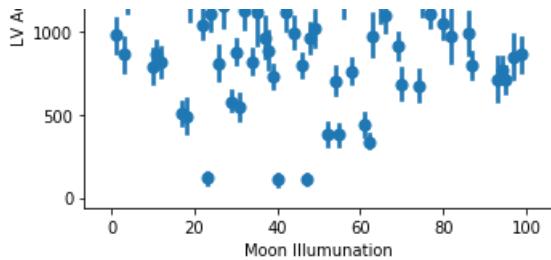
In [147]:

```
sns.lmplot(y="LV ActivePower (kW)", x="Moon Illumunation", data=df, x_bins=5000)
#Electric Production and Moon Illumination does not have a linear relationship
```

Out[147]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4da0b1888>
```



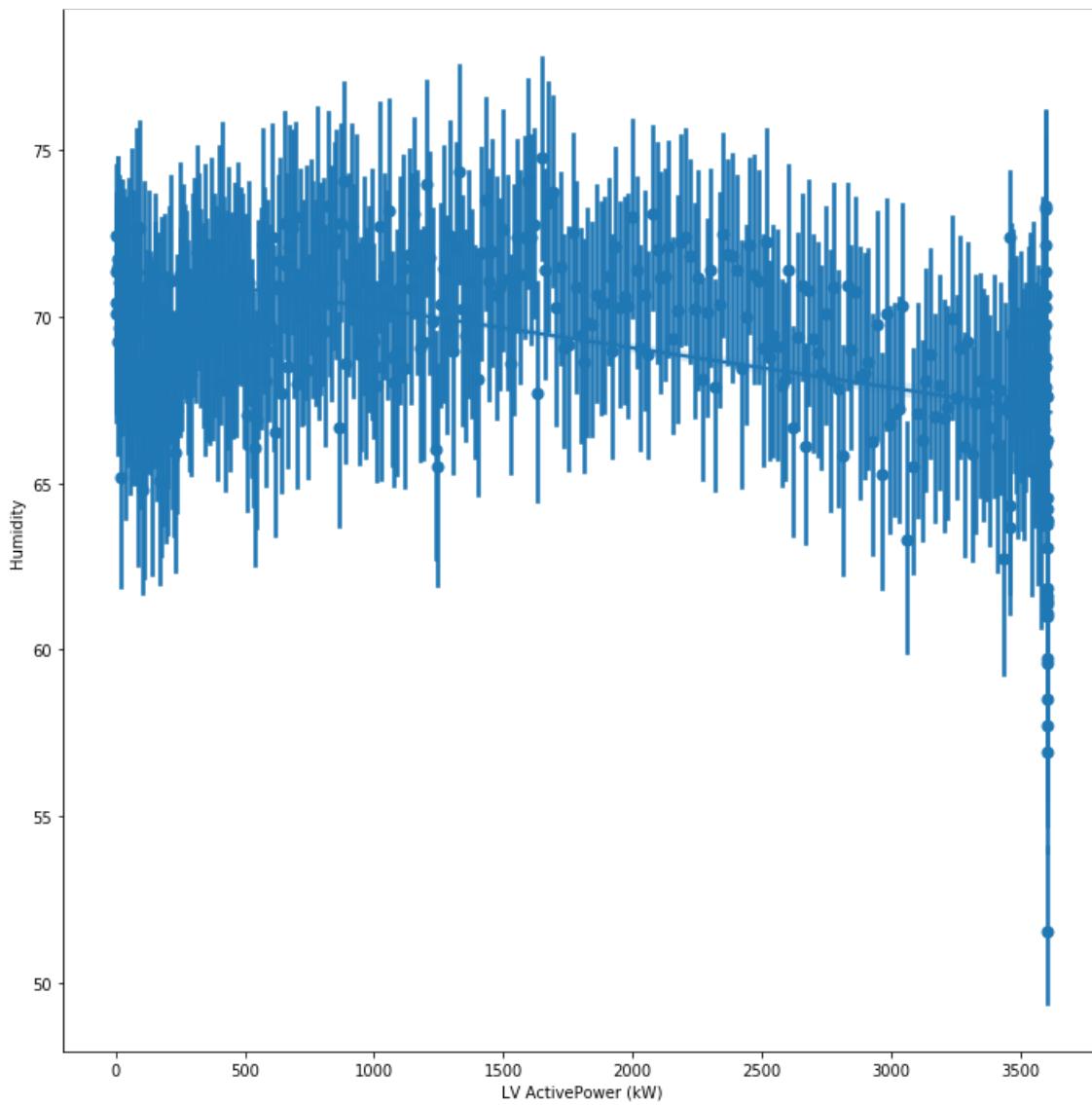


In [115]:

```
sns.lmplot(x="LV ActivePower (kW)", y="Humidity", data=df, height=10, x_bins=500)  
#Humidity and Electric Production seem to have negative linear relationship
```

Out[115]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d370b048>
```

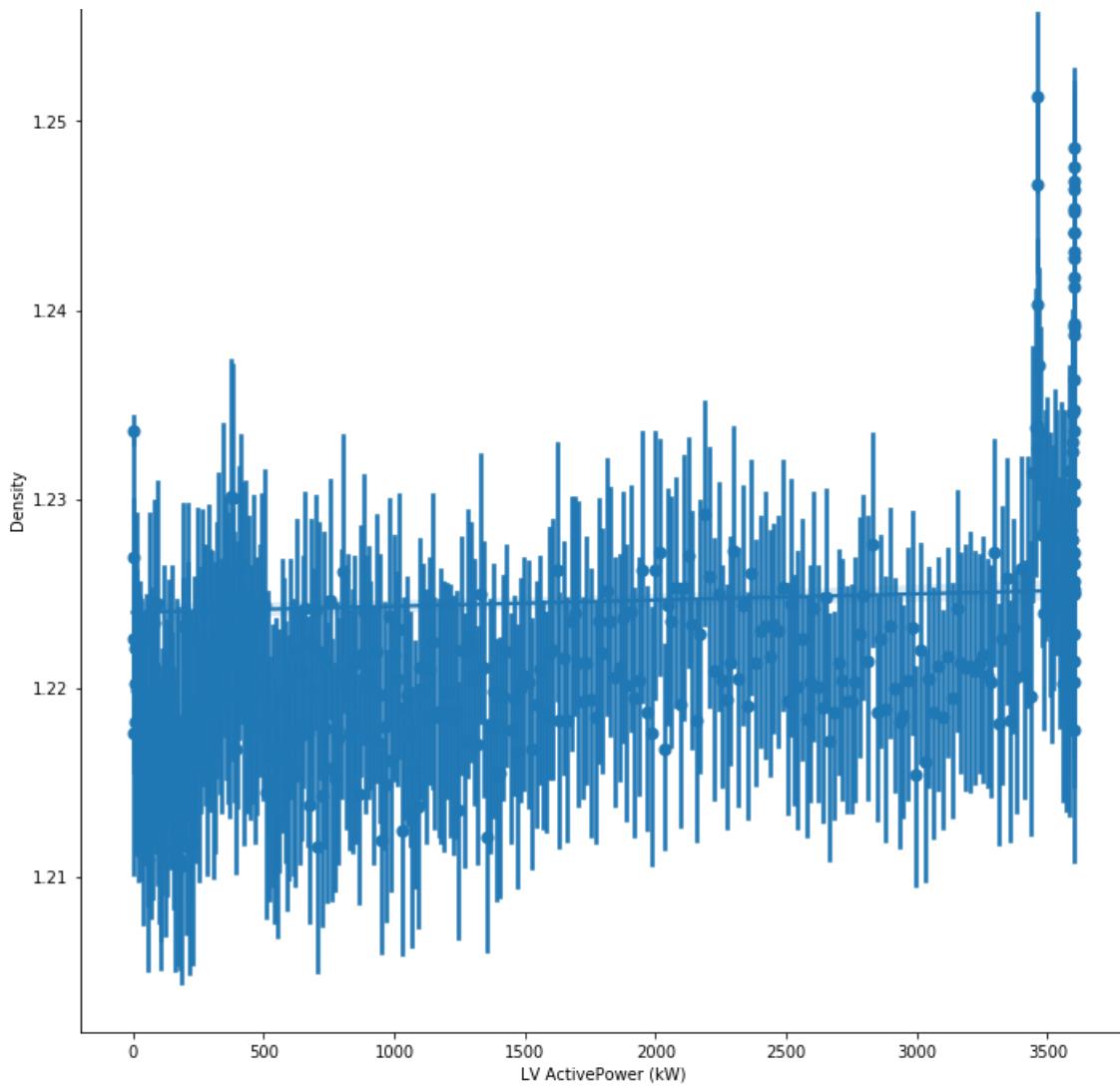


In [109]:

```
sns.lmplot(x="LV ActivePower (kW)", y="Density", data=df, height=10, x_bins=500)  
#Density and Electric Production seem to have little or no linear relation
```

Out[109]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d0073b88>
```

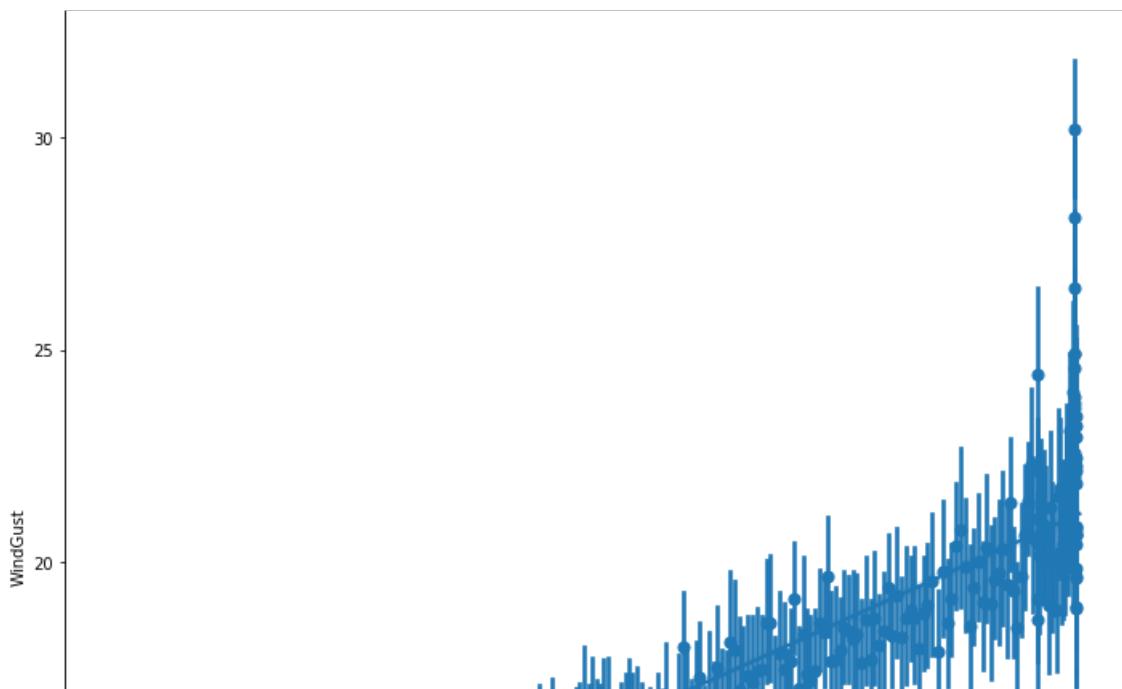


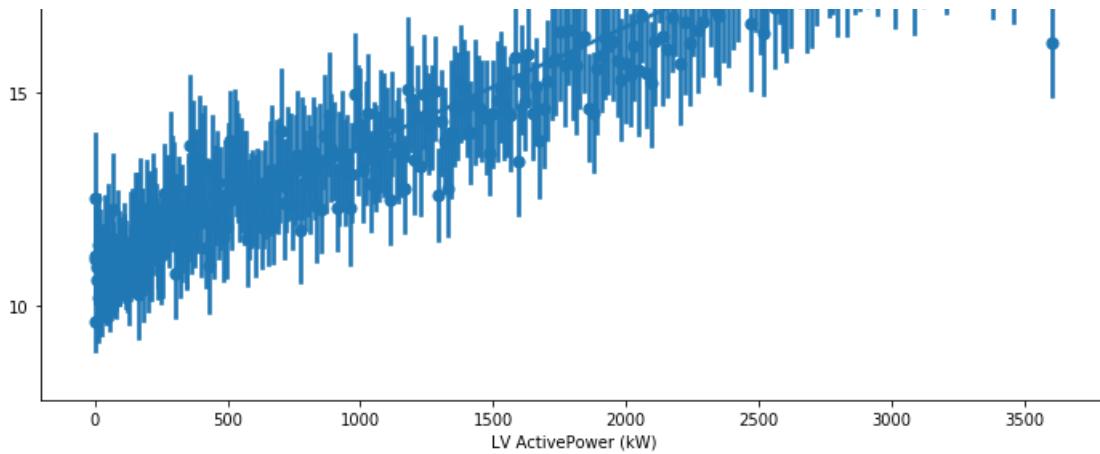
In [110]:

```
sns.lmplot(x="LV ActivePower (kW)",y="WindGust",data=df,height=10,x_bins=500)
#Wind gust and production seem to have strong linear relationship
```

Out[110]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4a06bb388>
```



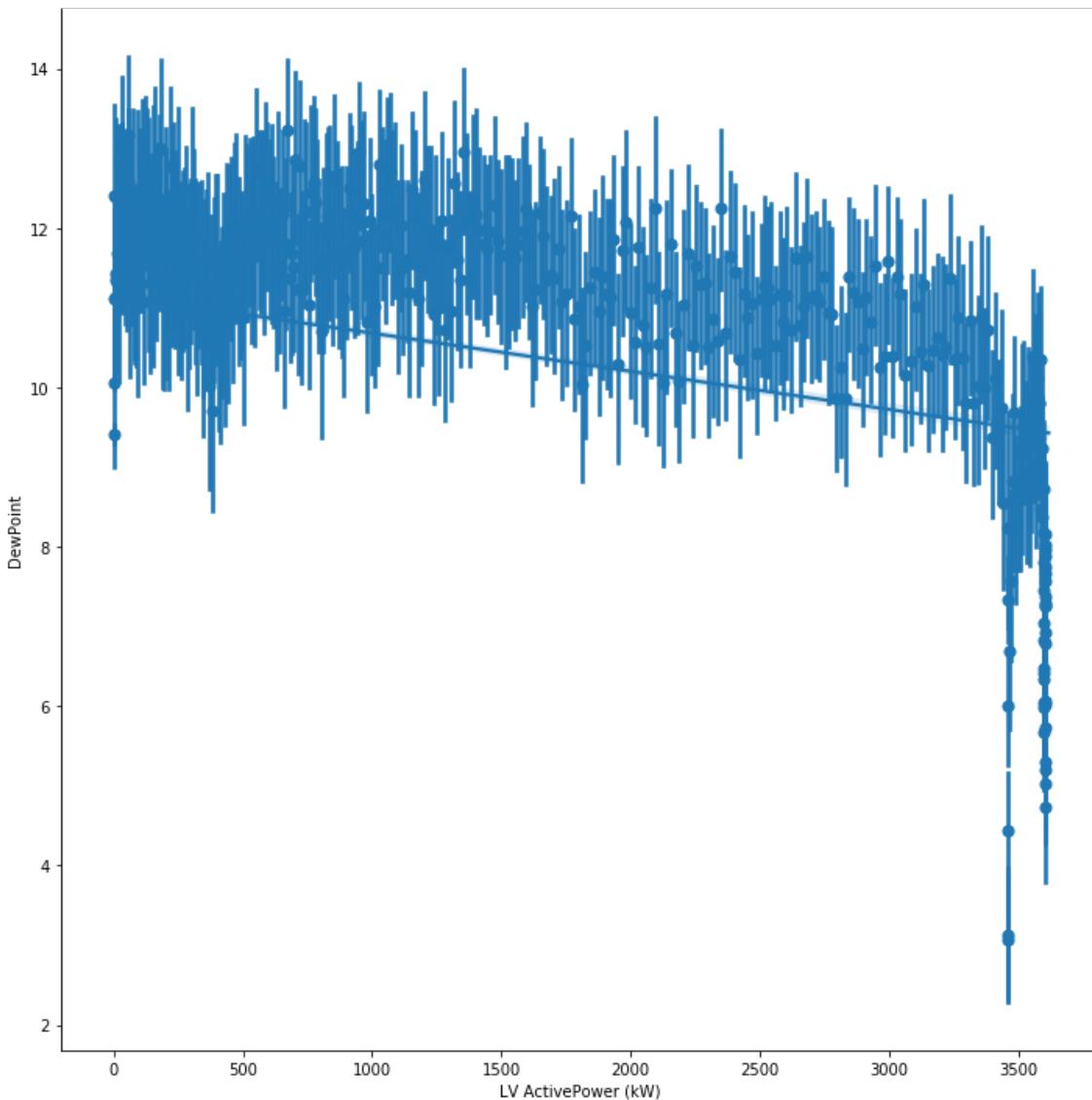


In [111]:

```
sns.lmplot(x="LV ActivePower (kW)",y="DewPoint",data=df,height=10,x_bins=500)
#Dew point and production seem to have a negative linear relationship
```

Out [111]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d1267b08>
```

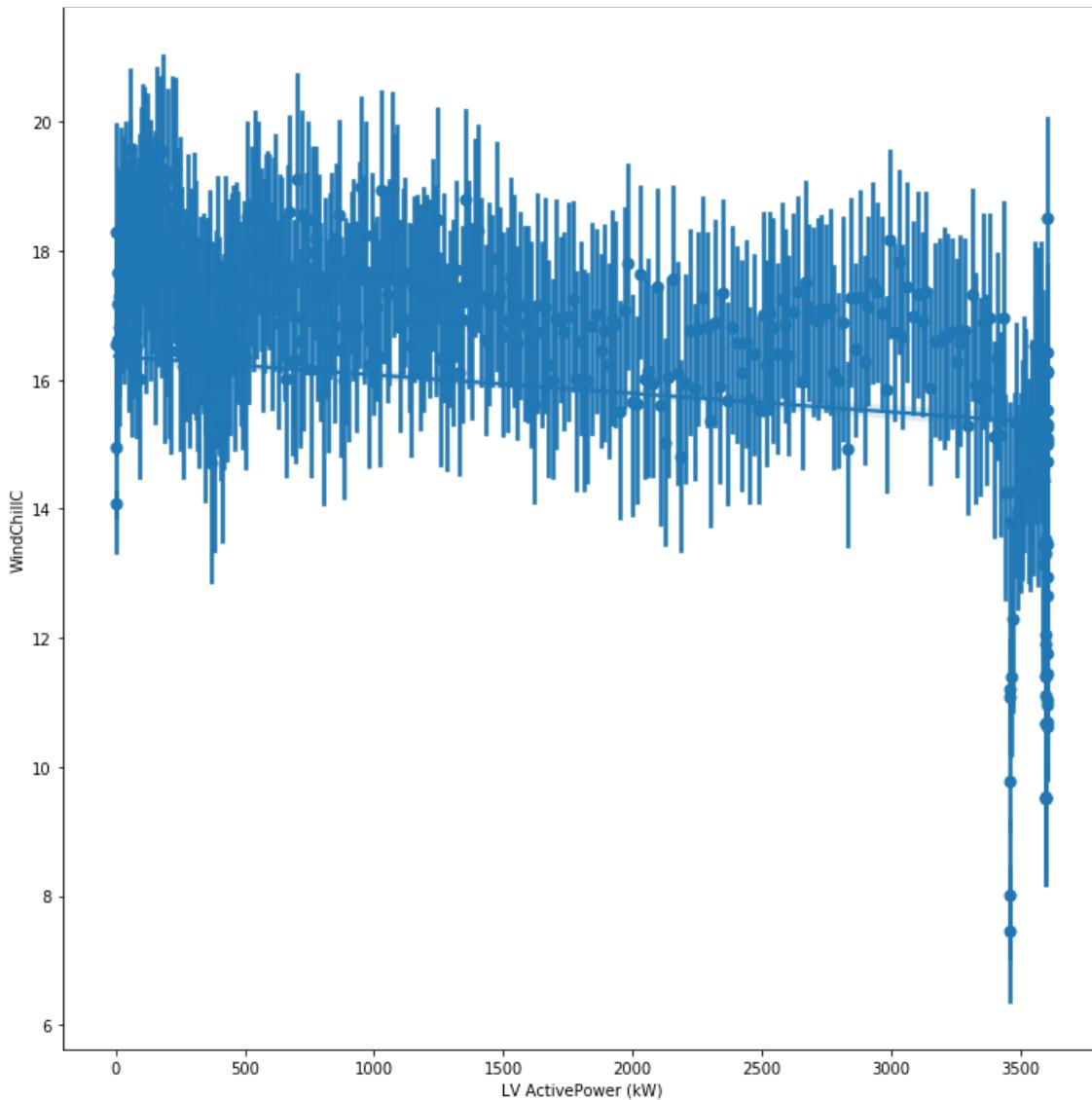


In [112]:

```
sns.lmplot(x="LV ActivePower (kW)",y="WindChillC",data=df,height=10,x_bins=500)
#Electric Production and Wind Chill seem to have a weak negative linear relationship
```

Out [112] :

```
<seaborn.axisgrid.FacetGrid at 0x1c4d1b9ab48>
```

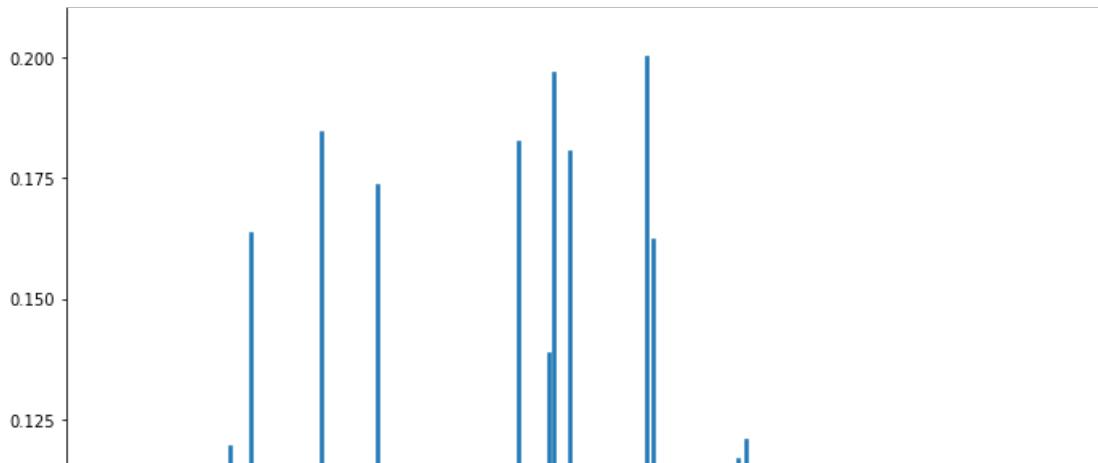


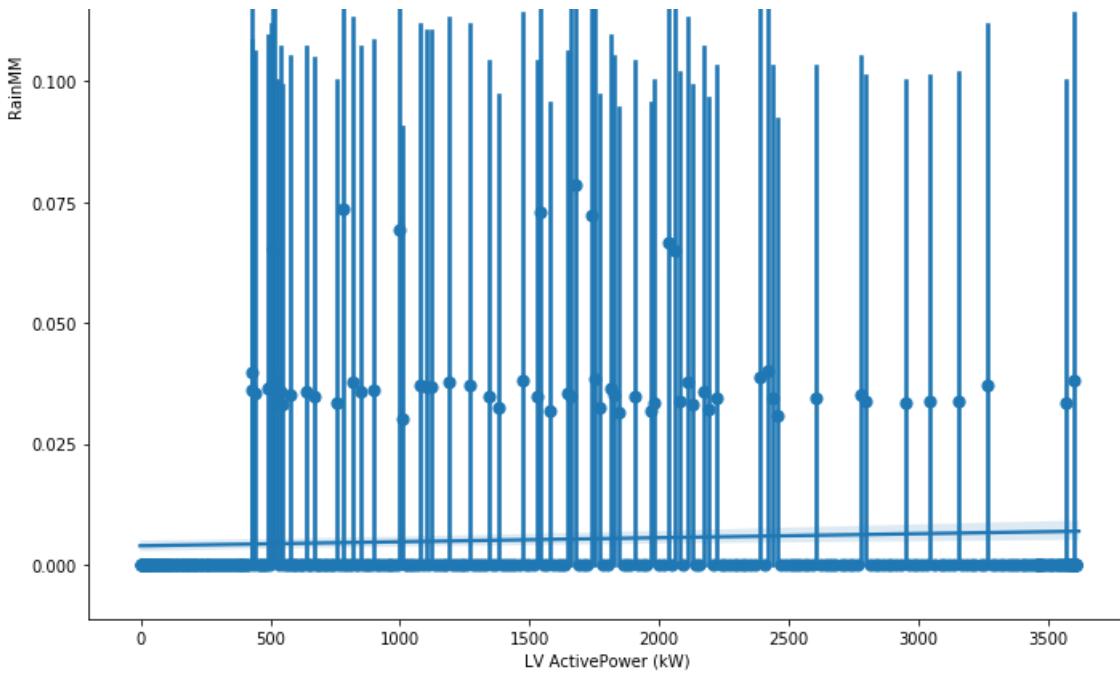
In [113] :

```
sns.lmplot(x="LV ActivePower (kW)",y="RainMM",data=df,height=10,x_bins=500)
#Since most of the rain data is nearly 0 there seem to be no relationship between
#Rain and Production.
```

Out [113] :

```
<seaborn.axisgrid.FacetGrid at 0x1c4d250b048>
```



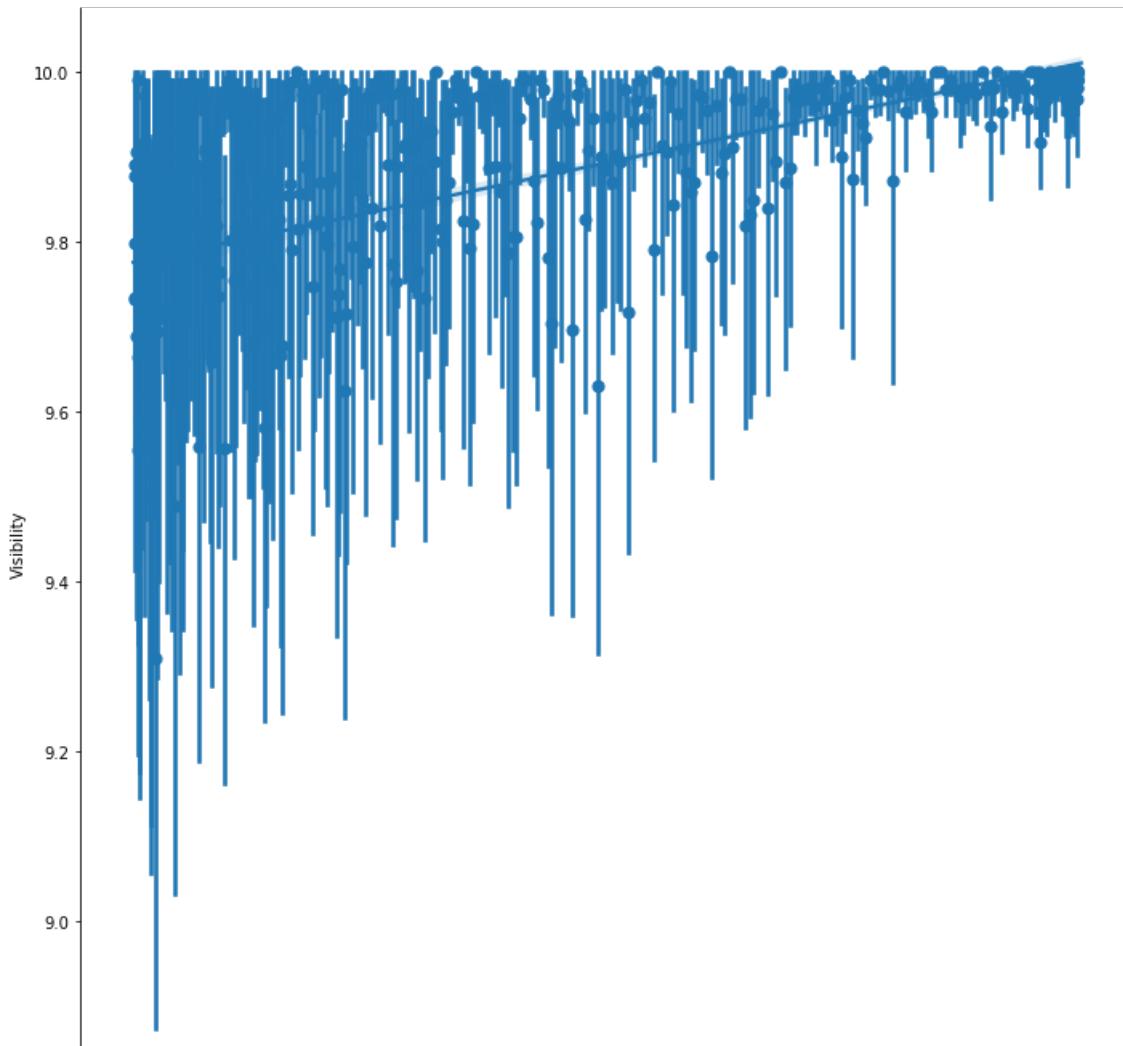


In [114]:

```
sns.lmplot(x="LV ActivePower (kW)",y="Visibility",data=df,height=10,x_bins=500)
#Visibility and Electric Production seem to have a relation but it is not clear
#since much of the visibility scores are aggregated around the same value(10)
#There may be also a non linear relationship as seen from graph
```

Out[114]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d24d4b88>
```



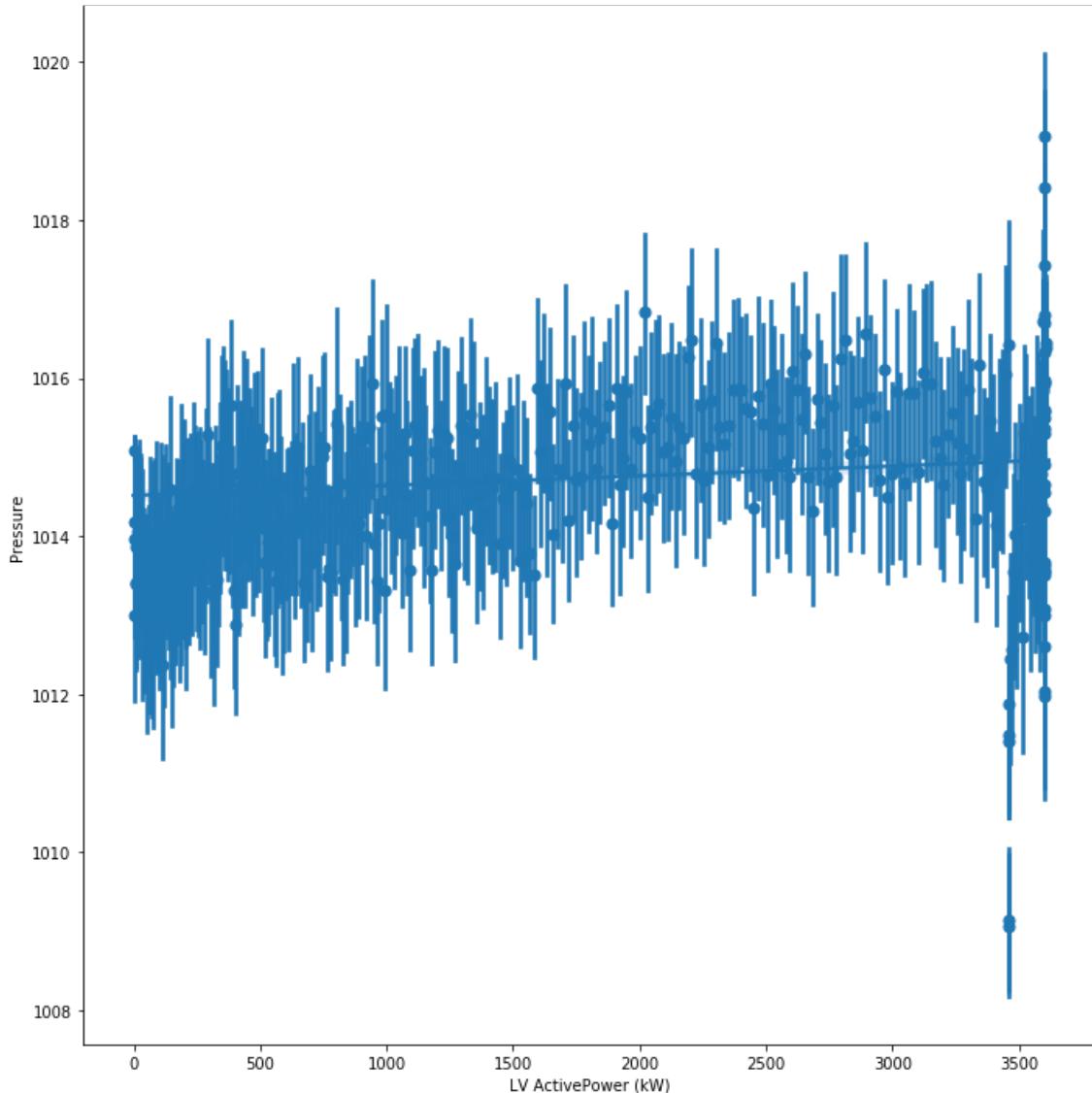
0 500 1000 1500 2000 2500 3000 3500  
LV ActivePower (kW)

In [116]:

```
sns.lmplot(x="LV ActivePower (kW)", y="Pressure", data=df, height=10, x_bins=500)  
#Electric Production and Pressure seem to have a very weak linear relationship
```

Out[116]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d39d0a48>
```

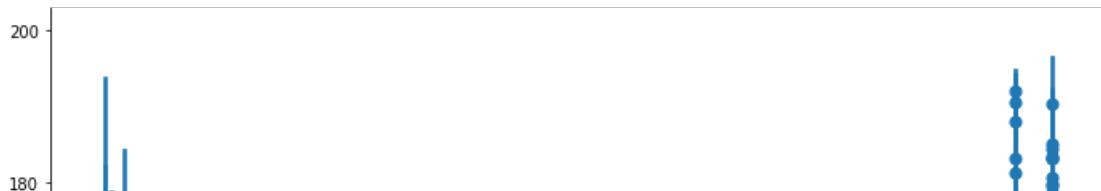


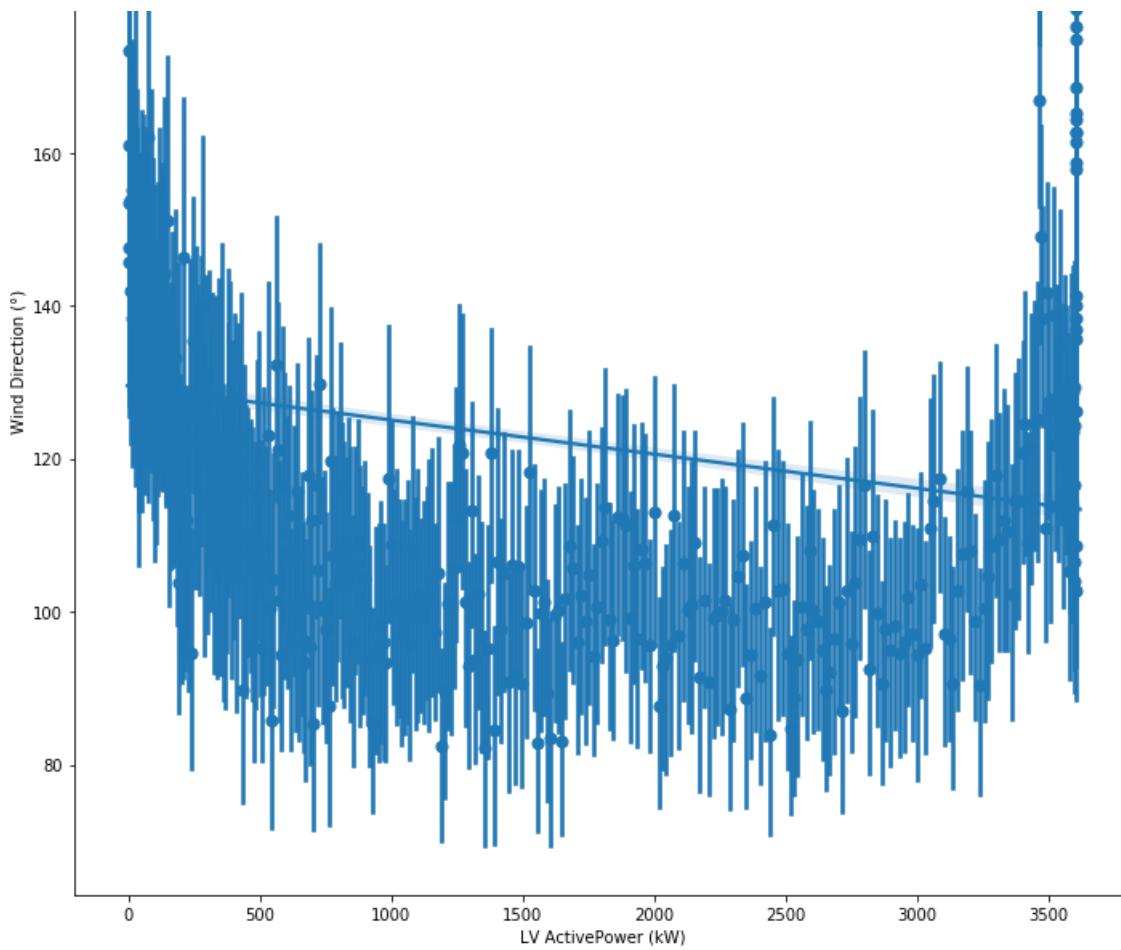
In [142]:

```
sns.lmplot(x="LV ActivePower (kW)", y="Wind Direction (°)", data=df, height=10, x_bins=500)  
#Wind Degrees and Electric Production seem to have a relation but not a linear one. It  
#may be a quadratic relationship as seen from the graph
```

Out[142]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d75e7648>
```





In [139]:

```
import math
import numpy as np
rad=[]
for items in df["Wind Direction (°)"]:
    rad.append(math.degrees(items))

sin_values=np.sin(rad)
sin_values #Converting degrees to sin values to have a normalization on the wind direction degree
df2=pd.DataFrame([df["LV ActivePower (kW)"],sin_values])
df2=df2.T
df2.columns=["LV ActivePower (kW)","Sin"]
df2.head()
```

Out[139]:

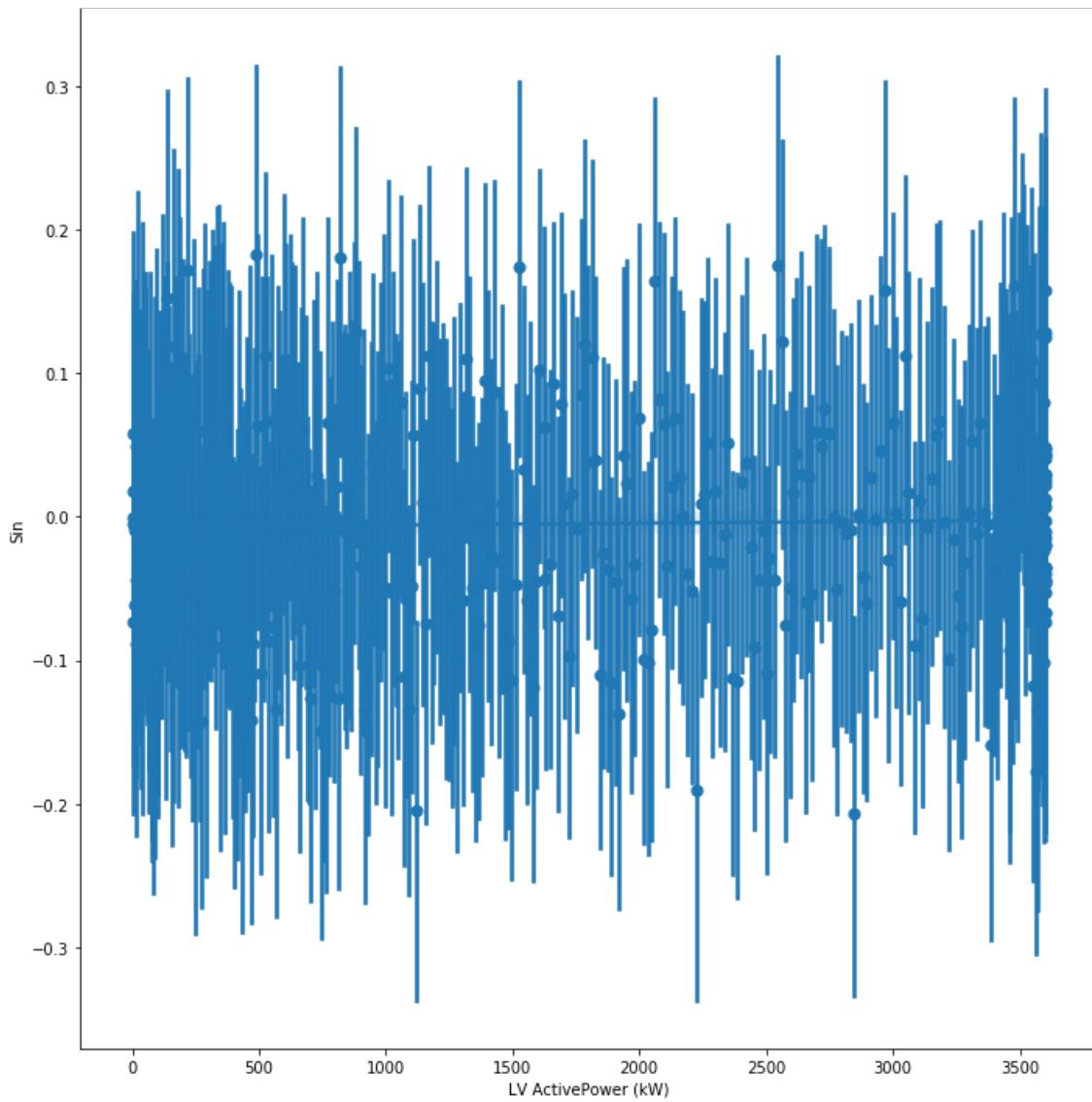
	LV ActivePower (kW)	Sin
0	380.047791	-0.732299
1	453.769196	-0.973390
2	306.376587	0.045024
3	419.645904	-0.465941
4	380.650696	-0.840876

In [141]:

```
sns.lmplot(x="LV ActivePower (kW)",y="Sin",height=10,x_bins=500,data=df2)
#The sinus values of wind direction degrees and Electric Production seem to have
#no relation
```

Out[141]:

<seaborn.axisgrid.FacetGrid at 0x1c4d72bba88>

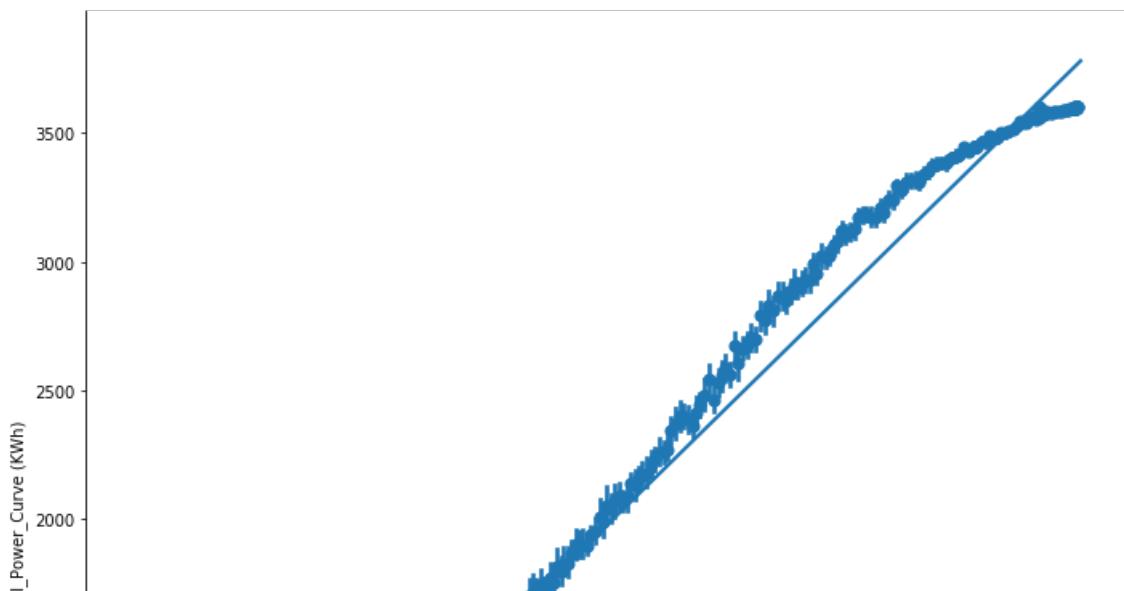


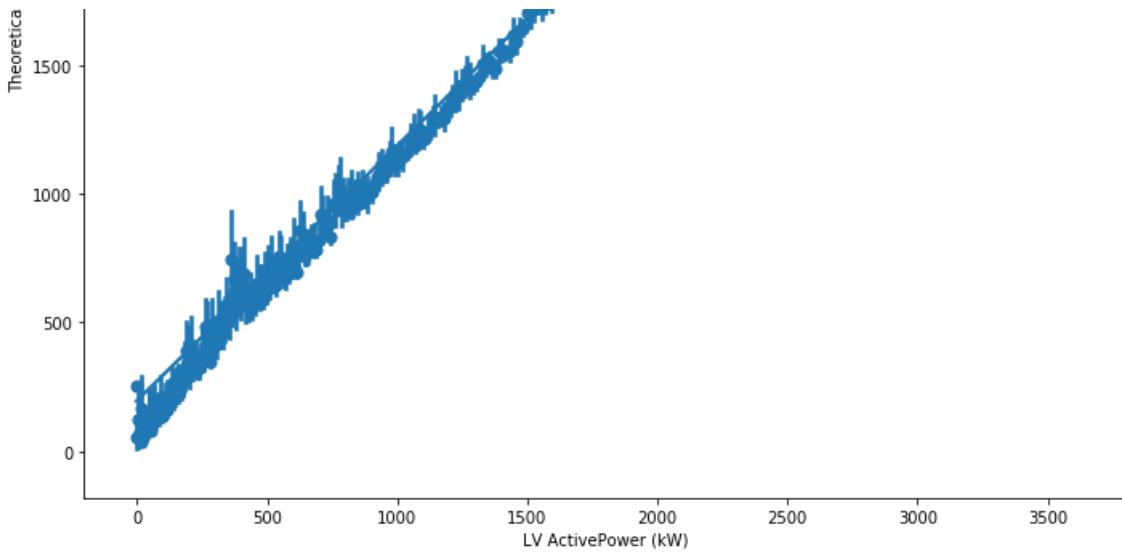
In [143]:

```
sns.lmplot(x="LV ActivePower (kW)",y="Theoretical_Power_Curve (KWh)",height=10,x_bins=500,data=df)
#Theoretical Calculations are directly guess values for Electric production it is not a surprise
#them to have very strong linear relationship
```

Out [143]:

```
<seaborn.axisgrid.FacetGrid at 0x1c4d84d5f48>
```



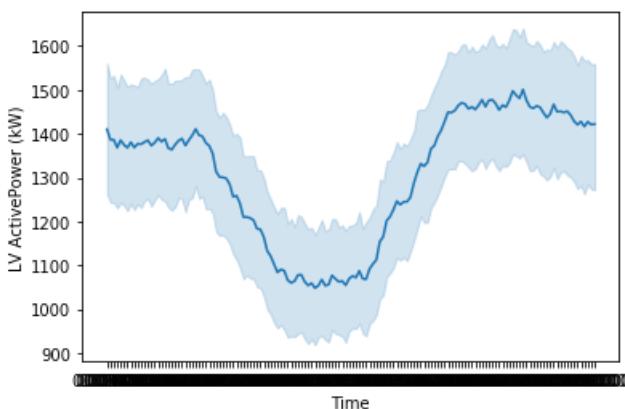


In [64]:

```
sns.lineplot(y=df["LV ActivePower (kW)"],x=df["Time"].astype("str"),data=df) #Change of the Average Electric Production  
# According to time s...
```

Out [64]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c431b68f48>
```

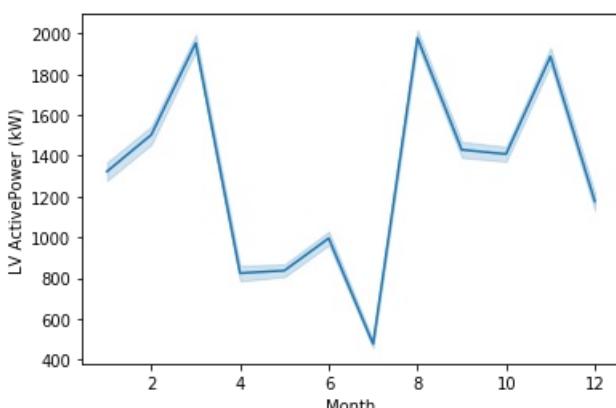


In [65]:

```
sns.lineplot(y=df["LV ActivePower (kW)"],x=df["Month"],data=df)
```

Out [65]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c431aa4048>
```



In [ ]:

In [ ]:

```
In [2]:
```

```
import pandas as pd
import seaborn as sns
```

```
In [3]:
```

```
data=pd.read_excel("C:/Users/Furkan/Desktop/Final Data.xlsx") #Reading Data From Excel File
df=data.copy() # Copying data for just in case
```

```
In [3]:
```

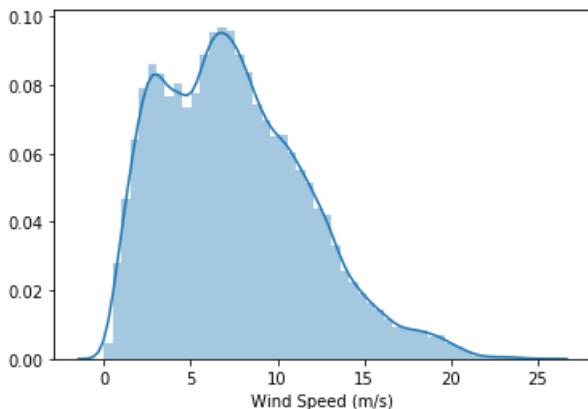
```
#Plots For Wind Speed
```

```
In [4]:
```

```
sns.distplot(df["Wind Speed (m/s)"])
```

```
Out[4]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa520fd88>
```

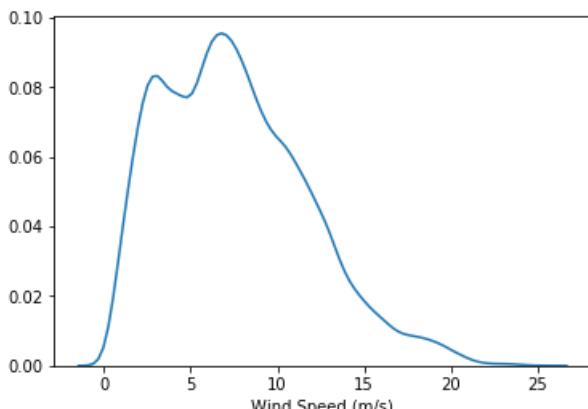


```
In [6]:
```

```
sns.distplot(df["Wind Speed (m/s)"],hist=False)
```

```
Out[6]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa7a05b88>
```



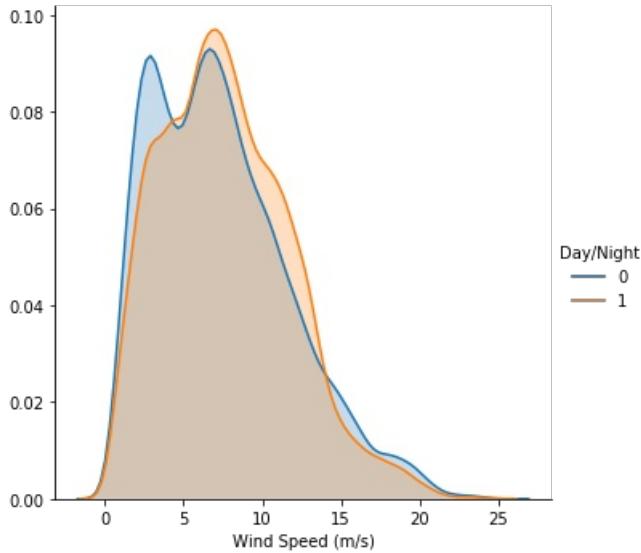
```
In [7]:
```

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Wind Speed (m/s)",shade=True).add_legend()
#When looking in to more detail we can see that our original distribution of the values
```

```
#When looking in to more detail we can see that our original distribution of the values  
#Has 2 peaks but the day sub category has only one peak. The first peak is caused by the night val  
ues
```

Out[7]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa7cda3c8>
```

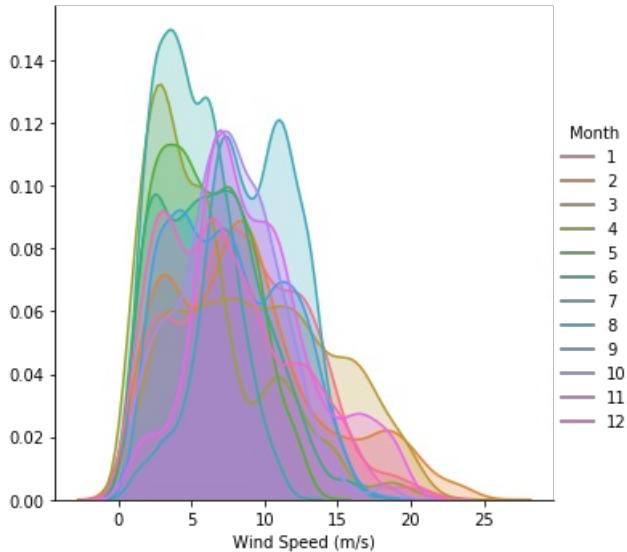


In [8]:

```
sns.FacetGrid(df,hue="Month",height=5).map(sns.kdeplot,"Wind Speed (m/s)",shade=True).add_legend()
```

Out[8]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa7d55bc8>
```



In [ ]:

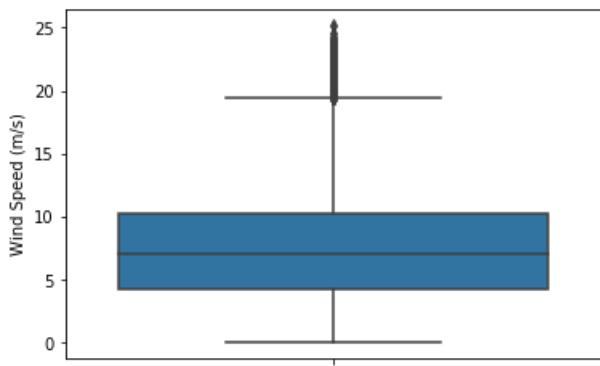
```
#When looking into months, it is far more interesting. We can see there are very different distri  
butions  
#In monthly evaluations
```

In [10]:

```
sns.boxplot(df["Wind Speed (m/s)"],orient="v")
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa8237408>
```



In [11]:

```
#We can see that wind speed data has lots of outliers.
```

In [13]:

```
df[\"Wind Speed (m/s)\"].describe() #After checking statistics, we can see median and mean are similar in this data.
```

Out[13]:

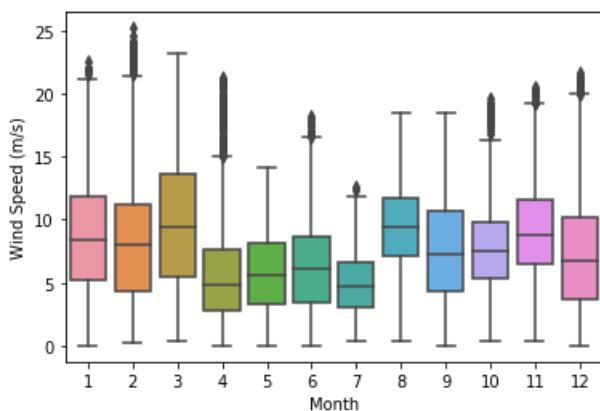
count	50530.000000
mean	7.557952
std	4.227166
min	0.000000
25%	4.201395
50%	7.104594
75%	10.300020
max	25.206011
Name:	Wind Speed (m/s), dtype: float64

In [16]:

```
sns.boxplot(x="Month",y=df[\"Wind Speed (m/s)\"],orient="v",data=df)
#In month from 4 to 7 it seems that wind speeds decrease. It may be because of the seasons effect.
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa1cd3848>
```

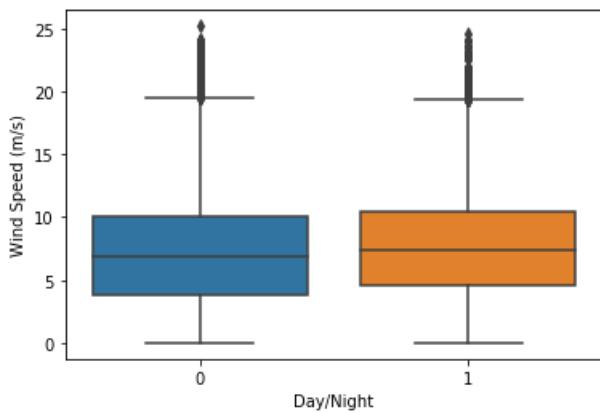


In [17]:

```
sns.boxplot(x="Day/Night",y=df[\"Wind Speed (m/s)\"],orient="v",data=df)
#Day-Night Data seems very similar for Wind Speed
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa3daa908>
```

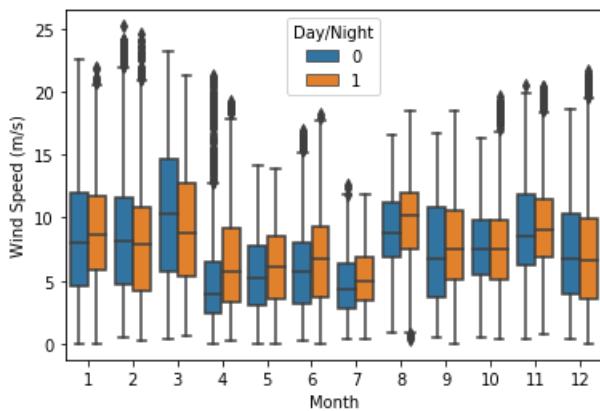


In [18]:

```
sns.boxplot(x="Month",y=df["Wind Speed (m/s)"],orient="v",data=df,hue="Day/Night")
#General Picture
```

Out [18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa321a208>
```



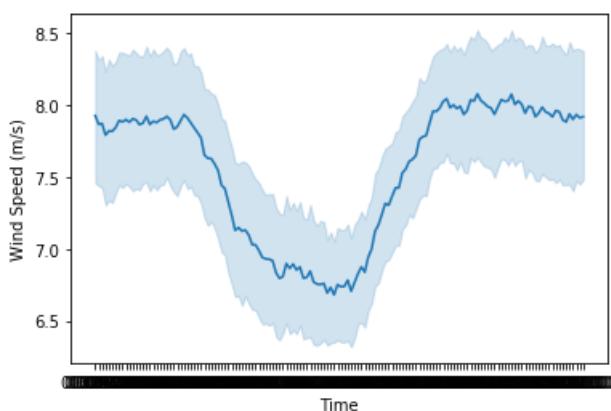
In [19]:

```
sns.lineplot(y=df["Wind Speed (m/s)"],x=df["Time"].astype("str"),data=df) #Change of the Average Electric Production
# According to time slot

#It shows a very similar distribution to LV Active Power -electric production- data.
```

Out [19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa44b3588>
```

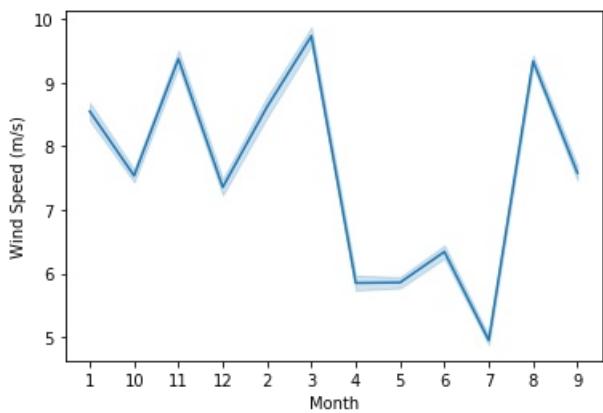


In [20]:

```
sns.lineplot(y=df["Wind Speed (m/s)"],x=df["Month"].astype("str"),data=df)
```

Out [20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa3af69c8>
```

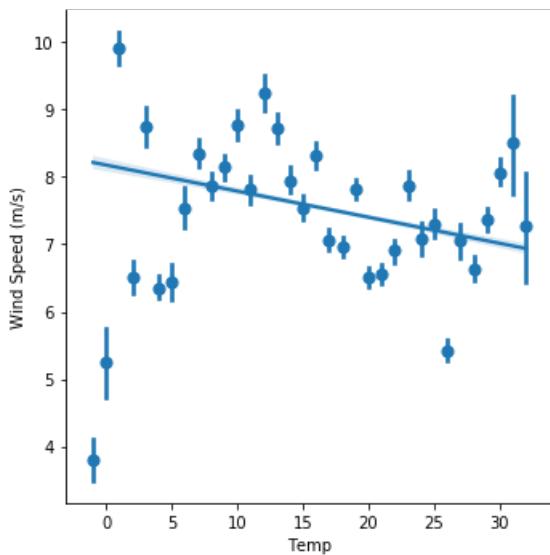


In [31]:

```
sns.lmplot(y="Wind Speed (m/s)",x="Temp",data=df,x_bins=5000)  
#Wind and Temperature does not seem to have a relationship
```

Out [31]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa3420488>
```



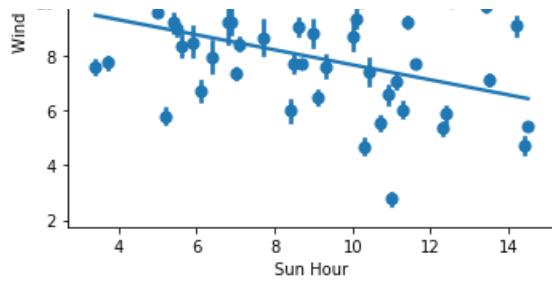
In [79]:

```
sns.lmplot(y="Wind Speed (m/s)",x="Sun Hour",data=df,x_bins=5000)  
#Wind and Sun Hour seem to have a negative linear relationship
```

Out [79]:

```
<seaborn.axisgrid.FacetGrid at 0x21fc184a708>
```



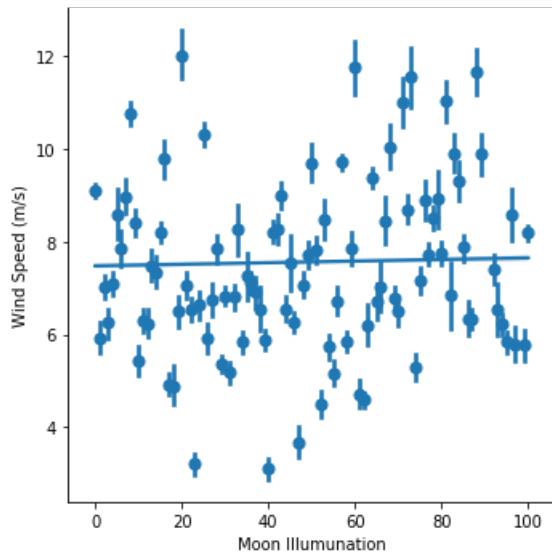


In [80]:

```
sns.lmplot(y="Wind Speed (m/s)",x="Moon Illumunation",data=df,x_bins=5000)
#Wind speed and Moon Illumination does not have a linear relationship
```

Out[80]:

```
<seaborn.axisgrid.FacetGrid at 0x21fc17ede08>
```

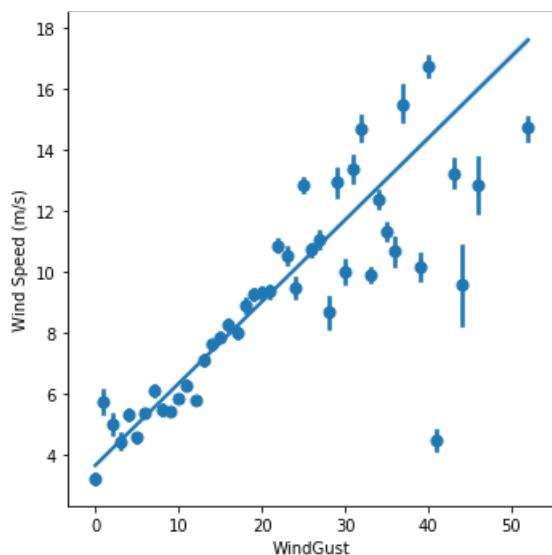


In [29]:

```
sns.lmplot(y="Wind Speed (m/s)",x="WindGust",data=df,x_bins=5000)
#Wind Gust and Wind Speed seem to have linear positive relationship.
#And it makes sense because wind gust measures the sudden speed burst of
#wind
```

Out[29]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa1bc6fc8>
```

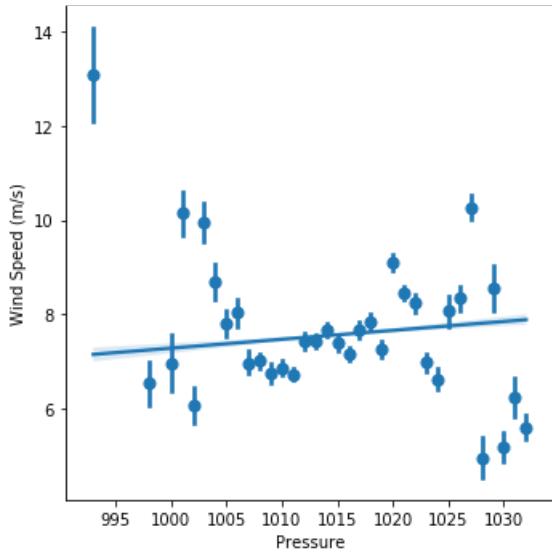


In [35]:

```
sns.lmplot(y="Wind Speed (m/s)",x="Pressure",data=df,x_bins=5000)
#Pressure and wind speed does not seem to have a strong linear relationship
```

Out[35]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa0541608>
```

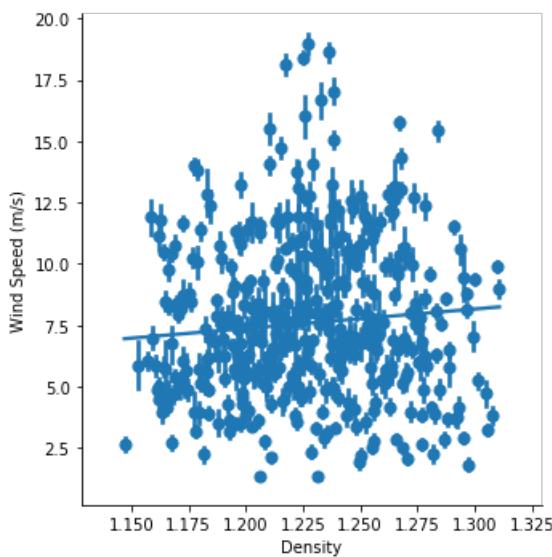


In [36]:

```
sns.lmplot(y="Wind Speed (m/s)",x="Density",data=df,x_bins=5000)
#Wind speed and air density does not show a clear linear relationship
```

Out[36]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa055a9c8>
```

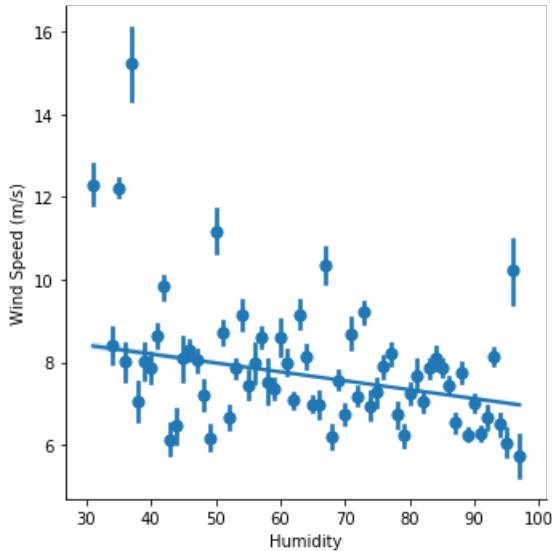


In [37]:

```
sns.lmplot(y="Wind Speed (m/s)",x="Humidity",data=df,x_bins=5000)
#Humidity and Wind Speed seem to have a negative linear relationship
```

Out[37]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa2749188>
```

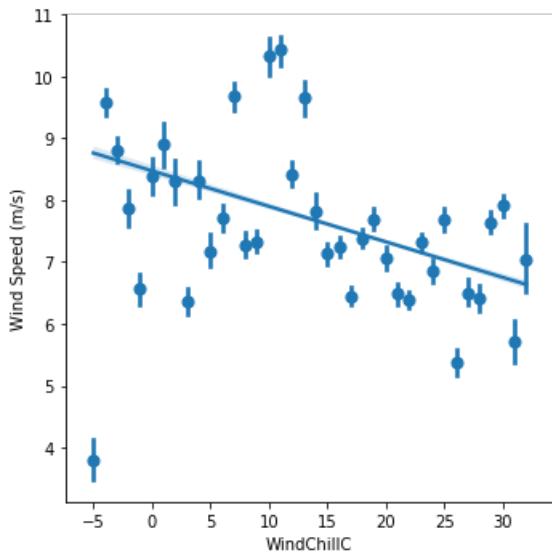


In [39]:

```
sns.lmplot(y="Wind Speed (m/s)",x="WindChillC",data=df,x_bins=5000)
#WindChill and Wind Speed seem to have a negative linear relationship
```

Out [39]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa2d9cc88>
```

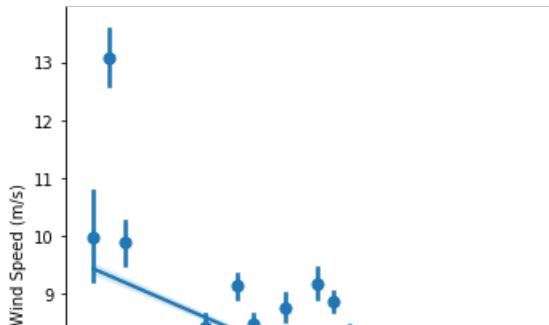


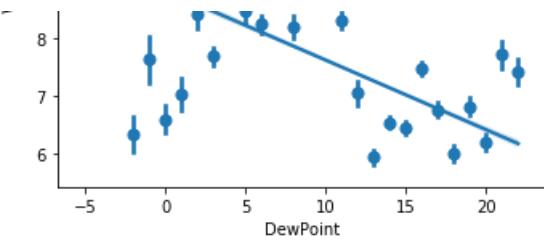
In [40]:

```
sns.lmplot(y="Wind Speed (m/s)",x="DewPoint",data=df,x_bins=5000)
#Dew and Wind Speed seem to have a negative linear relationship
```

Out [40]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa2eeb688>
```



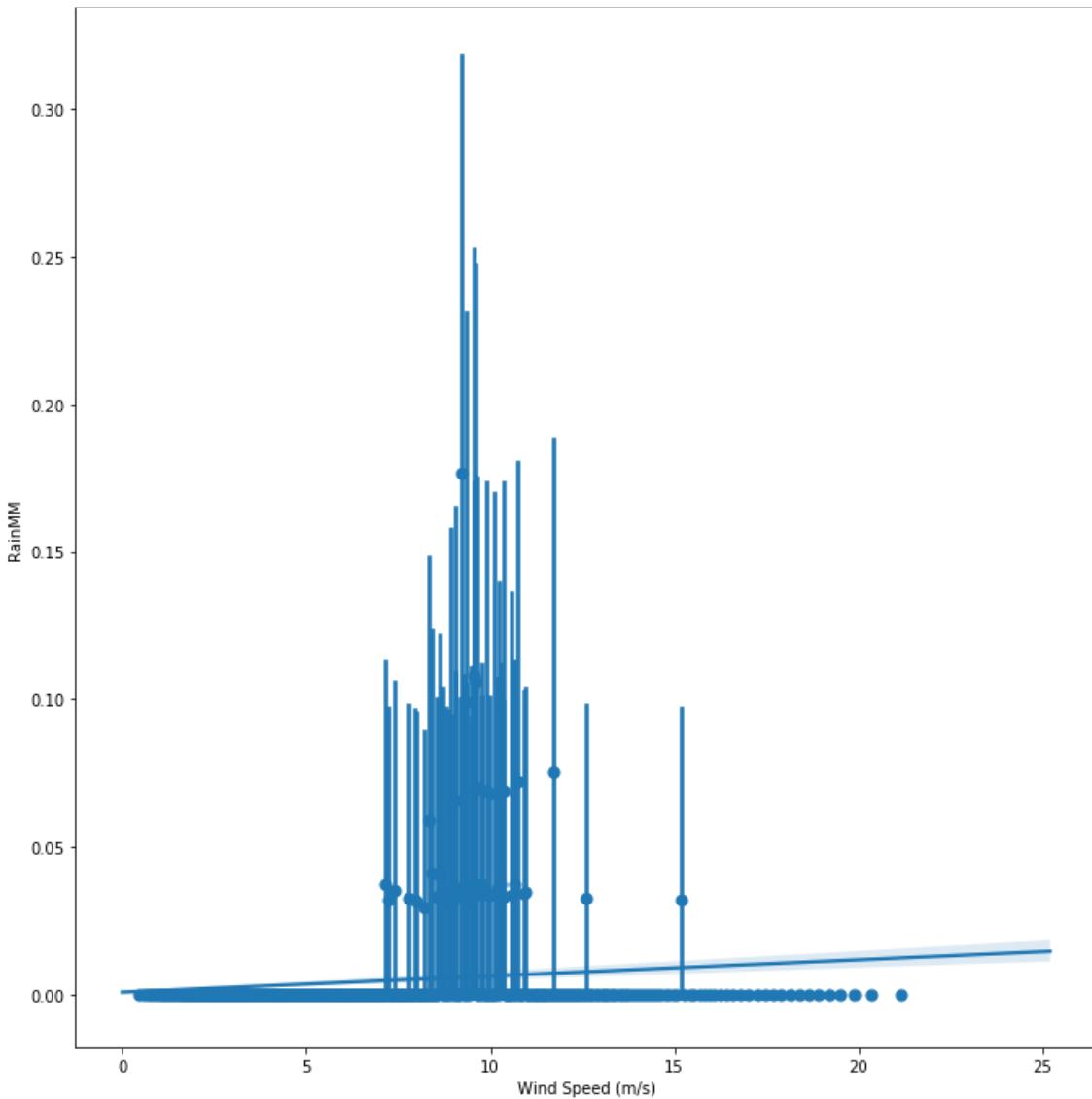


In [81]:

```
sns.lmplot(x="Wind Speed (m/s)",y="RainMM",data=df,height=10,x_bins=500)
#Wind Speed and rain variables does not show a meaningful relationship
```

Out[81]:

```
<seaborn.axisgrid.FacetGrid at 0x21fc1c16c48>
```

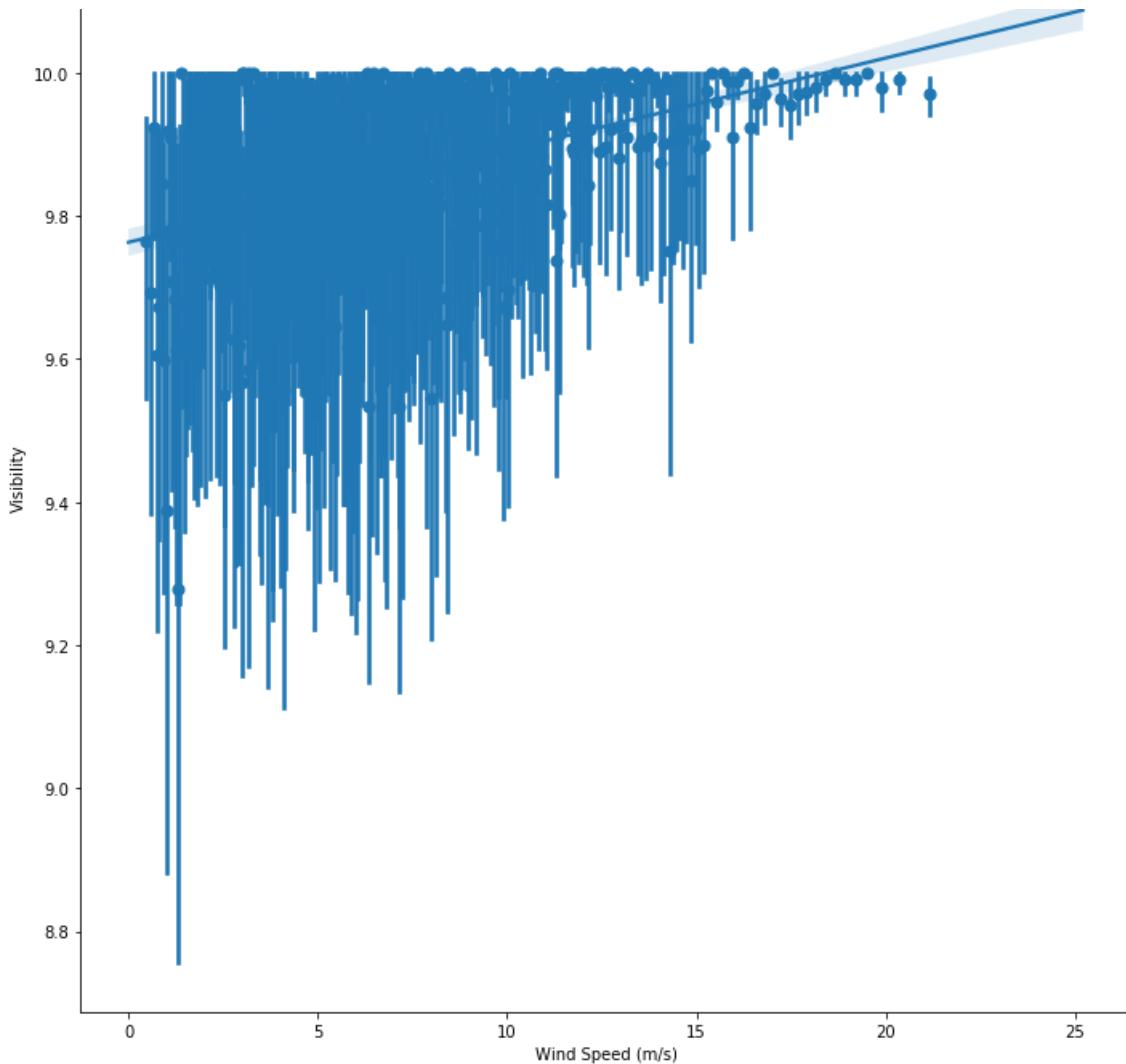


In [82]:

```
sns.lmplot(x="Wind Speed (m/s)",y="Visibility",data=df,height=10,x_bins=500)
#Wind Speed and Visibility seem to have a positive linear relationship
```

Out[82]:

```
<seaborn.axisgrid.FacetGrid at 0x21fc278ffc8>
```



In [41]:

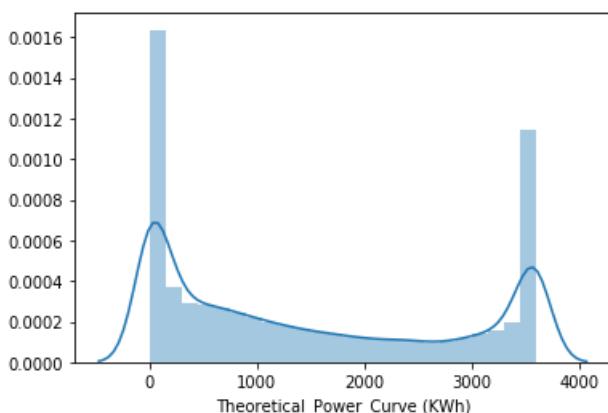
```
#Theoretical_Power_Curve (KWh) Plots  
#The theoretical calculations for Electric Production Values
```

In [43]:

```
sns.distplot(df["Theoretical_Power_Curve (KWh)"])  
#The distribution seems quite similar to real electric production values
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa3159e88>
```

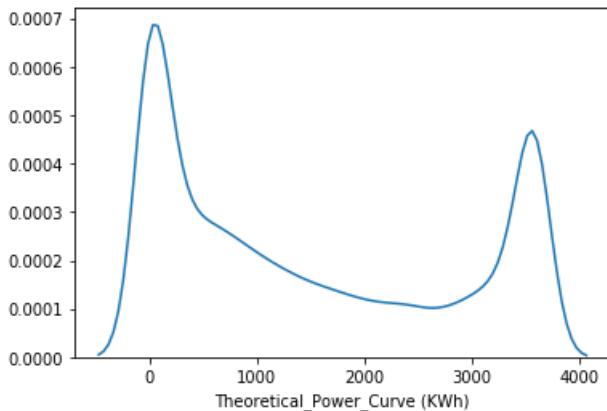


In [44]:

```
sns.distplot(df["Theoretical Power Curve (KWh)"], hist=False)
```

Out [44]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa061ca08>
```

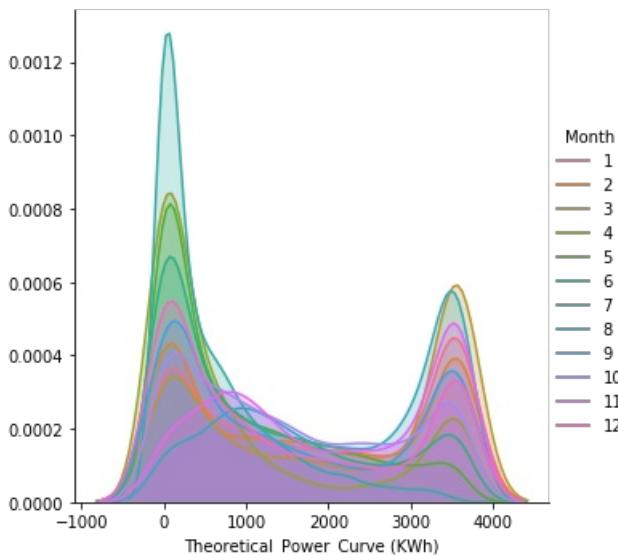


In [45]:

```
sns.FacetGrid(df,hue="Month",height=5).map(sns.kdeplot,"Theoretical_Power_Curve (KWh)",shade=True).add_legend()
```

Out [45]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa2065788>
```

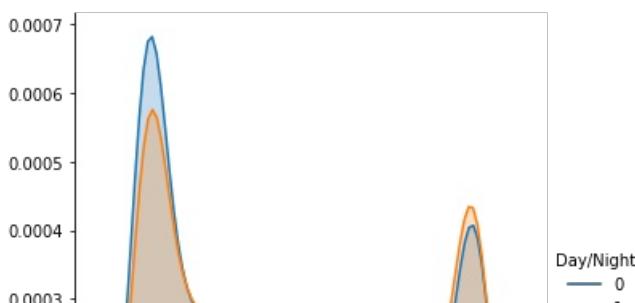


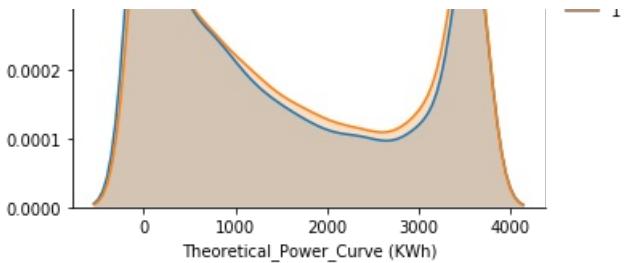
In [46]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Theoretical_Power_Curve (KWh)",shade=True).add_legend()
```

Out [46]:

```
<seaborn.axisgrid.FacetGrid at 0x21fa2ad8fc8>
```



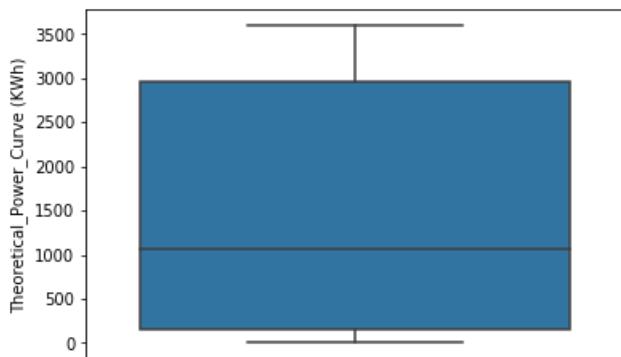


In [47]:

```
sns.boxplot(df["Theoretical_Power_Curve (KWh)"], orient="v")
```

Out[47]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fa23fdd08>
```

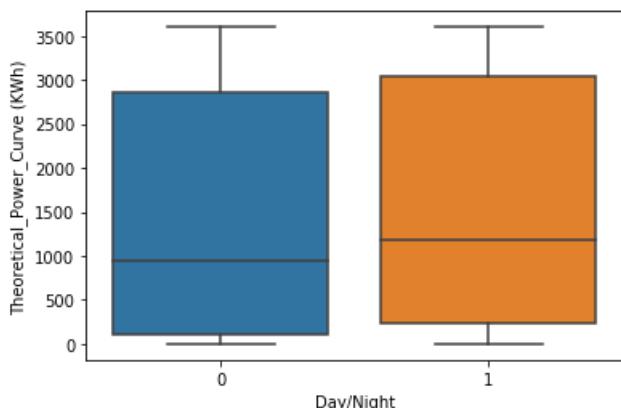


In [53]:

```
sns.boxplot(x="Day/Night", y=df["Theoretical_Power_Curve (KWh)"], orient="v", data=df)
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fb593d688>
```

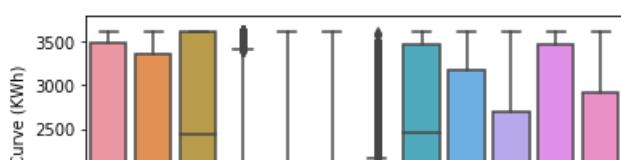


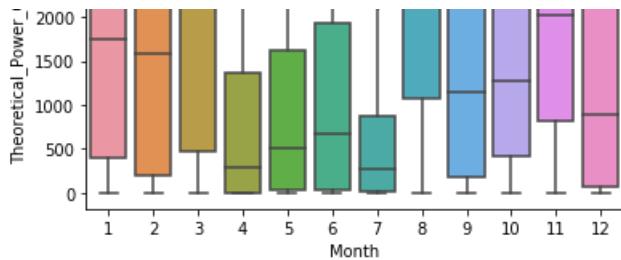
In [54]:

```
sns.boxplot(x="Month", y=df["Theoretical_Power_Curve (KWh)"], orient="v", data=df)
```

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fb7467508>
```



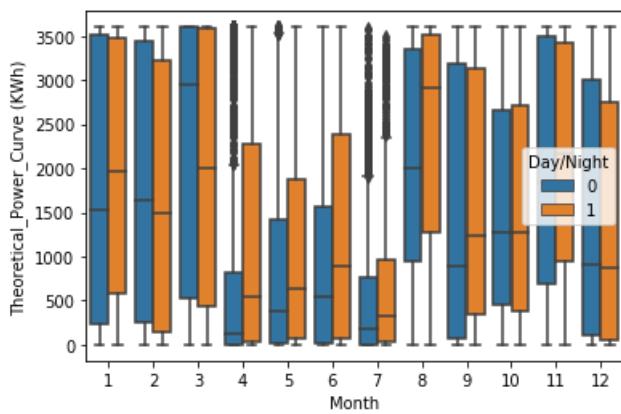


In [55]:

```
sns.boxplot(x="Month",y=df["Theoretical_Power_Curve (KWh)"],orient="v",data=df,hue="Day/Night")
```

Out [55]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fb7a2ed08>
```

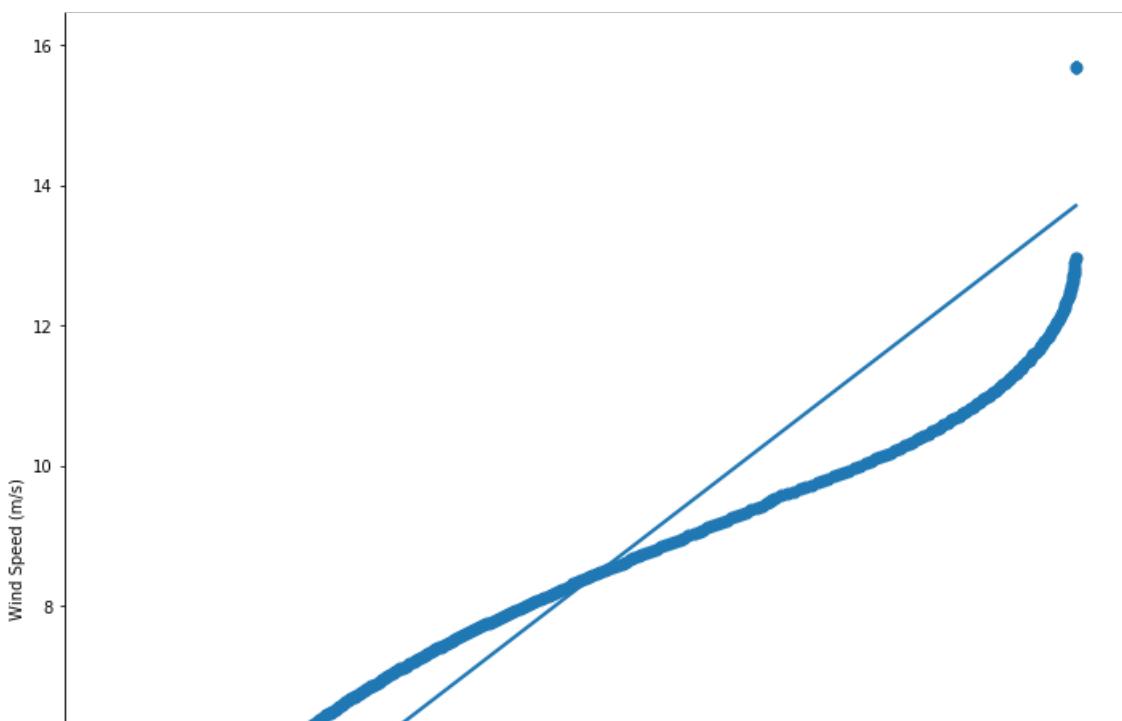


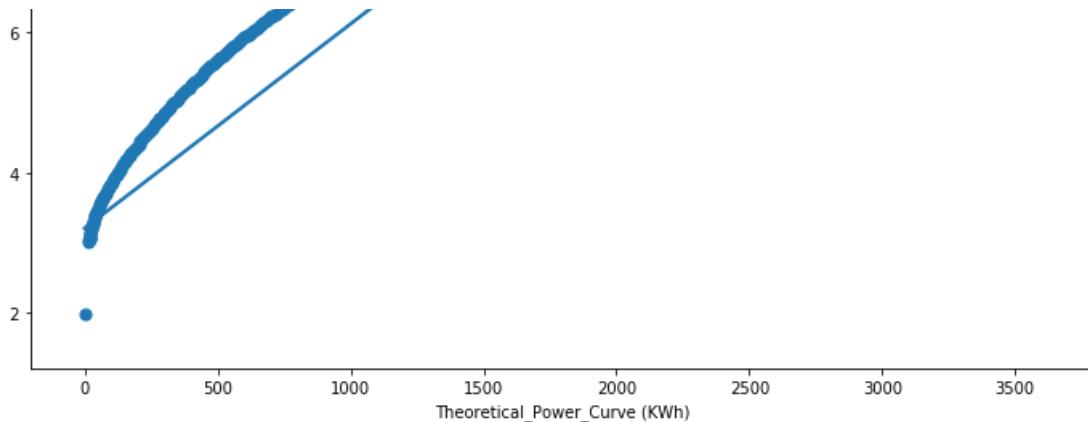
In [56]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)",y="Wind Speed (m/s)",data=df,height=10,x_bins=500)
#It seems a bit more diverse than the LV Active Power-Wind Graph , it may be because of the
#theoretical formula does not imply enough weight to the wind but in real life the wind speed actually
#matters more
```

Out [56]:

```
<seaborn.axisgrid.FacetGrid at 0x21fb8007648>
```



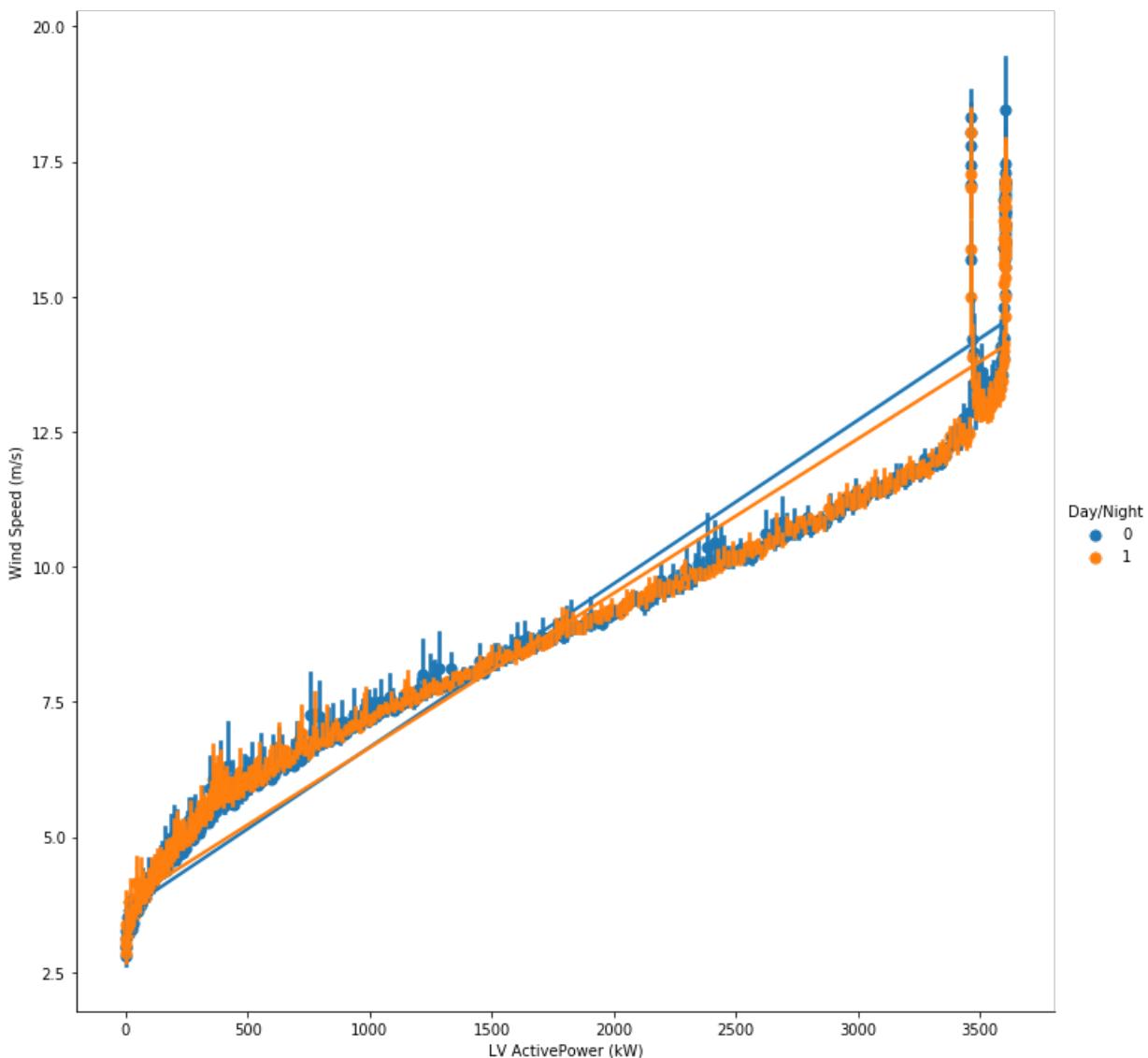


In [57]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="Wind Speed (m/s)", data=df, height=10, x_bins=500, hue="Day/Night")
```

Out[57]:

```
<seaborn.axisgrid.FacetGrid at 0x21fc4a8a0c8>
```

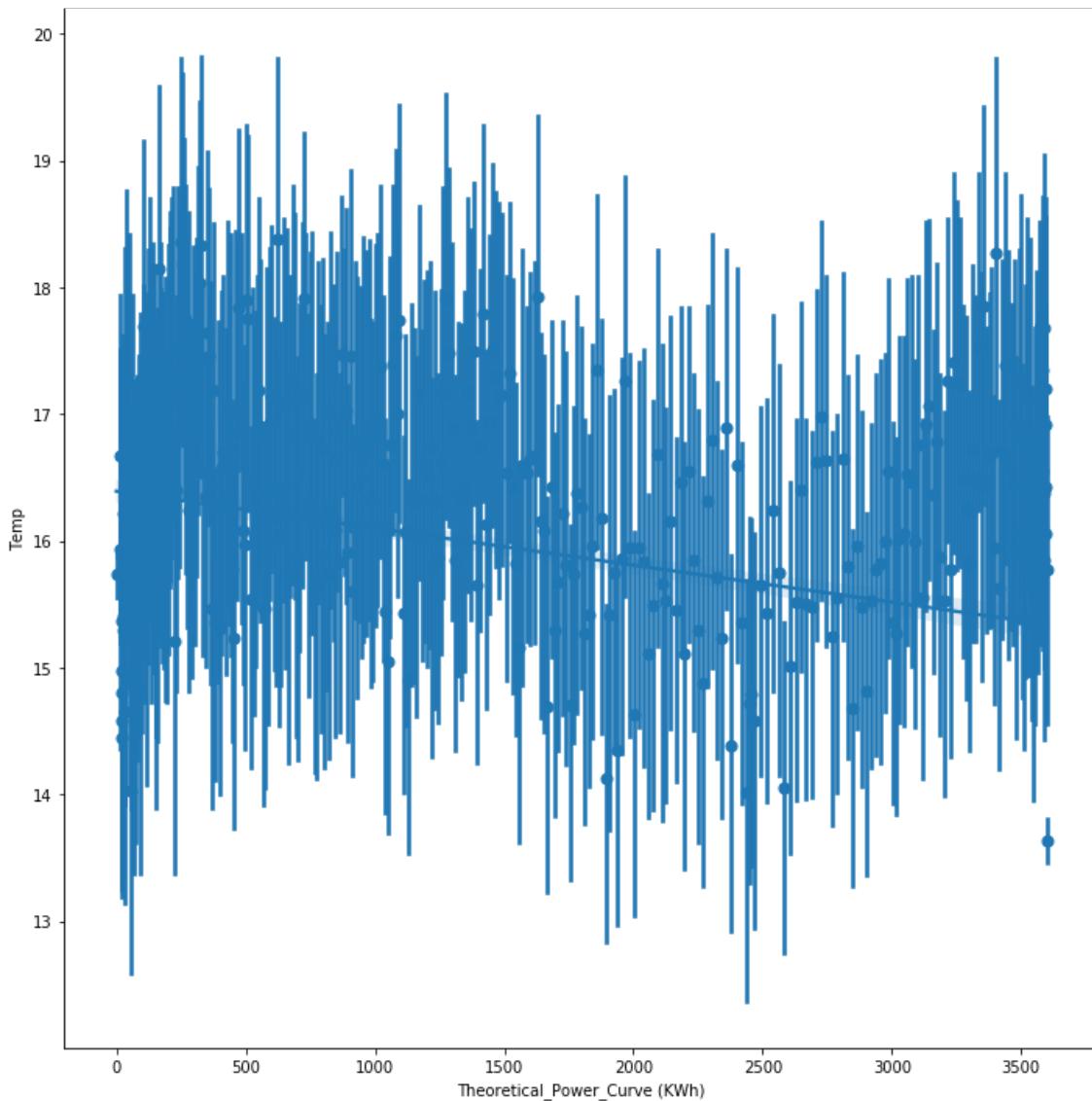


In [60]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="Temp", data=df, height=10, x_bins=500)
#Theoretical Value shows stronger linear relationship with temperature than the real production values
```

Out [60]:

```
<seaborn.axisgrid.FacetGrid at 0x21fc433cf8>
```

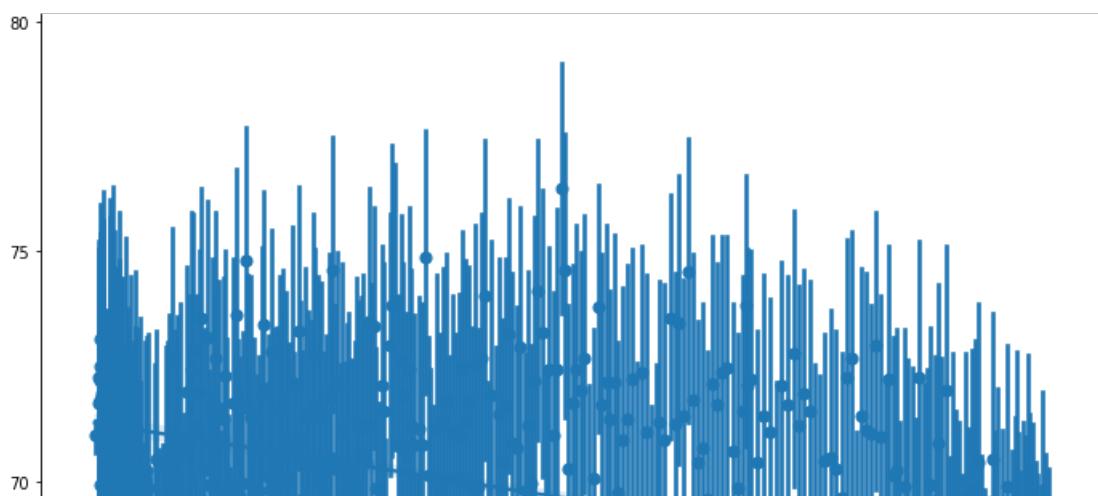


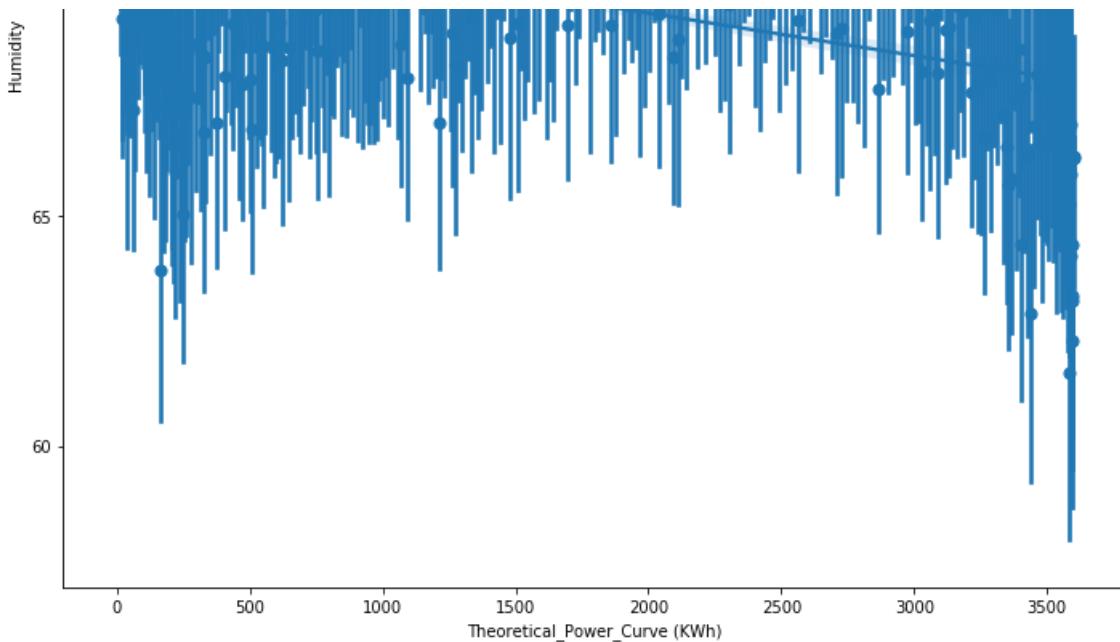
In [62]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="Humidity", data=df, height=10, x_bins=500)  
#Humidity and Theoretic value seem to have negative linear relationship
```

Out [62]:

```
<seaborn.axisgrid.FacetGrid at 0x21fbdb466e48>
```



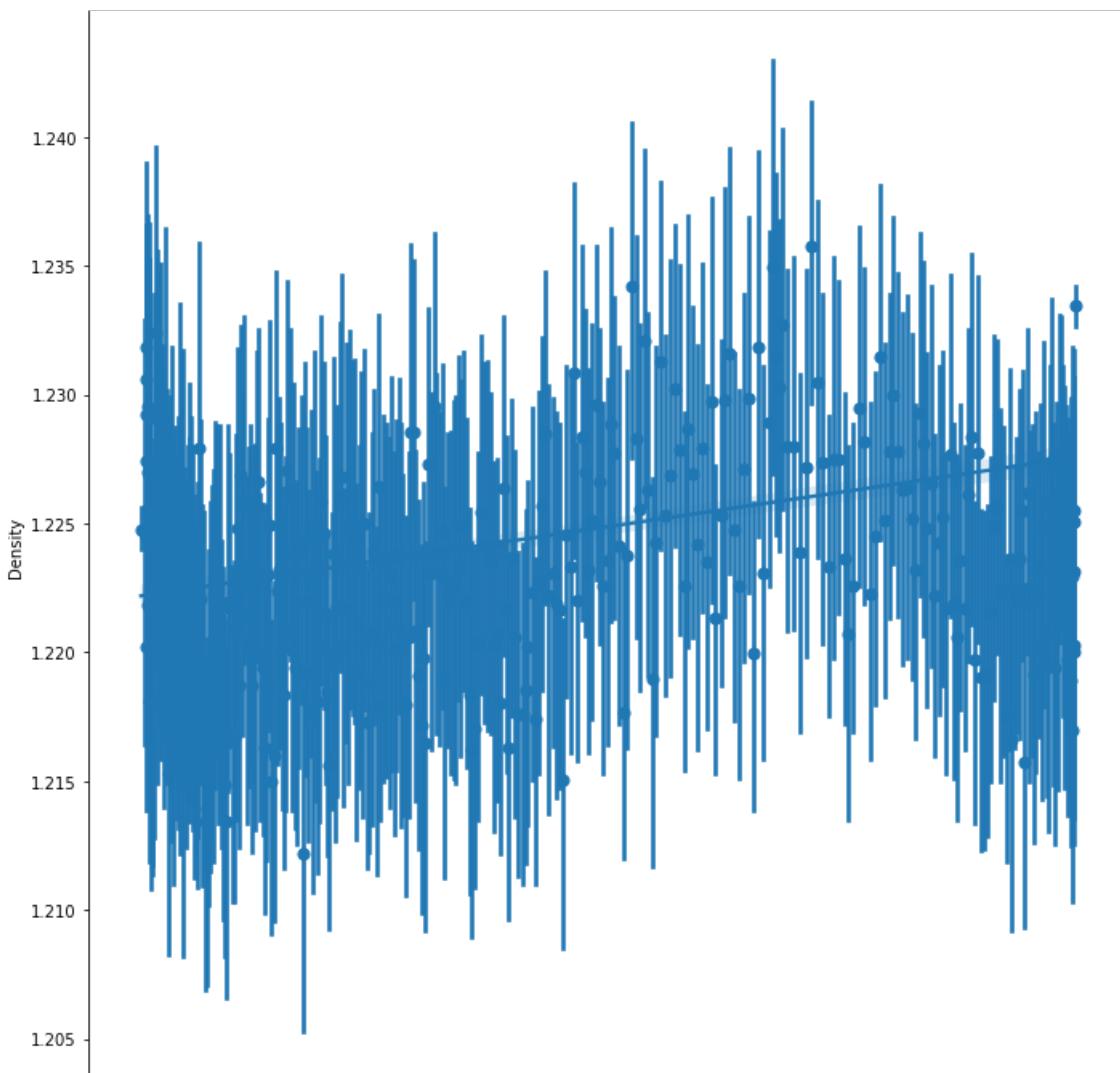


In [63]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="Density", data=df, height=10, x_bins=500)
#Density and Theoretical value seem to have linear relationship. The real production values
#Little or no relationship with the Density. These differences may be the flaws of theoretical
#calculation
```

Out[63]:

```
<seaborn.axisgrid.FacetGrid at 0x21fc2e16888>
```



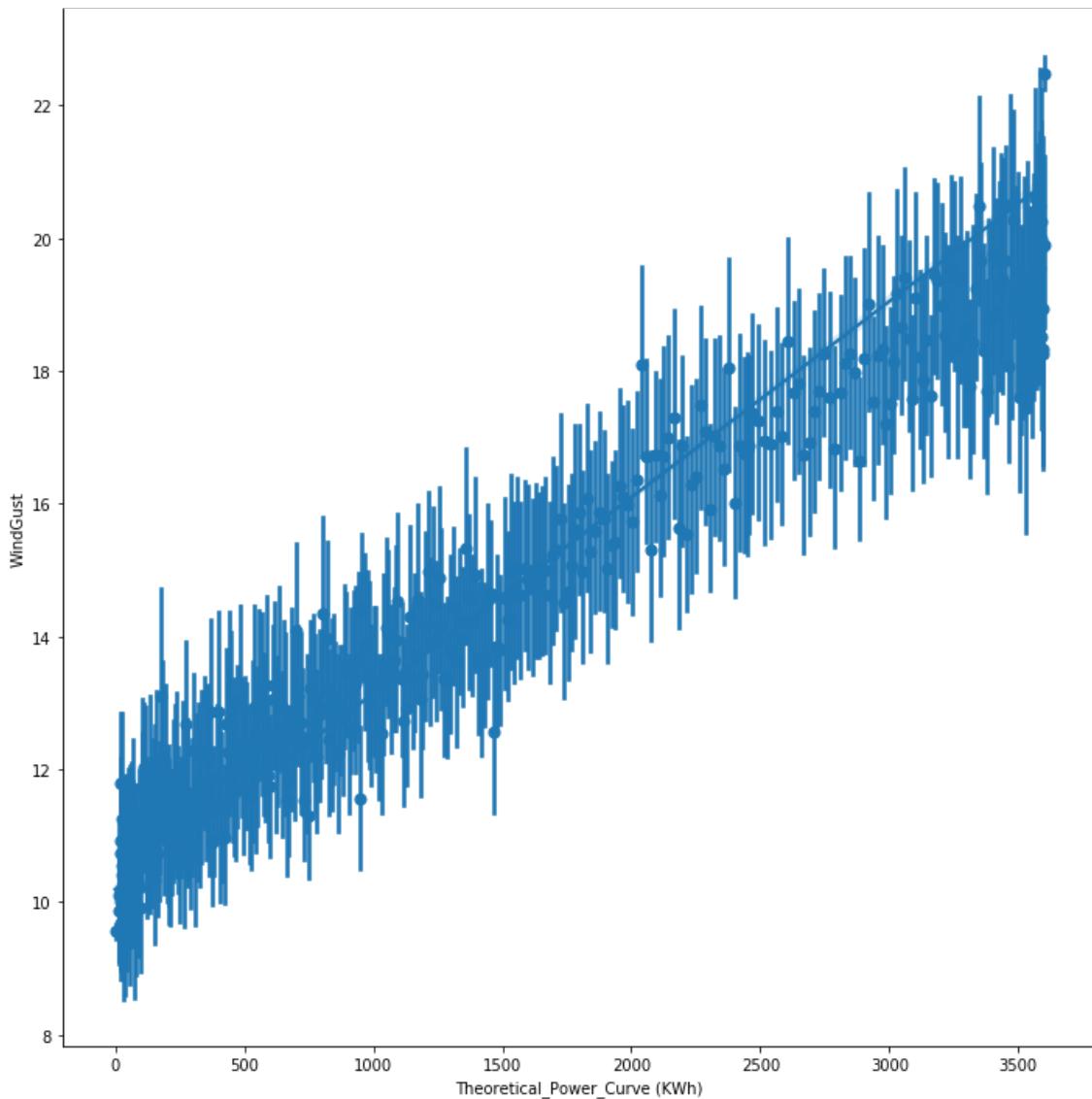


In [64]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="WindGust", data=df, height=10, x_bins=500)
#Wind gust and theoretical value seem to have strong linear relationship. The relationship
#is stronger than the real production values' relation ship with wind gust. But also it involves
#More error than the real production values-Windgust plot
```

Out [64]:

<seaborn.axisgrid.FacetGrid at 0x21fb9a13088>



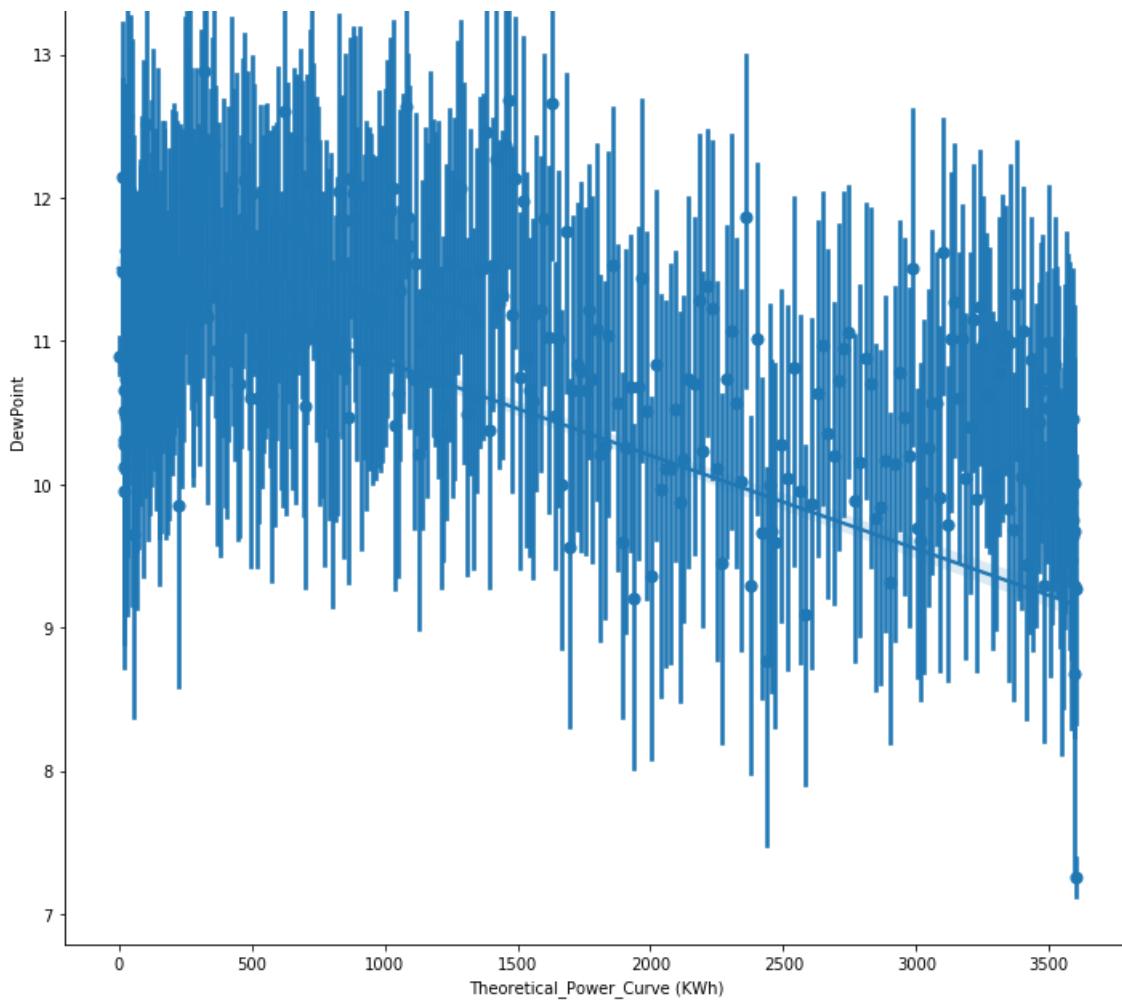
In [65]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="DewPoint", data=df, height=10, x_bins=500)
#Dew point and theoretical value seem to have a negative linear relationship similar to the procti
on values' plot
#with dew point but not exactly the same
```

Out [65]:

<seaborn.axisgrid.FacetGrid at 0x21fba2ae848>



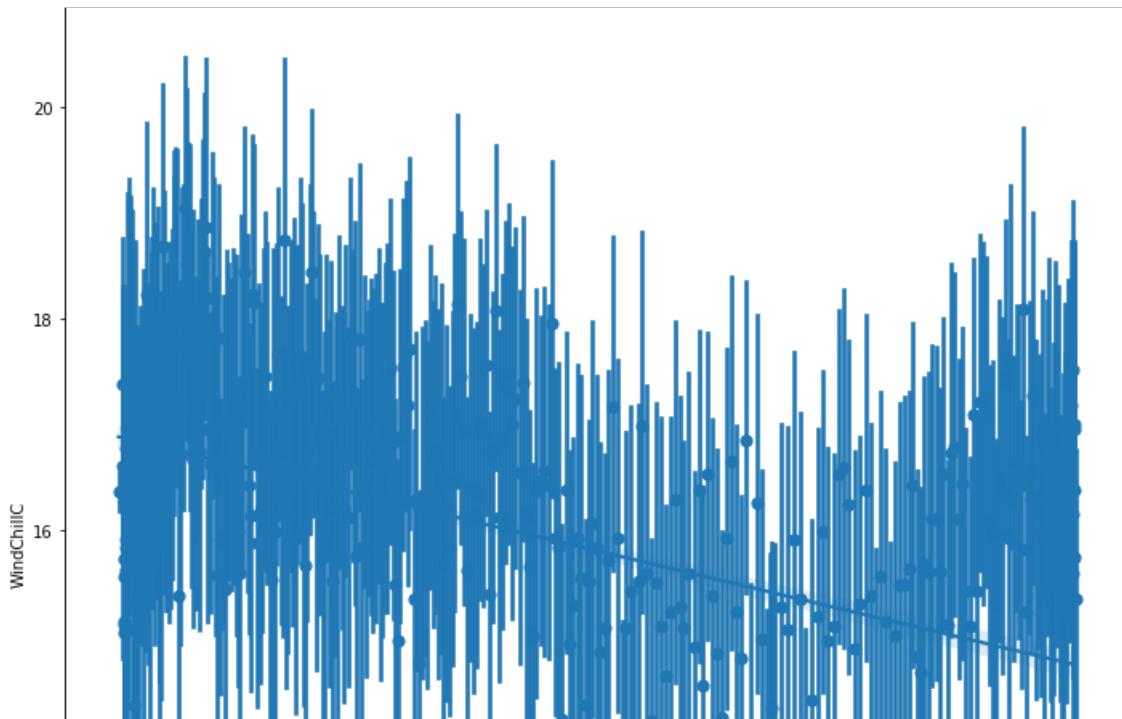


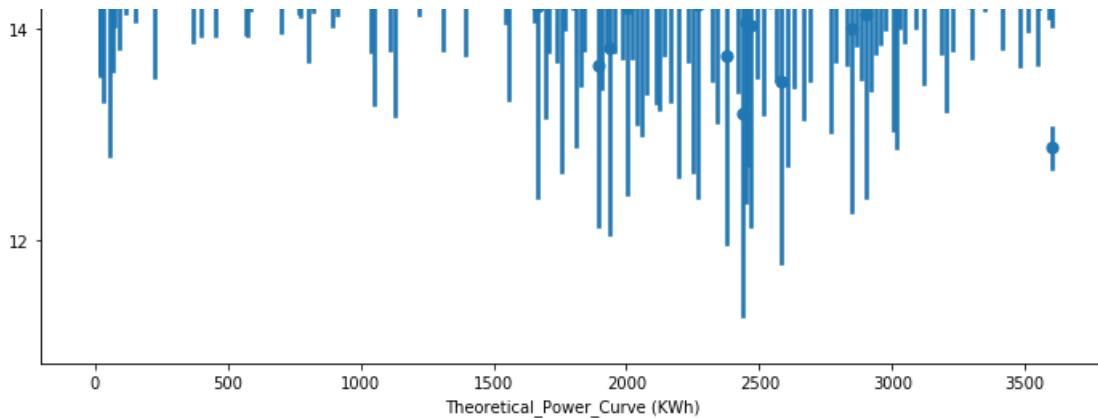
In [66]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="WindChillC", data=df, height=10, x_bins=500)
#Theoretical value and Wind Chill seem to have a stronger negative linear relationship than the real values
#and windchill
```

Out [66]:

```
<seaborn.axisgrid.FacetGrid at 0x21fbabe2e08>
```



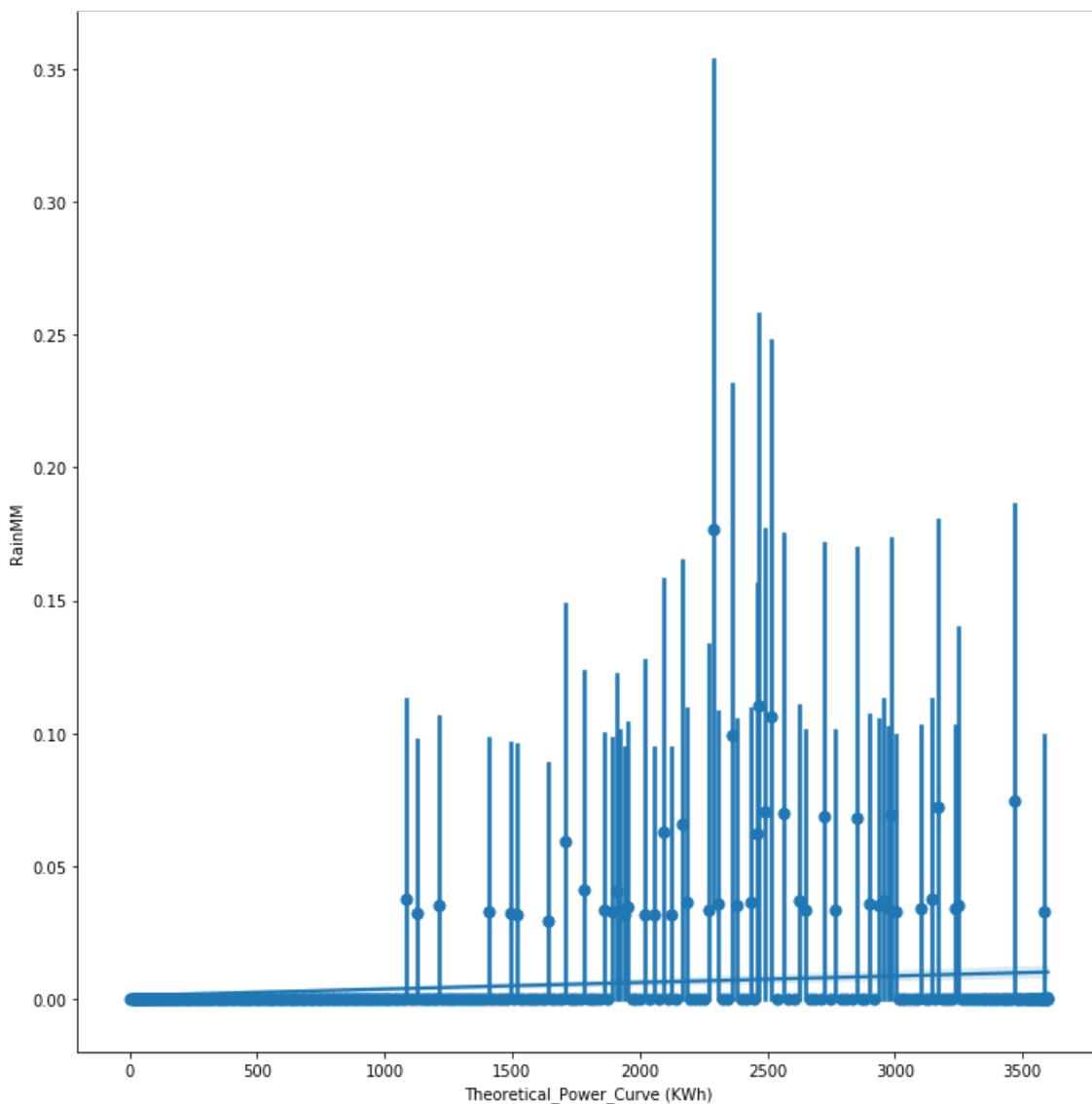


In [67]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="RainMM", data=df, height=10, x_bins=500)
#Since most of the rain data is nearly 0 there seem to be no relationship between
#Rain and Theoretical values. It is very similar to production values-Rain plot.
```

Out[67]:

```
<seaborn.axisgrid.FacetGrid at 0x21fbb479648>
```

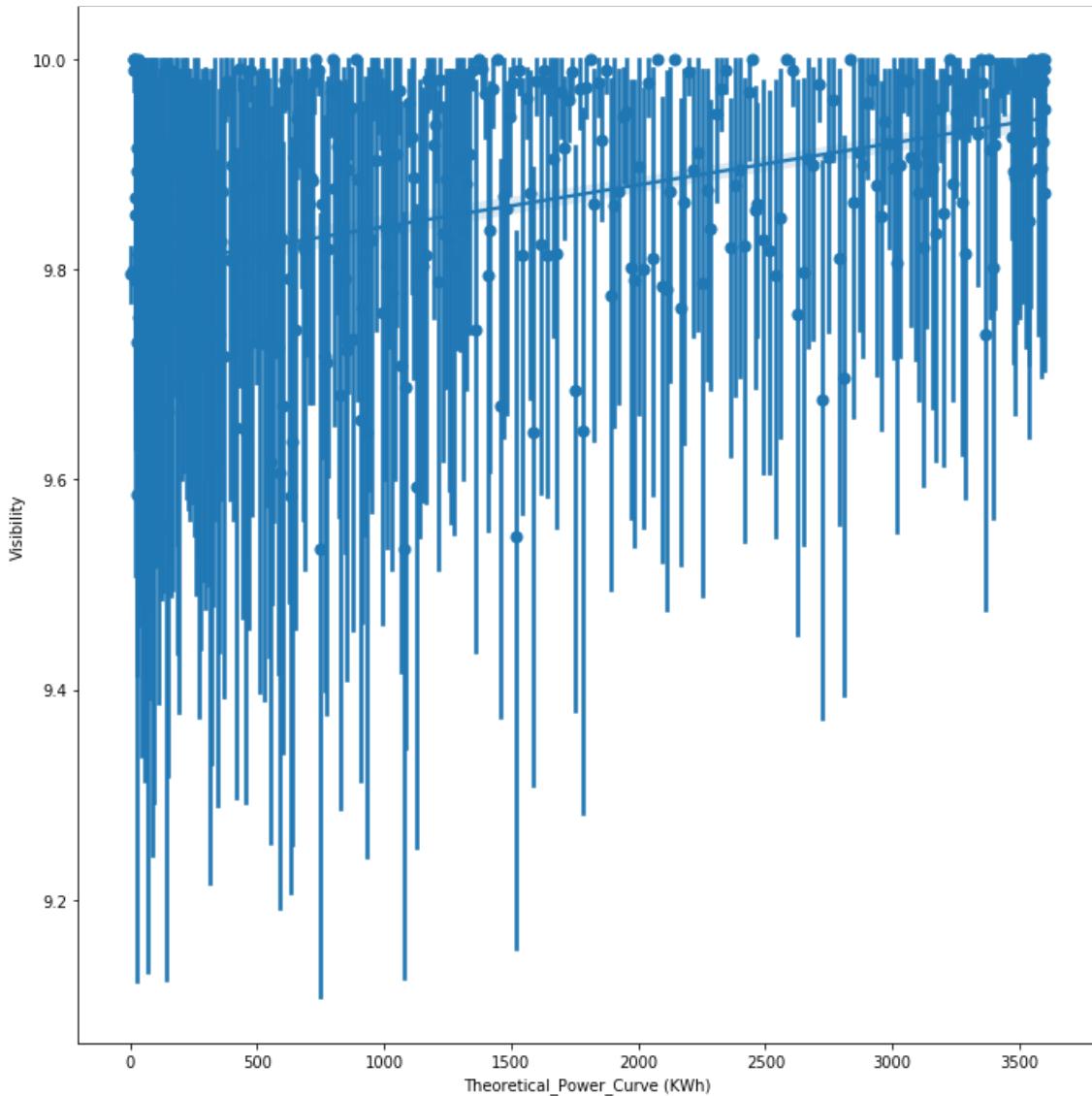


In [69]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="Visibility", data=df, height=10, x_bins=500)
#This graph is also very similar to the Production values-Visibility graph.
```

Out[69]:

```
<seaborn.axisgrid.FacetGrid at 0x21fbdb6b6e88>
```

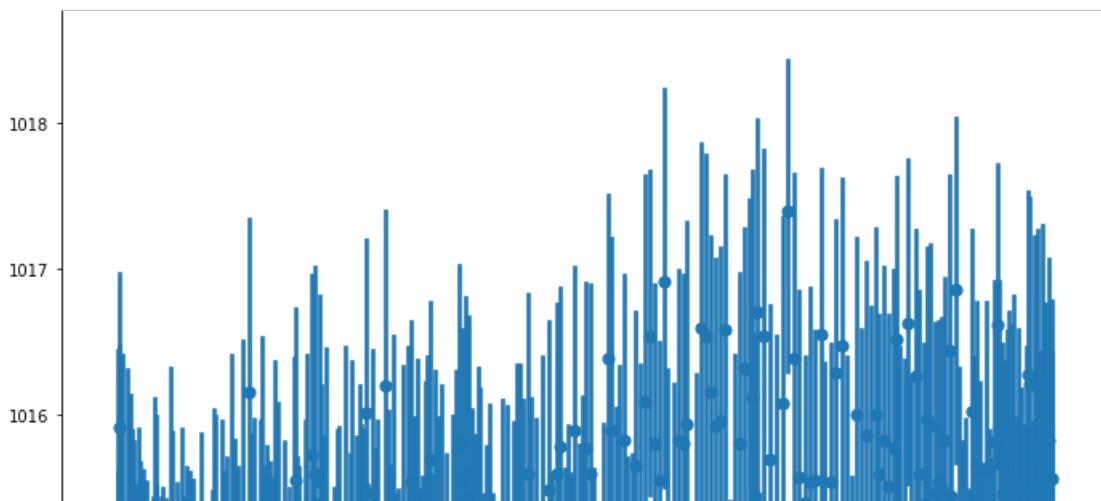


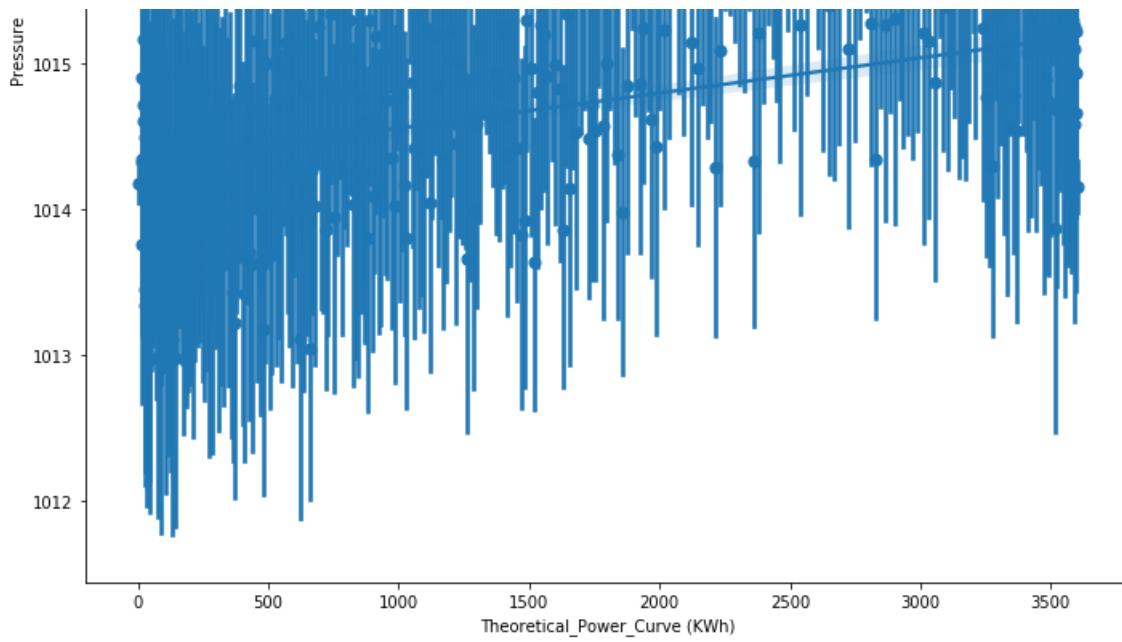
In [70]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)",y="Pressure",data=df,height=10,x_bins=500)
#This graph is also quite similar to the Electric Production-Pressure Graph
```

Out[70]:

```
<seaborn.axisgrid.FacetGrid at 0x21fbe01f748>
```



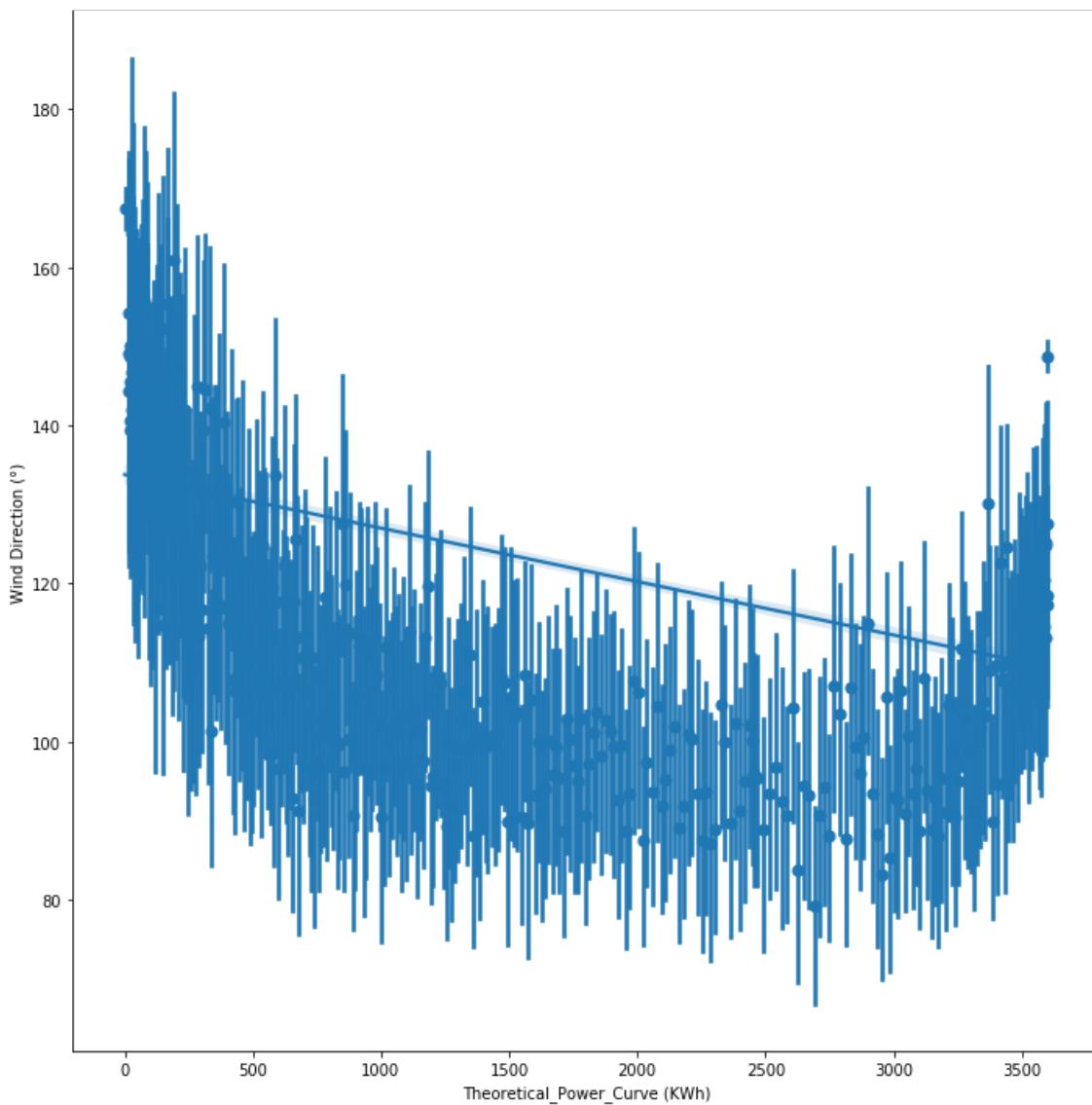


In [72]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)", y="Wind Direction (°)", data=df, height=10, x_bins=500)
#This graph is also quite similar to the Electric Production-Wind Direction Graph
```

Out[72]:

```
<seaborn.axisgrid.FacetGrid at 0x21fbf2b2488>
```



In [74]:

```
import math
import numpy as np
rad=[]
for items in df["Wind Direction (°)"]:
    rad.append(math.degrees(items))

sin_values=np.sin(rad)
sin_values #Converting degrees to sin values to have a normalization on the wind direction degree
df2=pd.DataFrame([df["Theoretical_Power_Curve (KWh)",sin_values]])
df2=df2.T
df2.columns=["Theoretical_Power_Curve (KWh)","Sin"]
df2.head()
```

Out[74]:

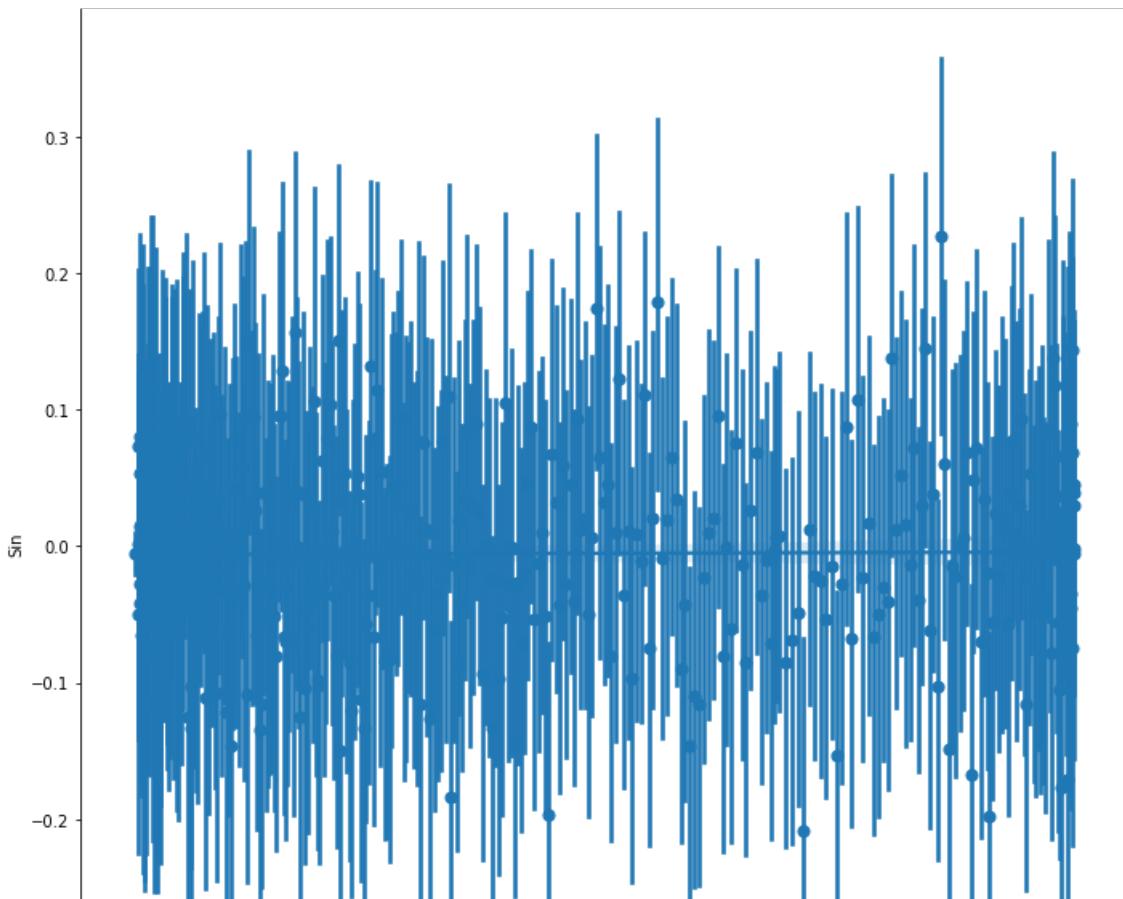
	Theoretical_Power_Curve (KWh)	Sin
0	416.328908	-0.732299
1	519.917511	-0.973390
2	390.900016	0.045024
3	516.127569	-0.465941
4	491.702972	-0.840876

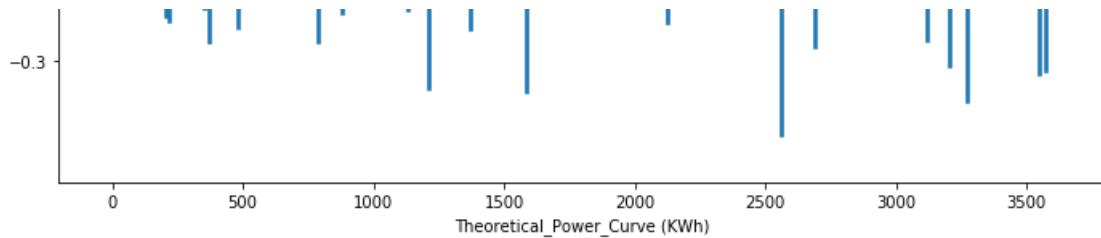
In [75]:

```
sns.lmplot(x="Theoretical_Power_Curve (KWh)",y="Sin",height=10,x_bins=500,data=df2)
#The sinus values of wind direction degrees and Theoretical value seem to have
#no relation
```

Out[75]:

```
<seaborn.axisgrid.FacetGrid at 0x21fbfb50ac8>
```



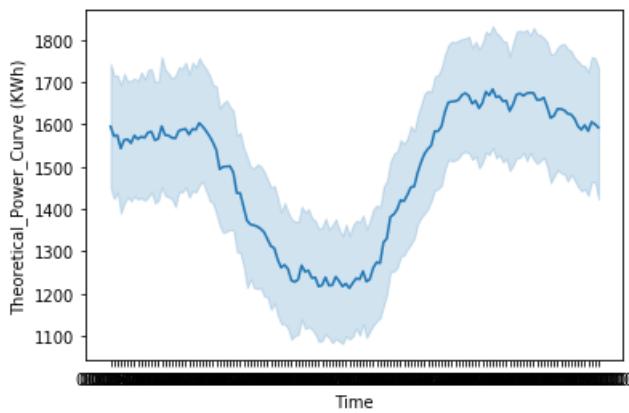


In [77]:

```
sns.lineplot(y=df["Theoretical_Power_Curve (KWh)"],x=df["Time"].astype("str"),data=df) #Change of the Average Electric Production  
# According to time s.
```

Out[77]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fc1858b08>
```

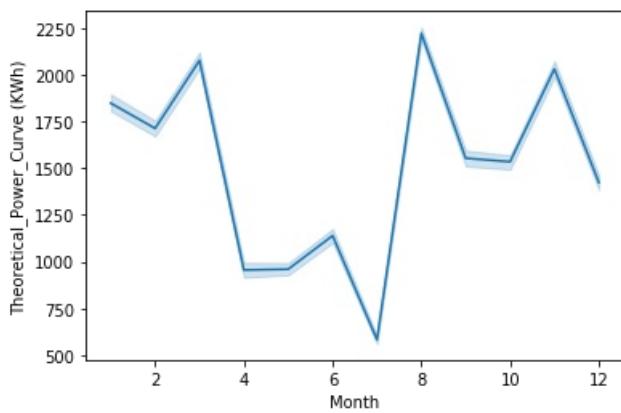


In [78]:

```
sns.lineplot(y=df["Theoretical_Power_Curve (KWh)"],x=df["Month"],data=df)
```

Out[78]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21fc19d9608>
```



In [ ]:

```
In [1]:
```

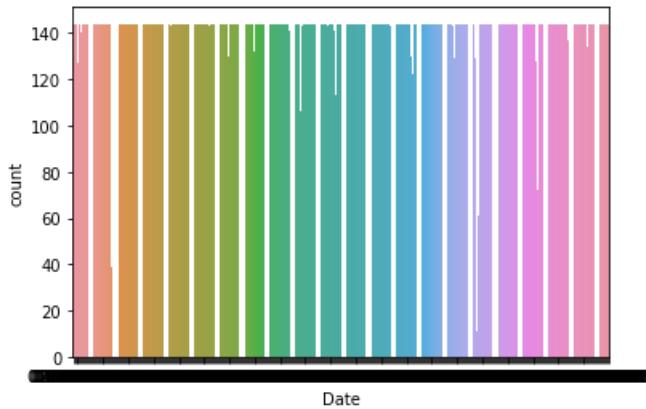
```
import pandas as pd
import seaborn as sns
```

```
In [5]:
```

```
sns.countplot(df["Date"]) #Date variable is our dates of the corresponding measures. And since we
have measures for every
#10 minutes ideally we should have 144 measures everyday. This means our
frequency for days
#should be equal and 144. Our plot shows that we are very close to the i
ideal case.
```

```
Out[5]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21b887b5ac8>
```

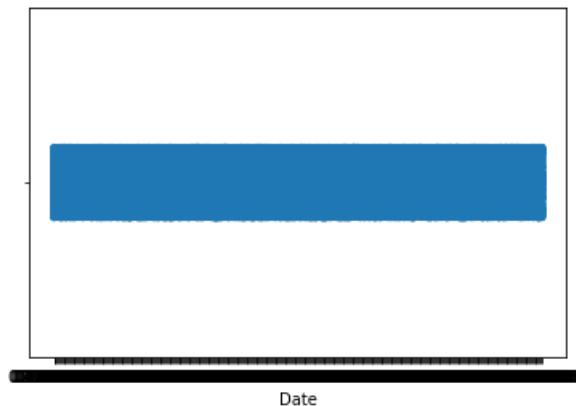


```
In [6]:
```

```
sns.stripplot(df["Date"])
```

```
Out[6]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21b887ed308>
```

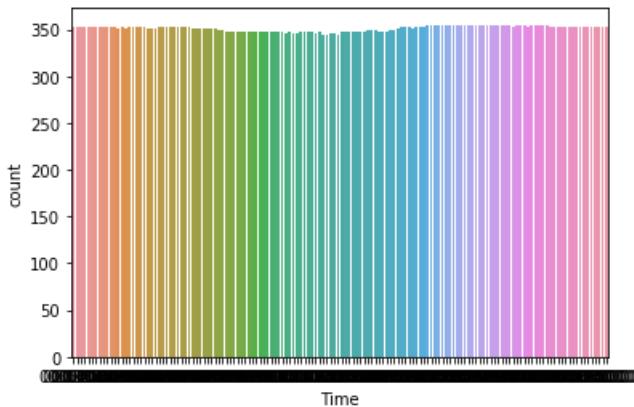


```
In [34]:
```

```
sns.countplot(df["Time"]) #For Time variable the situation is quite similar to the date variable,it
is the specific
# times that measurements took place ideally for every time period there s
hould be 365 measurements.
#It seems that we do not have 365 measurements for any of our points but at
least they seem to be
#uniformly distributed, for every time periot number of measurements are re
ally close.
```

Out [34]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09a737a08>
```

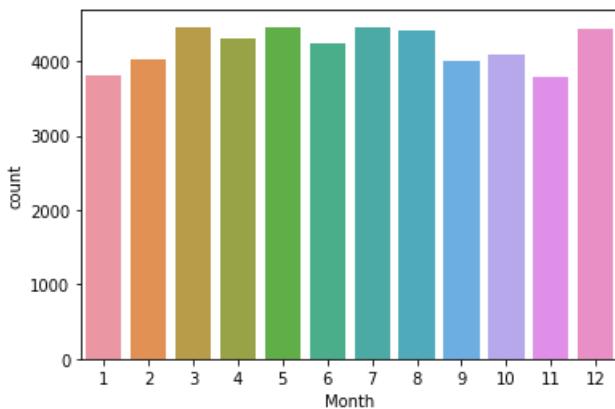


In [36]:

```
sns.countplot(df["Month"]) #The month variable represents the month since every month has  
#different number of days and  
# and due to that different number of obervations number of counts will  
not be the same. Also there  
# inequalities because of the missing observations, (not equal 144 for e  
every hour or 365 for everyday)  
#Because of these factors frequency of month categories are not the same  
but close to each other
```

Out [36]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09dd9b588>
```

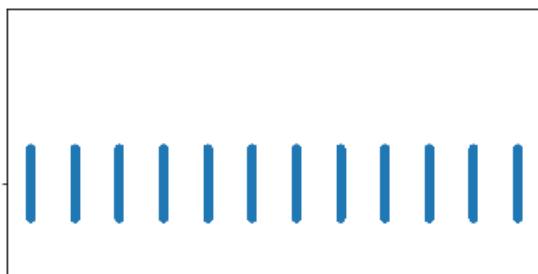


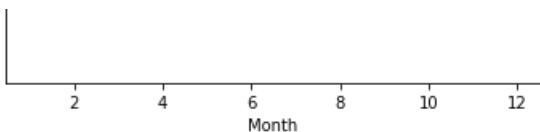
In [39]:

```
sns.stripplot(df["Month"]) #Also we can see here that the number of observations per month or in o  
ther words, frequency of the  
#month categories are very close to each other.
```

Out [39]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09e215ec8>
```



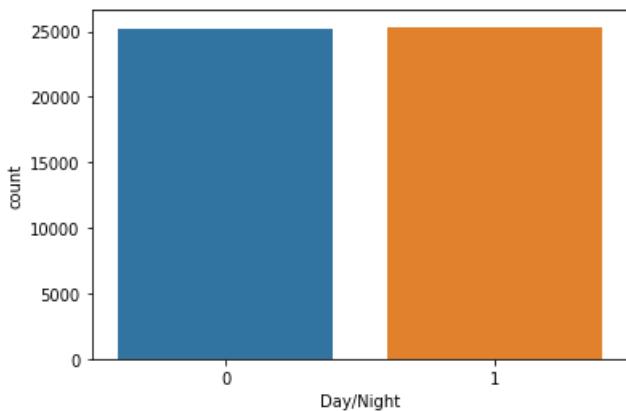


In [43]:

```
sns.countplot(df["Day/Night"]) # Number of observations per day and night seems very close.
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09b7f0388>
```

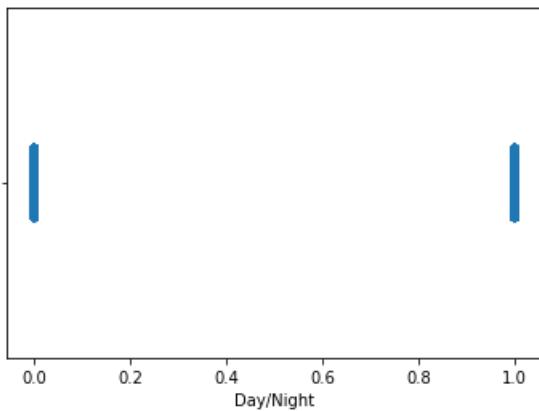


In [44]:

```
sns.stripplot(df["Day/Night"]) #Also here we can see that they are very close.
```

Out[44]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09b0c9d88>
```



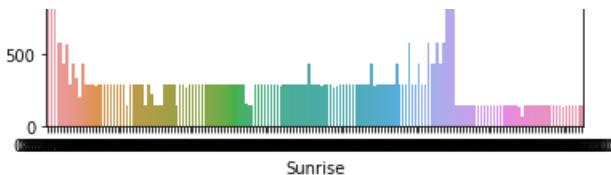
In [50]:

```
sns.countplot(df["Sunrise"]) #Frequencies of the sunrise times.
```

Out[50]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c08d65c288>
```



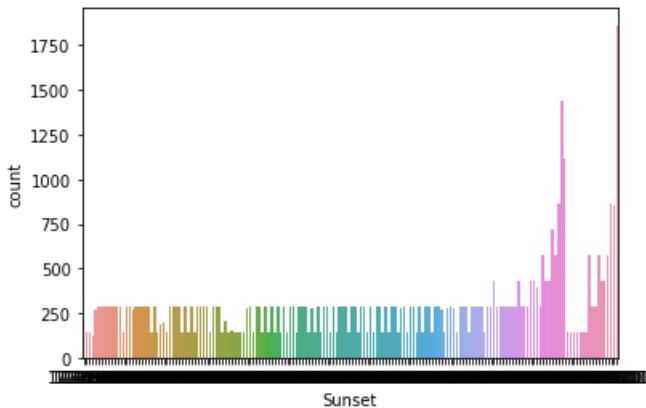


In [46]:

```
sns.countplot(df["Sunset"]) #Frequencies of the sunset times.
```

Out [46]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09a0ec388>
```

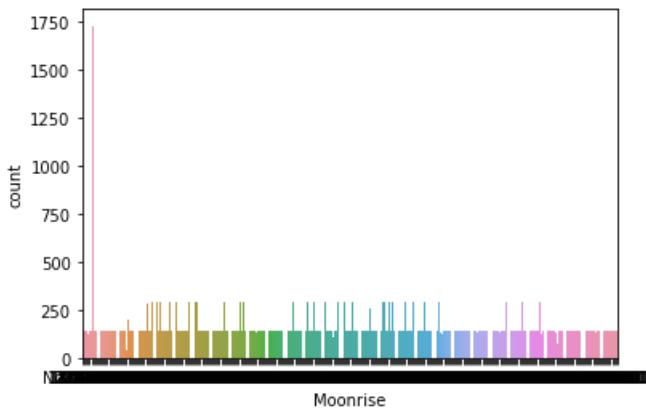


In [48]:

```
sns.countplot(df["Moonrise"]) #Frequencies of the Moonrise times.
```

Out [48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09a01afc8>
```



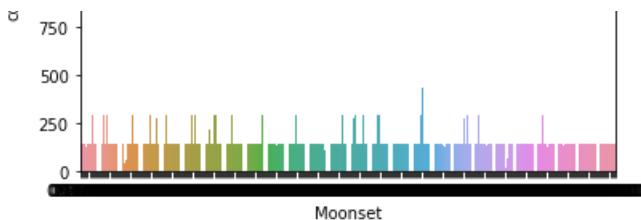
In [49]:

```
sns.countplot(df["Moonset"]) #Frequencies of the Moonset times.
```

Out [49]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c09a03f748>
```





In [52]:

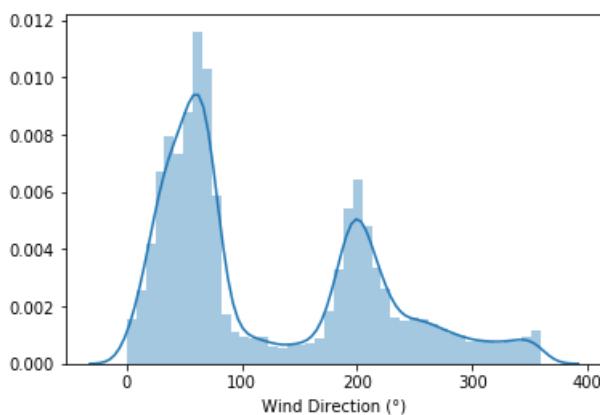
```
#Wind Direction
```

In [53]:

```
sns.distplot(df["Wind Direction (°)"]) #Our wind direction is angles in terms of degrees. So it has to be between
                                         # 0-360. There are 2 peaks one is around 100 degrees and the
other peak is very
                                         #close to 200 degrees.
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c08d714188>
```

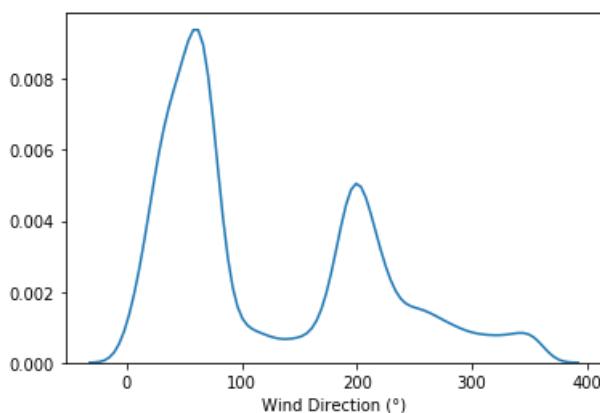


In [54]:

```
sns.distplot(df["Wind Direction (°)"], hist=False)
```

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c08d6ed8c8>
```

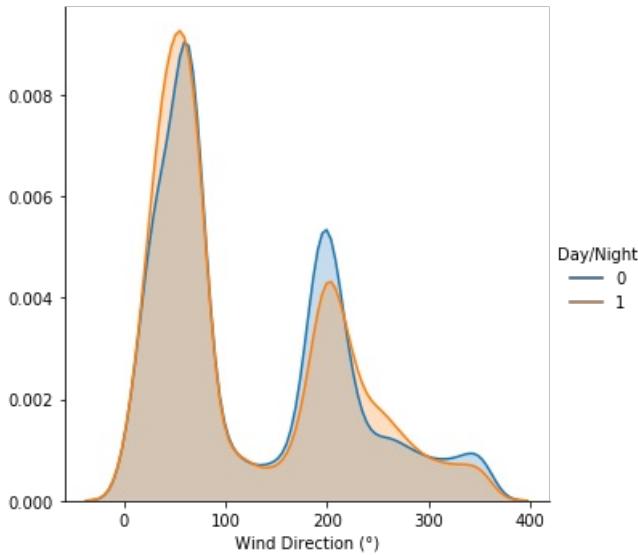


In [63]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Wind Direction (°)",shade=True).add_legend()
# Day and night distributions are very similar.
```

Out[63]:

```
<seaborn.axisgrid.FacetGrid at 0x1c099d70548>
```

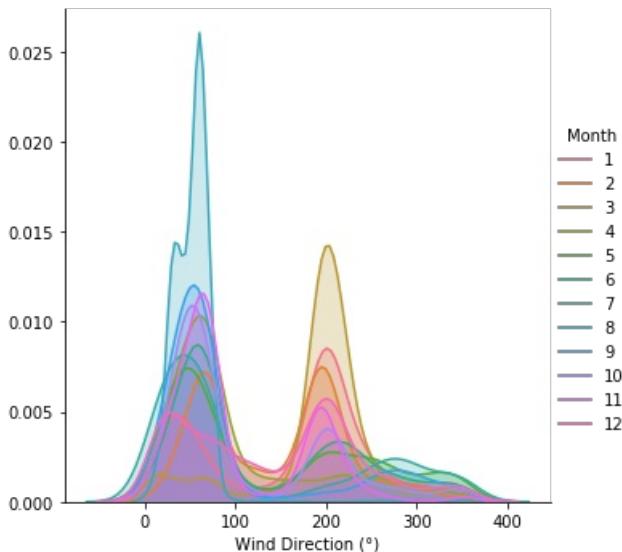


In [64]:

```
sns.FacetGrid(df,hue="Month",height=5).map(sns.kdeplot,"Wind Direction (°)",shade=True).add_legend()  
# Other than months 4 and 9 other seem to distribute similarly, both month 4 and 9 have really high  
# peaks, they may be the reason of the peaks at the general distribution.
```

Out[64]:

```
<seaborn.axisgrid.FacetGrid at 0x1c099d3a7c8>
```



In [14]:

```
import math  
import numpy as np  
rad=[]  
for items in df["Wind Direction (°)"]:  
    rad.append(math.degrees(items))  
  
sin_values=np.sin(rad)  
sin_values #Converting degrees to sin values to have a normalization on the wind direction degree  
s  
df2=pd.DataFrame([df["LV ActivePower (kW)"],sin_values])  
df2=df2.T  
df2.columns=["T17 ActivePower (kW)" "Sin"]
```

```
airz.columns = [ 'LV ActivePower (kW)', 'Sin' ]  
df2.head()
```

Out[14]:

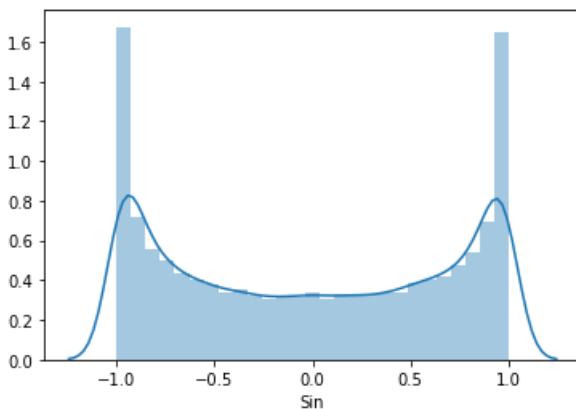
	LV ActivePower (kW)	Sin
0	380.047791	-0.732299
1	453.769196	-0.973390
2	306.376587	0.045024
3	419.645904	-0.465941
4	380.650696	-0.840876

In [57]:

```
sns.distplot(df2["Sin"]) #Since in degree terms the degrees 0 and 360 means the same,to change it  
and have a normalization  
#effect. We calculated the sin(x) for our wind direction angles.To have va-  
lues between -1 and 1.  
#The distribution seems a little bit different but it also has 2 peaks and  
both of them around  
#the extremes. The reason of it probably our angle distribution peaks was  
around 100 and 200  
#and sin90 and sin180are the extreme values. Probably we have a high freq-  
ency of 90 and 180 degrees.
```

Out[57]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c08bca43c8>
```

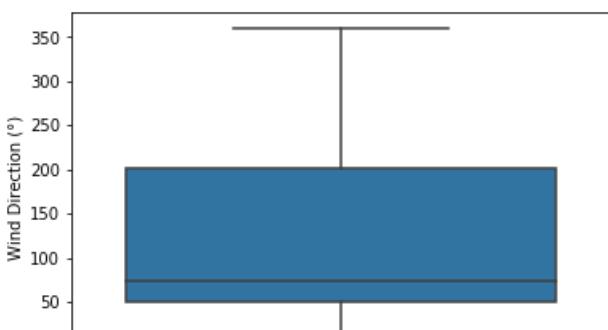


In [67]:

```
sns.boxplot(df["Wind Direction (°)"],orient="v") # Probably as we seen on the distribution plots,  
because of the high frequency  
#of the values around 100 and below our median is 73 but our  
mean is 123.  
#We have a relatively high standard deviation.
```

Out[67]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c08f7be188>
```





In [62]:

```
df["Wind Direction (°)"].describe()
```

Out[62]:

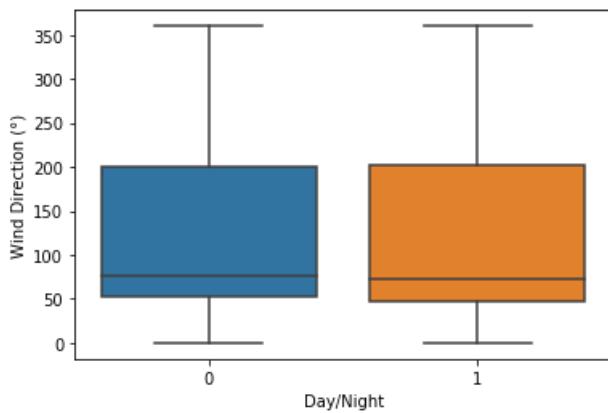
```
count      50530.000000
mean       123.687559
std        93.443736
min        0.000000
25%       49.315437
50%       73.712978
75%      201.696720
max      359.997589
Name: Wind Direction (°), dtype: float64
```

In [9]:

```
sns.boxplot(x=df["Day/Night"],y=df["Wind Direction (°)"],orient="v",data=df)
#Day and night seem to have very similar distribution
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21baa4c0ac8>
```

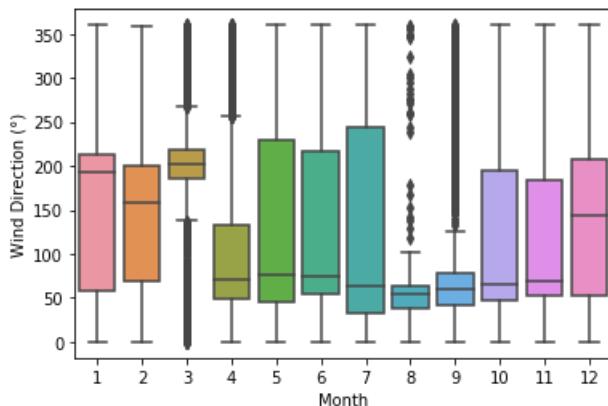


In [12]:

```
sns.boxplot(x=df["Month"],y=df["Wind Direction (°)"],data=df,orient="v") #Day and night seem to have very similar distribution
#Months 3-4 and 8-9 has a lot of outliers.
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21baaa3b0c8>
```

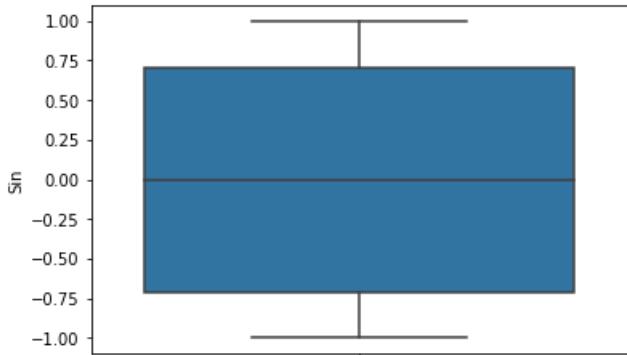


In [17]:

```
sns.boxplot(df2["Sin"], orient="v")
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21ba0251488>
```



In [19]:

```
df2["Sin"].describe() #Sin box plot seems more regular and consistent than the actual wind direction data.
```

Out[19]:

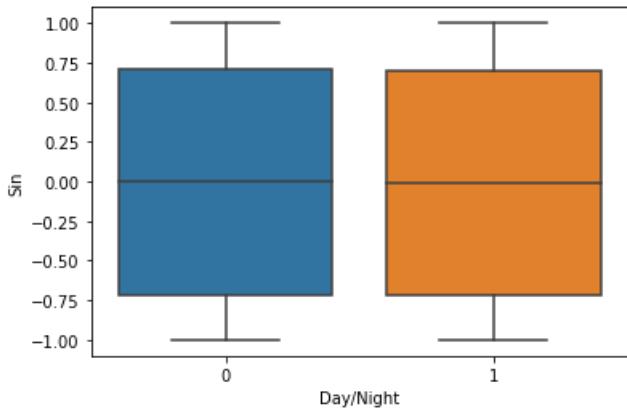
```
count    50530.000000
mean     -0.005429
std      0.707683
min     -1.000000
25%    -0.713920
50%    -0.005971
75%     0.701460
max     1.000000
Name: Sin, dtype: float64
```

In [16]:

```
sns.boxplot(x=df["Day/Night"], y=df2["Sin"], orient="v") #Day-Night observations are nearly same in sin(x) graph
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21baa03d248>
```

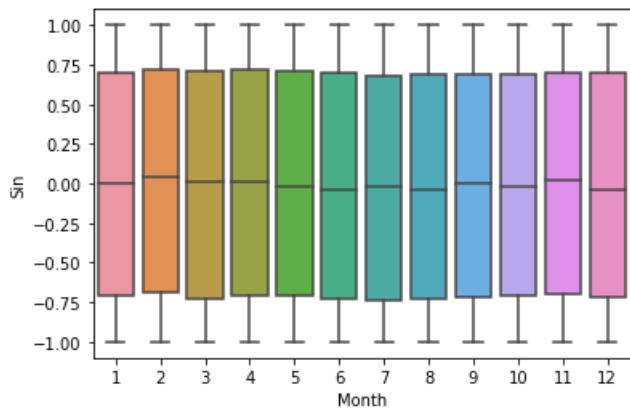


In [18]:

```
sns.boxplot(x=df["Month"], y=df2["Sin"], orient="v") #Different than the actual angle graph, sin(x) graph has no outlier
#On the month categories
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21b9e8ab208>
```

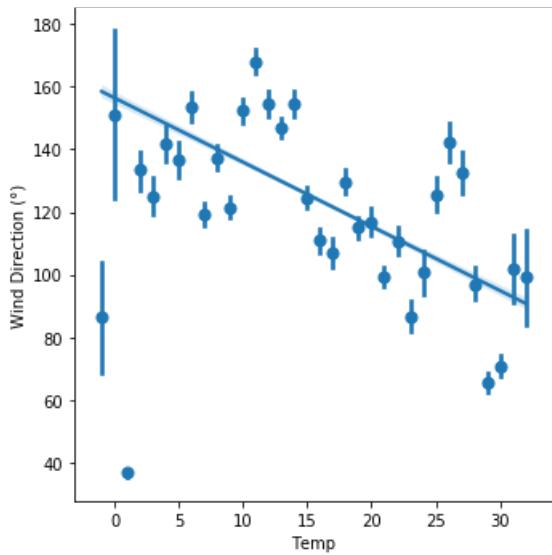


In [22]:

```
sns.lmplot(y="Wind Direction (°)",x="Temp",data=df,x_bins=5000)
#These graph of wind direction with other variables can not interpreted as same as other
#variables because angles are not like numeric values exactly. 160 degree does not imply it is bigger than 120 degree
#It actually implies the directions.
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x21b9d903fc8>
```

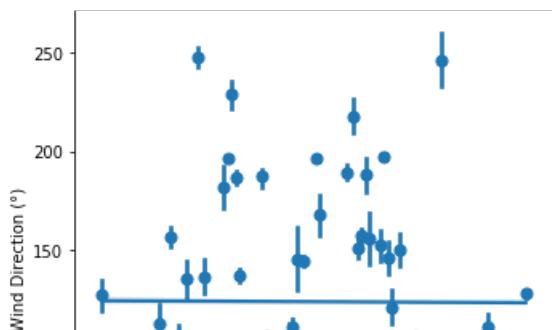


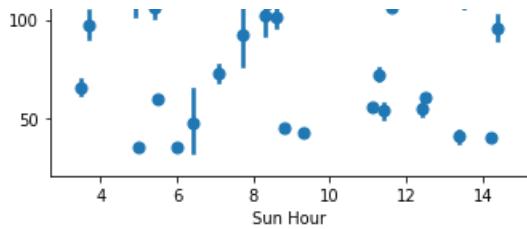
In [24]:

```
sns.lmplot(y="Wind Direction (°)",x="Sun Hour",data=df,x_bins=5000)
```

Out[24]:

```
<seaborn.axisgrid.FacetGrid at 0x21b8d4db7c8>
```



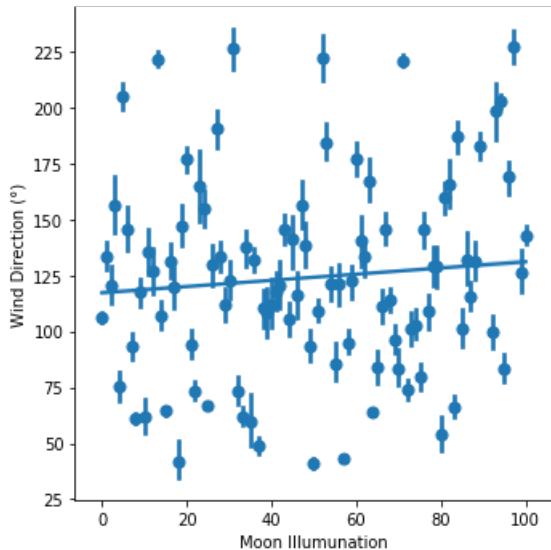


In [28]:

```
sns.lmplot(y="Wind Direction (°)", x="Moon Illumunation", data=df, x_bins=5000)
```

Out [28]:

```
<seaborn.axisgrid.FacetGrid at 0x21b8d646348>
```

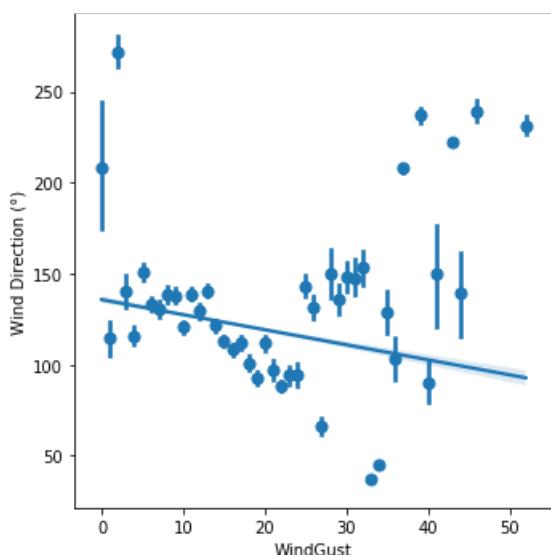


In [29]:

```
sns.lmplot(y="Wind Direction (°)", x="WindGust", data=df, x_bins=5000)
```

Out [29]:

```
<seaborn.axisgrid.FacetGrid at 0x21b8d7db208>
```

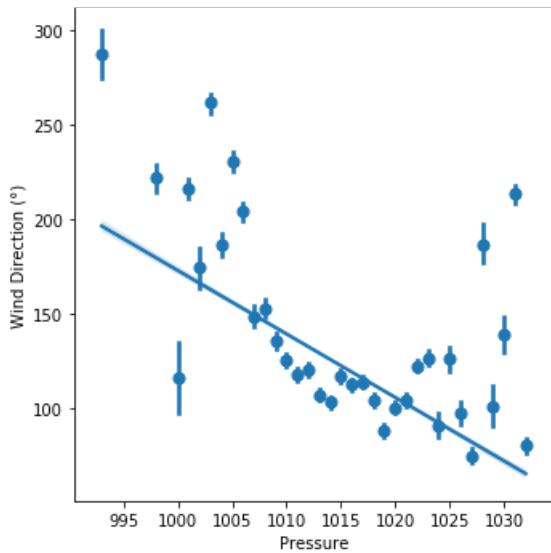


In [30]:

```
sns.lmplot(y="Wind Direction (°)", x="Pressure", data=df, x_bins=5000)
```

Out [30]:

```
<seaborn.axisgrid.FacetGrid at 0x21b8d8bac08>
```

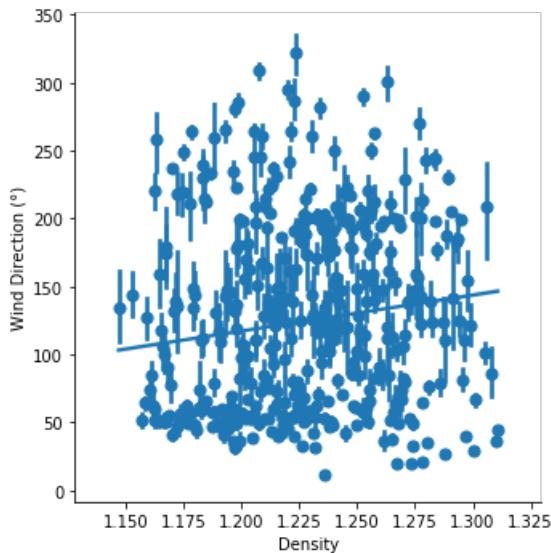


In [31]:

```
sns.lmplot(y="Wind Direction (°)", x="Density", data=df, x_bins=5000)
```

Out [31]:

```
<seaborn.axisgrid.FacetGrid at 0x21b8d988288>
```

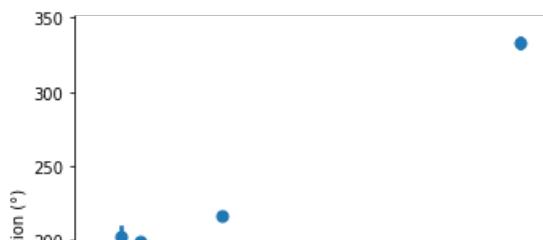


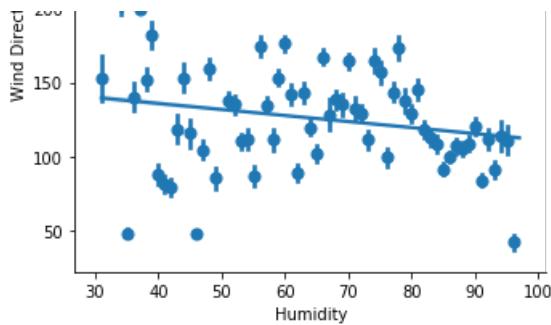
In [32]:

```
sns.lmplot(y="Wind Direction (°)", x="Humidity", data=df, x_bins=5000)
```

Out [32]:

```
<seaborn.axisgrid.FacetGrid at 0x21b9e8b4f08>
```



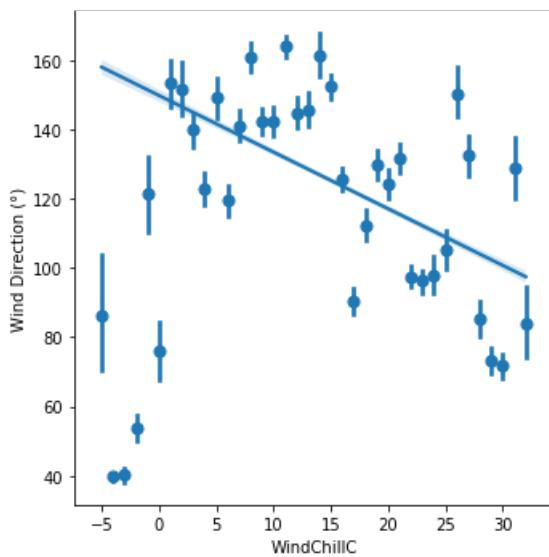


In [33]:

```
sns.lmplot(y="Wind Direction (°)",x="WindChillC",data=df,x_bins=5000)  
#WindChill and Wind Speed seem to have a negative linear relationship
```

Out[33]:

```
<seaborn.axisgrid.FacetGrid at 0x21b8b610348>
```

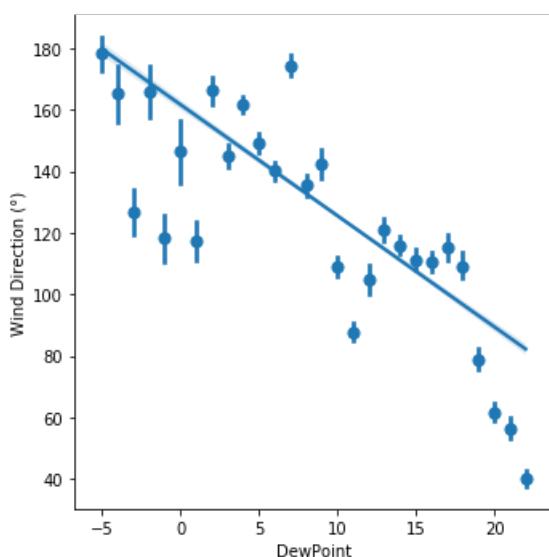


In [34]:

```
sns.lmplot(y="Wind Direction (°)",x="DewPoint",data=df,x_bins=5000)
```

Out[34]:

```
<seaborn.axisgrid.FacetGrid at 0x21b9aa64c48>
```

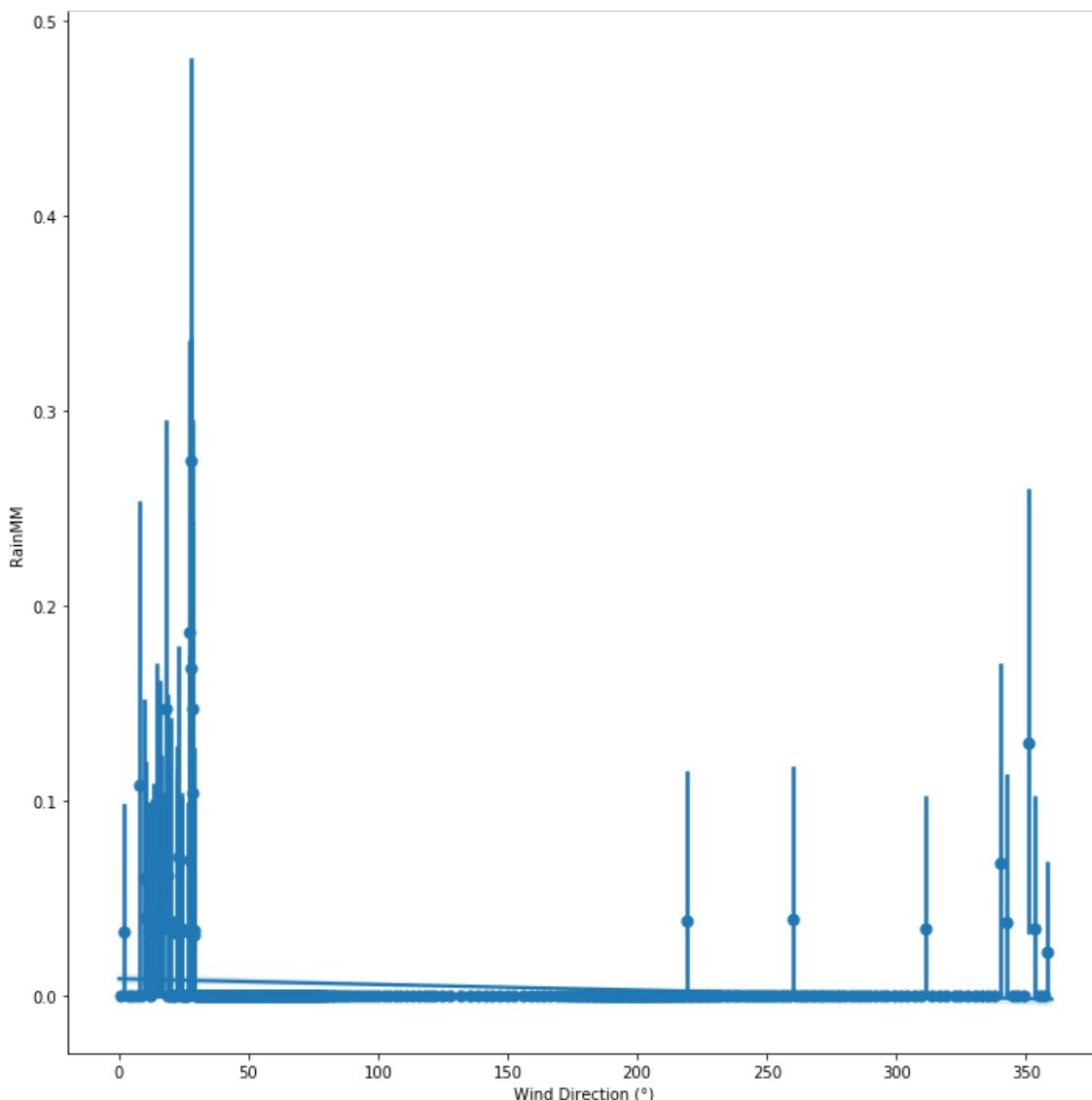


In [35]:

```
sns.lmplot(x="Wind Direction (°)",y="RainMM",data=df,height=10,x_bins=500)
```

Out[35]:

```
<seaborn.axisgrid.FacetGrid at 0x21b8761bc88>
```

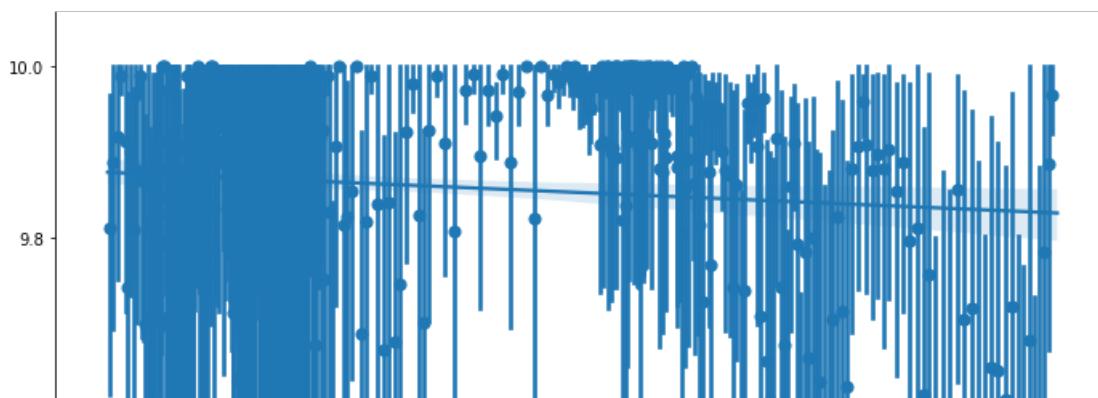


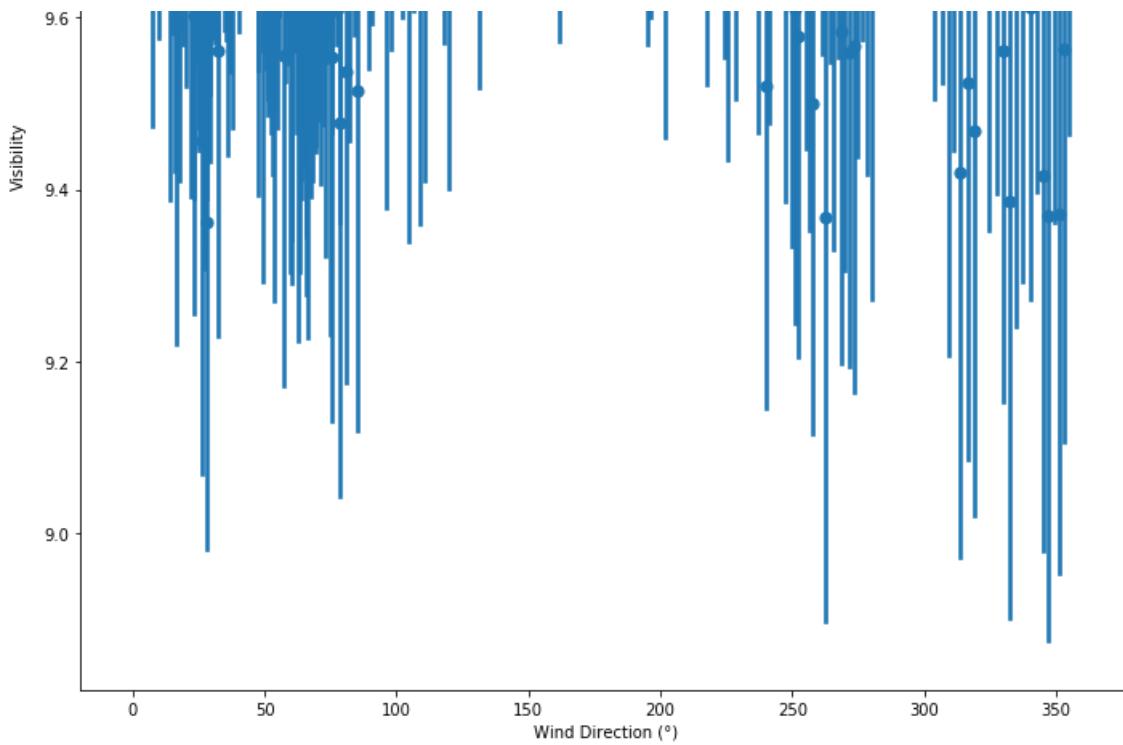
In [36]:

```
sns.lmplot(x="Wind Direction (°)",y="Visibility",data=df,height=10,x_bins=500)
```

Out[36]:

```
<seaborn.axisgrid.FacetGrid at 0x21b9bba9d08>
```



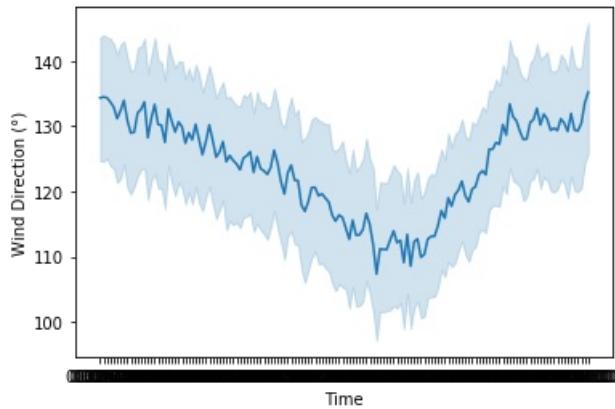


In [37]:

```
sns.lineplot(y=df["Wind Direction (°)"],x=df["Time"].astype("str"),data=df)
```

Out[37]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21b841d3848>
```

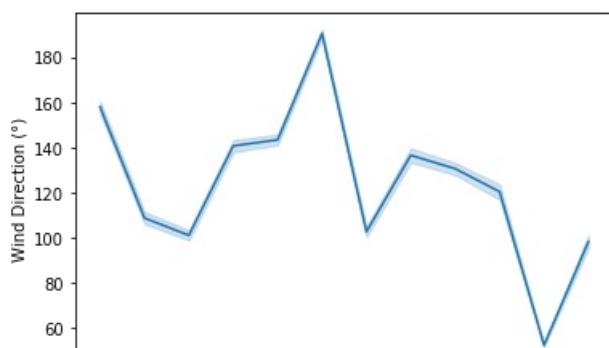


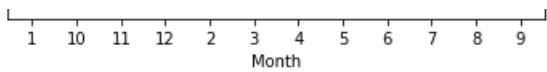
In [38]:

```
sns.lineplot(y=df["Wind Direction (°)"],x=df["Month"].astype("str"),data=df)
```

Out[38]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21b87f4d808>
```





In [ ]:

In [16]:

```
import pandas as pd
import seaborn as sns
```

In [17]:

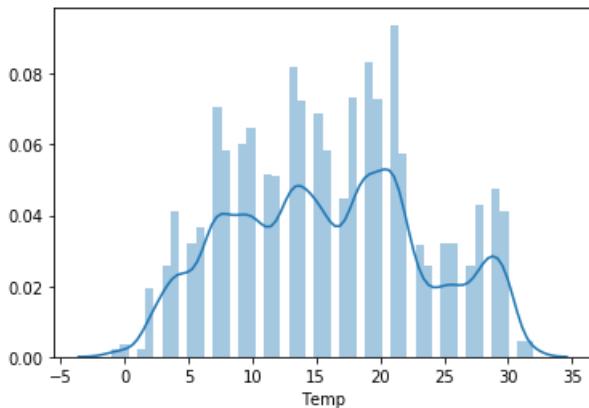
```
data=pd.read_excel("C:/Users/Furkan/Desktop/Final Data.xlsx") #Reading Data From Excel File
df=data.copy() # Copying data for just in case
```

In [3]:

```
sns.distplot(df["Temp"])
```

Out[3]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25124551fc8>
```

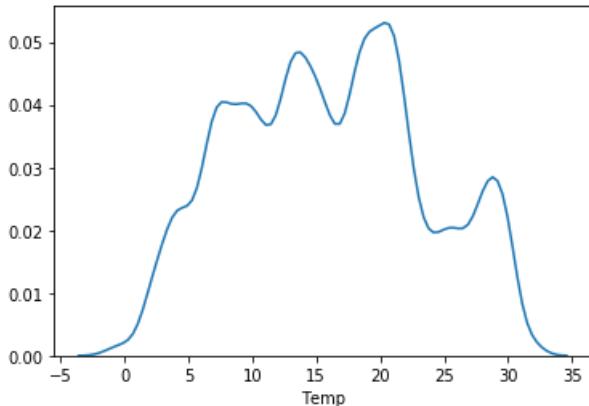


In [4]:

```
sns.distplot(df["Temp"], hist=False)
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25127a17348>
```

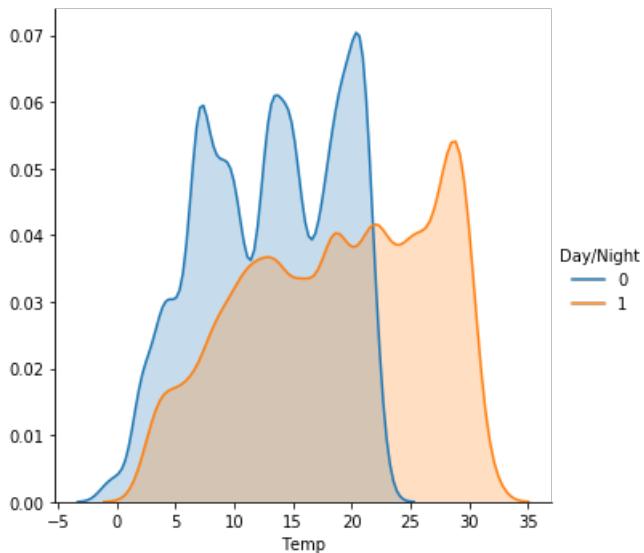


In [5]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Temp",shade=True).add_legend()
# Day and night distributions seem very different as expected.
```

Out[5]:

```
<seaborn.axisgrid.FacetGrid at 0x25124691308>
```

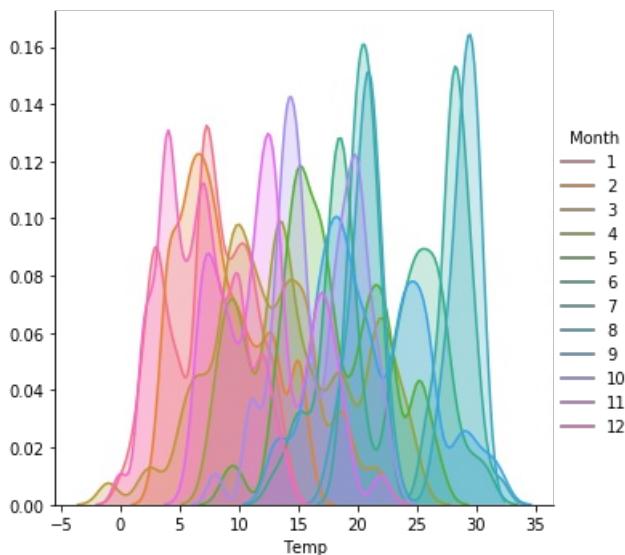


In [6]:

```
sns.FacetGrid(df,hue="Month",height=5).map(sns.kdeplot,"Temp",shade=True).add_legend()
# Different moths shows different distributions as expected.
```

Out [6]:

<seaborn.axisgrid.FacetGrid at 0x2512a099ac8>

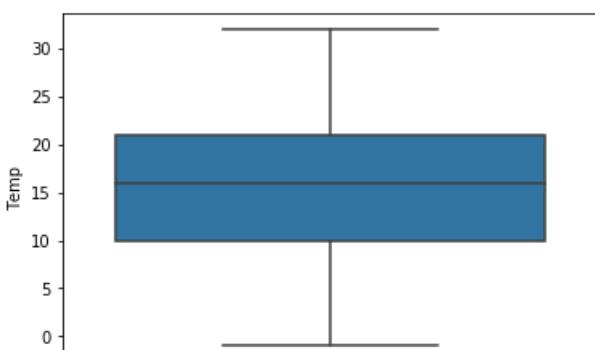


In [7]:

```
sns.boxplot(df["Temp"],orient="v")
```

Out [7]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x25123ec3d48>

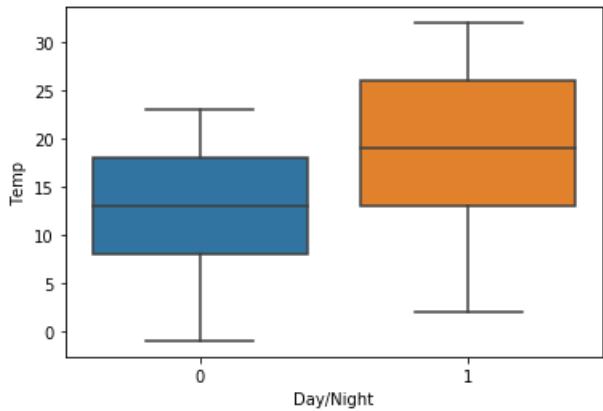


In [8]:

```
sns.boxplot(x=df["Day/Night"], y=df["Temp"], orient="v")
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2512951e5c8>
```

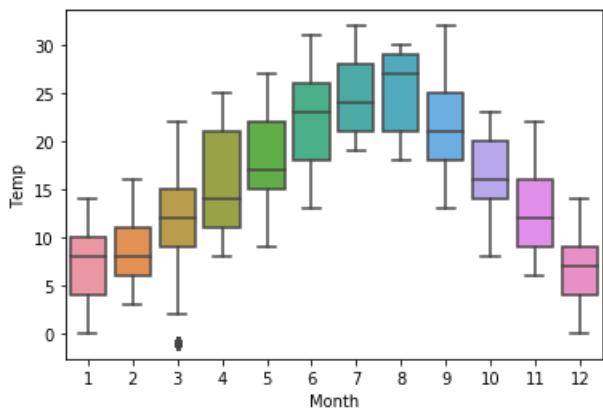


In [9]:

```
sns.boxplot(x=df["Month"], y=df["Temp"], orient="v")
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2512c602508>
```

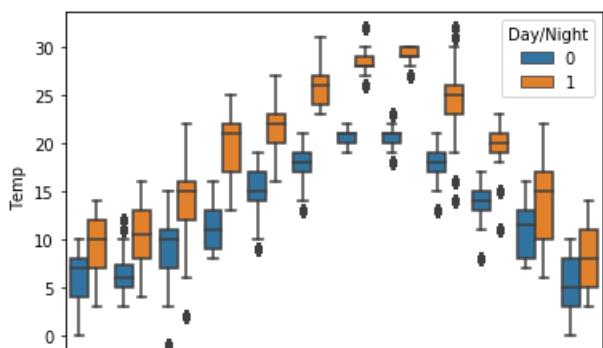


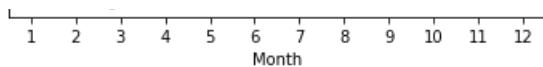
In [10]:

```
sns.boxplot(x=df["Month"], y=df["Temp"], orient="v", hue=df["Day/Night"])
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2512609f048>
```



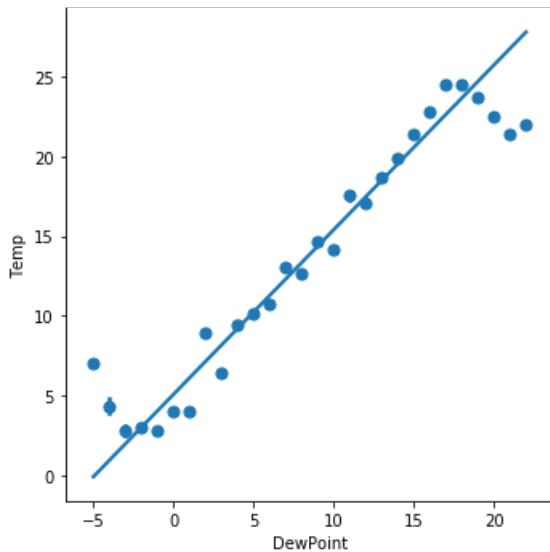


In [11]:

```
sns.lmplot(y="Temp",x="DewPoint",data=df,x_bins=5000)
#Temperature and dew point seem to have a strong positive linear relationship
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x25128b51f88>
```

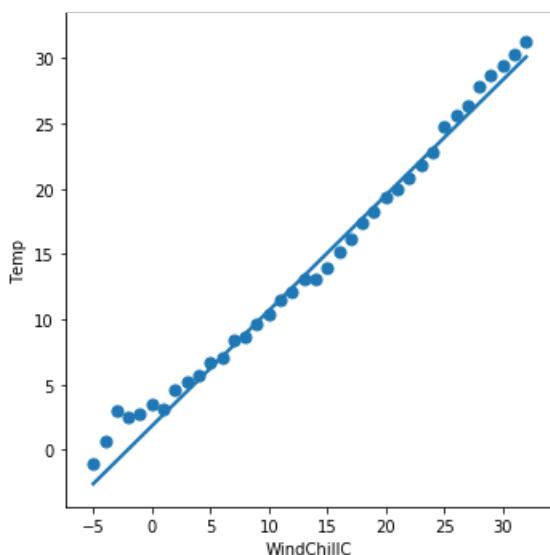


In [12]:

```
sns.lmplot(y="Temp",x="WindChillC",data=df,x_bins=5000)
#Windchill and temp has a very strong linear relationship
```

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x25128cd44c8>
```



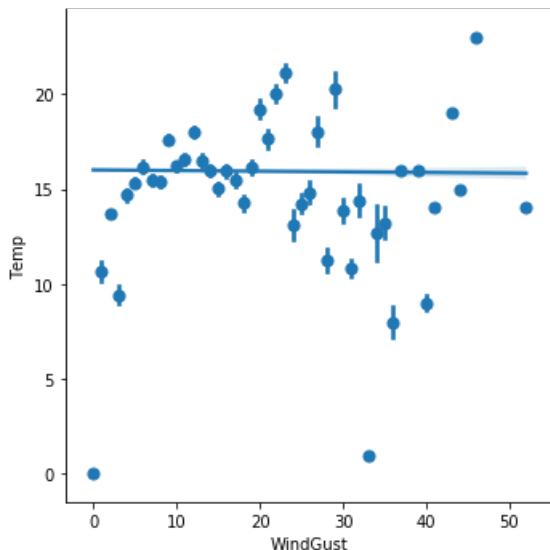
In [13]:

```
sns.lmplot(y="Temp",x="WindGust",data=df,x_bins=5000)
#Temp and windgust does not seem to have a relationship
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x25128d41128>
```

```
<seaborn.axisgrid.FacetGrid at 0x25128f41dc8>
```

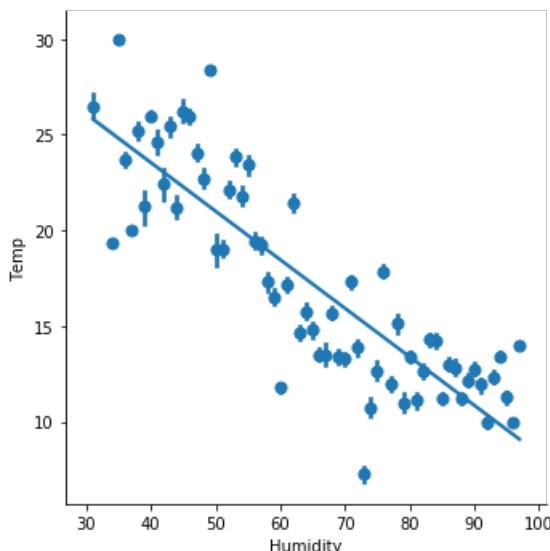


```
In [14]:
```

```
sns.lmplot(y="Temp",x="Humidity",data=df,x_bins=5000)
#Temperature and humidity seem to have a negative linear relationship
```

```
Out[14]:
```

```
<seaborn.axisgrid.FacetGrid at 0x25129e63708>
```

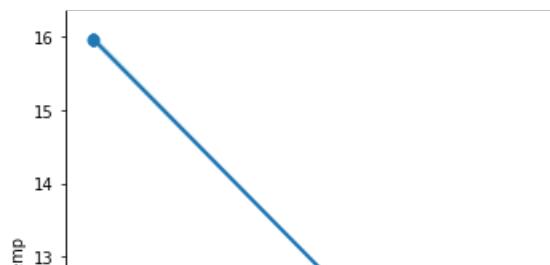


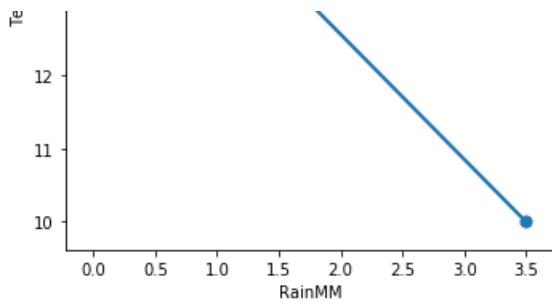
```
In [15]:
```

```
sns.lmplot(y="Temp",x="RainMM",data=df,x_bins=5000)
#Temp and Rain do not seem to have a relationship
```

```
Out[15]:
```

```
<seaborn.axisgrid.FacetGrid at 0x25128f374c8>
```



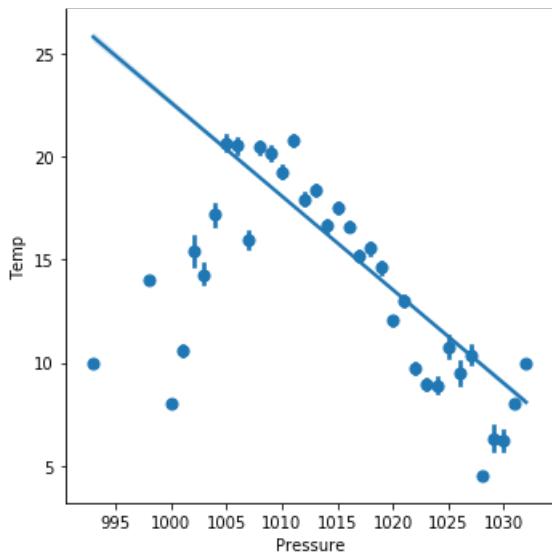


In [16]:

```
sns.lmplot(y="Temp",x="Pressure",data=df,x_bins=5000)  
#Temp and Pressure have negative linear relationship
```

Out[16]:

```
<seaborn.axisgrid.FacetGrid at 0x25129652048>
```

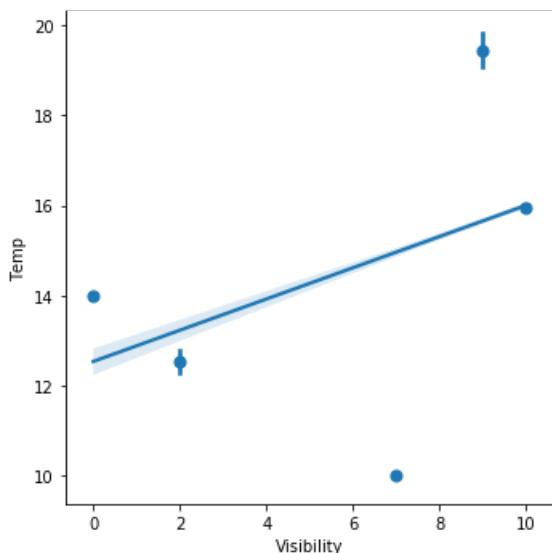


In [17]:

```
sns.lmplot(y="Temp",x="Visibility",data=df,x_bins=5000)  
#Temp and Visibility do not seem to have a relationship
```

Out[17]:

```
<seaborn.axisgrid.FacetGrid at 0x2512998d608>
```

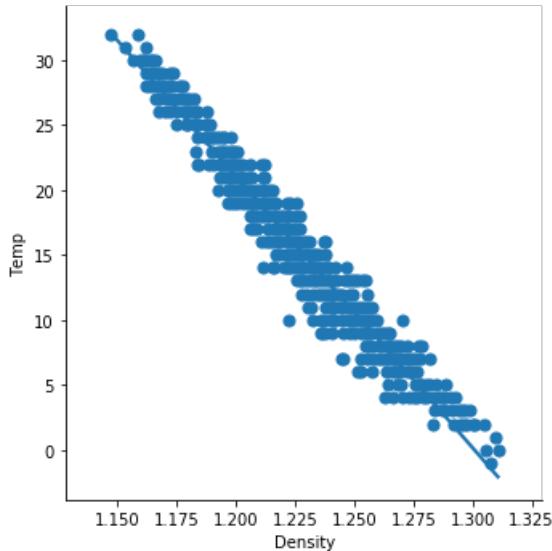


In [18]:

```
sns.lmplot(y="Temp",x="Density",data=df,x_bins=5000)
#Temp and Air Density seem to have a strong negative linear relationship
```

Out[18]:

```
<seaborn.axisgrid.FacetGrid at 0x2512c24bcc8>
```

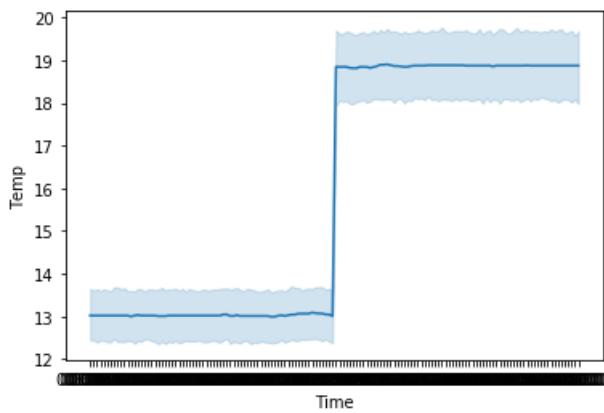


In [19]:

```
sns.lineplot(y=df["Temp"],x=df["Time"].astype("str"),data=df)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25125e8d948>
```

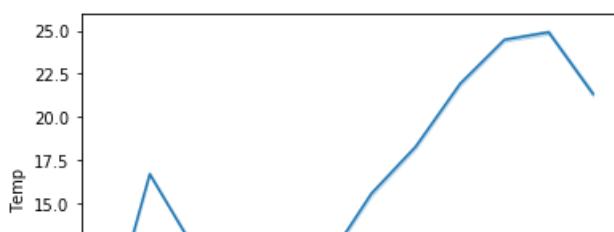


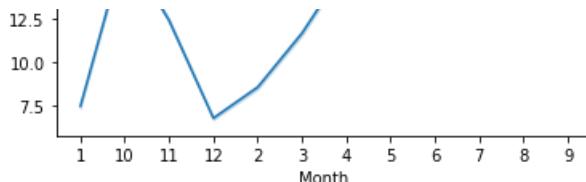
In [20]:

```
sns.lineplot(y=df["Temp"],x=df["Month"].astype("str"),data=df)
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25126a84208>
```





In [4]:

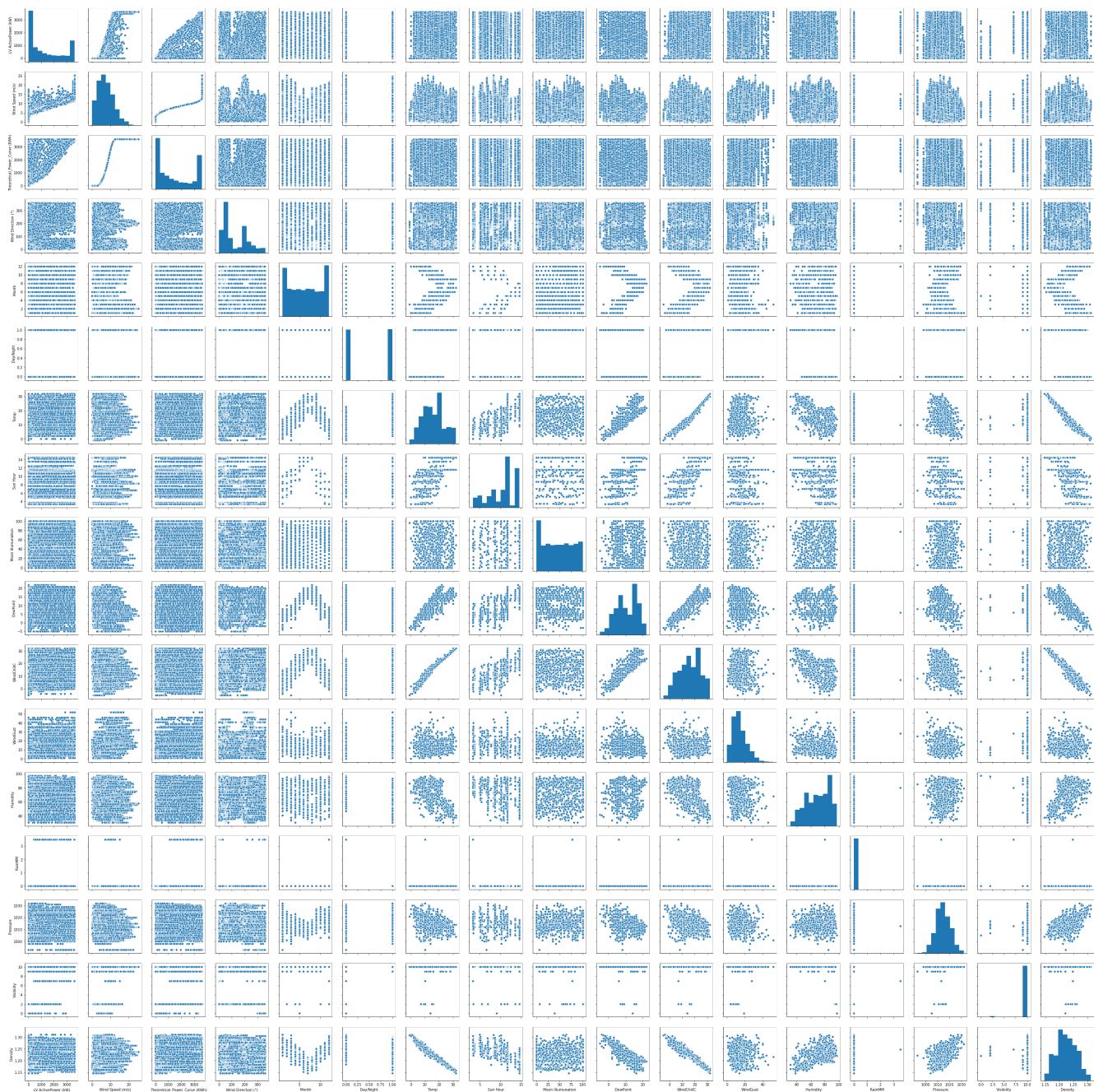
```
df2=df.drop(["Date","Time","Sunrise","Sunset","Moonrise","Moonset"],axis=1)
```

In [5]:

```
sns.pairplot(df2)
```

Out[5]:

```
<seaborn.axisgrid.PairGrid at 0x2251a18ca88>
```



```
In [1]:
```

```
import pandas as pd
import seaborn as sns
```

```
In [73]:
```

```
data=pd.read_excel("/home/yasin/Desktop/Data.xlsx") #Reading Data From Excel File
```

```
In [3]:
```

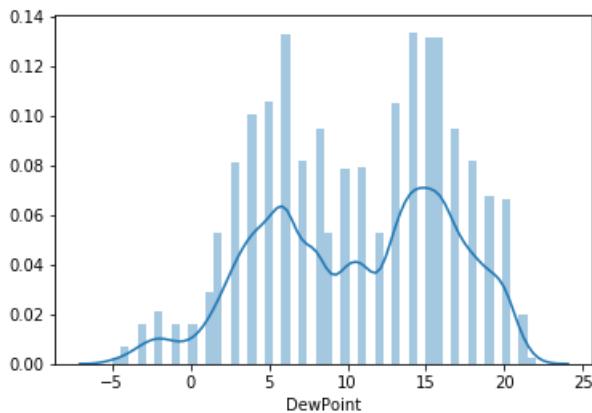
```
df=data.copy() # Copying data for just in case
```

```
In [29]:
```

```
sns.distplot(df["DewPoint"])
```

```
Out[29]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c5ed2d790>
```

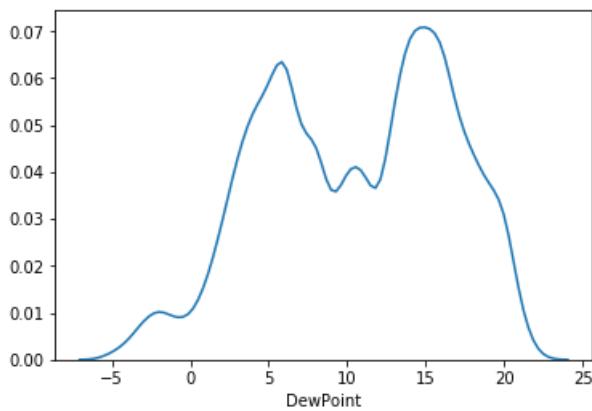


```
In [66]:
```

```
sns.distplot(df["DewPoint"], hist=False)
```

```
Out[66]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4e109cd0>
```



```
In [ ]:
```

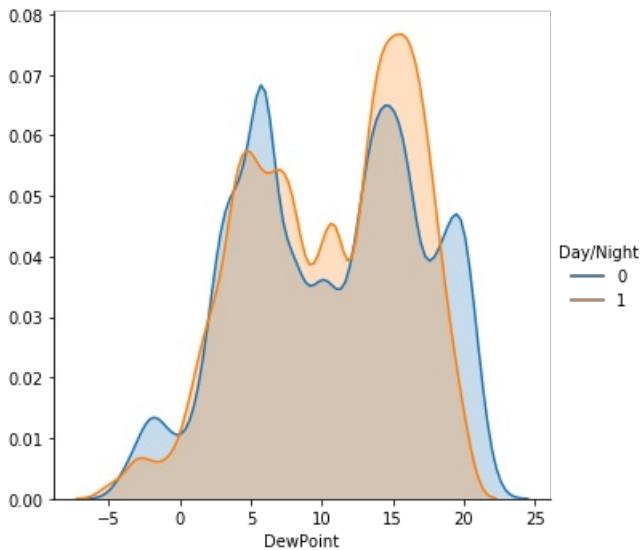
```
#Distribution of DewPoint
```

In [80]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"DewPoint",shade=True).add_legend()  
#Distribution of DewPoint by Day-Night
```

Out[80]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4cfb0c10>
```

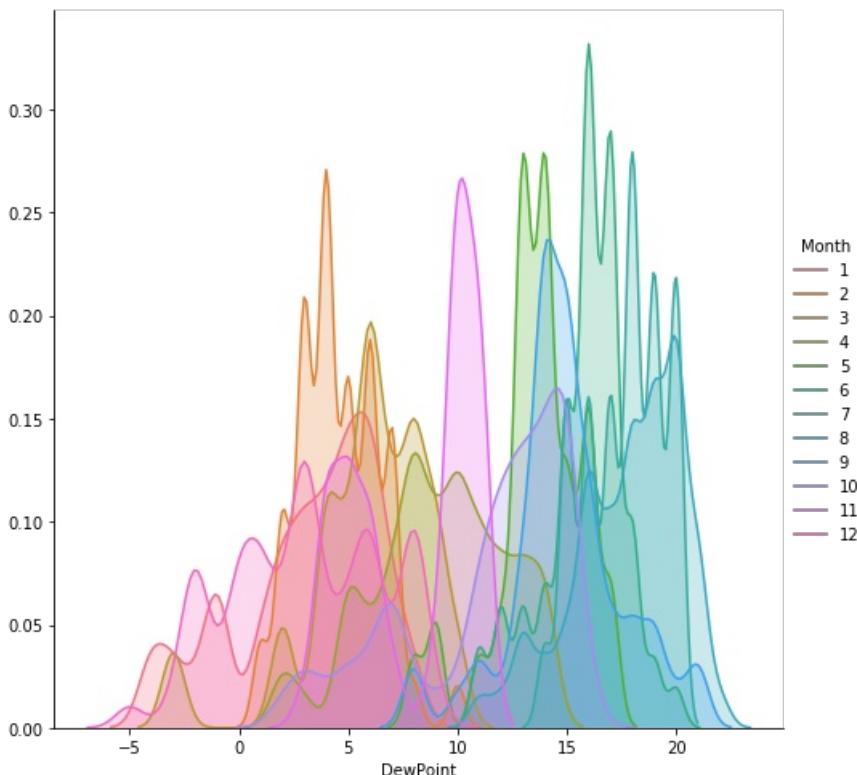


In [84]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"DewPoint",shade=True).add_legend()  
#Distribution of DewPoint, According to Months  
#It shows a lot of variation by months
```

Out[84]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4cbf3c50>
```



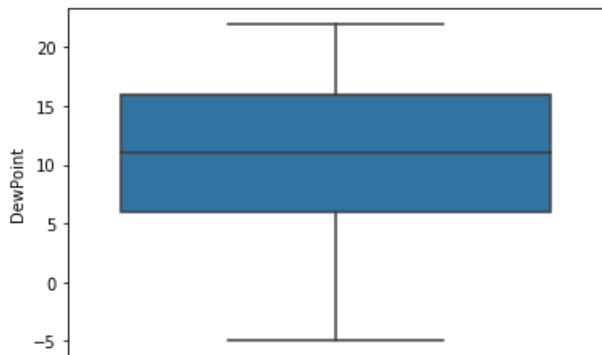
In [104]:

```
sns.boxplot(df["DewPoint"],orient="v")
```

```
# It seems -5 and max points, data seems no outlier value
```

Out[104]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4c799d90>
```



In [105]:

```
df["DewPoint"].describe() #Our boxplot shows that our data is distributed generally between 40%-75% range.  
# There seem to be no outliers in the boxplot.  
# Our median or 50% value -11- and mean value -10.5- seems highly close value.
```

Out[105]:

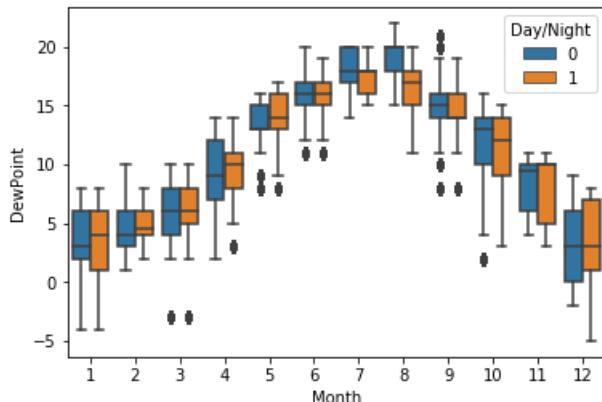
```
count    50530.000000  
mean     10.533089  
std      5.966731  
min     -5.000000  
25%      6.000000  
50%      11.000000  
75%      16.000000  
max     22.000000  
Name: DewPoint, dtype: float64
```

In [106]:

```
sns.boxplot(y=df["DewPoint"], orient="v", x="Month", data=df, hue="Day/Night")  
# We can also see here that throughout the months, day and night Dewpoint values  
# Generally Dewpoint is higher in the 5 - 8 month range  
# and Range of Dewpoint Day and night is so close  
# Some montyh have lots of outliers
```

Out[106]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4c71ba10>
```

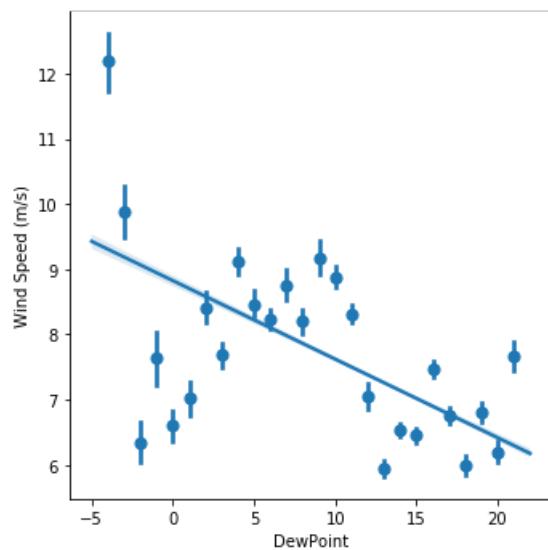


In [30]:

```
sns.lmplot(x="DewPoint",y="Wind Speed (m/s)",data=df,height=5,x_bins=500)
```

Out [30]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c8fc50b50>
```



In [ ]:

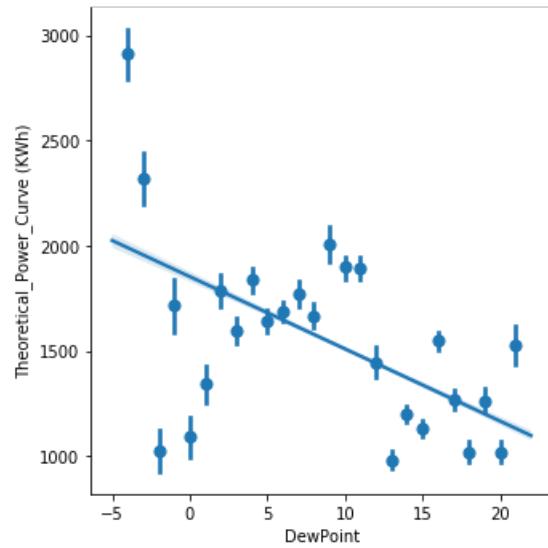
```
#DewPoint and Wind Speed seem to have a negative linear relationship
```

In [31]:

```
sns.lmplot(x="DewPoint",y="Theoretical_Power_Curve (KWh)",data=df,height=5,x_bins=500)
```

Out [31]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c7c7056d0>
```



In [ ]:

```
#DewPoint and Theoretical_Power_Curve (KWh) seem to have a weak linear relationship
```

In [ ]:

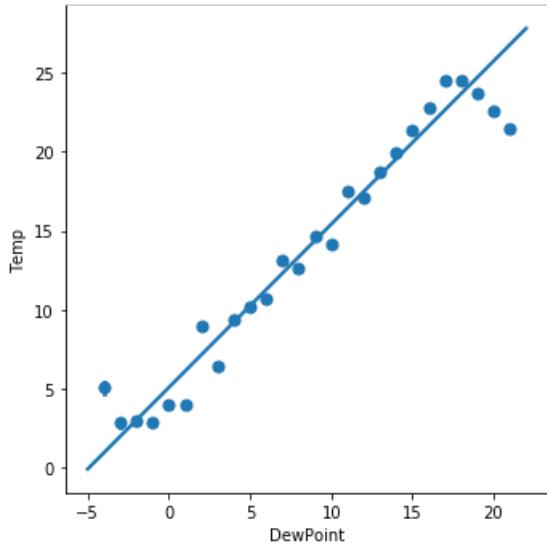
```
#But it is not so clear in this graph
```

In [32]:

```
sns.lmplot(x="DewPoint",y="Temp",data=df,height=5,x_bins=500)
```

Out [32]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c8f3a9a50>
```



In [ ]:

```
#DewPoint and Temperature seem to have a strong linear relationship
```

In [ ]:

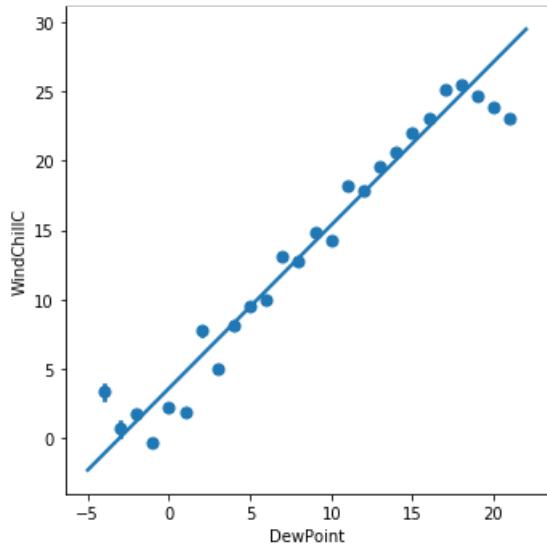
```
#They connected each other
```

In [34]:

```
sns.lmplot(x="DewPoint",y="WindChillC",data=df,height=5,x_bins=500)
```

Out [34]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c8f3b3090>
```



In [ ]:

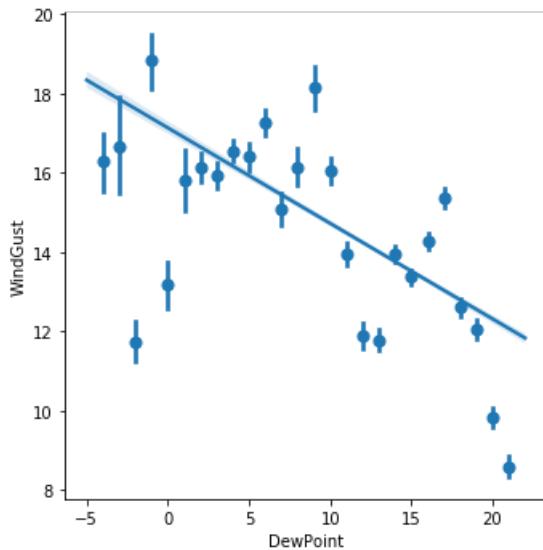
```
#DewPoint and Wind Chill seem to have a strong linear relationship
```

```
In [35]:
```

```
sns.lmplot(x="DewPoint",y="WindGust",data=df,height=5,x_bins=500)
```

```
Out[35]:
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8c8f25ea90>
```



```
In [ ]:
```

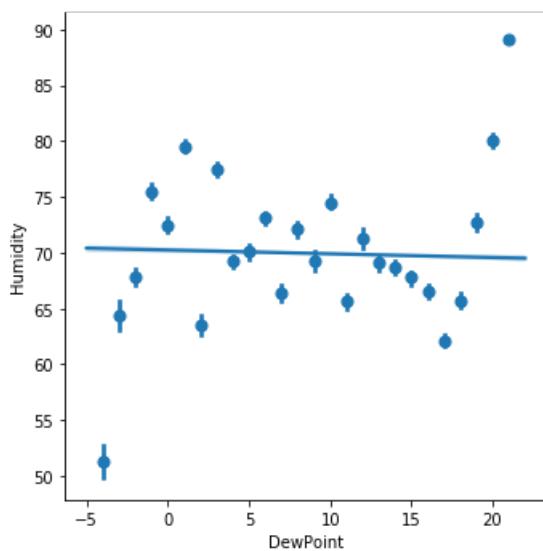
```
#DewPoint and Wind Gust seem to have a negative linear relationship
```

```
In [36]:
```

```
sns.lmplot(x="DewPoint",y="Humidity",data=df,height=5,x_bins=500)
```

```
Out[36]:
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8c8f0fddd0>
```



```
In [ ]:
```

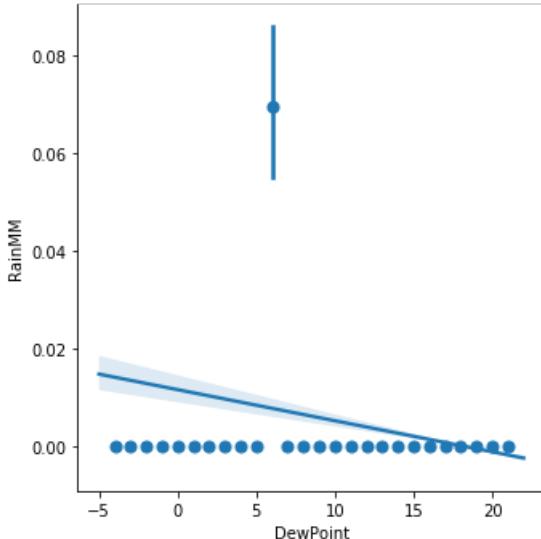
```
#DewPoint and Humidity seem to have not relationship
```

```
In [37]:
```

```
sns.lmplot(x="DewPoint",y="RainMM",data=df,height=5,x_bins=500)
```

Out [37] :

```
<seaborn.axisgrid.FacetGrid at 0x7f8c8f093750>
```



In [ ]:

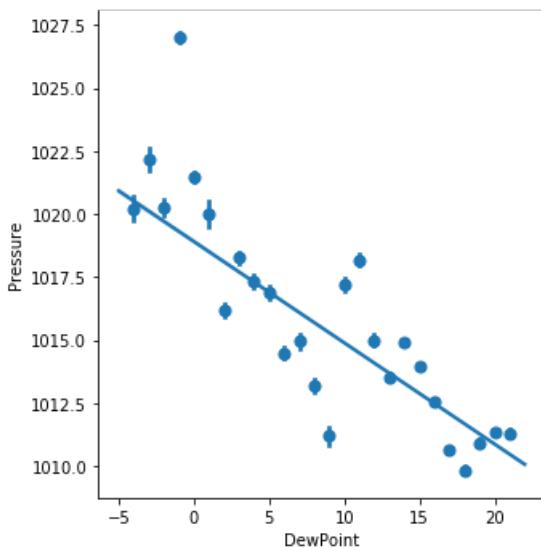
```
#DewPoint and Rain MM seem to have not relationship
```

In [38] :

```
sns.lmplot(x="DewPoint",y="Pressure",data=df,height=5,x_bins=500)
```

Out [38] :

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4feb23d0>
```



In [ ]:

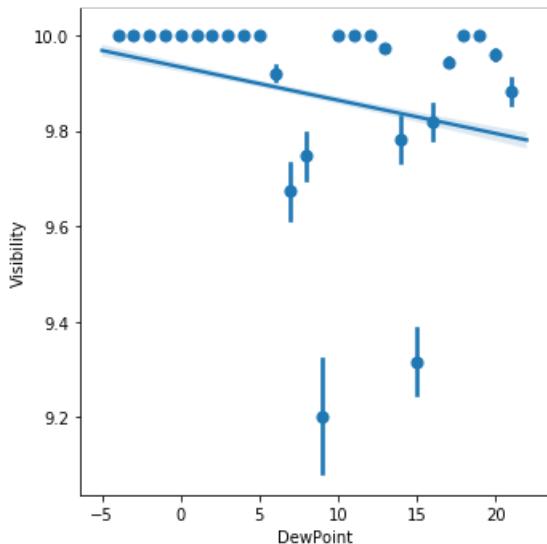
```
#DewPoint and Pressure seem to have a strong linear relationship
```

In [39] :

```
sns.lmplot(x="DewPoint",y="Visibility",data=df,height=5,x_bins=500)
```

Out [39] :

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4feb2610>
```



In [ ]:

```
#DewPoint and Visibility seem to have not relationship or little relationship
```

In [ ]:

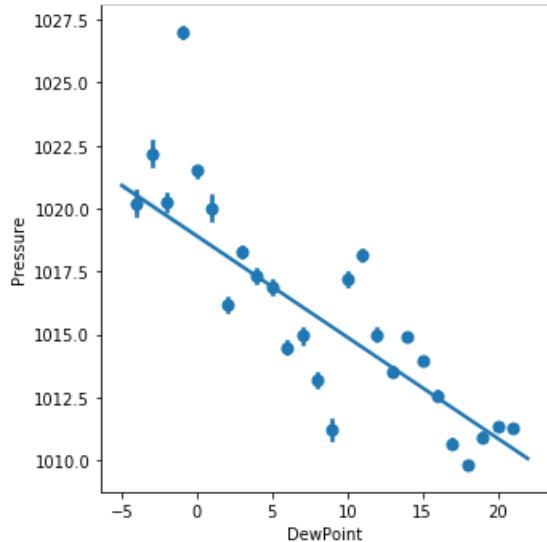
```
#not enough information to interpret
```

In [40]:

```
sns.lmplot(x="DewPoint",y="Pressure",data=df,height=5,x_bins=500)
```

Out[40]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c8f1b6f10>
```



In [ ]:

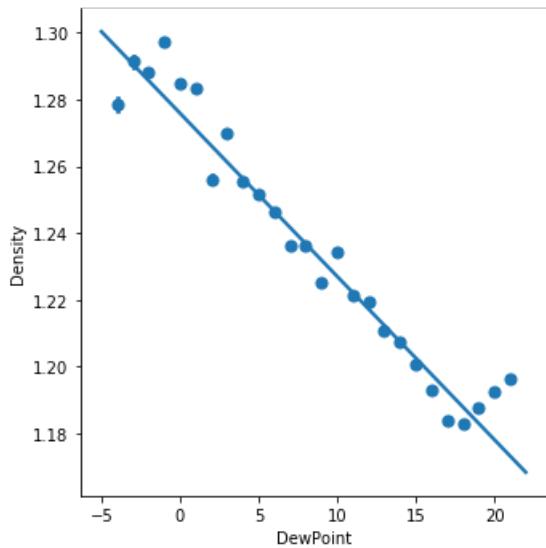
```
#DewPoint and Pressure seem to have strong relationship as negative
```

In [41]:

```
sns.lmplot(x="DewPoint",y="Density",data=df,height=5,x_bins=500)
```

Out[41]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4fe45350>
```



In [ ]:

```
#DewPoint and Density seem to have strong linear relationship as negative
```

In [ ]:

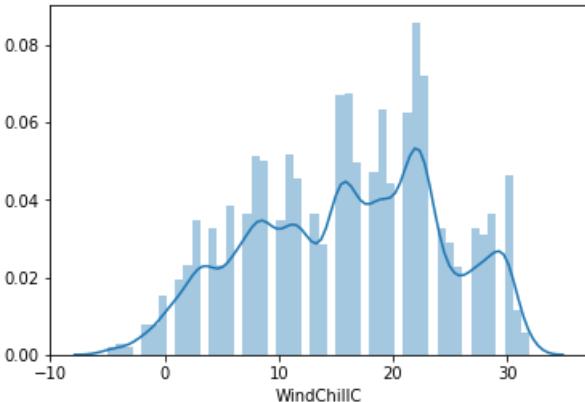
```
# Graph DataAnalysis Of WindChillC
```

In [70]:

```
sns.distplot(df["WindChillC"], hist=True)
```

Out[70]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4de6c0d0>
```

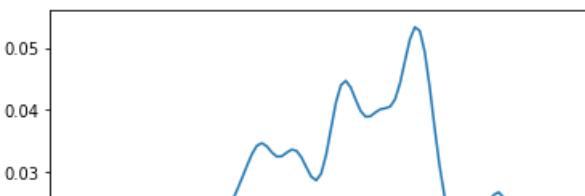


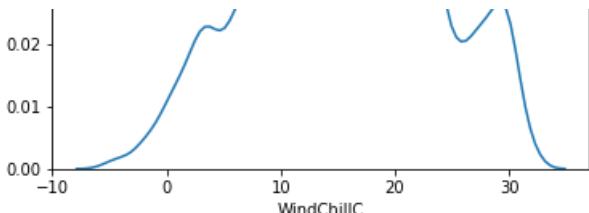
In [71]:

```
sns.distplot(df["WindChillC"], hist=False)
```

Out[71]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4dd5dd50>
```





In [ ]:

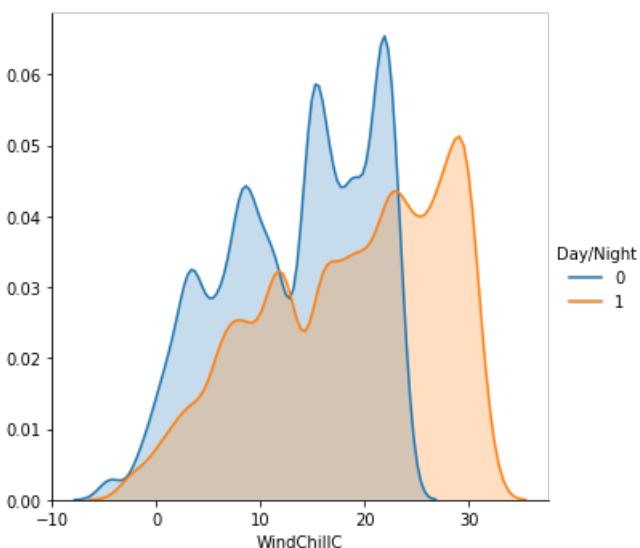
```
#Distribution of WindChill
```

In [81]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"WindChillC",shade=True).add_legend()  
#Distribution of WindChill by Day-Night
```

Out[81]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4cf8ded0>
```

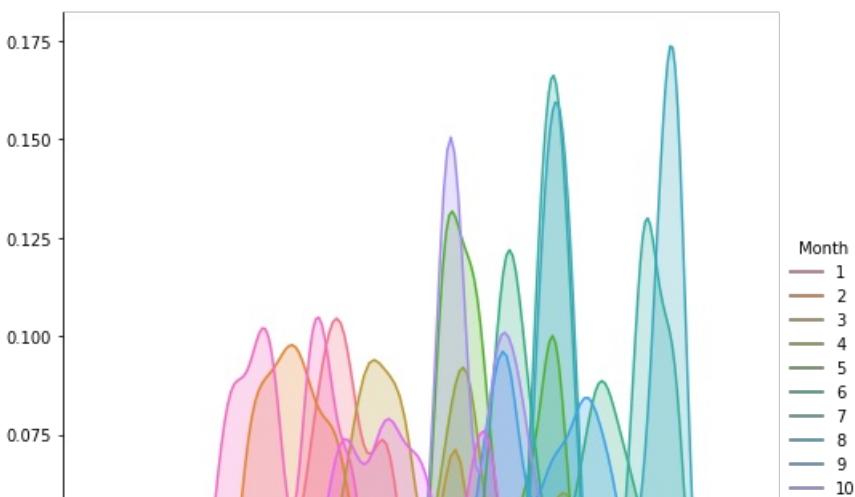


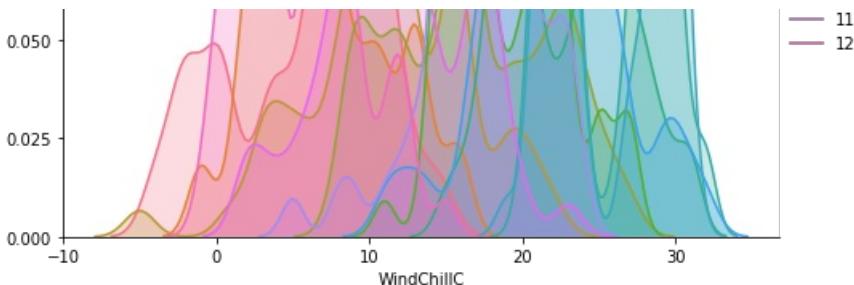
In [87]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"WindChillC",shade=True).add_legend()  
#Distribution of WindChillC, According to Months  
#It shows a lot of variation by months
```

Out[87]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c5e01f590>
```



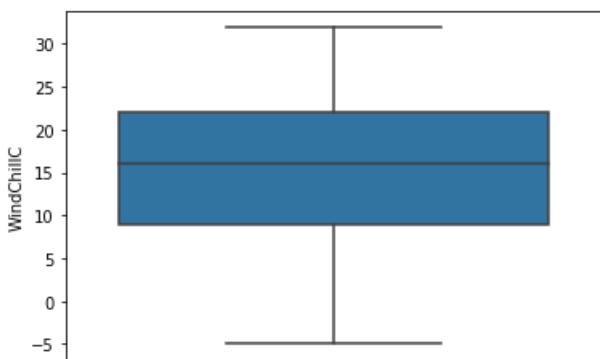


In [101]:

```
sns.boxplot(df["WindChillC"], orient="v")
# It seems -5 and max points, data seems no outlier value
```

Out[101]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4c828050>
```



In [103]:

```
df["WindChillC"].describe() #Our boxplot shows that our data is distributed generally between 25%-75% range.
# There seem to be no outliers in the boxplot.
# Our median or 50% value -16- and mean value -15.9- seems highly close value.
```

Out[103]:

```
count    50530.000000
mean     15.989234
std      8.341774
min     -5.000000
25%      9.000000
50%     16.000000
75%     22.000000
max     32.000000
Name: WindChillC, dtype: float64
```

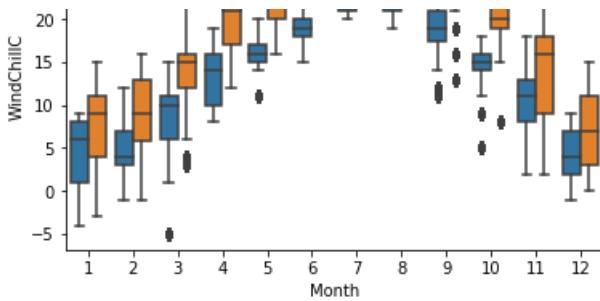
In [108]:

```
sns.boxplot(y=df["WindChillC"], orient="v", x="Month", data=df, hue="Day/Night")
# We can also see here that throughout the months, day and night WindChillC values
# Generally WindChillC is higher in the 5 - 10 month range
#and Range of Dewpoint Day and night is so different in the 5-6-7-8-9-10th months
#Some montyh have lots of outliers
```

Out[108]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4c232ad0>
```



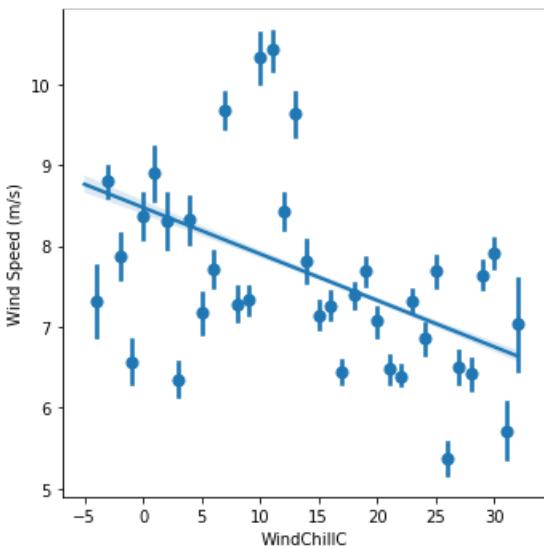


In [42]:

```
sns.lmplot(x="WindChillC", y="Wind Speed (m/s)", data=df, height=5, x_bins=500)
```

Out[42]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4fc29ed0>
```



In [ ]:

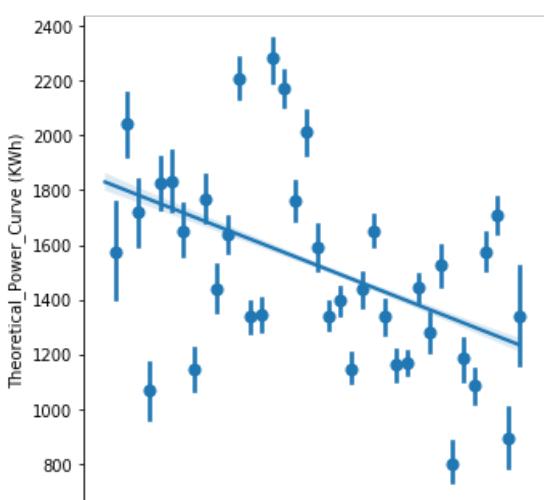
```
#Wind Chill and Wind Speed seem to have a negative linear relationship
```

In [43]:

```
sns.lmplot(x="WindChillC", y="Theoretical_Power_Curve (KWh)", data=df, height=5, x_bins=500)
```

Out[43]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4fa05710>
```





In [ ]:

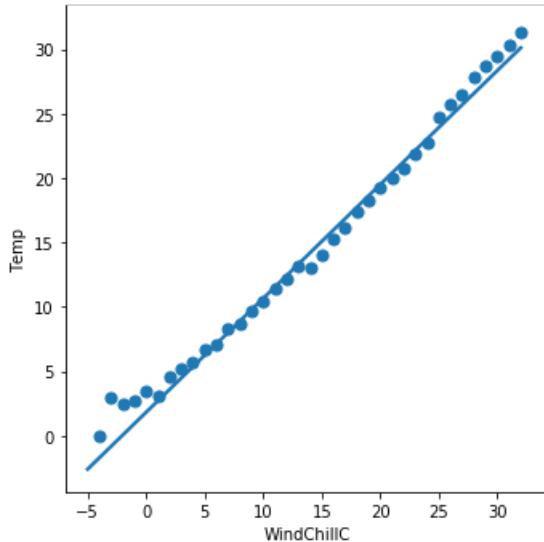
```
#Wind Chill and Theoretical_Power_Curve (KWh) seem to have a negative linear relationship
```

In [44]:

```
sns.lmplot(x="WindChillC",y="Temp",data=df,height=5,x_bins=500)
```

Out[44]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4fd3f650>
```



In [ ]:

```
#Wind Chill and Temp seem to have a strong linear relationship
```

In [ ]:

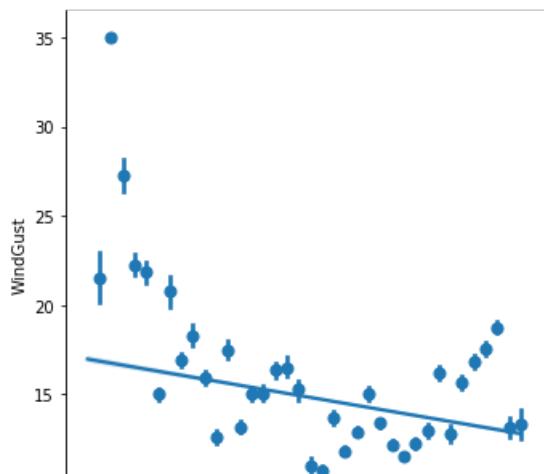
```
#Probably they connect each other directly
```

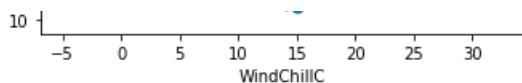
In [45]:

```
sns.lmplot(x="WindChillC",y="WindGust",data=df,height=5,x_bins=500)
```

Out[45]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4f8c1b10>
```





In [ ]:

```
#Wind Chill and Wind Gust seem to have little or not linear relationship
```

In [ ]:

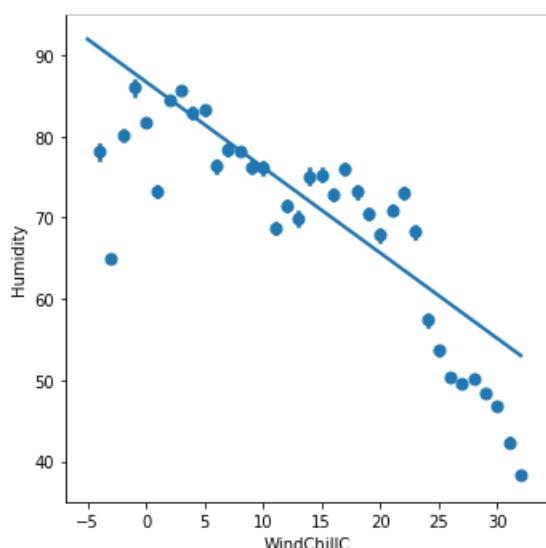
```
#not enough information to interpret
```

In [67]:

```
sns.lmplot(x="WindChillC", y="Humidity", data=df, height=5, x_bins=500)
```

Out[67]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4e0a2090>
```



In [ ]:

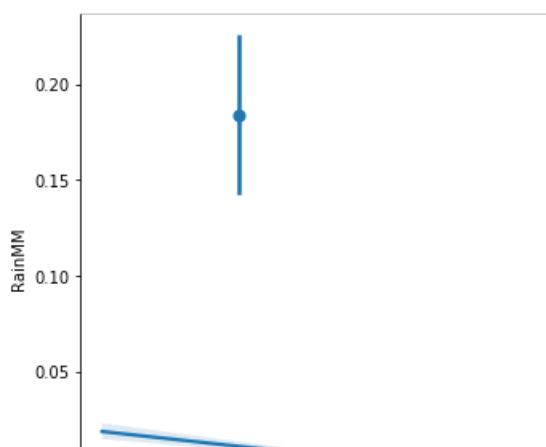
```
#Wind Chill and Humidity seem to have a negative linear relationship
```

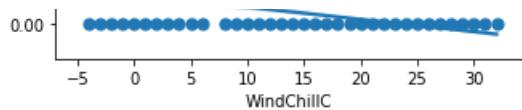
In [49]:

```
sns.lmplot(x="WindChillC", y="RainMM", data=df, height=5, x_bins=500)
```

Out[49]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4f51dc10>
```





In [ ]:

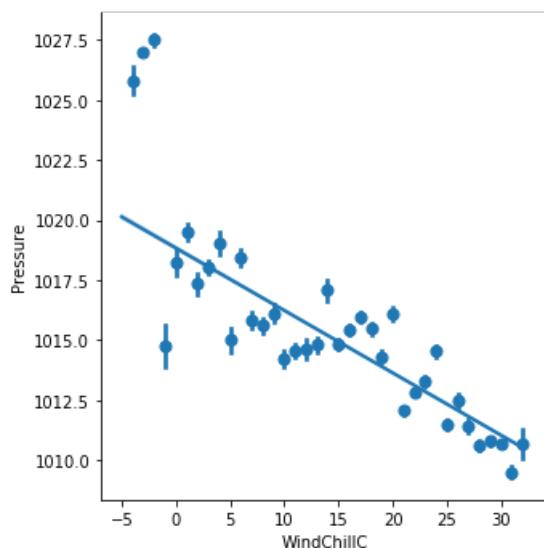
```
#Wind Chill and RainMM seem to have not relationship
```

In [50]:

```
sns.lmplot(x="WindChillC", y="Pressure", data=df, height=5, x_bins=500)
```

Out[50]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4f7c18d0>
```



In [ ]:

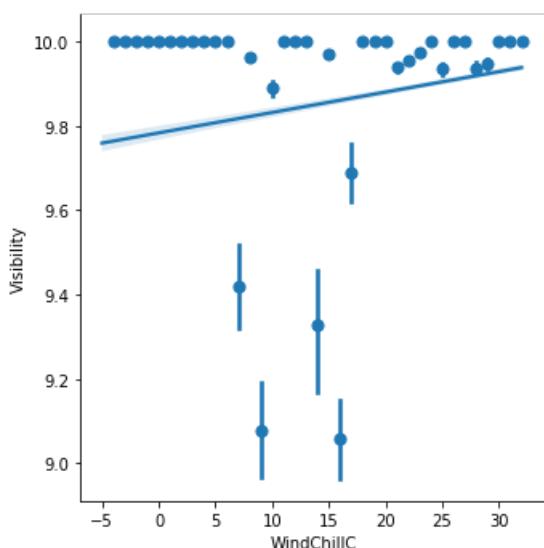
```
#Wind Chill and Pressure seem to have a negative linear relationship
```

In [47]:

```
sns.lmplot(x="WindChillC", y="Visibility", data=df, height=5, x_bins=500)
```

Out[47]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c50ad7410>
```



```
In [ ]:
```

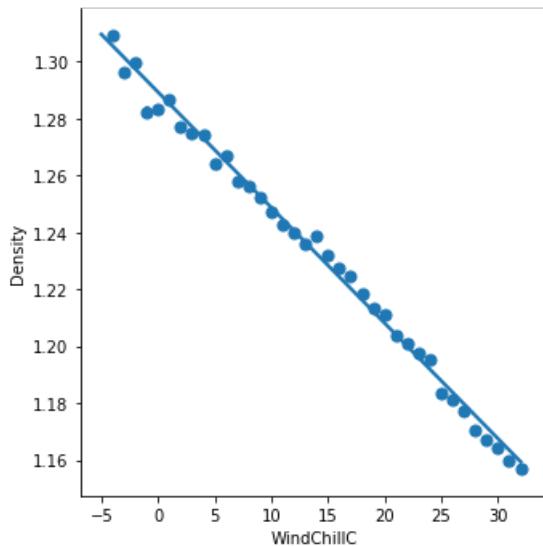
```
#Wind Chill and Visibility seem to have not relationship or little
```

```
In [48]:
```

```
sns.lmplot(x="WindChillC",y="Density",data=df,height=5,x_bins=500)
```

```
Out[48]:
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4f5623d0>
```



```
In [ ]:
```

```
#Wind Chill and Density seem to have strong relationship as negative
```

```
In [ ]:
```

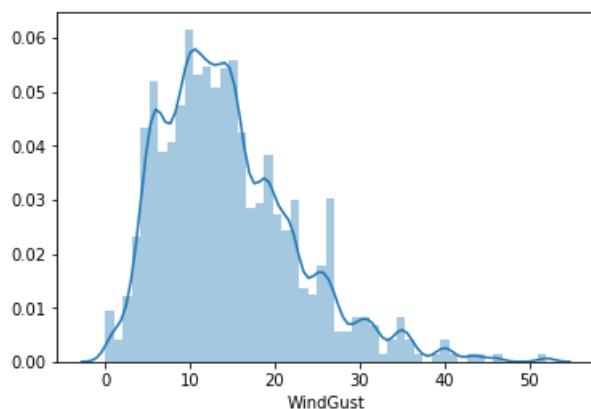
```
# Graph DataAnalysis of WindGust
```

```
In [91]:
```

```
sns.distplot(df["WindGust"],hist=True)
```

```
Out[91]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c5dc0d510>
```

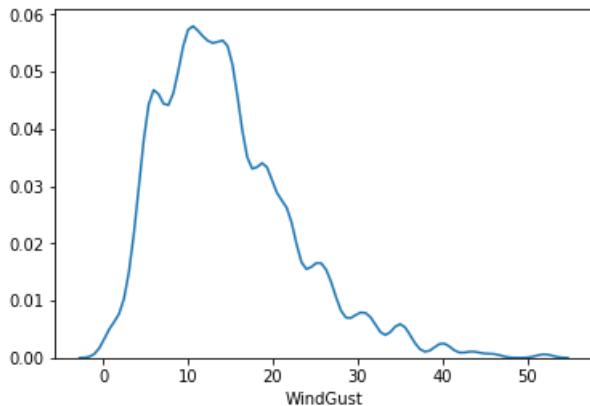


```
In [76]:
```

```
sns.distplot(df["WindGust"], hist=False)
```

Out[76]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4d517f50>
```



In [ ]:

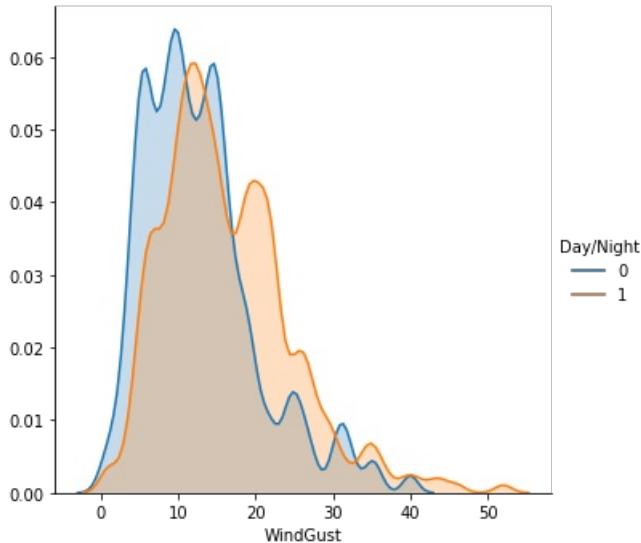
```
#Distribution Of WindGust
```

In [82]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"WindGust",shade=True).add_legend()  
#Distribution Of WindGust as DAY/NIGHT
```

Out[82]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4cdf6790>
```



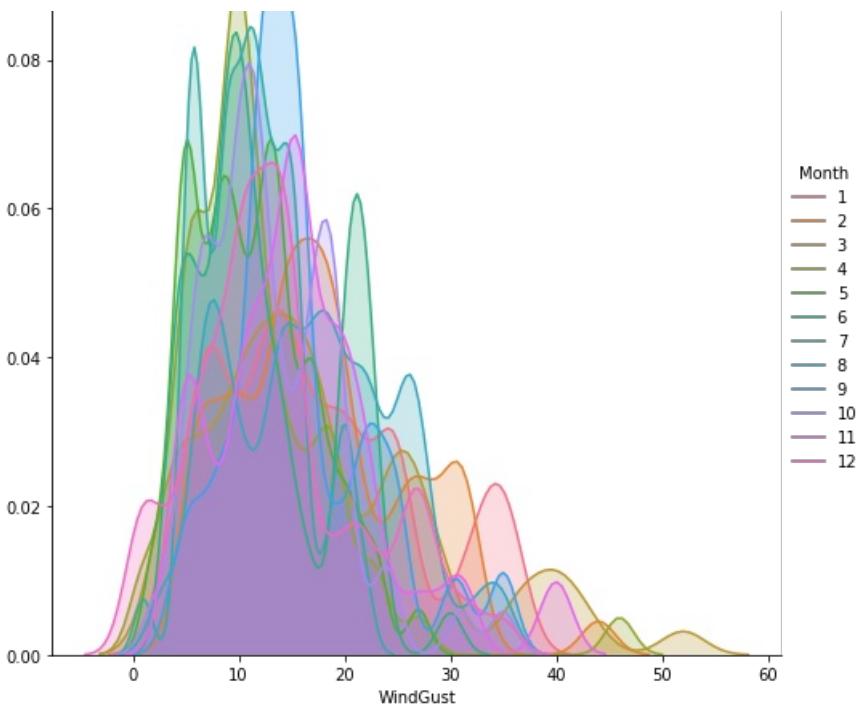
In [90]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"WindGust",shade=True).add_legend()  
#Distribution of WindGust, According to Months  
#It shows a lot of variation by months  
#When Windgust is in the 0-25 range, density so high
```

Out[90]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c5ddbe8d0>
```



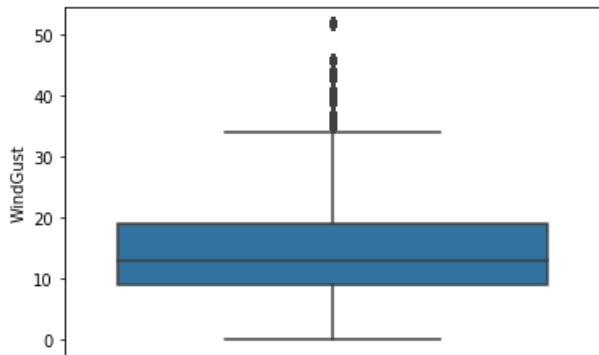


In [99]:

```
sns.boxplot(df["WindGust"], orient="v")
# It seems that other than 0 and max points, data seems much outlier value
```

Out[99]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4c896850>
```



In [100]:

```
df["WindGust"].describe() #Our boxplot shows that our data is distributed generally between 25%-40% range.
#There seem that much outliers in the boxplot.
# Our median or 50% value -13- and mean value -14.5- seems so
lose values.
```

Out[100]:

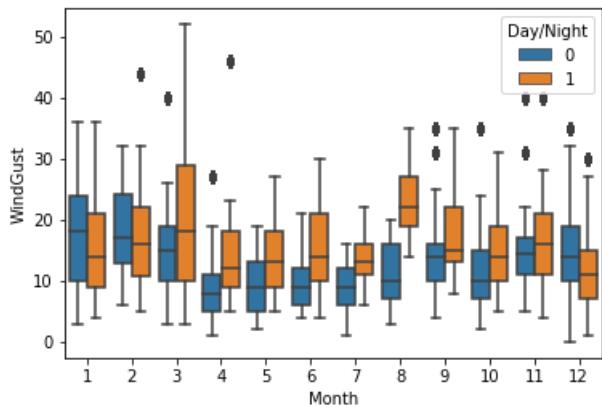
count	50530.000000
mean	14.597645
std	8.020789
min	0.000000
25%	9.000000
50%	13.000000
75%	19.000000
max	52.000000
Name:	WindGust, dtype: float64

In [109]:

```
sns.boxplot(y=df["WindGust"],orient="v",x="Month",data=df,hue="Day/Night")
# We can also see here that throughout the months, day and night WindGust values
#and Range of WindGust Day and night is so different in the 4 to 10th months
#Some montyh have lots of outliers
```

Out [109]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4c44d050>
```

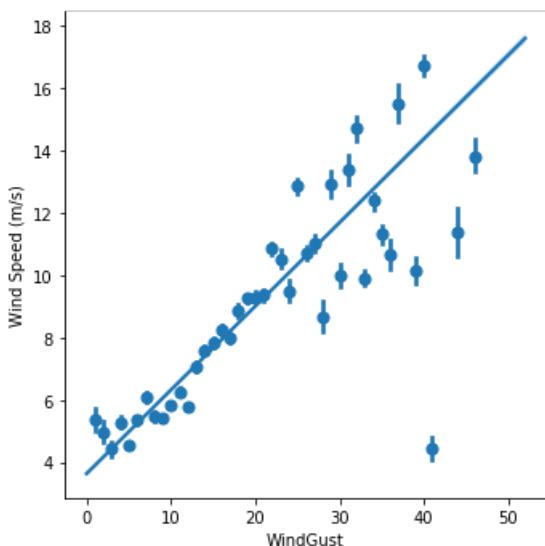


In [51]:

```
sns.lmplot(x="WindGust",y="Wind Speed (m/s)",data=df,height=5,x_bins=500)
```

Out [51]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4f31cf10>
```



In [ ]:

```
#WindGust and Wind Speed seem to have linear relationship
```

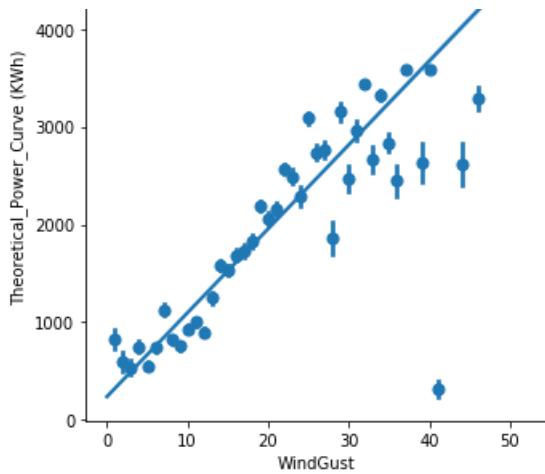
In [52]:

```
sns.lmplot(x="WindGust",y="Theoretical_Power_Curve (KWh)",data=df,height=5,x_bins=500)
```

Out [52]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4f1395d0>
```





In [ ]:

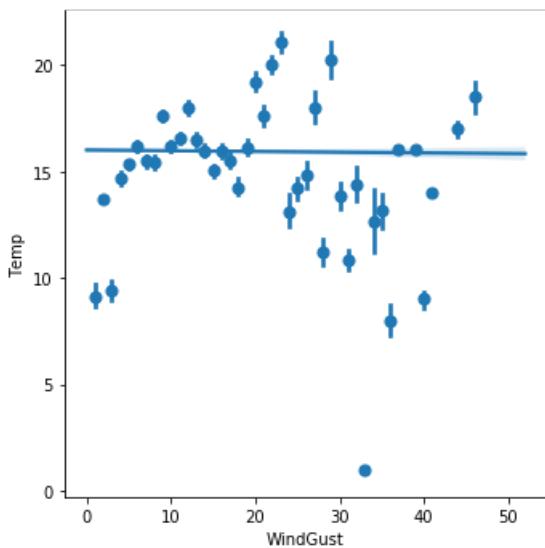
```
#Wind Gust and Theoretical_Power_Curve (KWh) seem to have strong linear relationship
```

In [53]:

```
sns.lmplot(x="WindGust",y="Temp",data=df,height=5,x_bins=500)
```

Out[53]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4f7c1710>
```



In [ ]:

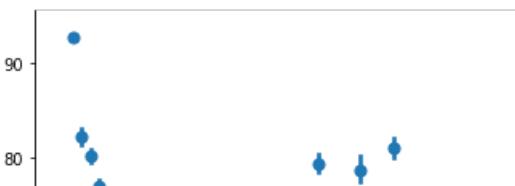
```
#Wind Gust and Temperature seem to have no relationship
```

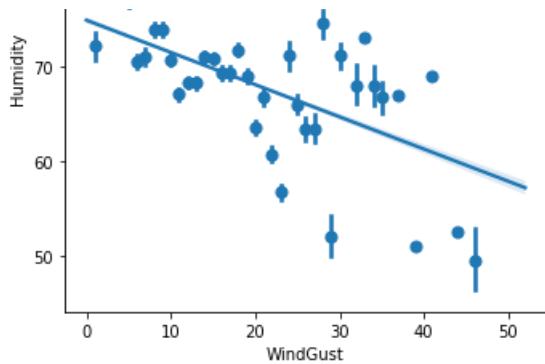
In [54]:

```
sns.lmplot(x="WindGust",y="Humidity",data=df,height=5,x_bins=500)
```

Out[54]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ef652d0>
```





In [ ]:

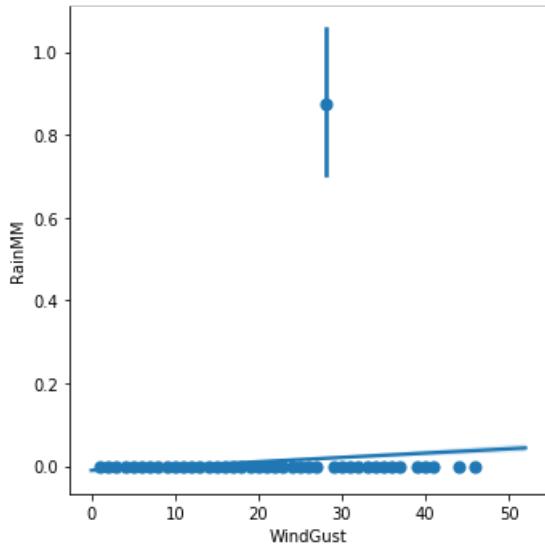
```
#Wind Gust and Humidity seem to have a relationship as negative
```

In [55]:

```
sns.lmplot(x="WindGust",y="RainMM",data=df,height=5,x_bins=500)
```

Out[55]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ee65110>
```



In [ ]:

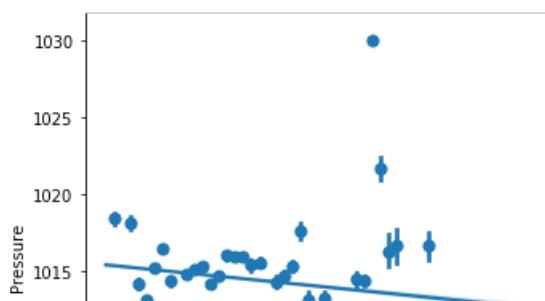
```
#Wind Gust and Rain MM seem to have not relationship
```

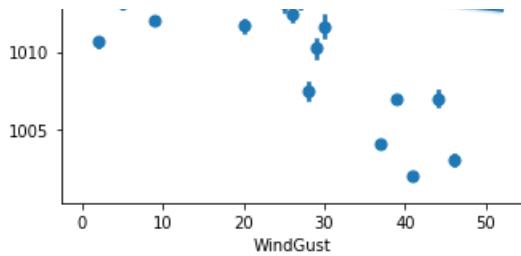
In [56]:

```
sns.lmplot(x="WindGust",y="Pressure",data=df,height=5,x_bins=500)
```

Out[56]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ee0a410>
```





In [ ]:

```
#Wind Gust and Pressure seem to have not or little relationship
```

In [ ]:

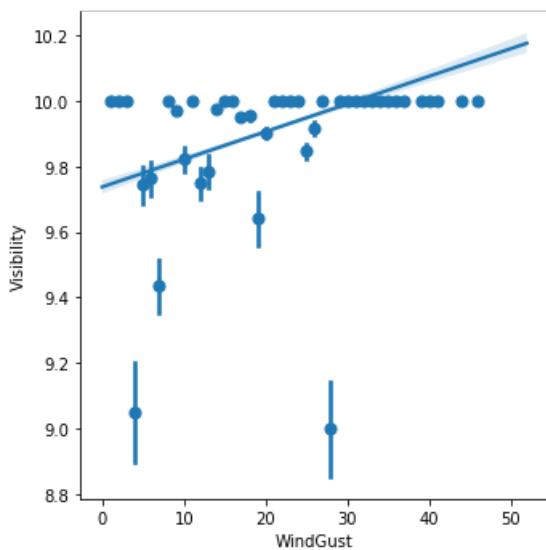
```
#not enough information to interpret
```

In [57]:

```
sns.lmplot(x="WindGust",y="Visibility",data=df,height=5,x_bins=500)
```

Out[57]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ed229d0>
```



In [ ]:

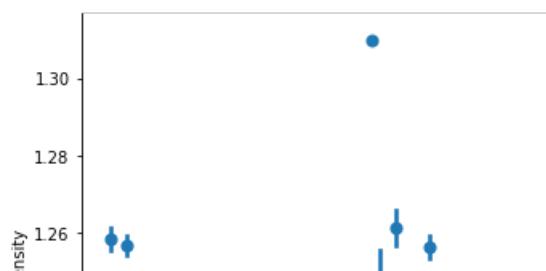
```
#Wind Gust and Visibility seem to have a linear relationship
```

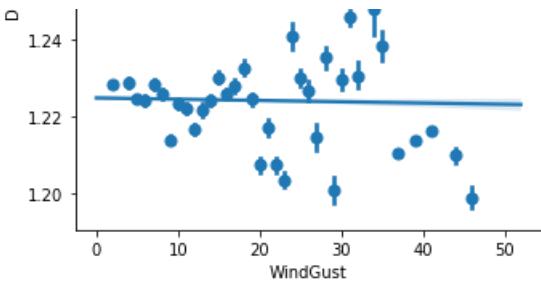
In [58]:

```
sns.lmplot(x="WindGust",y="Density",data=df,height=5,x_bins=500)
```

Out[58]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ee38050>
```





In [ ]:

```
#Wind Gust and Density seem to have not relationship
```

In [ ]:

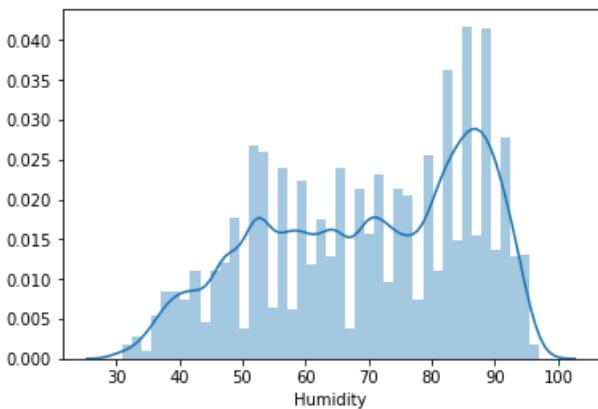
```
# Graph DataAnalysis Of Humidity
```

In [77]:

```
sns.distplot(df["Humidity"], hist=True)
```

Out[77]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4dd354d0>
```

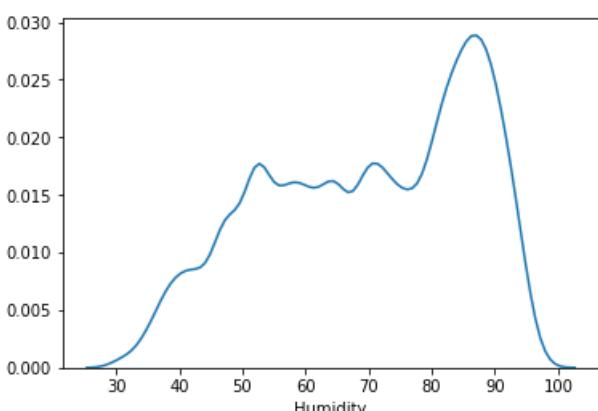


In [79]:

```
sns.distplot(df["Humidity"], hist=False)
```

Out[79]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4d0a9e10>
```



In [ ]:

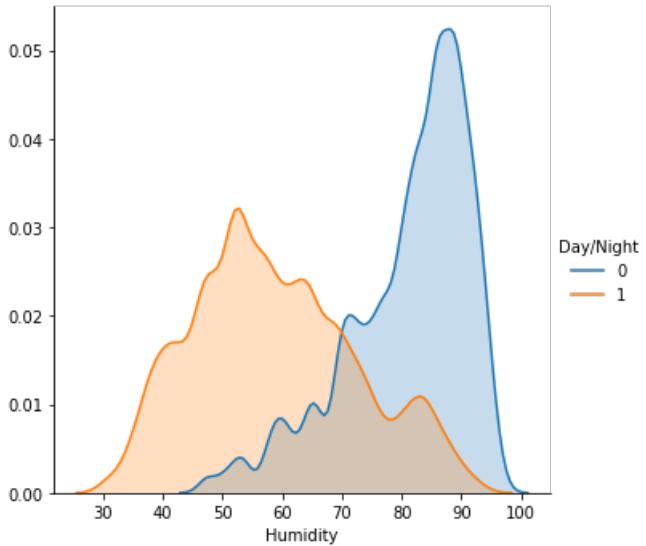
```
# Distribution Of Humidity
```

In [83]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Humidity",shade=True).add_legend()  
#Distribution Of Humidity as DAY/NIGHT
```

Out [83]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4cc96450>
```

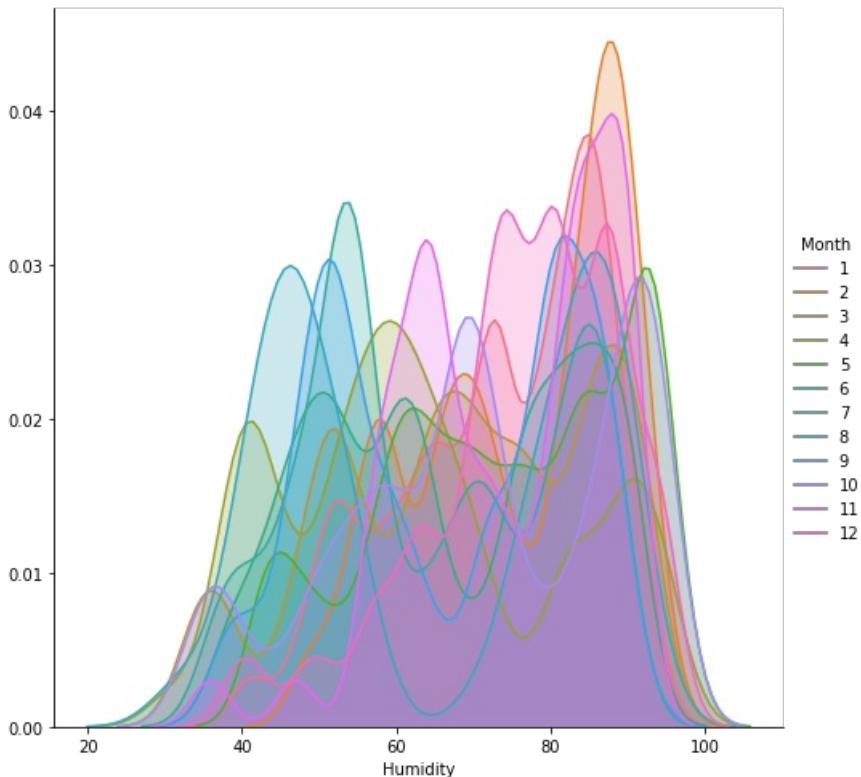


In [88]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"Humidity",shade=True).add_legend()  
#Distribution of Humidity, According to Months  
#It shows a lot of variation by months
```

Out [88]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c5df01ad0>
```

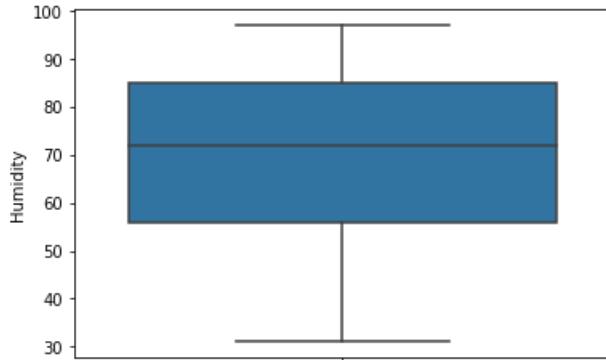


In [92]:

```
sns.boxplot(df["Humidity"],orient="v") # It seems that other than 30 and max points, data seems  
nearly equally  
# distributed.
```

Out[92]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c5dd0f5d0>
```



In [93]:

```
df["Humidity"].describe() #Our boxplot shows that our data is distributed generally between 40%-75  
% range.  
#There seem to be no outliers in the boxplot.  
# Our median or 50% value -72- and mean value -69.7- seems so  
lose values.
```

Out[93]:

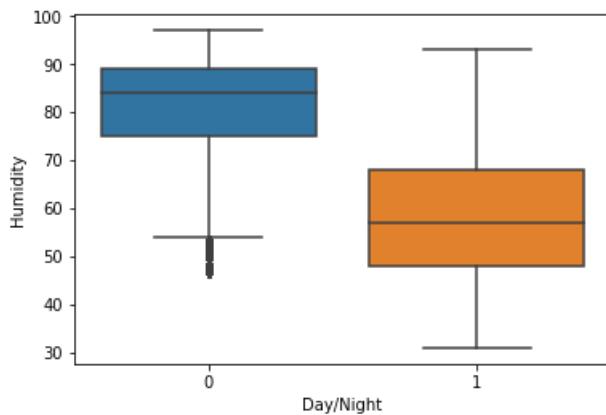
```
count    50530.000000  
mean     69.887592  
std      16.452907  
min      31.000000  
25%     56.000000  
50%     72.000000  
75%     85.000000  
max     97.000000  
Name: Humidity, dtype: float64
```

In [94]:

```
sns.boxplot(y=df["Humidity"],orient="v",x="Day/Night",data=df)  
#The difference between day and night humidity values
```

Out[94]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c5db3b9d0>
```

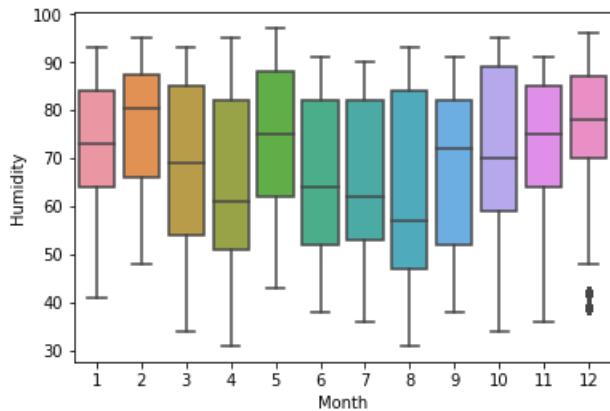


```
In [95]:
```

```
sns.boxplot(y=df["Humidity"], orient="v", x="Month", data=df)
#In months spectrum we can see that there is only Outlier in the 12th month
```

```
Out[95]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c5da48c10>
```

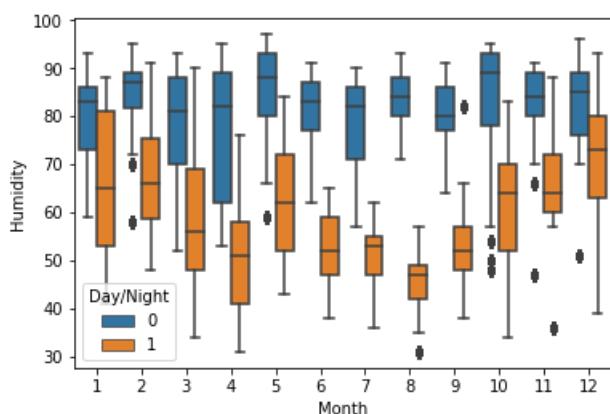


```
In [96]:
```

```
sns.boxplot(y=df["Humidity"], orient="v", x="Month", data=df, hue="Day/Night")
# We can also see here that humidity values the months, day and night
#different within the same month.
#we can see that some months have outlier value
#In generally, Humidity value is at higher values at night
```

```
Out[96]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c5da5eb90>
```

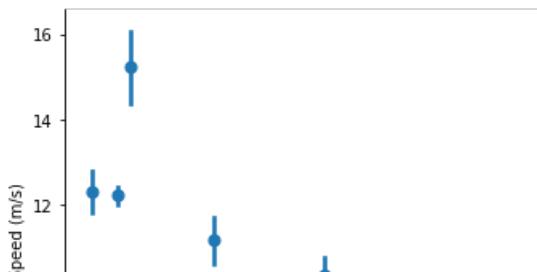


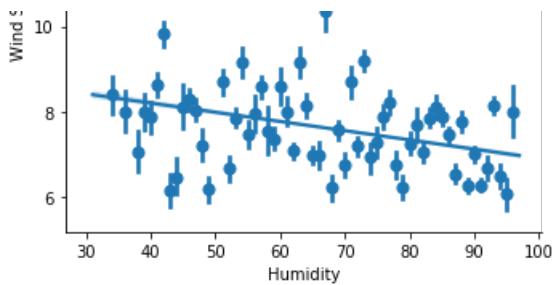
```
In [59]:
```

```
sns.lmplot(x="Humidity", y="Wind Speed (m/s)", data=df, height=5, x_bins=500)
```

```
Out[59]:
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4eab9350>
```





In [ ]:

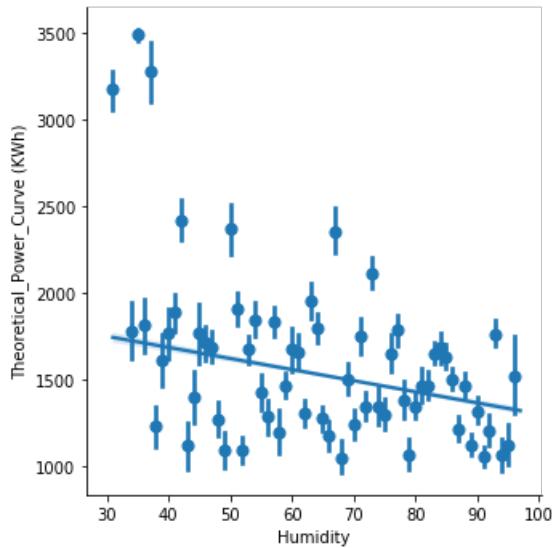
```
#Humidity and Wind Speed seem to have little linear relationship as negative
```

In [60]:

```
sns.lmplot(x="Humidity",y="Theoretical_Power_Curve (KWh)",data=df,height=5,x_bins=500)
```

Out[60]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4e9c7150>
```



In [ ]:

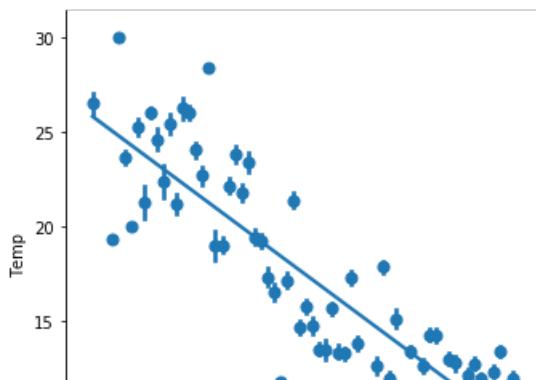
```
#Humidity and Theoretical_Power_Curve (KWh) seem to have little linear relationship as negative
```

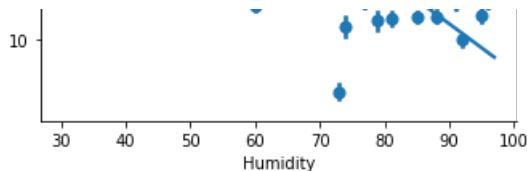
In [61]:

```
sns.lmplot(x="Humidity",y="Temp",data=df,height=5,x_bins=500)
```

Out[61]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ea06990>
```





In [ ]:

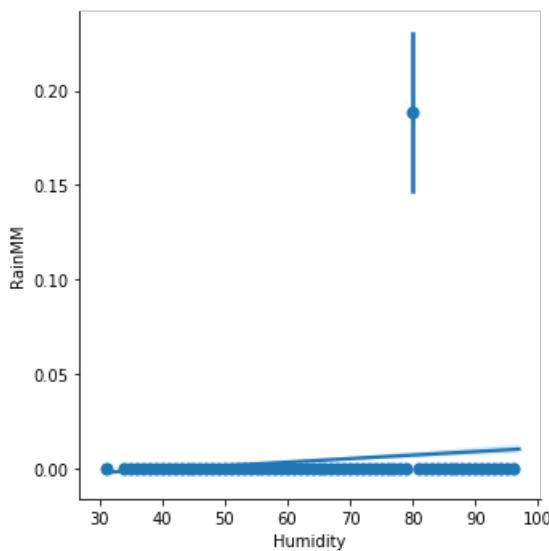
```
#Humidity and Temperature seem to have a strong linear relationship as negative
```

In [62]:

```
sns.lmplot(x="Humidity", y="RainMM", data=df, height=5, x_bins=500)
```

Out[62]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4e642bd0>
```



In [ ]:

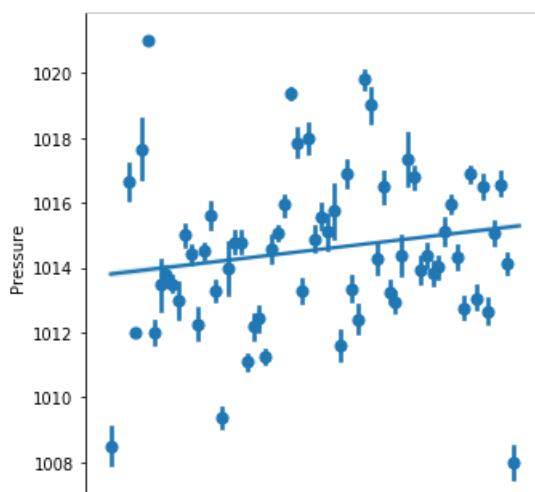
```
#Humidity and Rain MM seem to have not relationship
```

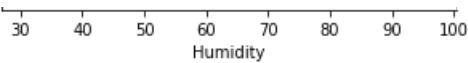
In [63]:

```
sns.lmplot(x="Humidity", y="Pressure", data=df, height=5, x_bins=500)
```

Out[63]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4e635390>
```





In [ ]:

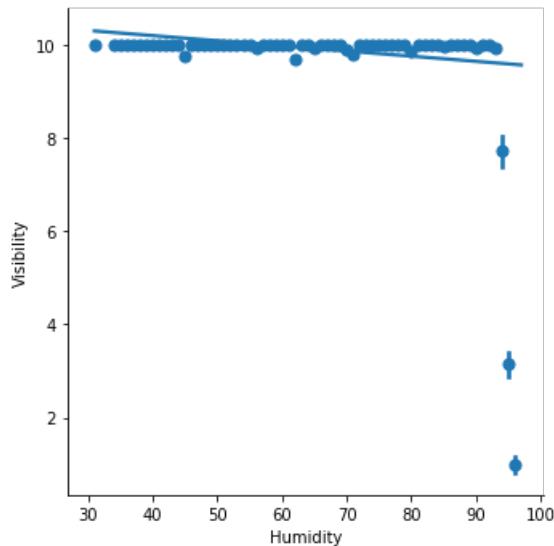
```
#Humidity and Pressure seem to have a linear relationship
```

In [64]:

```
sns.lmplot(x="Humidity",y="Visibility",data=df,height=5,x_bins=500)
```

Out[64]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4e399d10>
```



In [ ]:

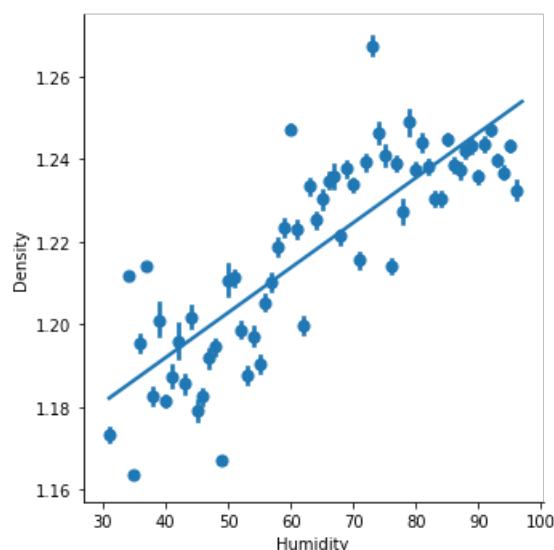
```
#Humidity and Visibility seem to have a not relationship
```

In [65]:

```
sns.lmplot(x="Humidity",y="Density",data=df,height=5,x_bins=500)
```

Out[65]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4e362f50>
```



```
In [ ]:
```

```
#Humidity and Density seem to have a linear relationship
```

```
In [ ]:
```

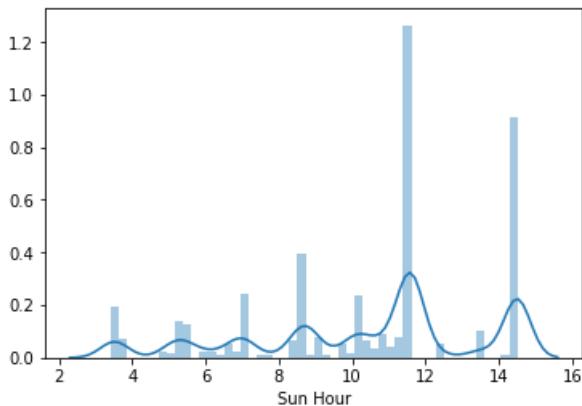
```
# Graph DataAnalysis Of SunHour
```

```
In [111]:
```

```
sns.distplot(df["Sun Hour"],hist=True)
```

```
Out[111]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4bca9dd0>
```

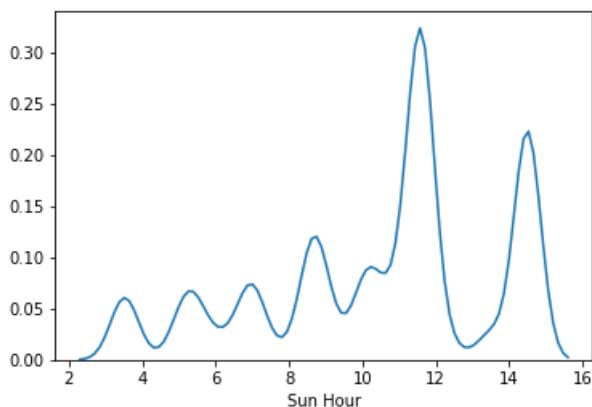


```
In [112]:
```

```
sns.distplot(df["Sun Hour"],hist=False)
```

```
Out[112]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c4bbc9f90>
```



```
In [ ]:
```

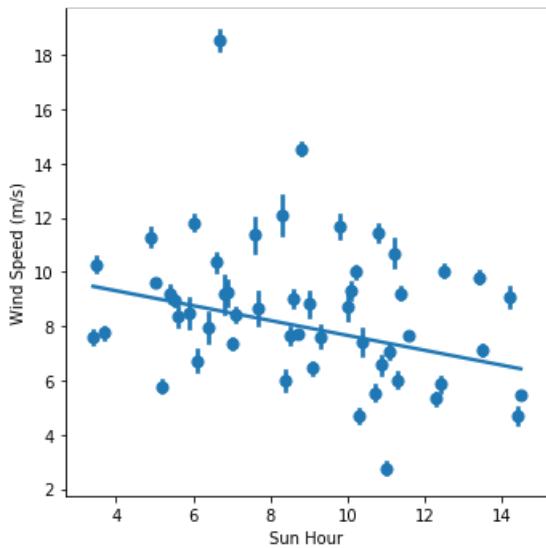
```
# Distribution Of SunOur
```

```
In [113]:
```

```
sns.lmplot(x="Sun Hour",y="Wind Speed (m/s)",data=df,height=5,x_bins=500)
```

```
Out[113]:
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ba91f10>
```



In [ ]:

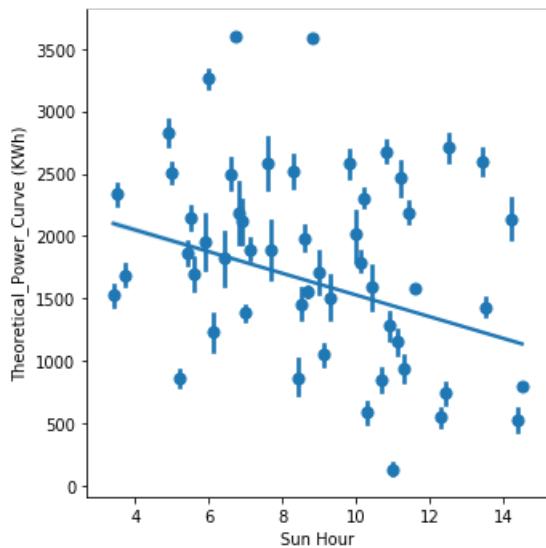
```
#Sun Hour and Wind Speed seem to have a negative linear relationship
```

In [114]:

```
sns.lmplot(x="Sun Hour",y="Theoretical_Power_Curve (KWh)",data=df,height=5,x_bins=500)
```

Out[114]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4ba03b90>
```



In [ ]:

```
#Sun Hour and Theoretical_Power_Curve (KWh) seem to have a negative linear relationship
```

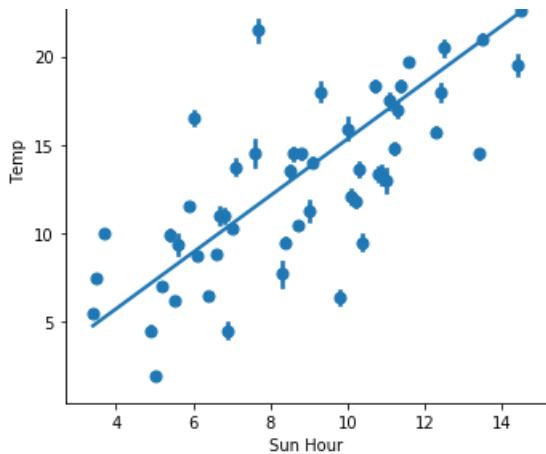
In [115]:

```
sns.lmplot(x="Sun Hour",y="Temp",data=df,height=5,x_bins=500)
```

Out[115]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4b802d10>
```





In [ ]:

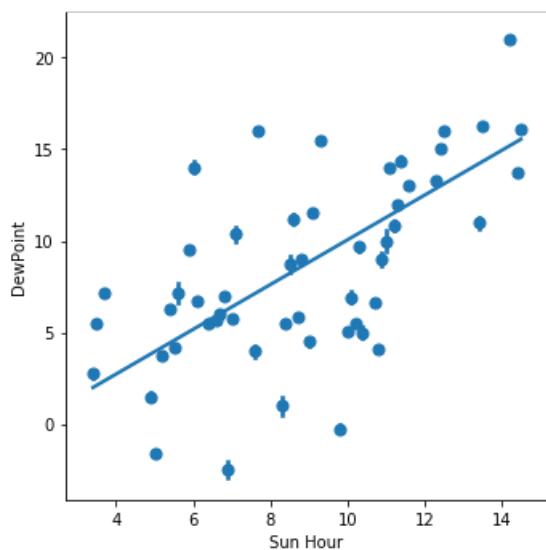
```
#Sun Hour and Temperature seem to have a linear relationship
```

In [116]:

```
sns.lmplot(x="Sun Hour",y="DewPoint",data=df,height=5,x_bins=500)
```

Out[116]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4b6d4e90>
```



In [ ]:

```
#Sun Hour and DewPoint seem to have a linear relationship
```

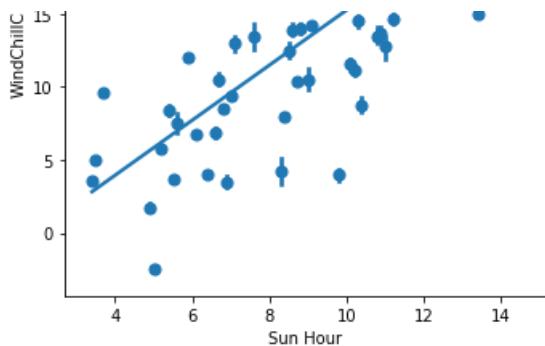
In [117]:

```
sns.lmplot(x="Sun Hour",y="WindChillC",data=df,height=5,x_bins=500)
```

Out[117]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4b787850>
```





In [ ]:

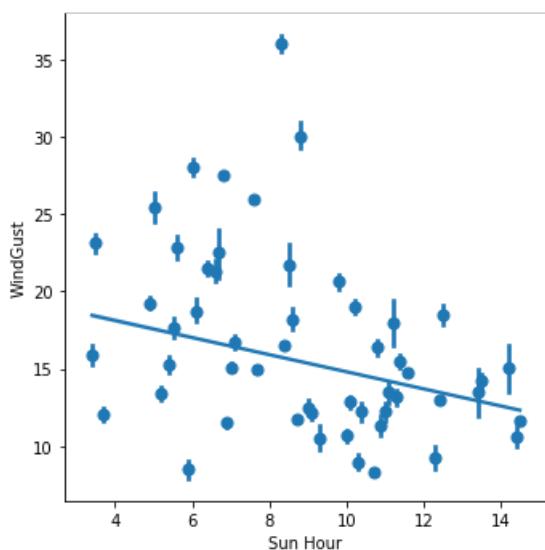
```
#Sun Hour and WindChillC seem to have a linear relationship
```

In [118]:

```
sns.lmplot(x="Sun Hour",y="WindGust",data=df,height=5,x_bins=500)
```

Out[118]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4b4934d0>
```



In [ ]:

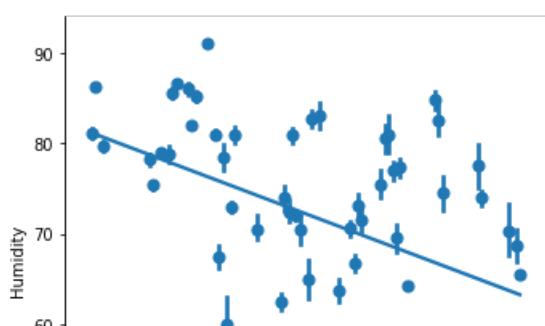
```
#Sun Hour and WindGust seem to have a linear relationship as negative
```

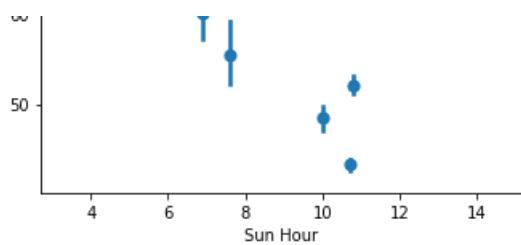
In [119]:

```
sns.lmplot(x="Sun Hour",y="Humidity",data=df,height=5,x_bins=500)
```

Out[119]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4b3fed50>
```





In [ ]:

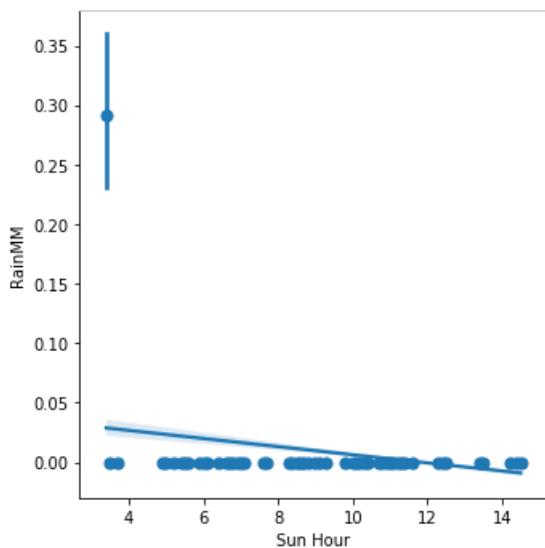
```
#Sun Hour and Humidity seem to have a linear relationship as negative
```

In [120]:

```
sns.lmplot(x="Sun Hour",y="RainMM",data=df,height=5,x_bins=500)
```

Out[120]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4b3fed10>
```



In [ ]:

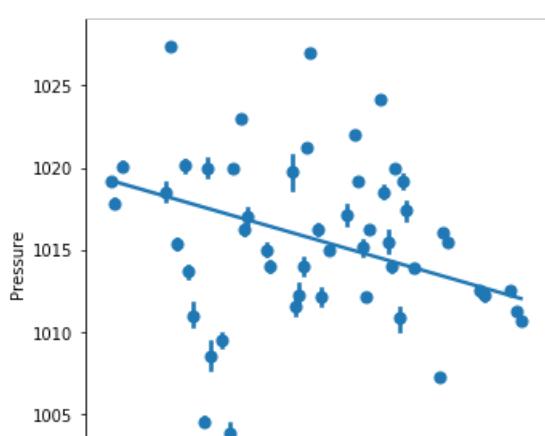
```
#Sun Hour and RainMM seem to have not relationship or little as negative
```

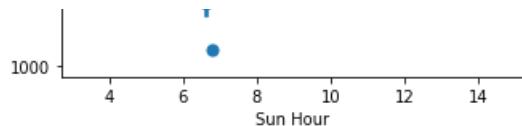
In [121]:

```
sns.lmplot(x="Sun Hour",y="Pressure",data=df,height=5,x_bins=500)
```

Out[121]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4b106e90>
```





In [ ]:

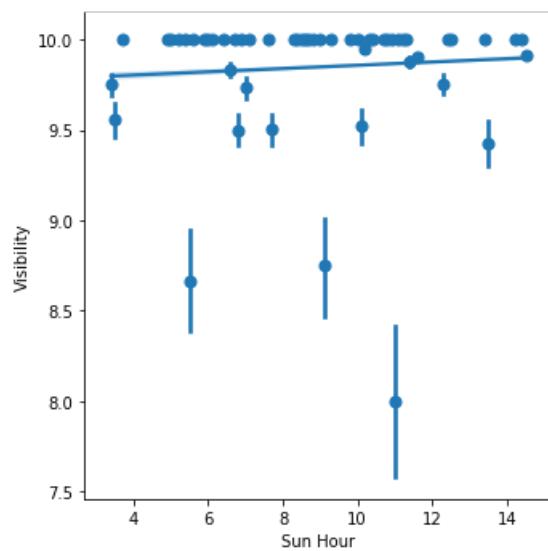
```
#Sun Hour and Pressure seem to have a linear relationship as negative
```

In [122]:

```
sns.lmplot(x="Sun Hour",y="Visibility",data=df,height=5,x_bins=500)
```

Out[122]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c4afbab10>
```



In [ ]:

```
#Sun Hour and Visibility seem to have little linear relationship or not
```

In [ ]:

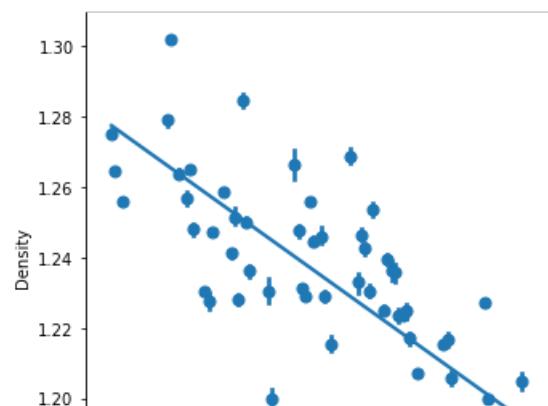
```
#not clear graph
```

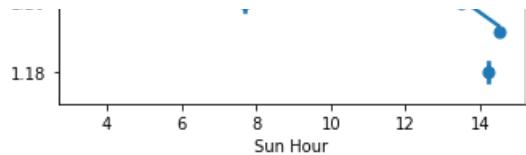
In [123]:

```
sns.lmplot(x="Sun Hour",y="Density",data=df,height=5,x_bins=500)
```

Out[123]:

```
<seaborn.axisgrid.FacetGrid at 0x7f8c5dc24490>
```





In [ ]:

```
#Sun Hour and Density seem to have a linear relationship as negative
```

In [33]:

```
df.columns
```

Out[33]:

```
Index([u'Unnamed: 0', u'Date',  
       u'Time', u'LV ActivePower (kW)',  
       u'Wind Speed (m/s)', u'Theoretical_Power_Curve (KWh)',  
       u'Wind Direction (°)', u'Month',  
       u'Day/Night', u'Temp',  
       u'Sun Hour', u'Moon Illumination',  
       u'Moonrise', u'Moonset',  
       u'Sunrise', u'Sunset',  
       u'DewPoint', u'WindChillC',  
       u'WindGust', u'Humidity',  
       u'RainMM', u'Pressure',  
       u'Visibility', u'Unnamed: 23',  
       u'Density'],  
      dtype='object')
```

In [ ]:

In [2]:

```
import pandas as pd
import seaborn as sns
```

In [4]:

```
df.Month=pd.Categorical(df.Month)
df[ "Day/Night" ]=pd.Categorical(df[ "Day/Night" ])
df.Date=pd.Categorical(df.Date)
df.Time=pd.Categorical(df.Time)
```

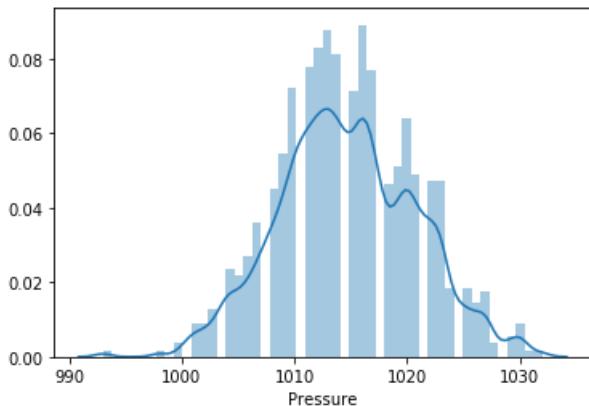
Plots for Our Target Variable

In [8]:

```
sns.distplot(df[ "Pressure" ])
```

Out [8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f68fd988>
```

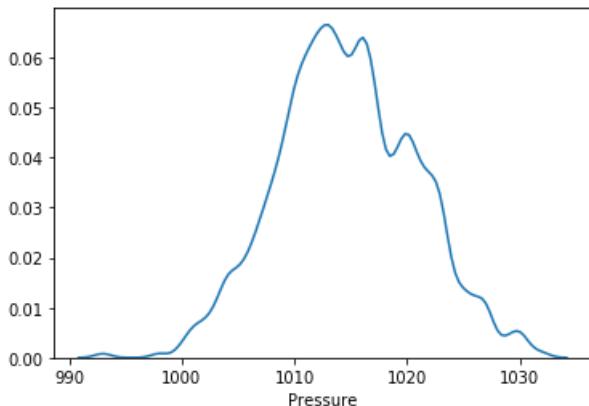


In [9]:

```
sns.distplot(df[ "Pressure" ],hist=False)
```

Out [9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f4290cc8>
```

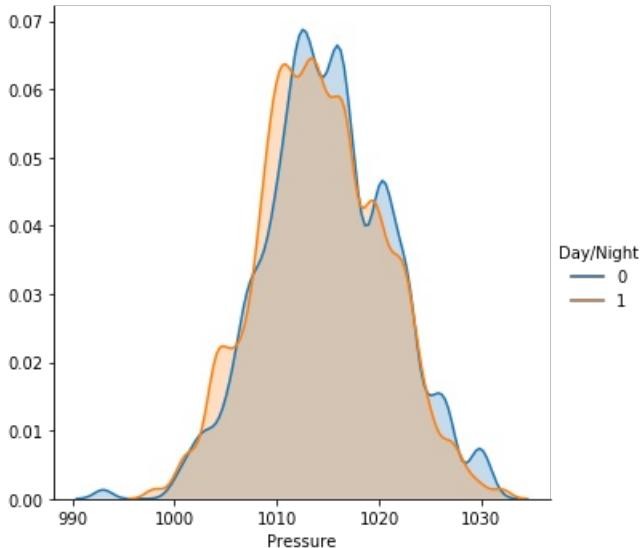


In [10]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Pressure",shade=True).add_legend()
#Distribution of Pressure by Day-Night
```

Out[10]:

```
<seaborn.axisgrid.FacetGrid at 0x240f5950cc8>
```

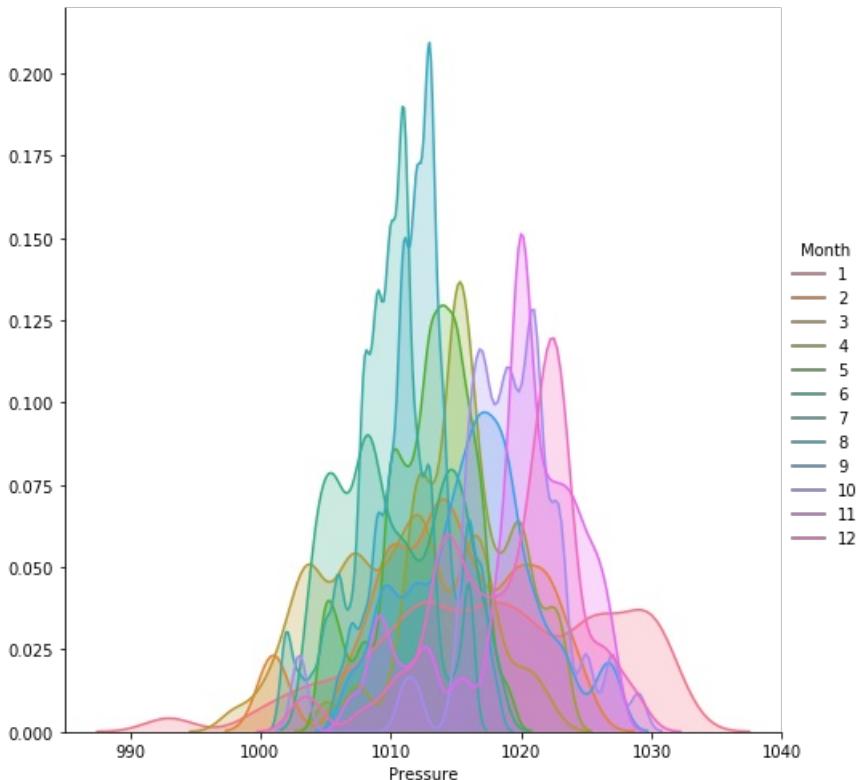


In [11]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"Pressure",shade=True).add_legend()  
#Distribution of Pressure, According to Months
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x240f730a148>
```

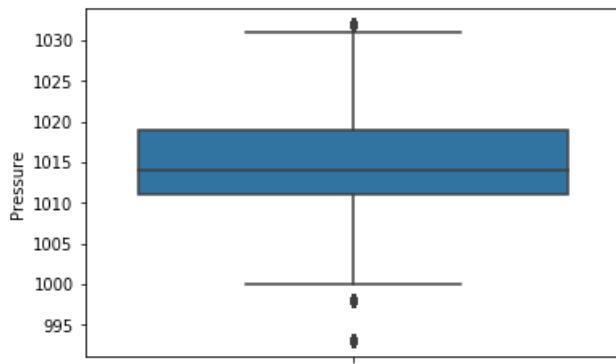


In [12]:

```
sns.boxplot(df["Pressure"],orient="v")  
# It seems that nearly equally distributed.
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f3f1cfcc8>
```



In [13]:

```
df["Pressure"].describe()
#Our boxplot shows that our data is distributed generally between 25%-75% range. There
# very less amount of values in the 0-25% range. There seem to
be no outliers in the
#boxplot. Our median or 50% value -1014- and mean value -1014-
seems nearly same.
```

Out[13]:

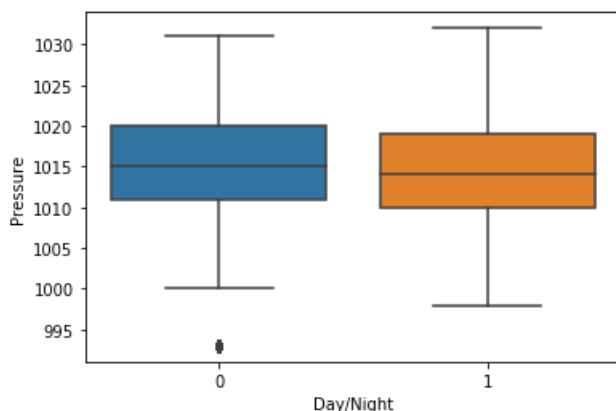
count	50530.000000
mean	1014.676727
std	6.112598
min	993.000000
25%	1011.000000
50%	1014.000000
75%	1019.000000
max	1032.000000
Name:	Pressure, dtype: float64

In [14]:

```
sns.boxplot(y=df["Pressure"], orient="v", x="Day/Night", data=df)
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f68a8a08>
```

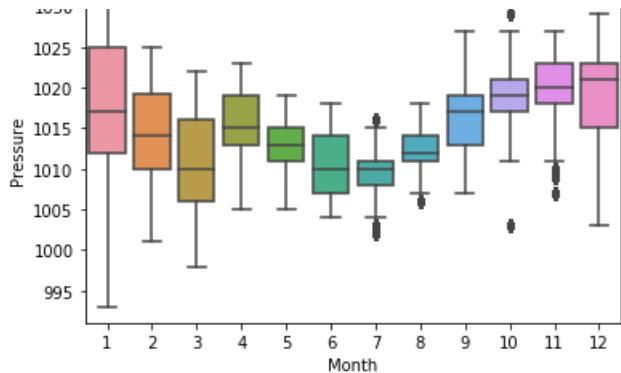


In [15]:

```
sns.boxplot(y=df["Pressure"], orient="v", x="Month", data=df)
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f6b12608>
```

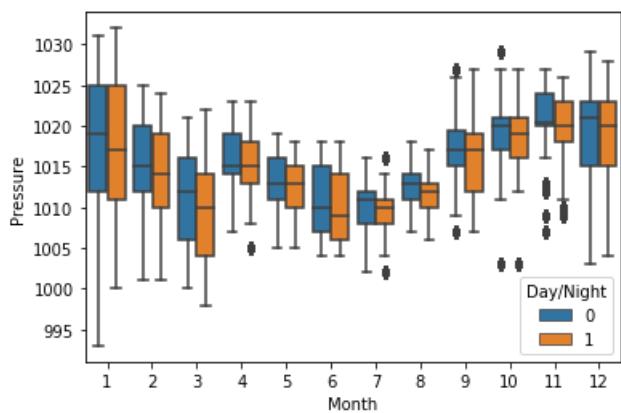


In [16]:

```
sns.boxplot(y=df["Pressure"], orient="v", x="Month", data=df, hue="Day/Night")
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f7871f88>
```

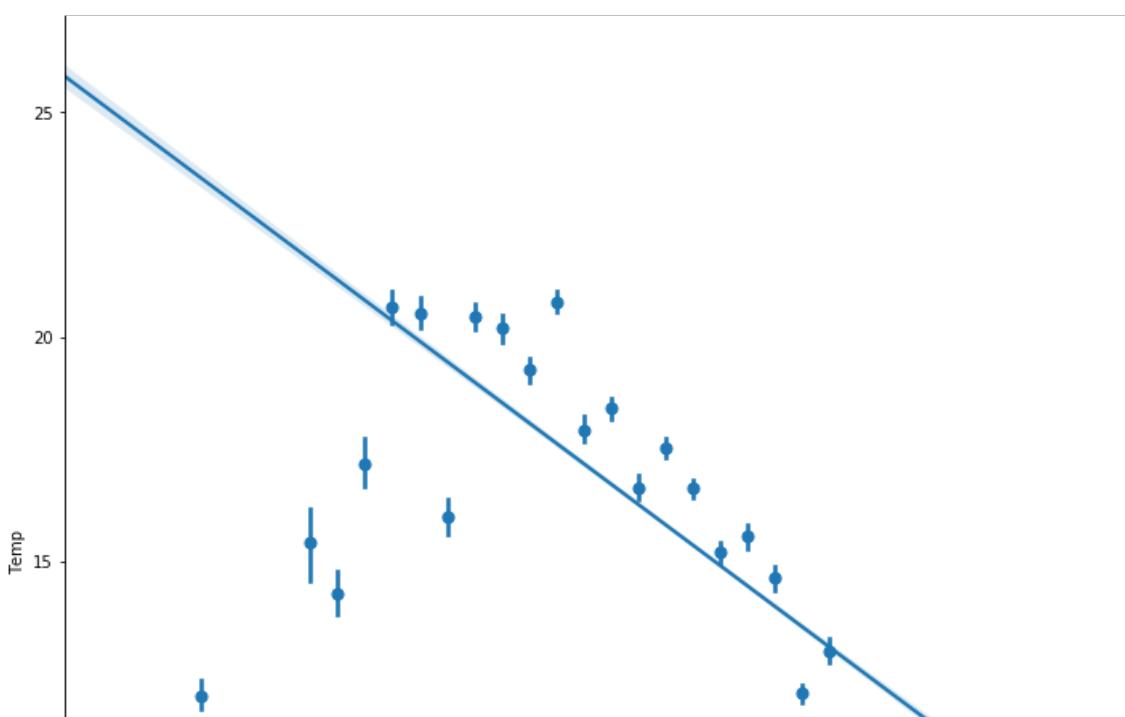


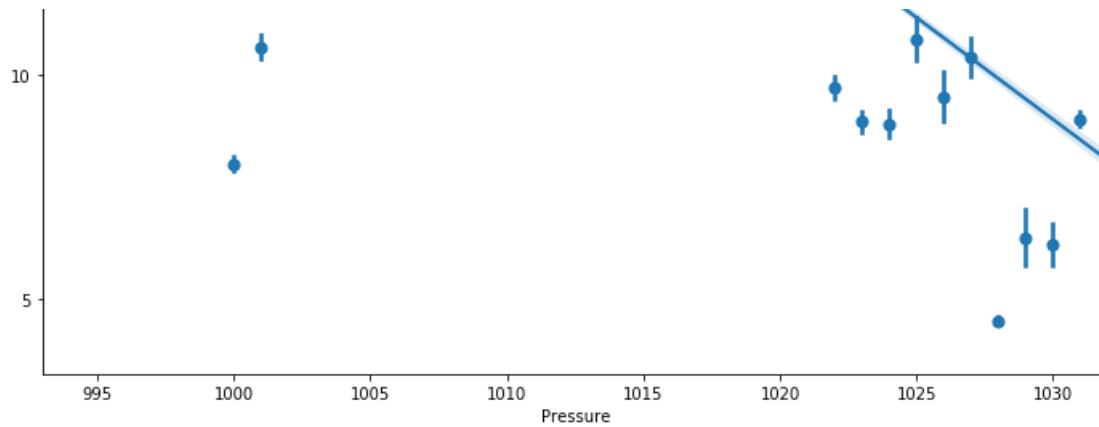
In [19]:

```
sns.lmplot(x="Pressure", y="Temp", data=df, height=10, x_bins=500)
```

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x240f6b30148>
```





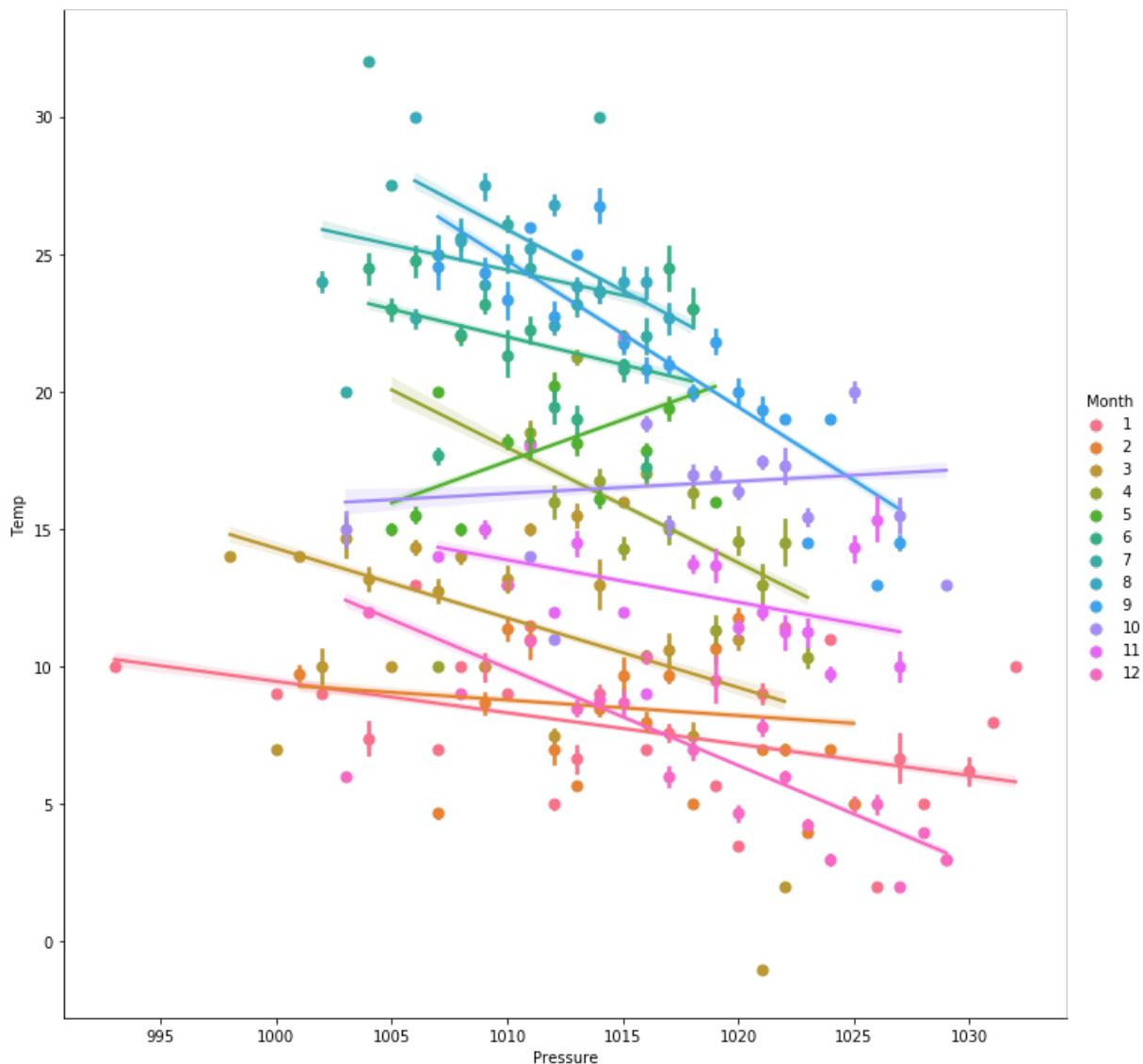
In [24]:

```
sns.lmplot(x="Pressure", y="Temp", data=df, height=10, x_bins=500, hue="Month")
```

#Seem to have a linear relationship for each month

Out [24]:

```
<seaborn.axisgrid.FacetGrid at 0x240f438ac48>
```

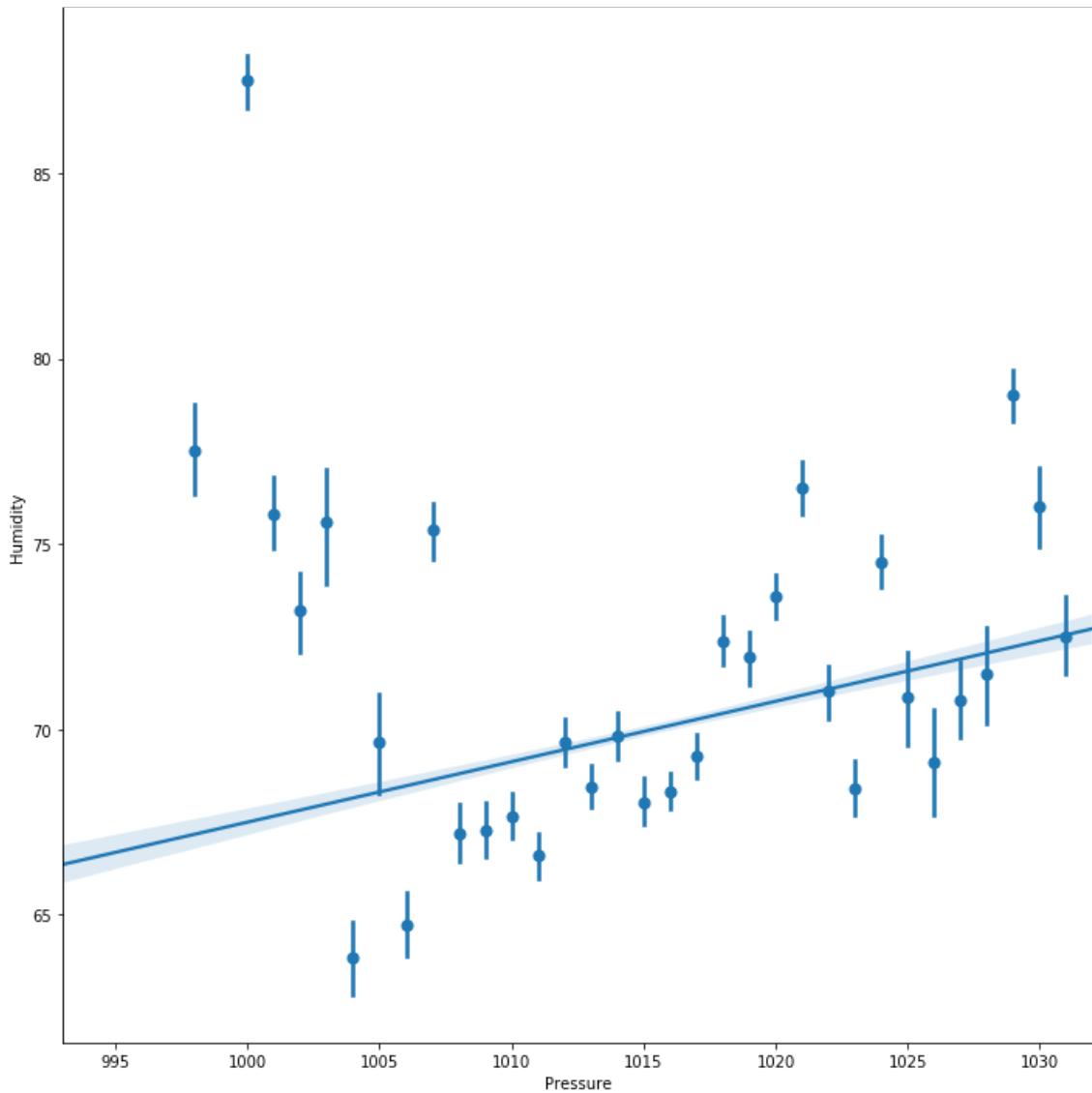


In [25]:

```
sns.lmplot(x="Pressure", y="Humidity", data=df, height=10, x_bins=500)
#Humidity and Pressure seem to have linear relationship
```

Out [25]:

```
<seaborn.axisgrid.FacetGrid at 0x240f4c84988>
```

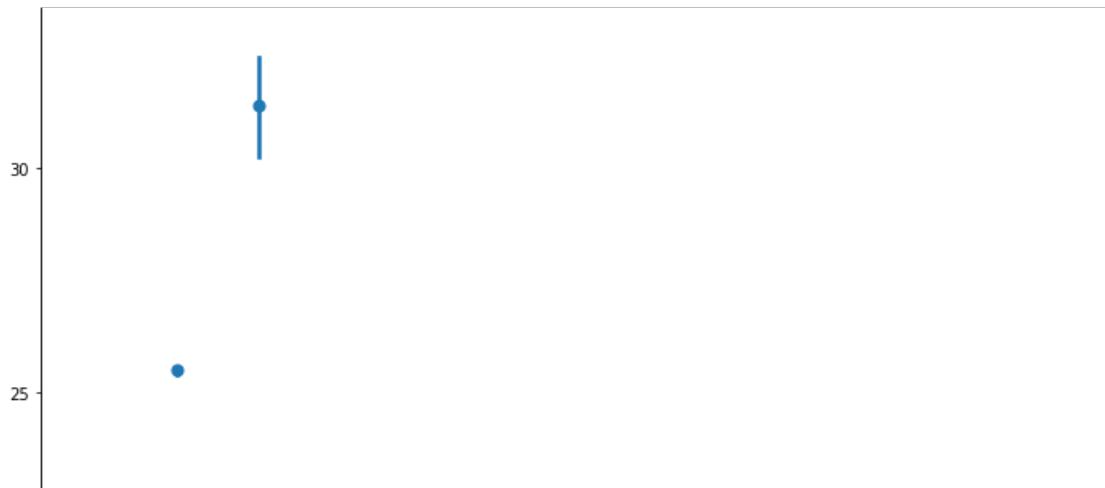


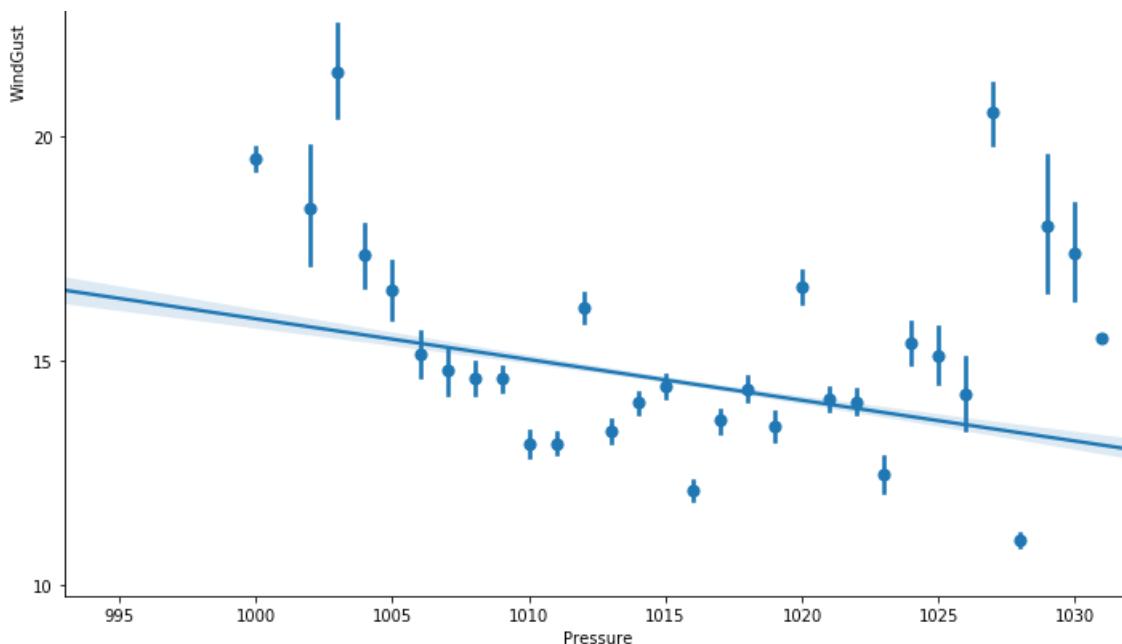
In [26]:

```
sns.lmplot(x="Pressure",y="WindGust",data=df,height=10,x_bins=500)
#Wind gust and Pressure seem to have a negative linear relationship
```

Out [26]:

```
<seaborn.axisgrid.FacetGrid at 0x240f5166788>
```



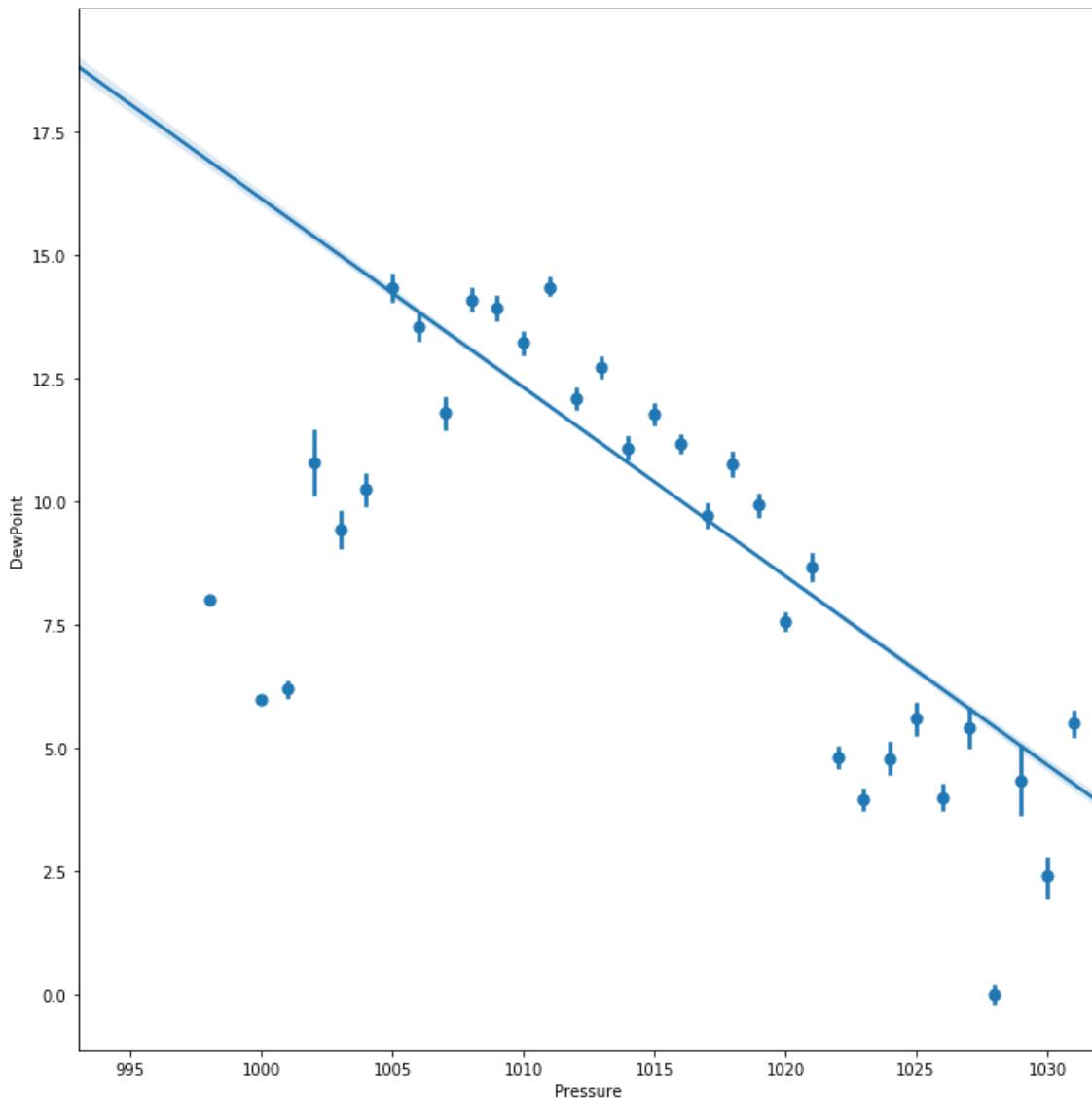


In [27]:

```
sns.lmplot(x="Pressure",y="DewPoint",data=df,height=10,x_bins=500)
#Dew point is directly dependent on pressure and humidity.
```

Out [27]:

```
<seaborn.axisgrid.FacetGrid at 0x240f52ca248>
```

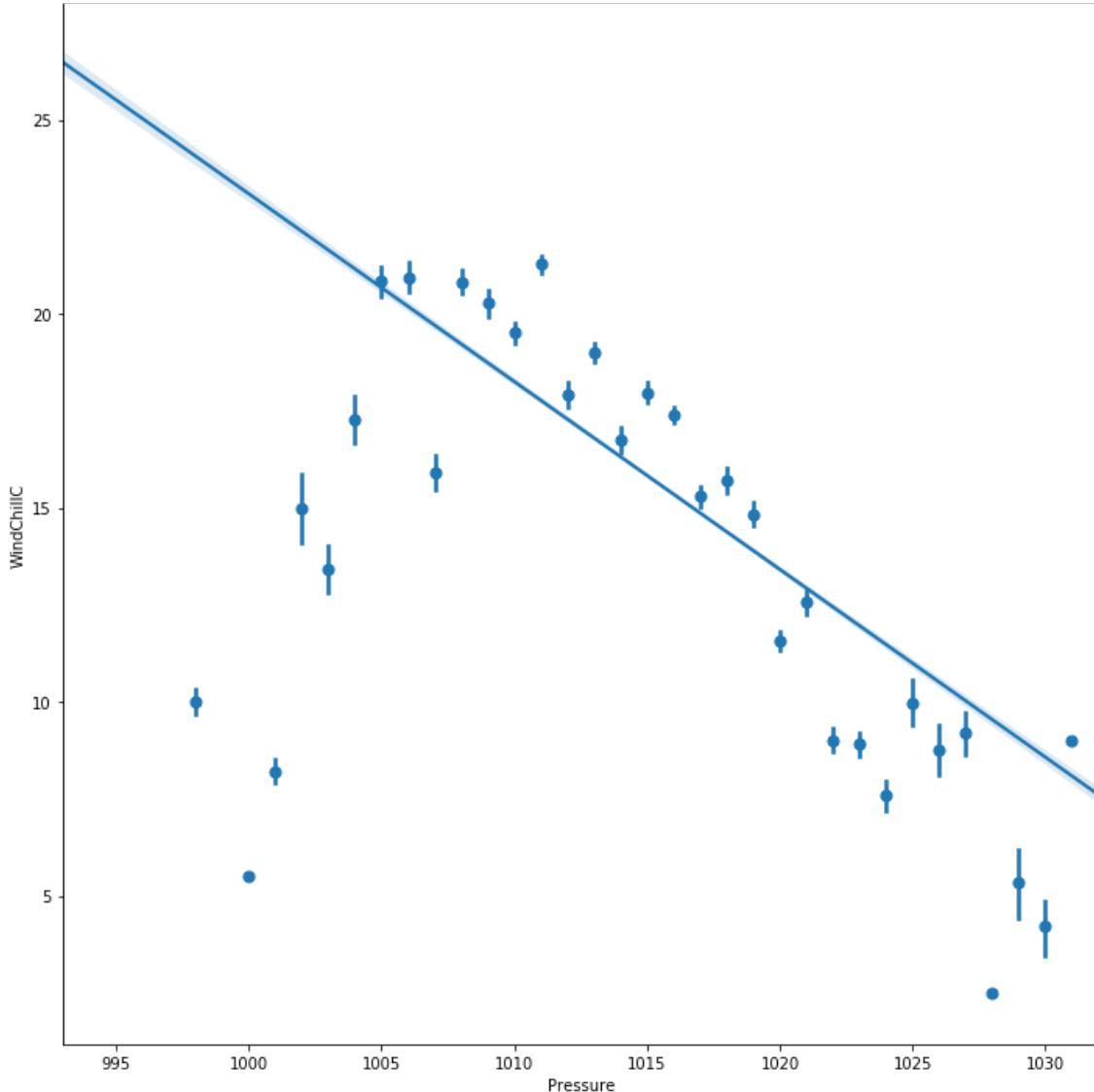


In [28]:

```
sns.lmplot(x="Pressure",y="WindChillC",data=df,height=10,x_bins=500)
#Pressure and Wind Chill seem to have a negative linear relationship
```

Out [28]:

```
<seaborn.axisgrid.FacetGrid at 0x240f54df148>
```



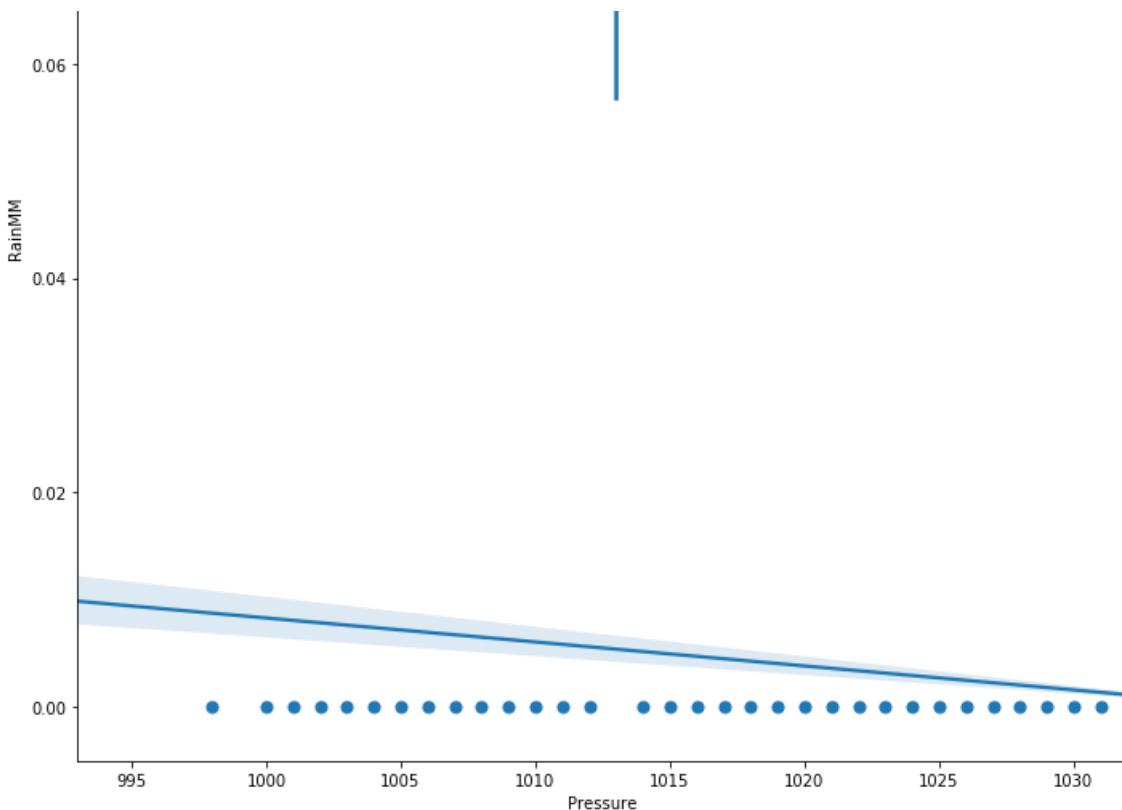
In [29]:

```
sns.lmplot(x="Pressure",y="RainMM",data=df,height=10,x_bins=500)
```

Out [29]:

```
<seaborn.axisgrid.FacetGrid at 0x240f561e888>
```



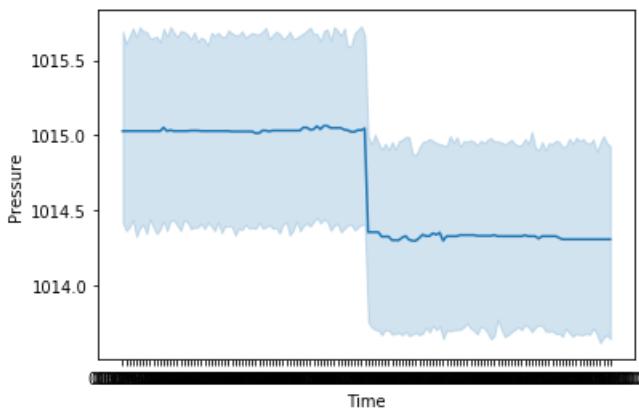


In [30]:

```
sns.lineplot(y=df["Pressure"],x=df["Time"].astype("str"),data=df) #Change of the Average Electric Production  
# According to time s.
```

Out[30]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f65d7708>
```

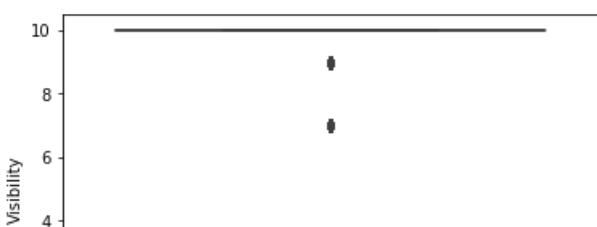


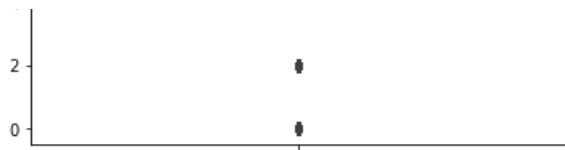
In [37]:

```
sns.boxplot(df["Visibility"],orient="v")
```

Out[37]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f6016748>
```





In [39]:

```
df["Visibility"].describe()
```

Out[39]:

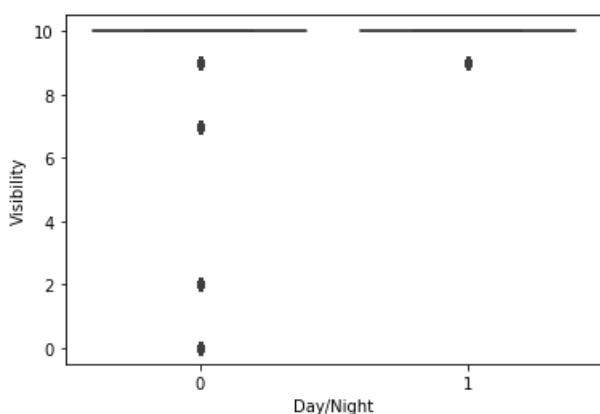
```
count    50530.000000
mean      9.860400
std       0.987456
min      0.000000
25%     10.000000
50%     10.000000
75%     10.000000
max     10.000000
Name: Visibility, dtype: float64
```

In [41]:

```
sns.boxplot(y=df["Visibility"], orient="v", x="Day/Night", data=df)
```

Out[41]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f7bd4b08>
```

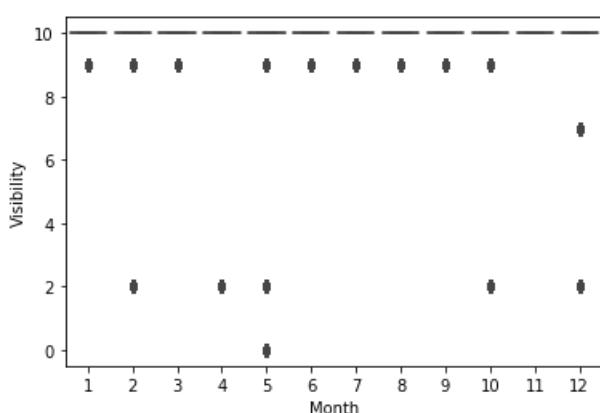


In [43]:

```
sns.boxplot(y=df["Visibility"], orient="v", x="Month", data=df)
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f7df3608>
```

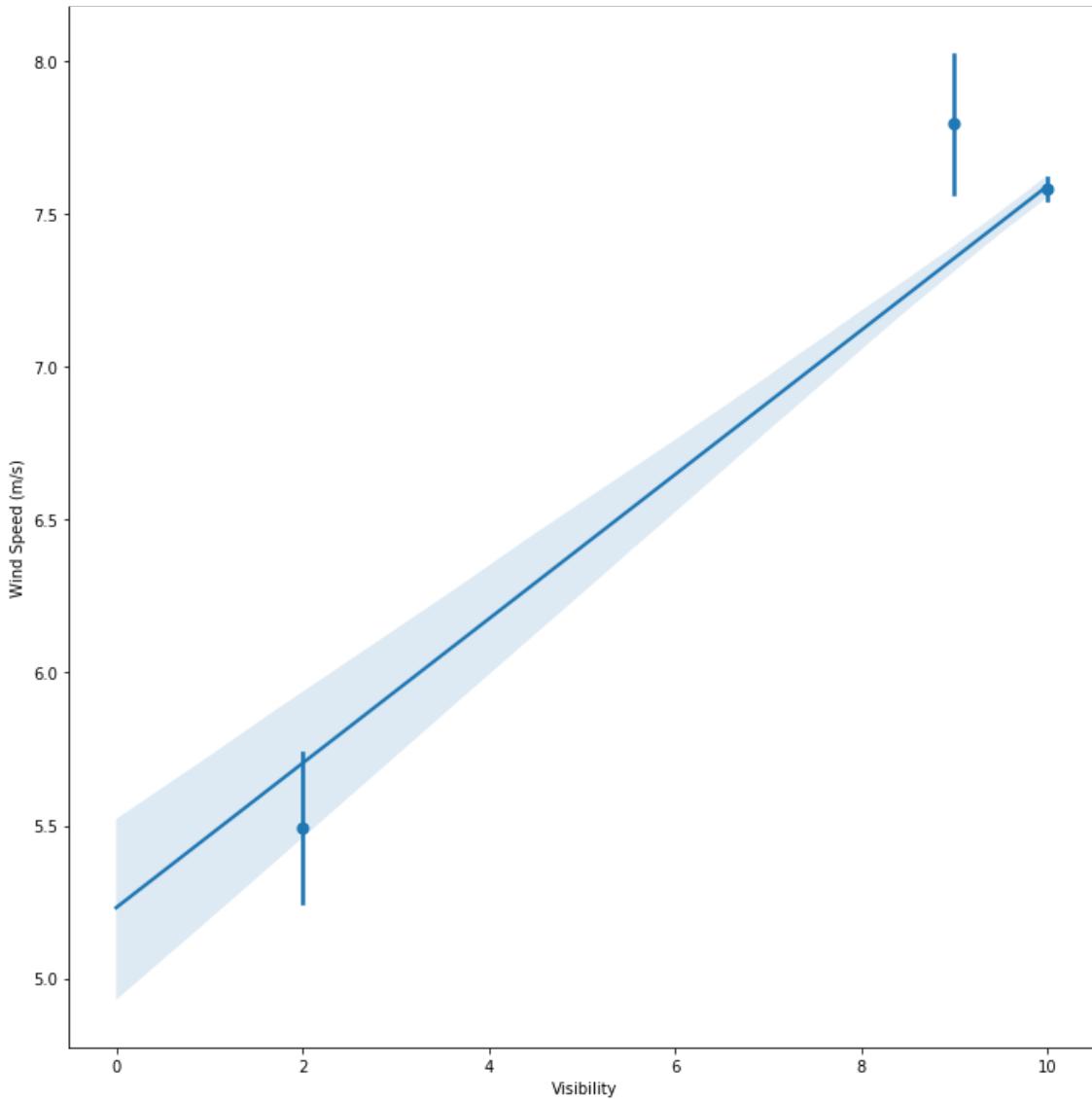


In [44]:

```
sns.lmplot(x="Visibility",y="Wind Speed (m/s)",data=df,height=10,x_bins=500)
```

Out[44]:

```
<seaborn.axisgrid.FacetGrid at 0x240f6d6dd48>
```

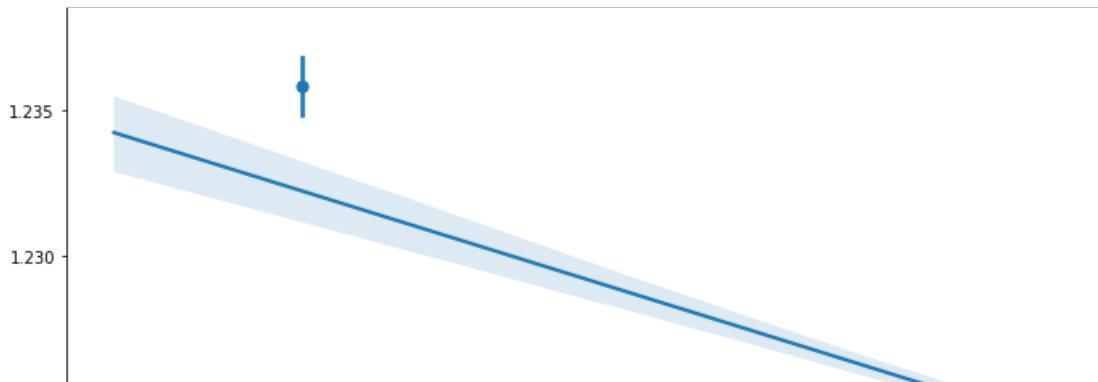


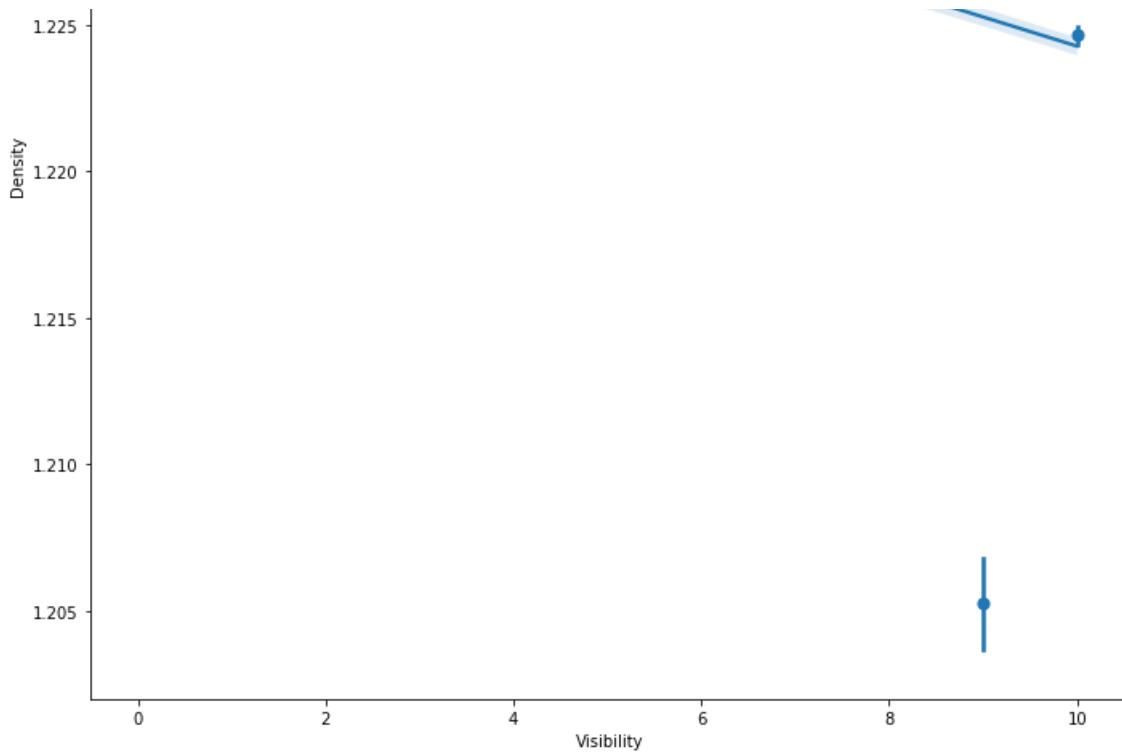
In [48]:

```
sns.lmplot(x="Visibility",y="Density",data=df,height=10,x_bins=500)
# Density and Visibility seem to have a negative linear relationship
```

Out[48]:

```
<seaborn.axisgrid.FacetGrid at 0x240f6c43948>
```



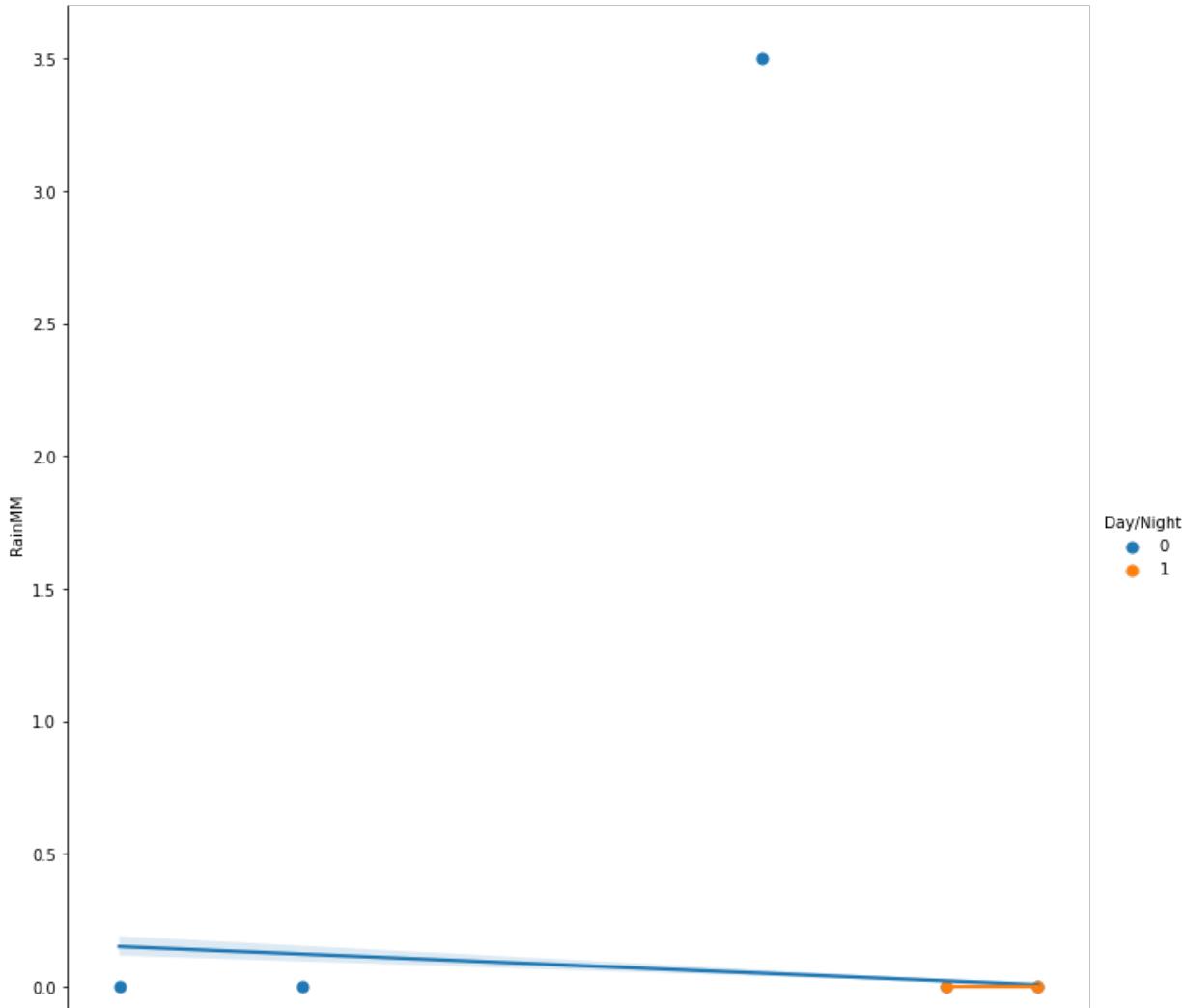


In [49]:

```
sns.lmplot(x="Visibility", y="RainMM", data=df, height=10, x_bins=500, hue="Day/Night")
```

Out [49]:

```
<seaborn.axisgrid.FacetGrid at 0x240f6bd00c8>
```



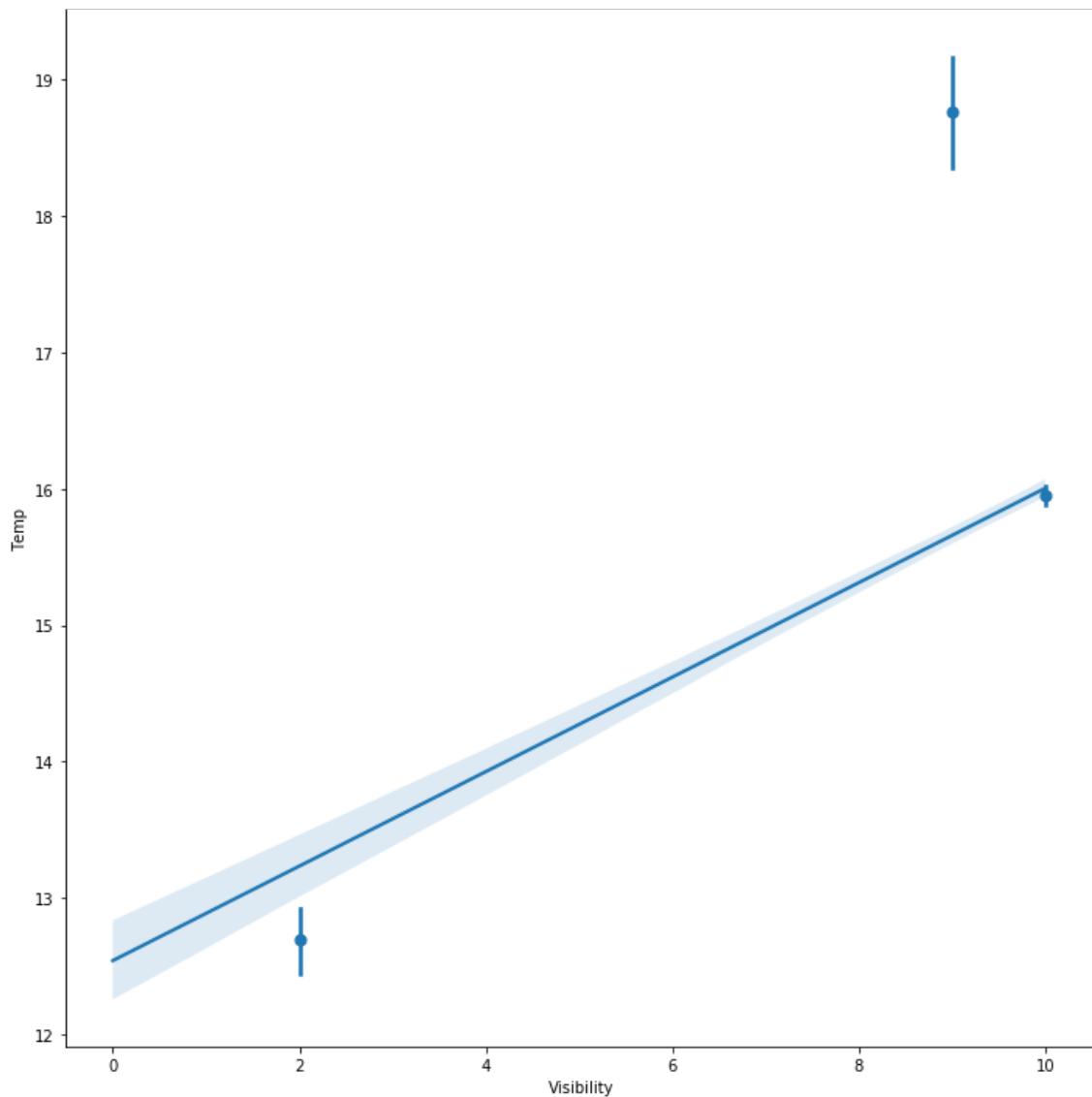


In [50]:

```
sns.lmplot(x="Visibility",y="Temp",data=df,height=10,x_bins=500)
# Temperature and Visibility seem to have a linear relationship
```

Out[50]:

```
<seaborn.axisgrid.FacetGrid at 0x240f5fa5dc8>
```

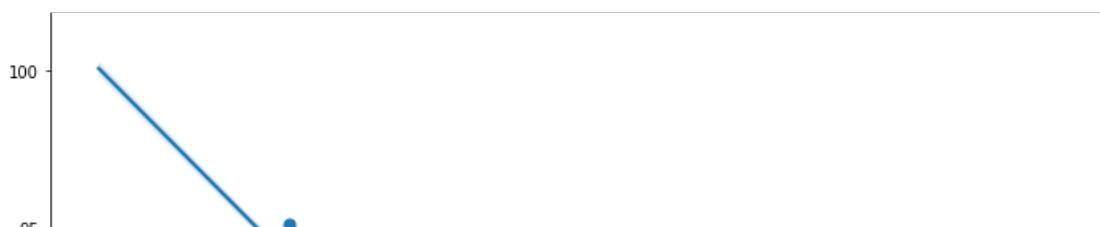


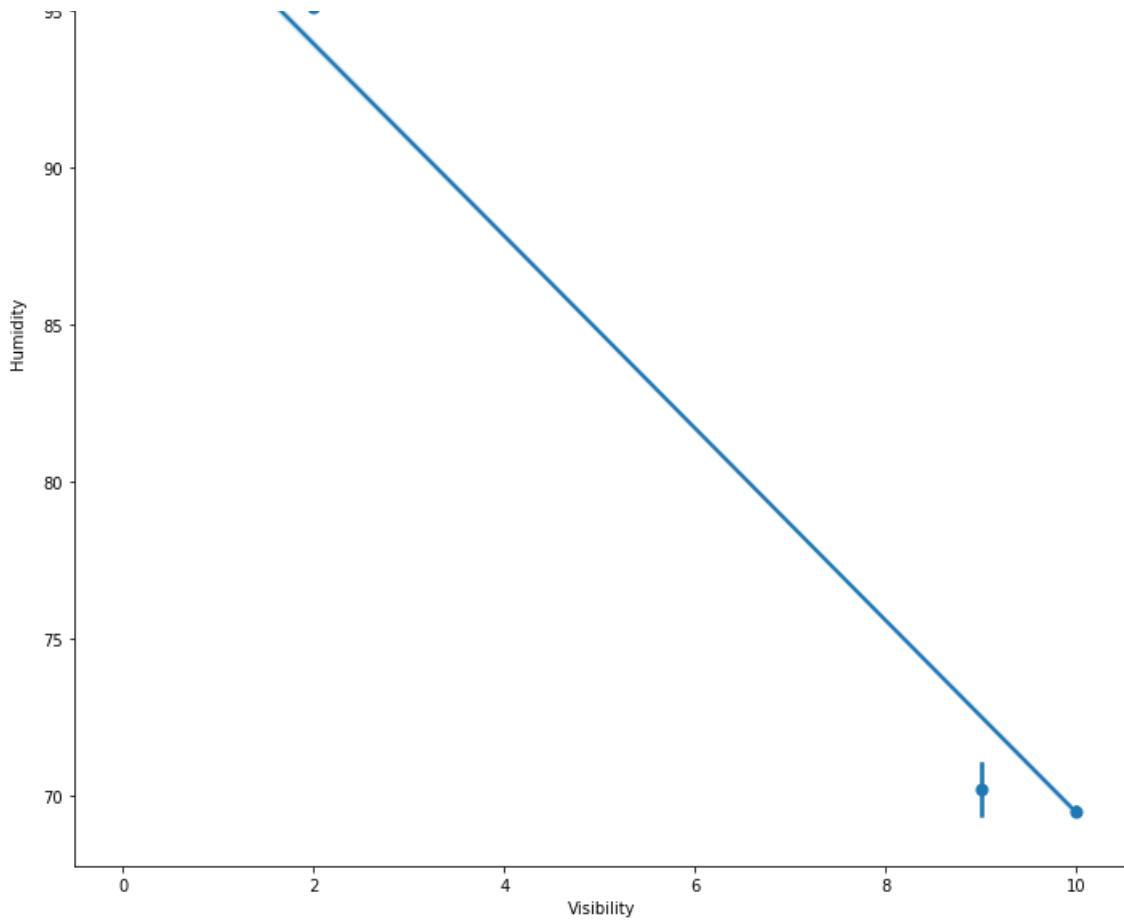
In [51]:

```
sns.lmplot(x="Visibility",y="Humidity",data=df,height=10,x_bins=500)
#Humidity and Visibility seem to have negative linear relationship
```

Out[51]:

```
<seaborn.axisgrid.FacetGrid at 0x240f816ddc8>
```



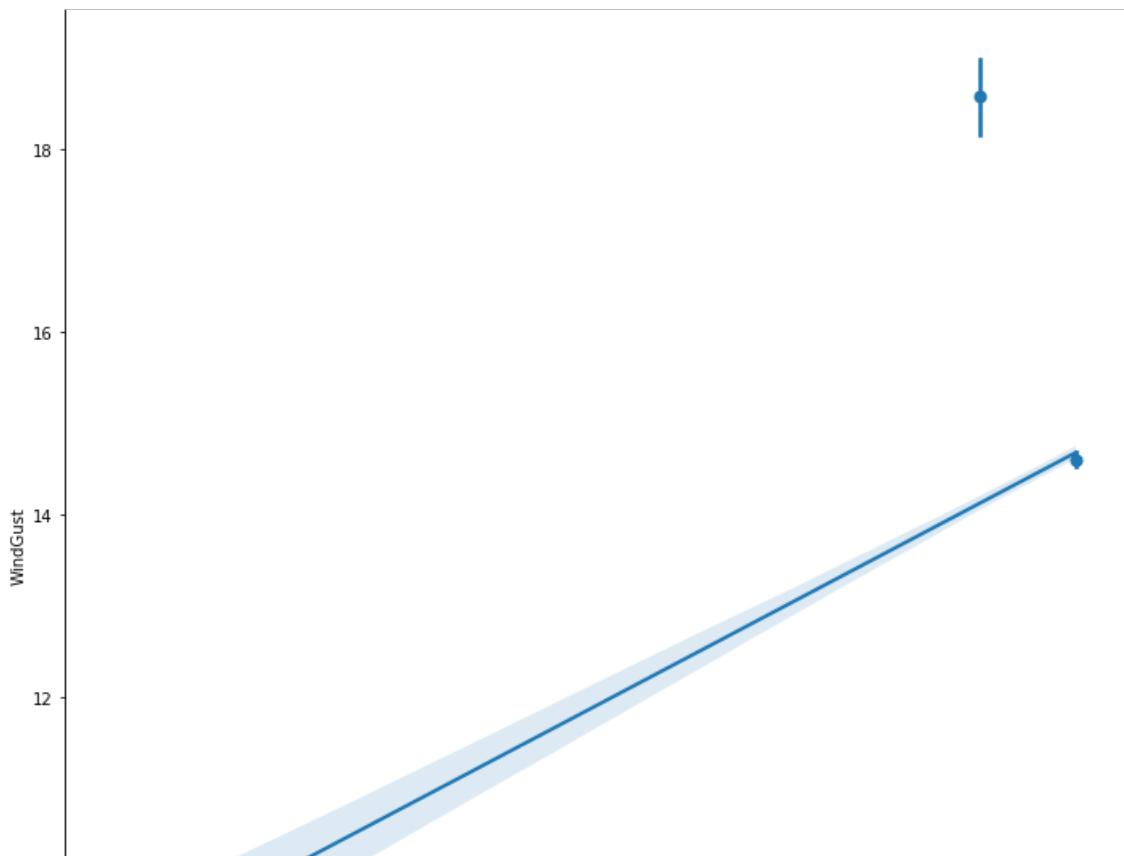


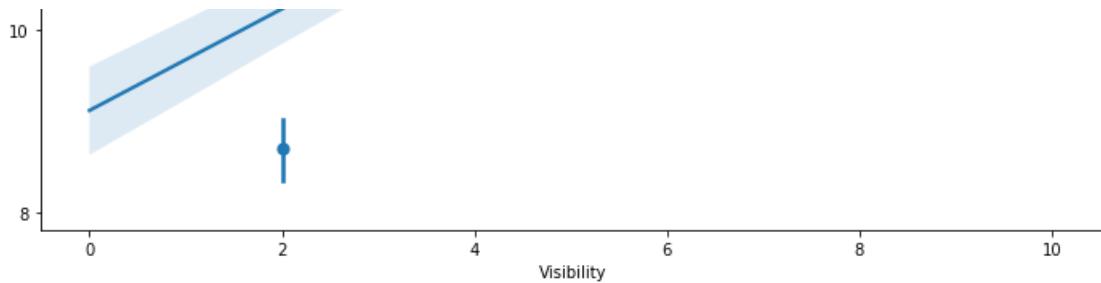
In [52]:

```
sns.lmplot(x="Visibility",y="WindGust",data=df,height=10,x_bins=500)
#Wind gust and visibility seem to have a linear relationship
```

Out [52]:

```
<seaborn.axisgrid.FacetGrid at 0x240f85ce448>
```



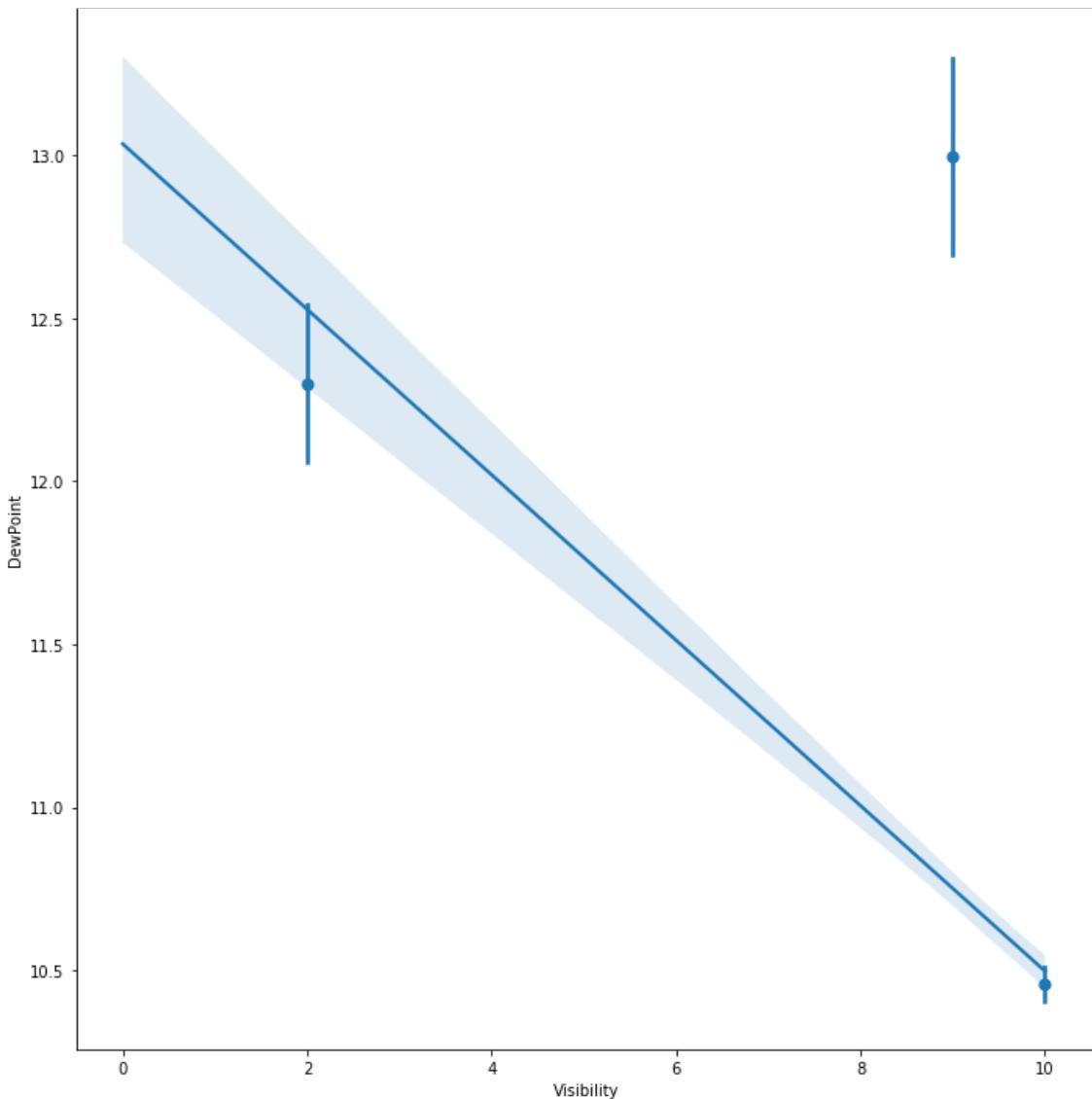


In [53]:

```
sns.lmplot(x="Visibility",y="DewPoint",data=df,height=10,x_bins=500)  
#Dew point and Visibility seem to have a negative linear relationship
```

Out [53]:

```
<seaborn.axisgrid.FacetGrid at 0x240f8690c48>
```

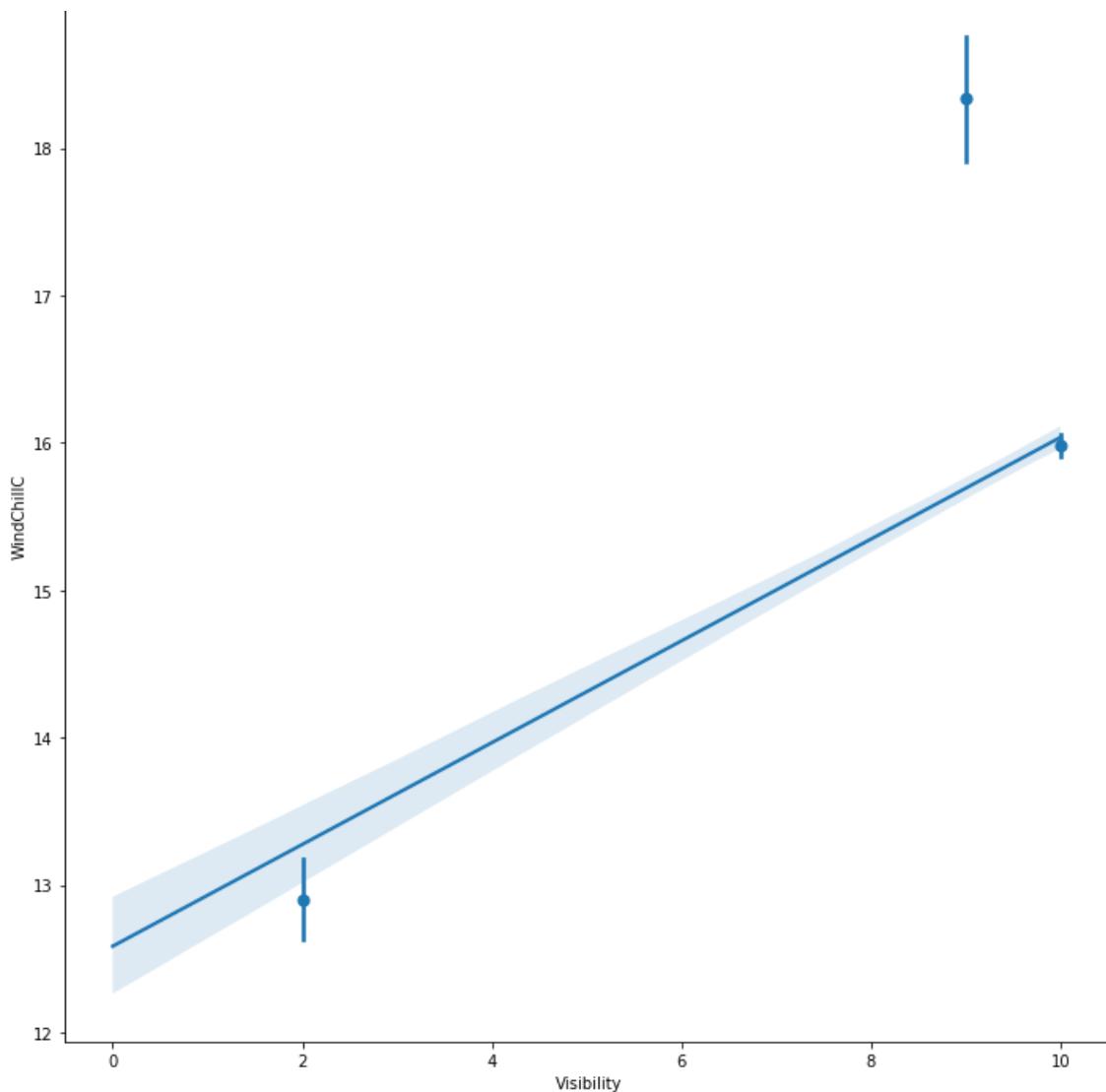


In [54]:

```
sns.lmplot(x="Visibility",y="WindChillC",data=df,height=10,x_bins=500)  
Visibility and Wind Chill seem to have a weak negative linear relationship
```

Out [54]:

```
<seaborn.axisgrid.FacetGrid at 0x240f8781908>
```

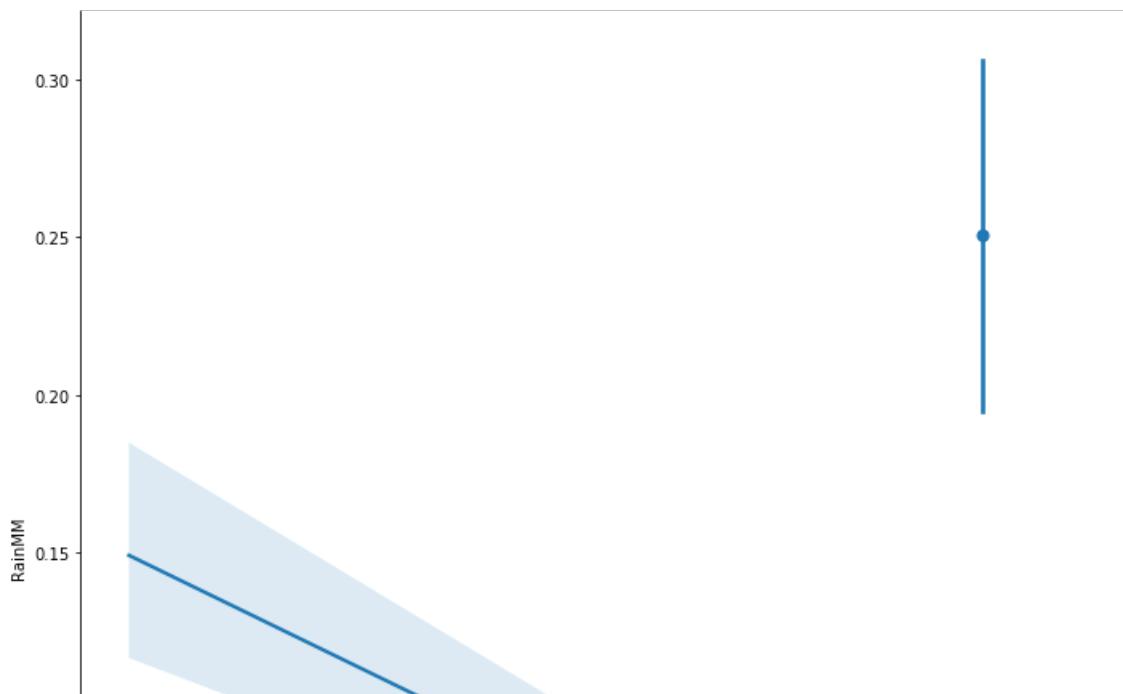


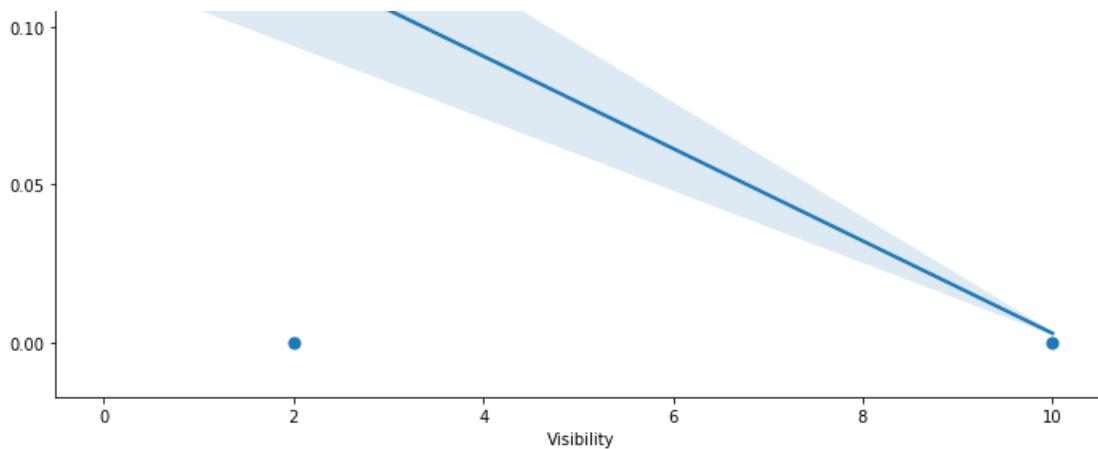
In [55]:

```
sns.lmplot(x="Visibility",y="WindChillC",data=df,height=10,x_bins=500)
```

Out [55]:

```
<seaborn.axisgrid.FacetGrid at 0x240f8898448>
```



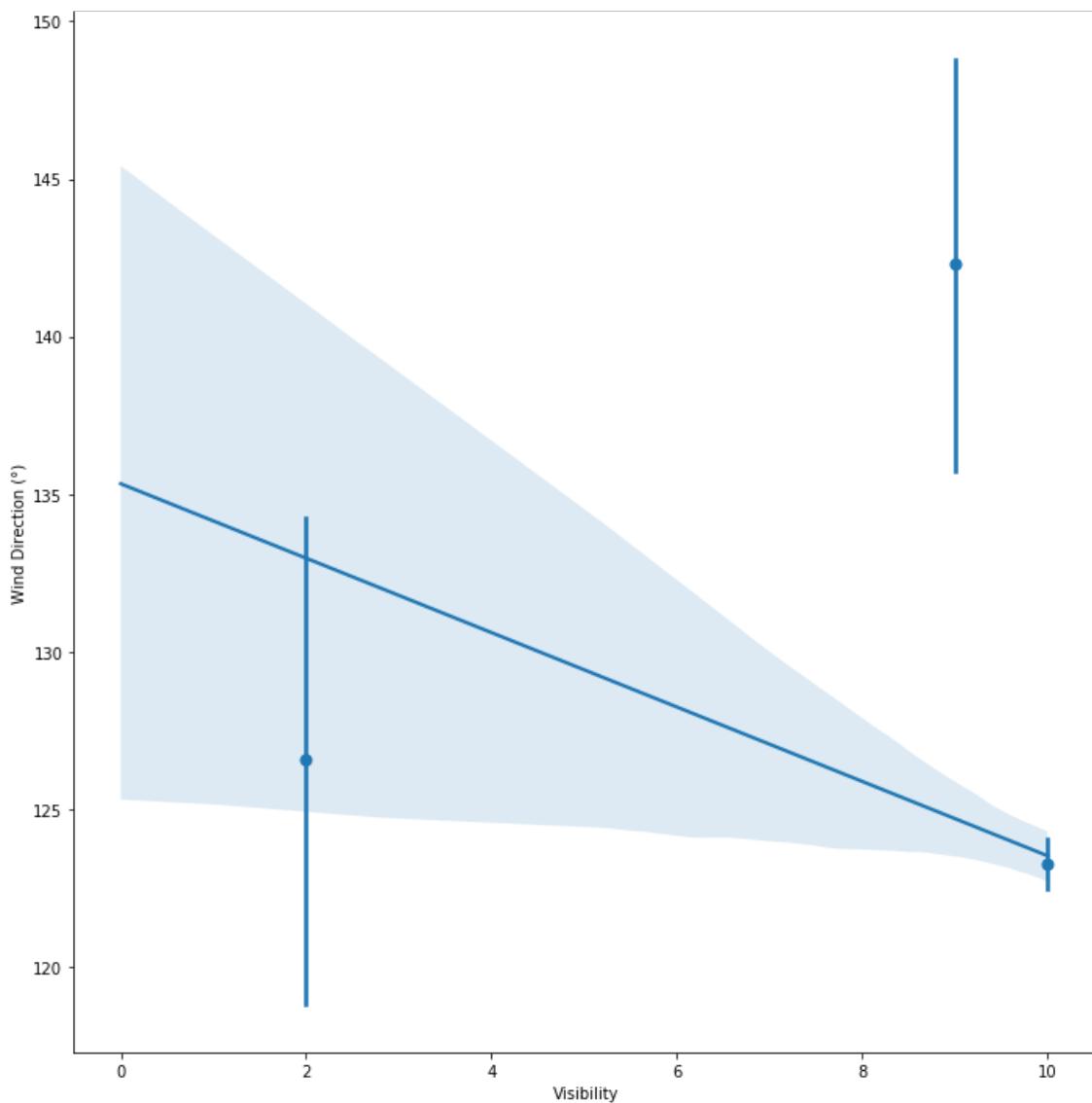


In [56]:

```
sns.lmplot(x="Visibility",y="Wind Direction ( °)",data=df,height=10,x_bins=500)
```

Out[56]:

```
<seaborn.axisgrid.FacetGrid at 0x240f8955fc8>
```

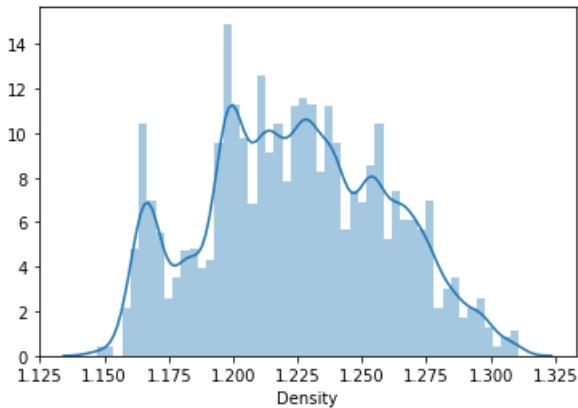


In [57]:

```
sns.distplot(df["Density"])
```

Out[57]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f8a11988>
```

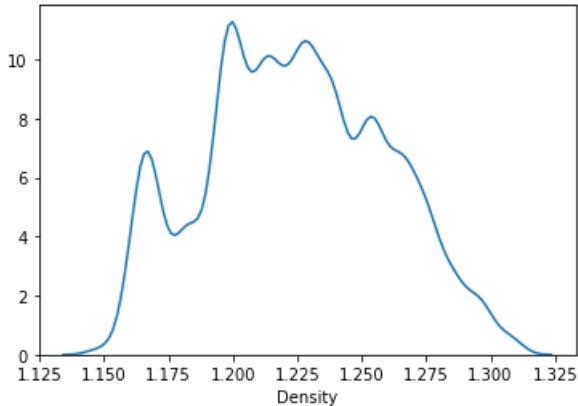


```
In [58]:
```

```
sns.distplot(df["Density"], hist=False)
```

```
Out[58]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f8b954c8>
```

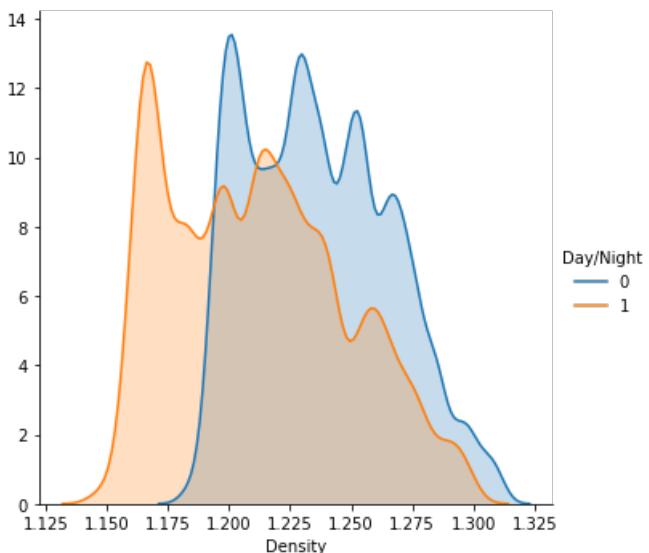


```
In [59]:
```

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Density",shade=True).add_legend()  
#Distribution of Density value by Day-Night
```

```
Out[59]:
```

```
<seaborn.axisgrid.FacetGrid at 0x240f8c8bc88>
```

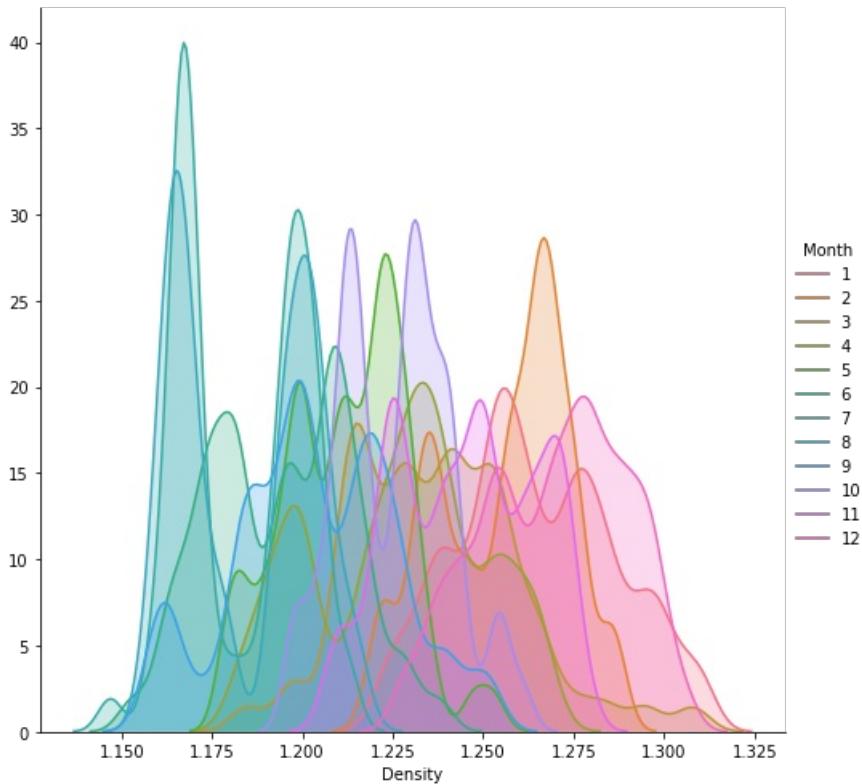


In [60]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"Density",shade=True).add_legend()
```

Out[60]:

```
<seaborn.axisgrid.FacetGrid at 0x240f8cdcec8>
```

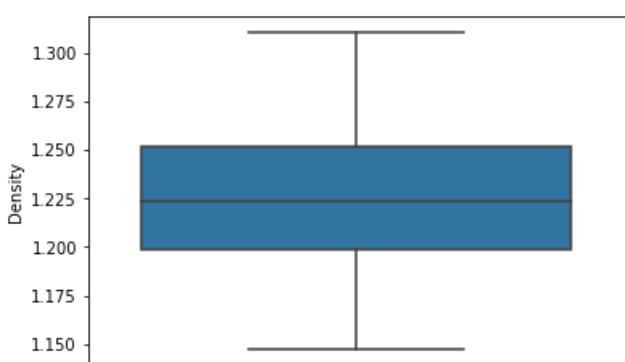


In [61]:

```
sns.boxplot(df["Density"],orient="v")
```

Out[61]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f565b9c8>
```



In [62]:

```
df["Density"].describe() #Our boxplot shows that our data is distributed generally between 25%-75%  
range. There  
be no outliers in the  
- seems same.  
# very less amount of values in the 0-25% range. There seem to  
#boxplot. Our median or 50% value -1.224- and mean value -1.224-
```

```
Out[62]:
```

```
count      50530.000000
mean       1.224415
std        0.035097
min        1.146970
25%        1.198842
50%        1.223732
75%        1.251411
max        1.310768
Name: Density, dtype: float64
```

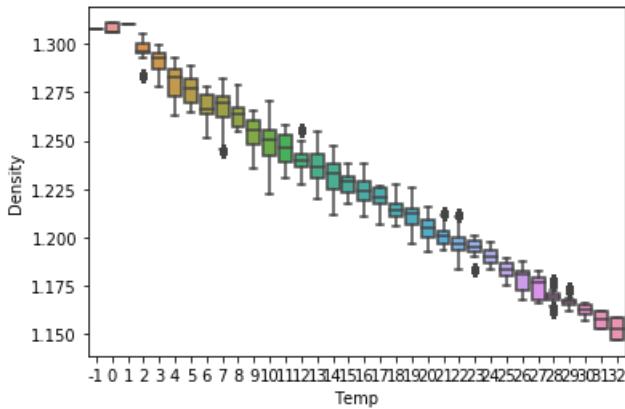
When we calculated the density we used temperature and pressure in formula. We can examine the relationship between them in ongoing graphs.

```
In [64]:
```

```
sns.boxplot(y=df["Density"], orient="v", x="Temp", data=df)
```

```
Out[64]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f64b0a48>
```

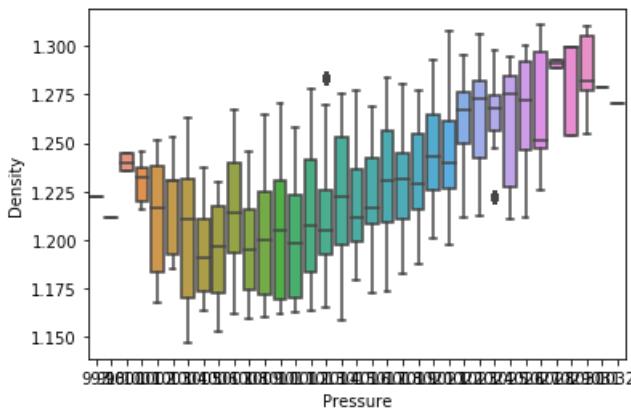


```
In [65]:
```

```
sns.boxplot(y=df["Density"], orient="v", x="Pressure", data=df)
```

```
Out[65]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x240f7e20b88>
```

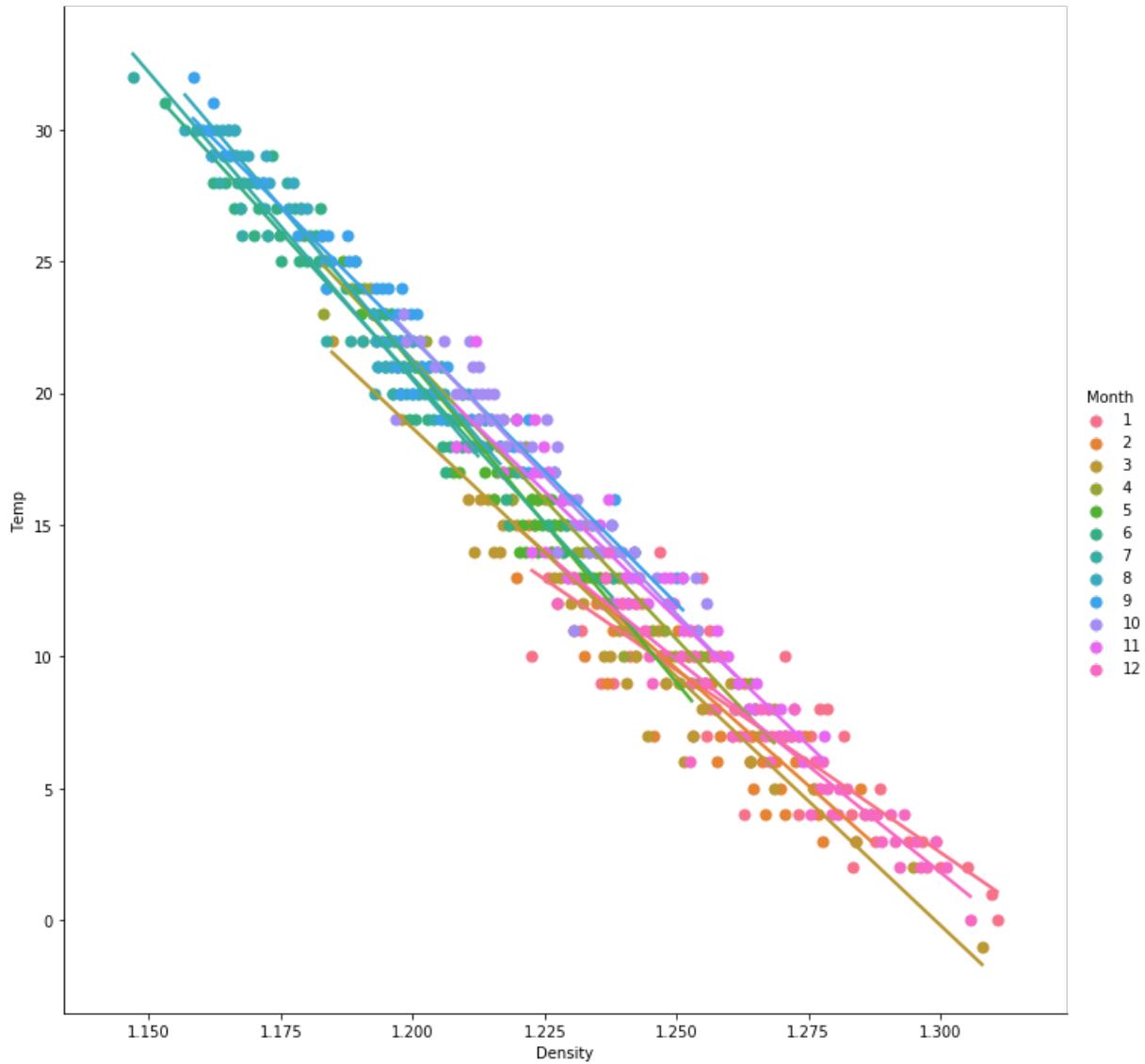


```
In [73]:
```

```
sns.lmplot(x="Density", y="Temp", data=df, height=10, x_bins=500, hue="Month")
```

```
Out[73]:
```

```
<seaborn.axisgrid.FacetGrid at 0x24090922388>
```



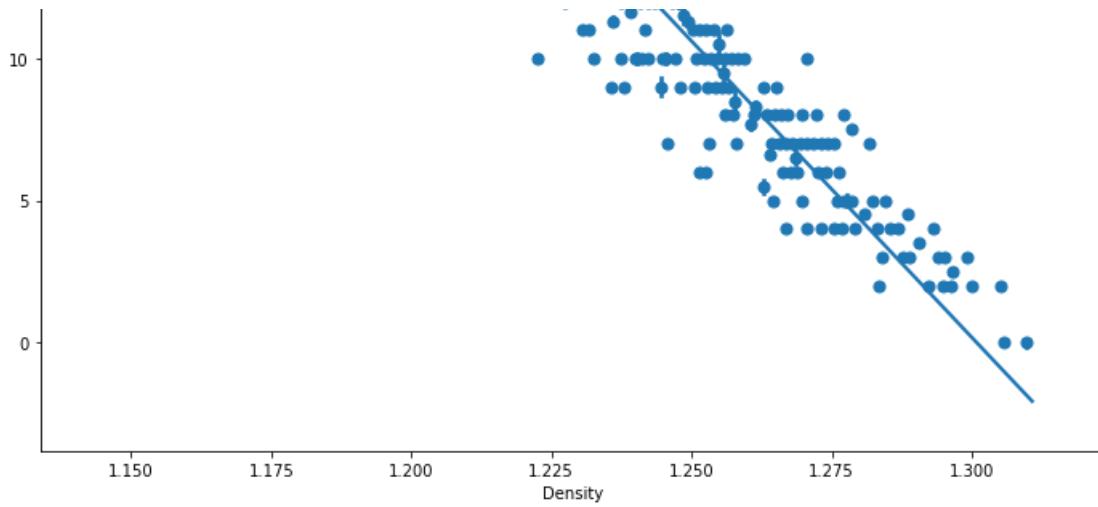
In [71]:

```
sns.lmplot(x="Density", y="Temp", data=df, height=10, x_bins=500)
```

Out[71]:

```
<seaborn.axisgrid.FacetGrid at 0x2408e7ac808>
```



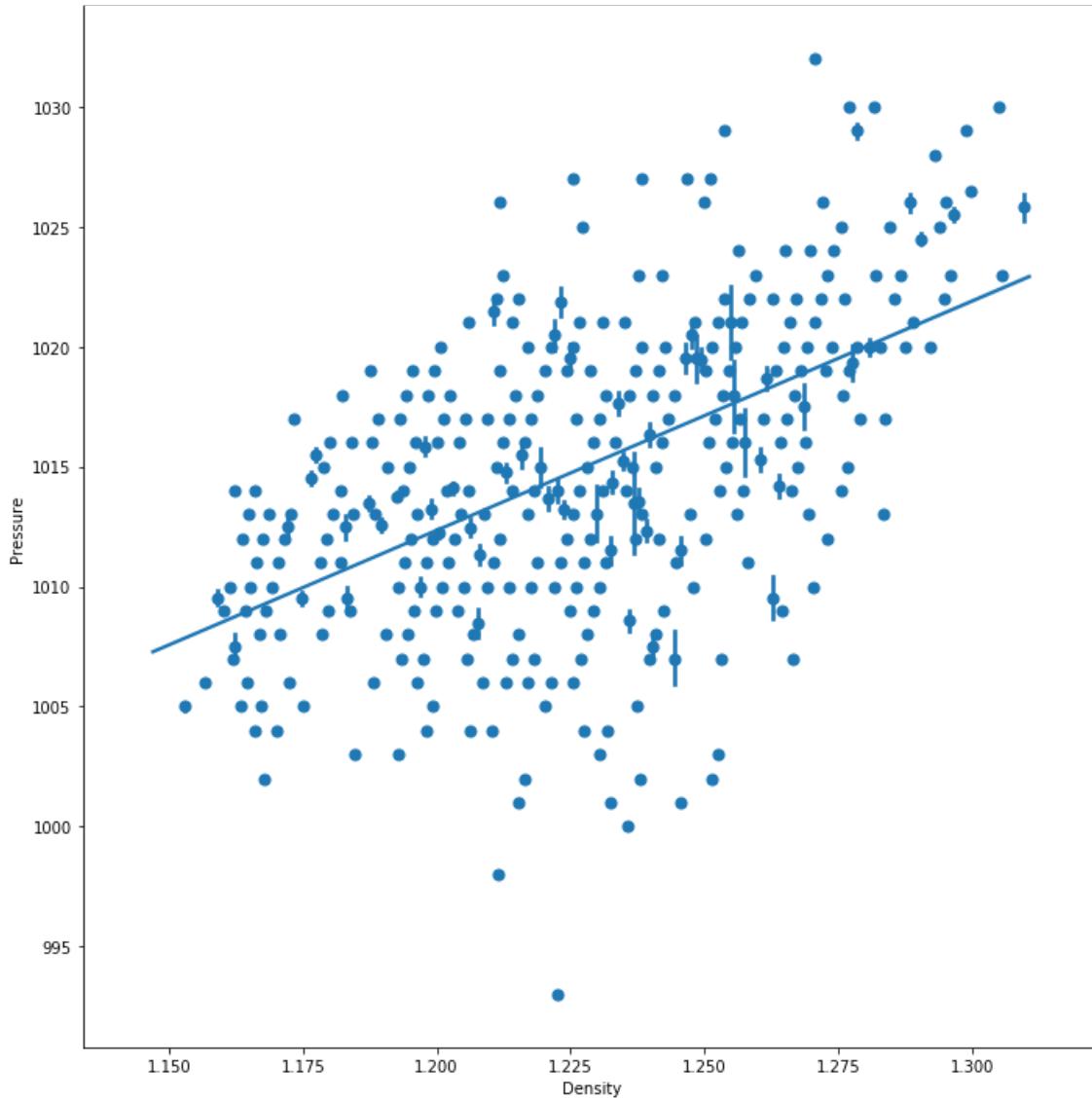


In [72]:

```
sns.lmplot(x="Density", y="Pressure", data=df, height=10, x_bins=500)
```

Out[72]:

```
<seaborn.axisgrid.FacetGrid at 0x2408f0bda08>
```

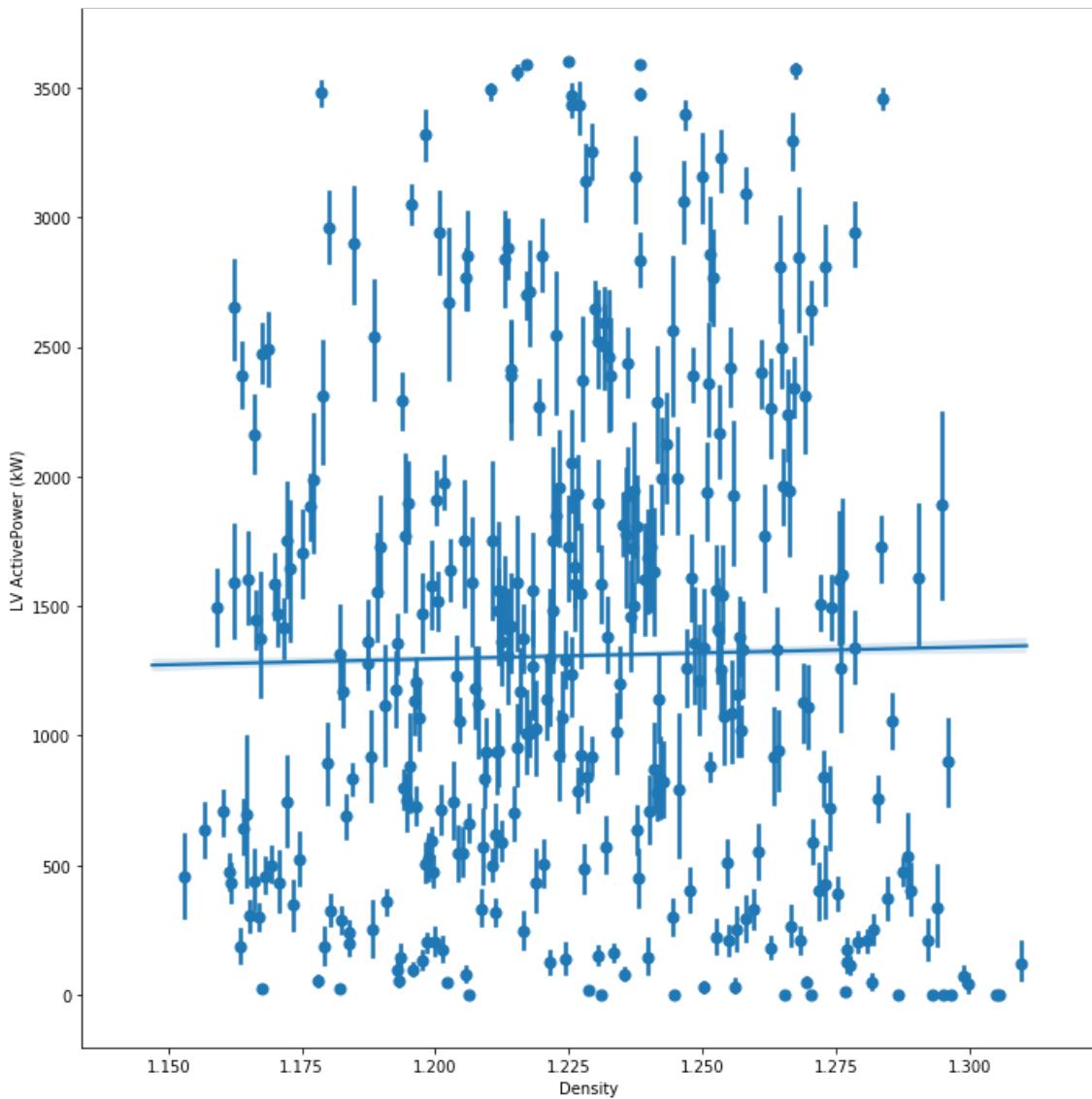


In [68]:

```
sns.lmplot(x="Density", y="LV ActivePower (kW)", data=df, height=10, x_bins=500)
```

Out [68]:

```
<seaborn.axisgrid.FacetGrid at 0x240f5763708>
```

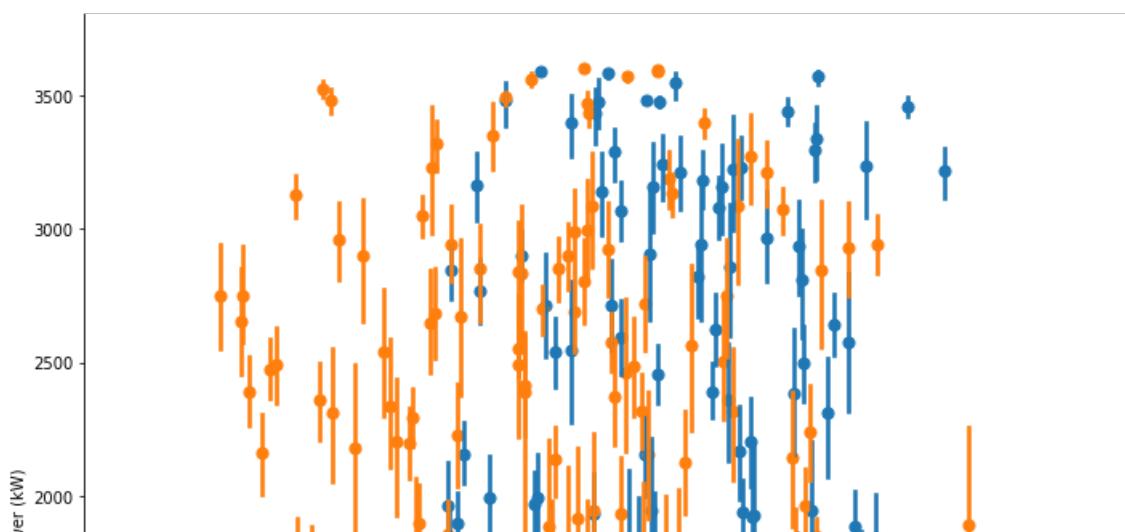


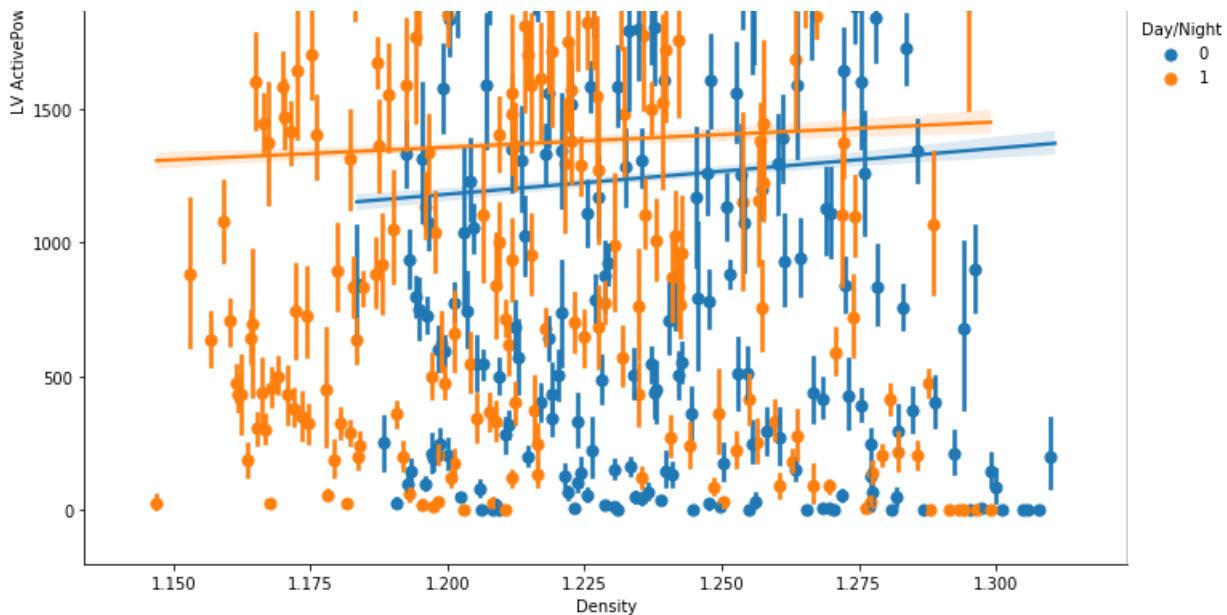
In [69]:

```
sns.lmplot(x="Density", y="LV ActivePower (kW)", data=df, height=10, x_bins=500, hue="Day/Night")
```

Out [69]:

```
<seaborn.axisgrid.FacetGrid at 0x2408d2cec88>
```



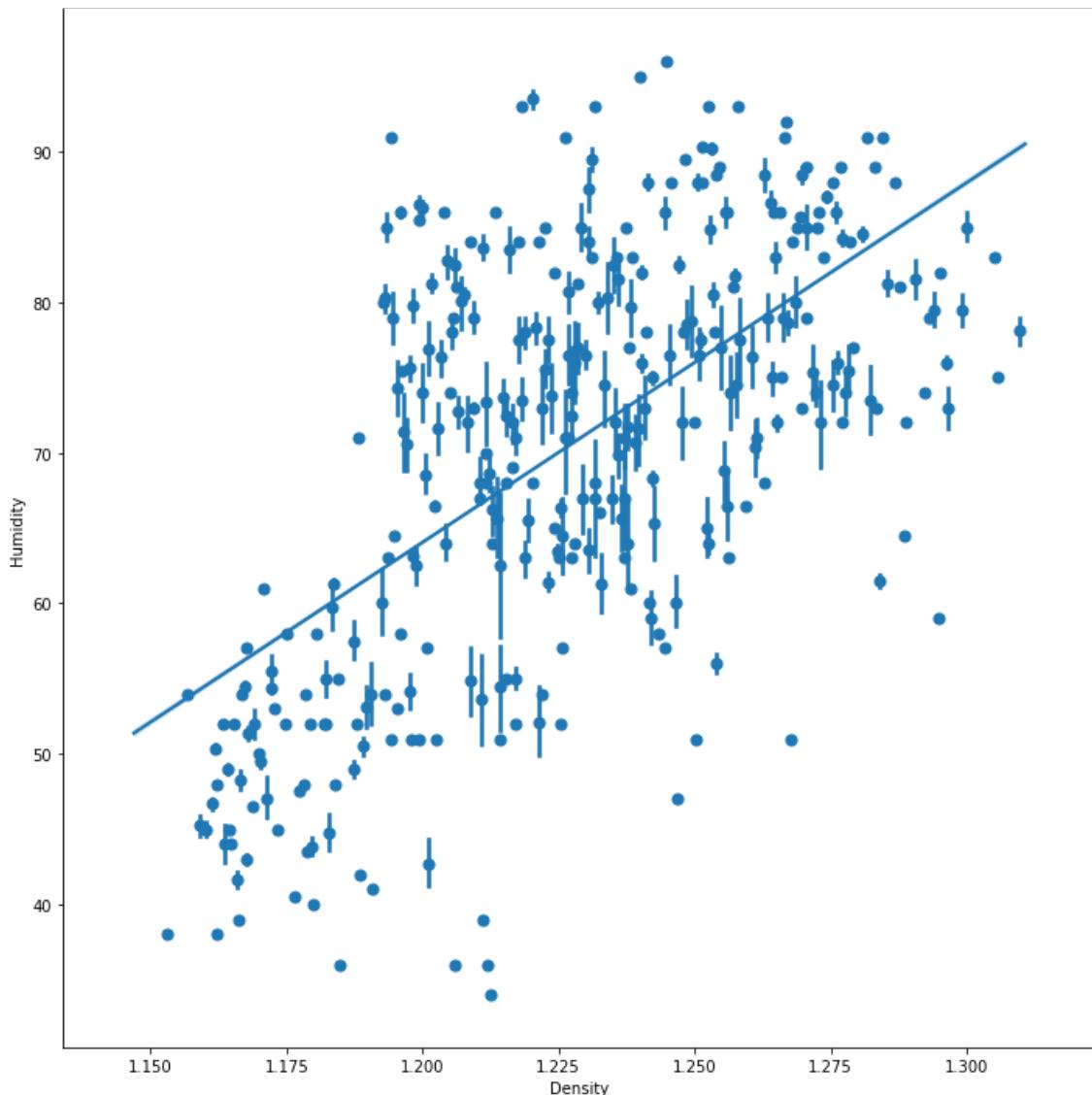


In [74]:

```
sns.lmplot(x="Density", y="Humidity", data=df, height=10, x_bins=500)
```

Out[74]:

```
<seaborn.axisgrid.FacetGrid at 0x2409091c748>
```

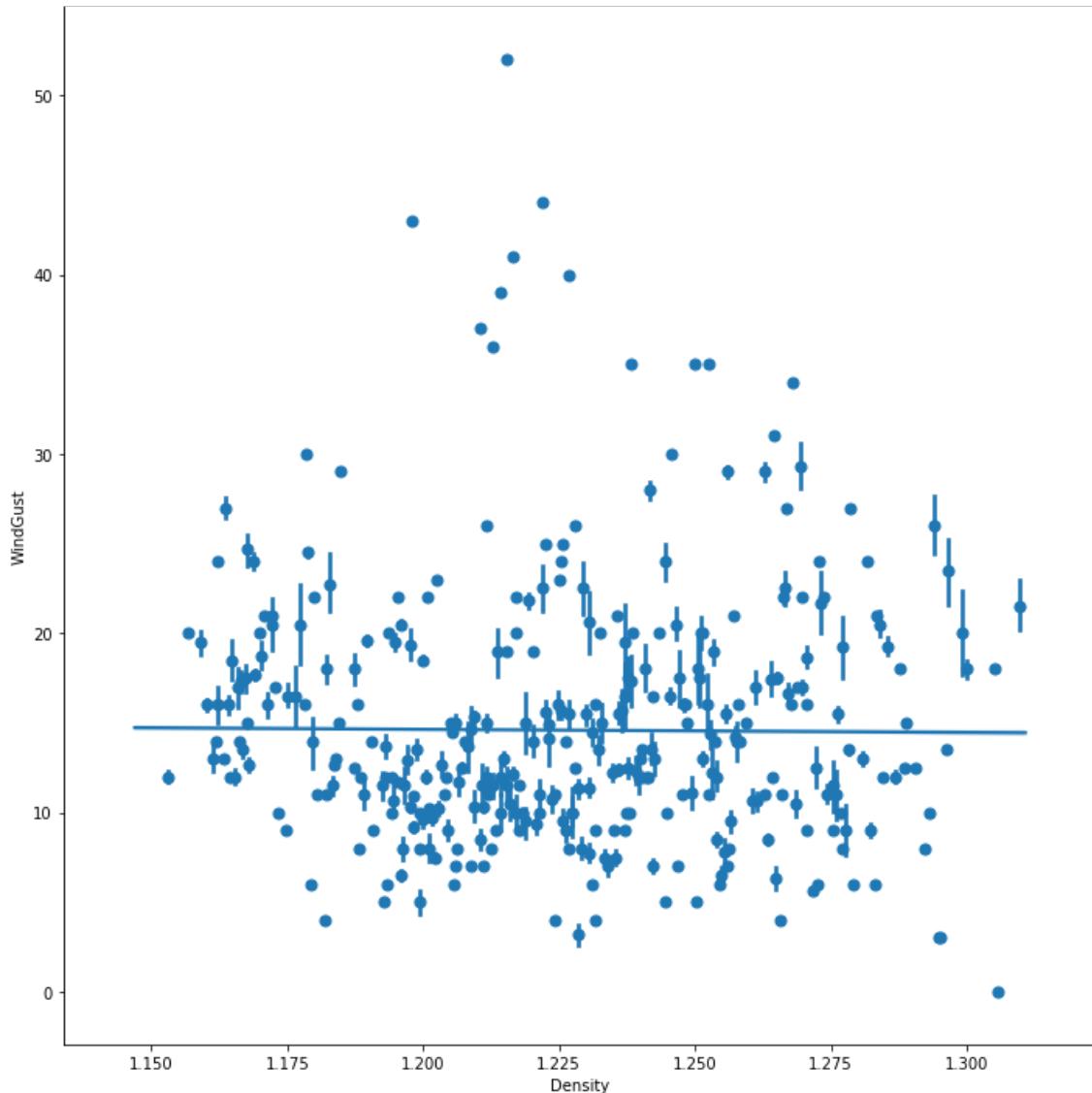


In [76]:

```
sns.lmplot(x="Density",y="WindGust",data=df,height=10,x_bins=500)
```

Out[76]:

```
<seaborn.axisgrid.FacetGrid at 0x24091e25808>
```



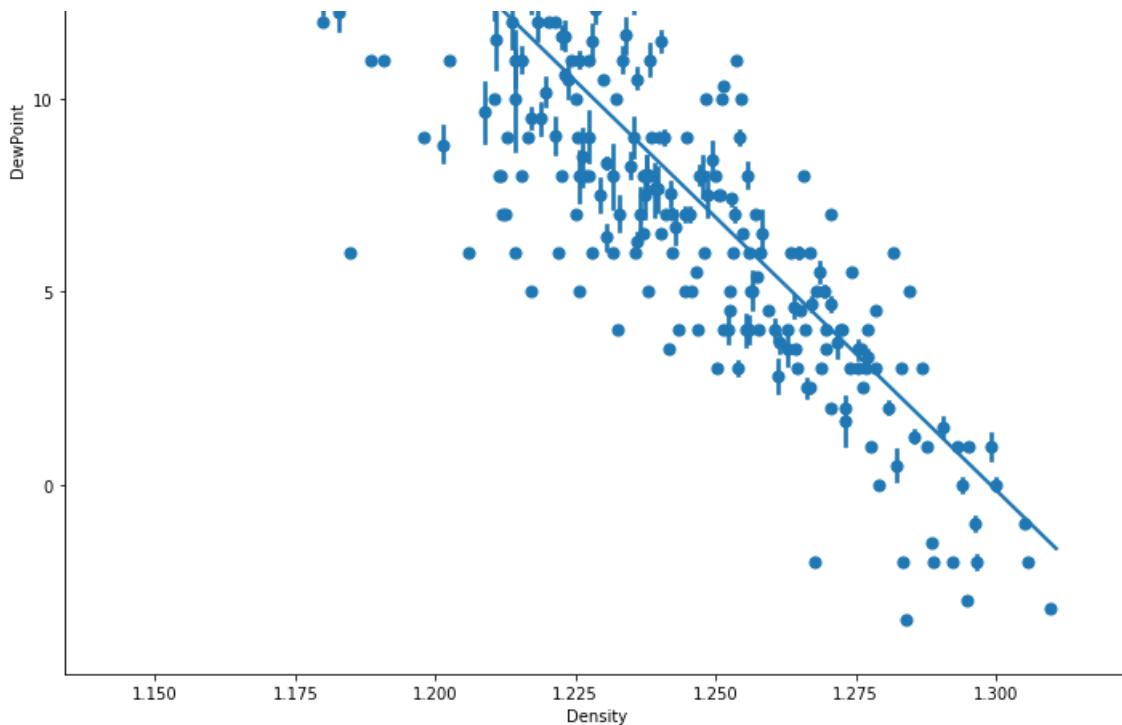
In [77]:

```
sns.lmplot(x="Density",y="DewPoint",data=df,height=10,x_bins=500)
```

Out[77]:

```
<seaborn.axisgrid.FacetGrid at 0x24091d85948>
```



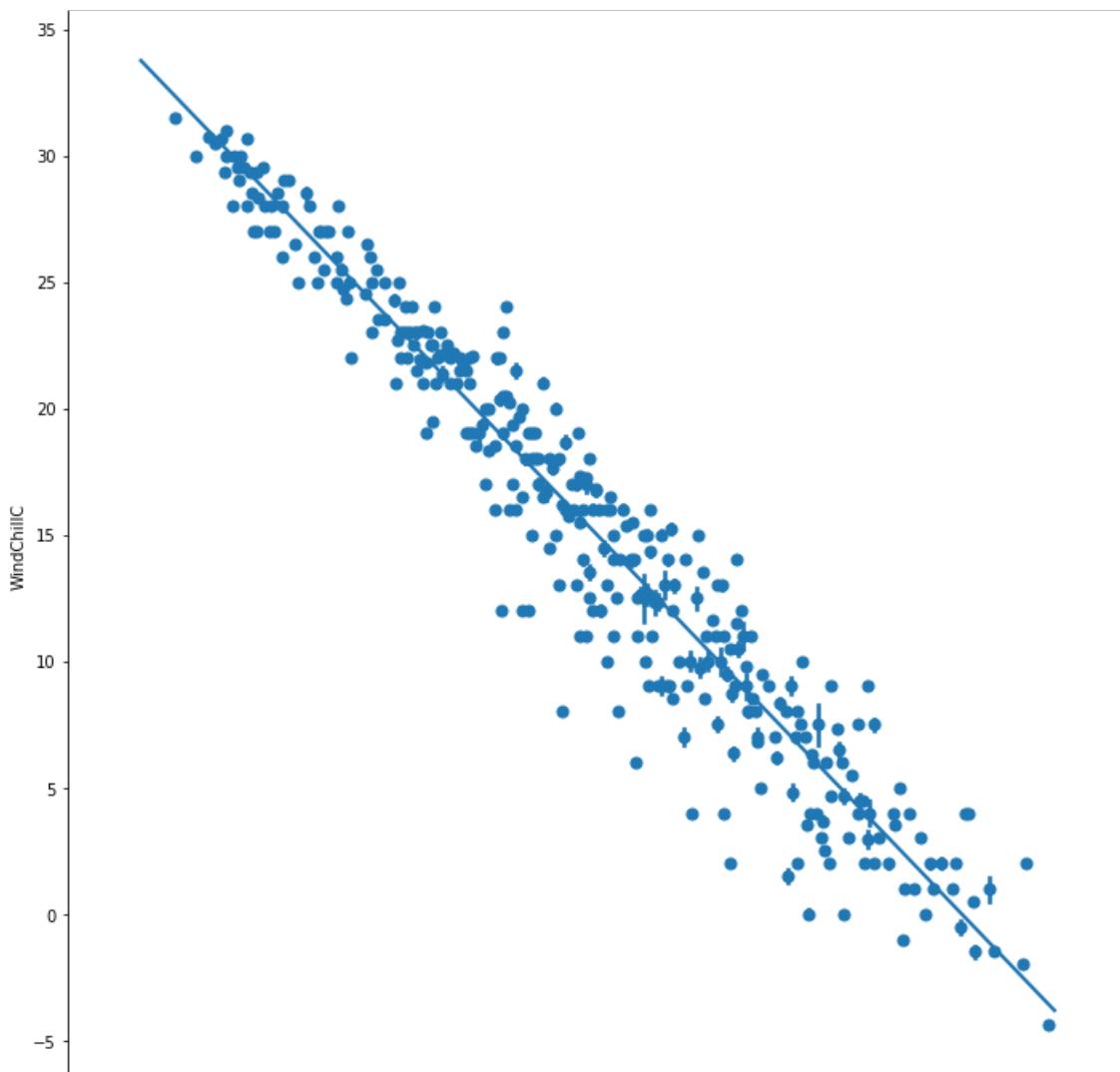


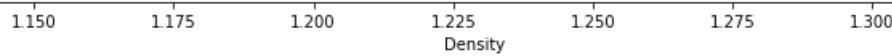
In [78]:

```
sns.lmplot(x="Density", y="WindChillC", data=df, height=10, x_bins=500)
```

Out[78]:

```
<seaborn.axisgrid.FacetGrid at 0x24092f2bd88>
```



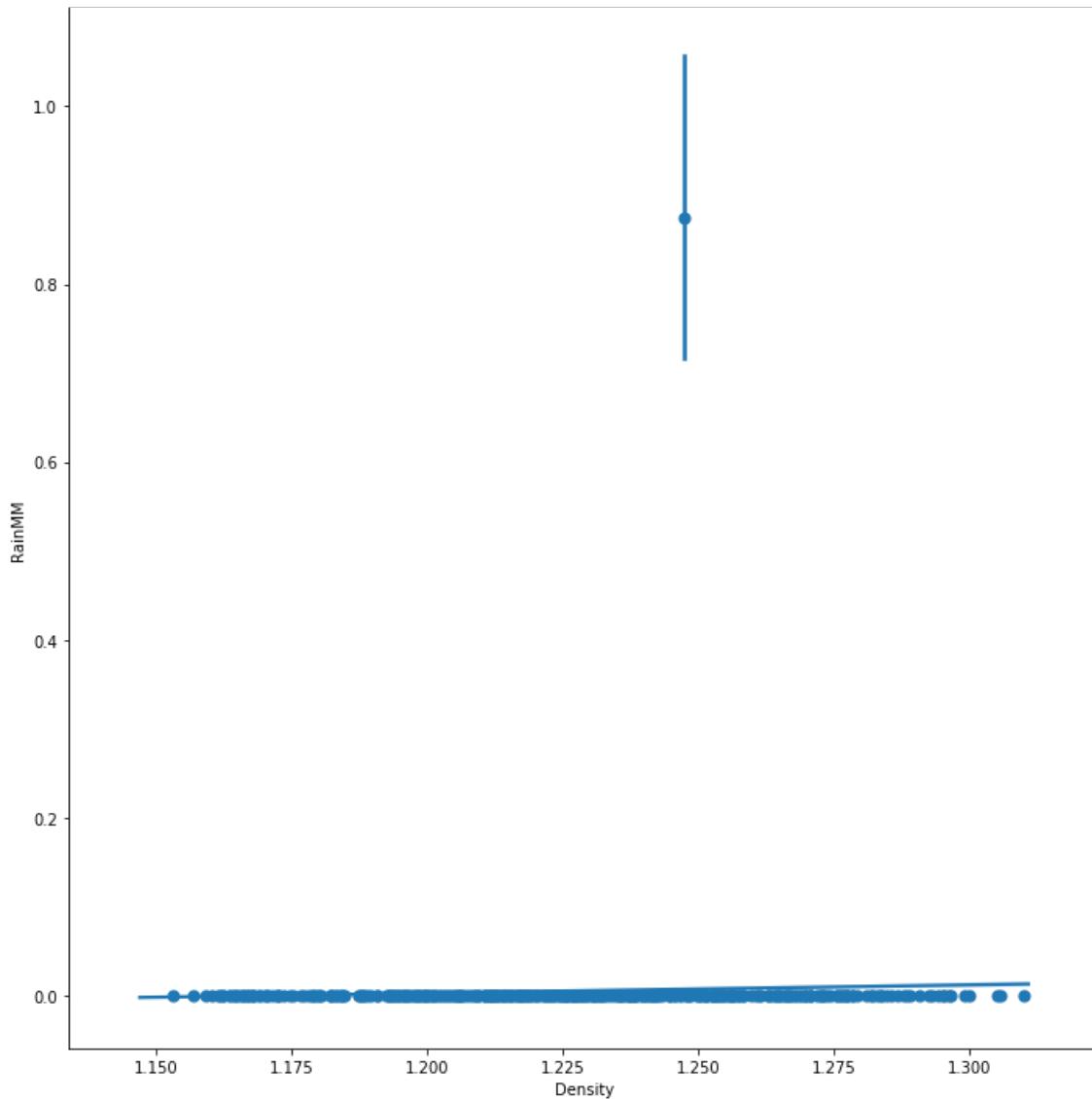


In [79]:

```
sns.lmplot(x="Density", y="RainMM", data=df, height=10, x_bins=500)
```

Out[79]:

```
<seaborn.axisgrid.FacetGrid at 0x2409478fd88>
```

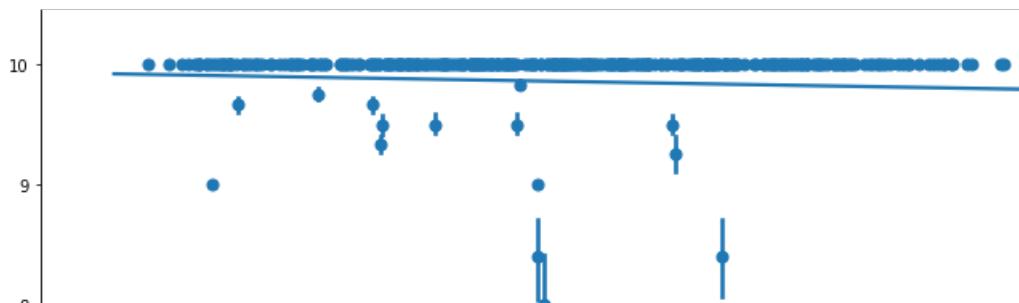


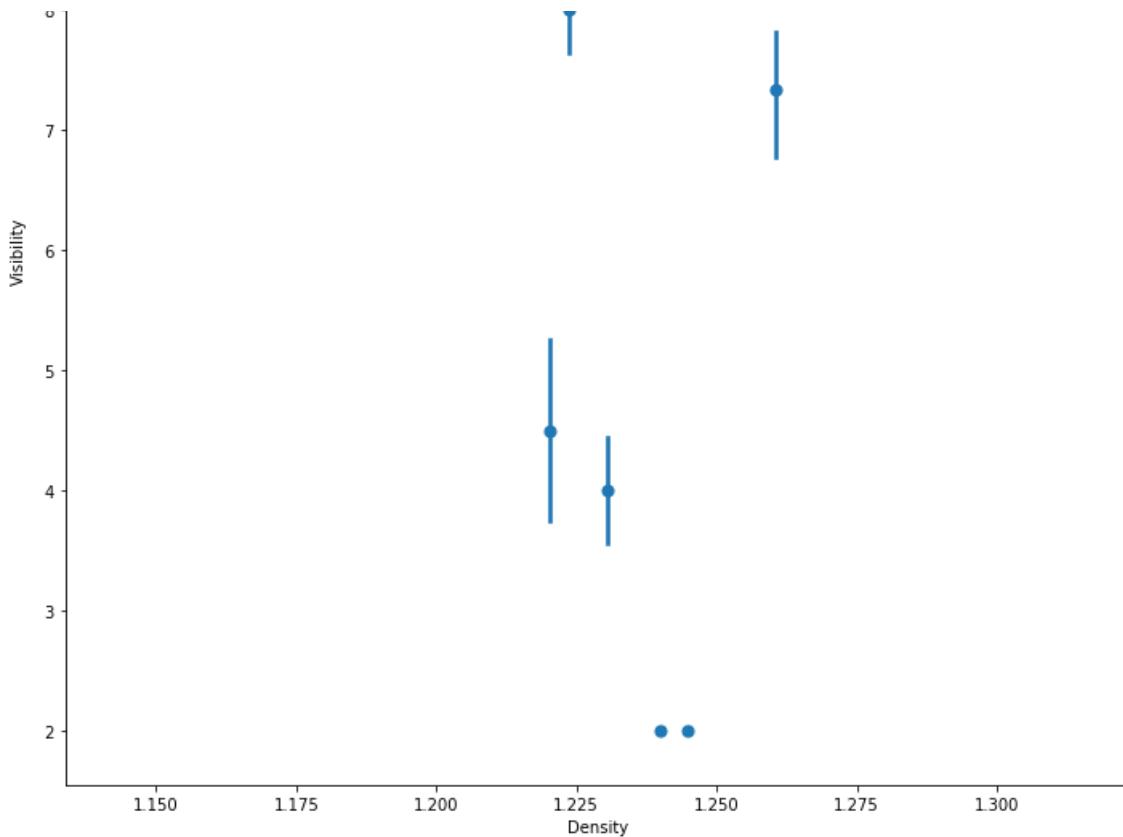
In [80]:

```
sns.lmplot(x="Density", y="Visibility", data=df, height=10, x_bins=500)
```

Out[80]:

```
<seaborn.axisgrid.FacetGrid at 0x240950037c8>
```



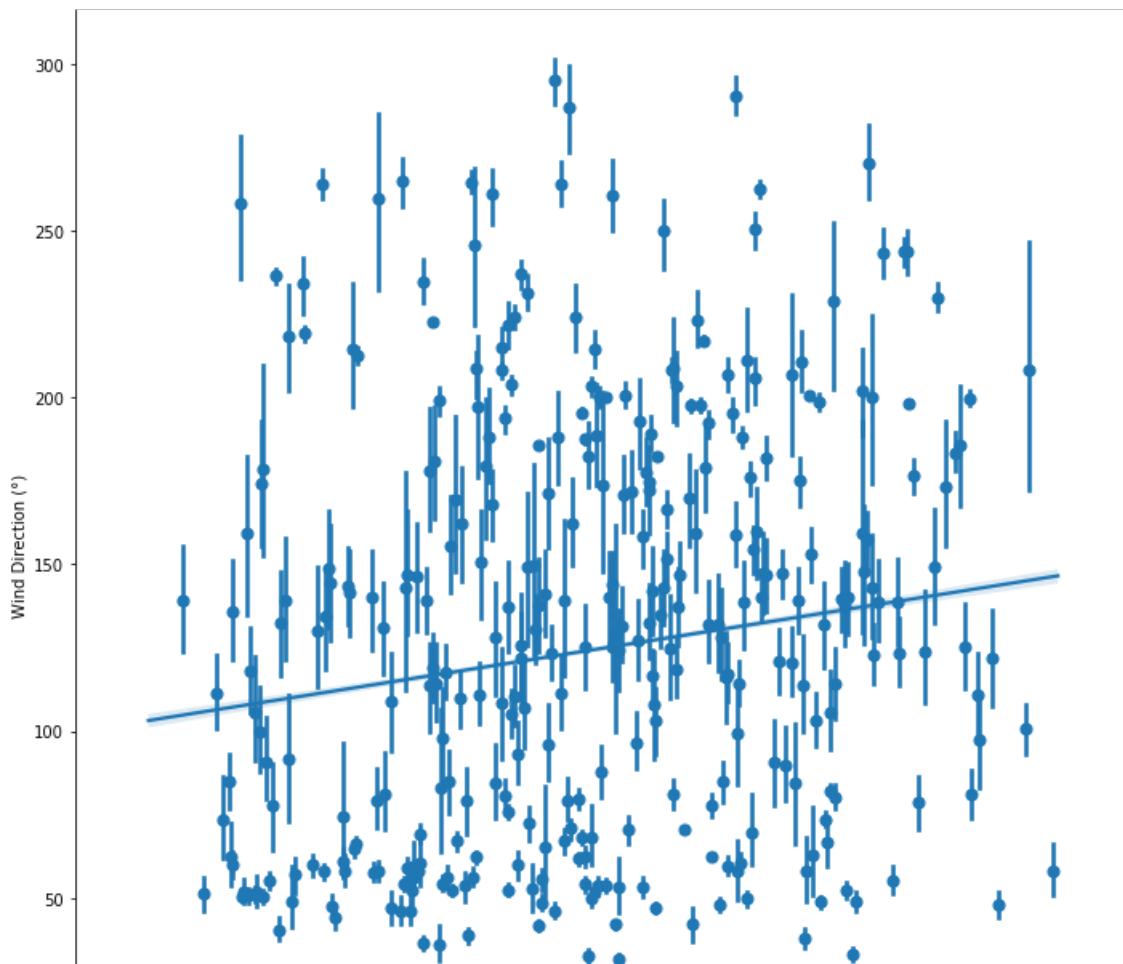


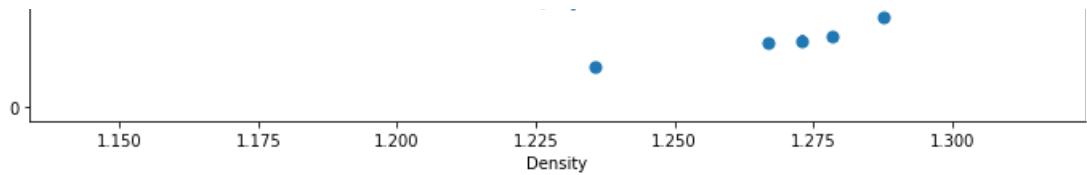
In [81]:

```
sns.lmplot(x="Density", y="Wind Direction (°)", data=df, height=10, x_bins=500)
```

Out[81]:

```
<seaborn.axisgrid.FacetGrid at 0x24095891588>
```



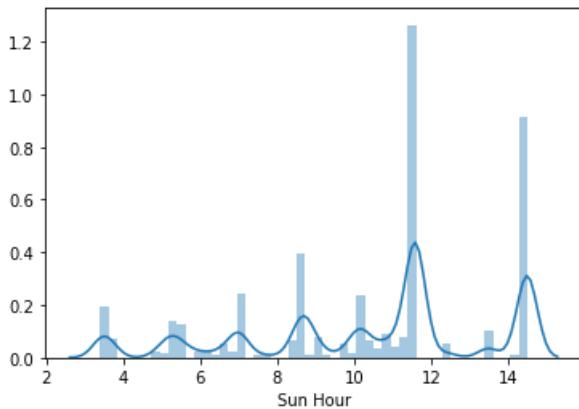


In [85]:

```
sns.distplot(df["Sun Hour"])
```

Out[85]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x240964a2608>
```

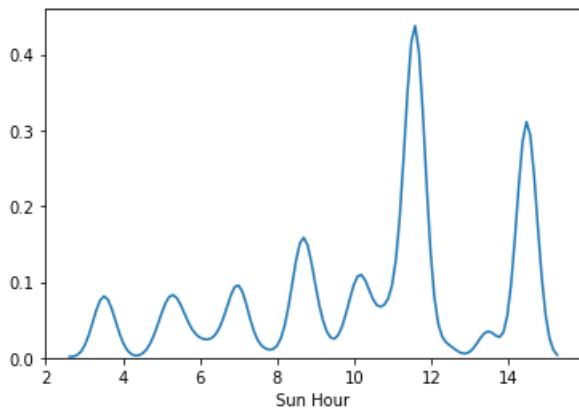


In [86]:

```
sns.distplot(df["Sun Hour"], hist=False)
```

Out[86]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24096470848>
```

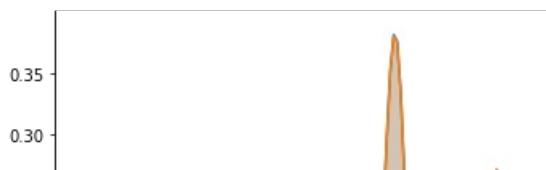


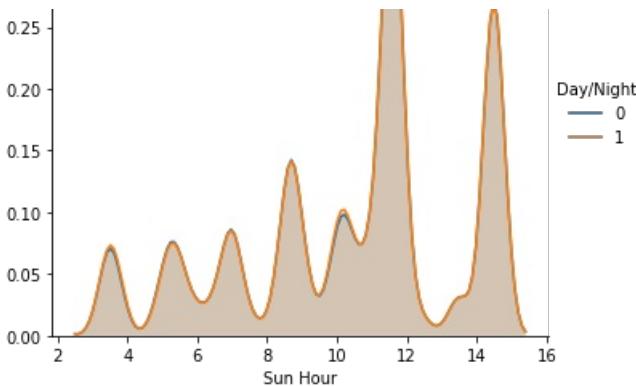
In [89]:

```
sns.FacetGrid(df,hue="Day/Night",height=5).map(sns.kdeplot,"Sun Hour", shade=True).add_legend()  
#Distribution of Electric Production by Day-Night
```

Out[89]:

```
<seaborn.axisgrid.FacetGrid at 0x2409670f048>
```





In [90]:

```
sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"Sun Hour",shade=True).add_legend()
#Distribution of Sun Hour, According to Months
```

```
-----
ValueError                                Traceback (most recent call last)
C:\anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py in kdensityfft(X, kernel, bw,
weights, gridsize, adjust, clip, cut, retgrid)
    450     try:
--> 451         bw = float(bw)
    452     except:
```

**ValueError:** could not convert string to float: 'scott'

During handling of the above exception, another exception occurred:

```
RuntimeError                               Traceback (most recent call last)
<ipython-input-90-f7feb9da8baa> in <module>
----> 1 sns.FacetGrid(df,hue="Month",height=7).map(sns.kdeplot,"Sun Hour",shade=True).add_legend()
      2 #Distribution of Electric Production, According to Months
      3 #It seems that 9th and 11th months does not fit the general distribution of data.
      4 #The months 9 and 11 does not have 2 peaks like the rest of the data
```

```
C:\anaconda3\lib\site-packages\seaborn\axisgrid.py in map(self, func, *args, **kwargs)
    765
    766     # Draw the plot
--> 767     self._facet_plot(func, ax, plot_args, kwargs)
    768
    769     # Finalize the annotations and layout
```

```
C:\anaconda3\lib\site-packages\seaborn\axisgrid.py in _facet_plot(self, func, ax, plot_args,
plot_kwargs)
    849
    850     # Draw the plot
--> 851     func(*plot_args, **plot_kwargs)
    852
    853     # Sort out the supporting information
```

```
C:\anaconda3\lib\site-packages\seaborn\distributions.py in kdeplot(data, data2, shade, vertical, k
ernel, bw, gridsize, cut, clip, legend, cumulative, shade_lowest, cbar, cbar_ax, cbar_kws, ax, **k
wargs)
    703         ax = _univariate_kdeplot(data, shade, vertical, kernel, bw,
    704                                     gridsize, cut, clip, legend, ax,
--> 705                                     cumulative=cumulative, **kwargs)
    706
    707     return ax
```

```
C:\anaconda3\lib\site-packages\seaborn\distributions.py in _univariate_kdeplot(data, shade,
vertical, kernel, bw, gridsize, cut, clip, legend, ax, cumulative, **kwargs)
    293         x, y = _statsmodels_univariate_kde(data, kernel, bw,
    294                                         gridsize, cut, clip,
--> 295                                         cumulative=cumulative)
    296     else:
    297         # Fall back to scipy if missing statsmodels
```

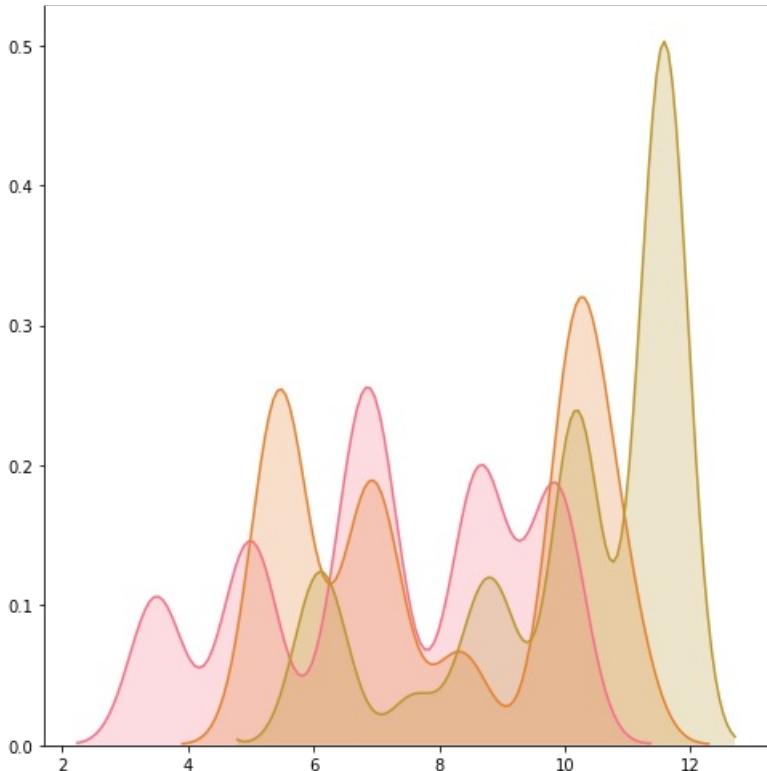
```
C:\anaconda3\lib\site-packages\seaborn\distributions.py in _statsmodels_univariate_kde(data,
kernel, bw, gridsize, cut, clip, cumulative)
    365         fft = kernel == "gau"
    366         kde = smnp.KDEUnivariate(data)
```

```

--> 367      kde.fit(kernel, bw, fft, gridsize=gridsize, cut=cut, clip=clip)
368      if cumulative:
369          grid, y = kde.support, kde.cdf
C:\anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py in fit(self, kernel, bw, fft,
weights, gridsize, adjust, cut, clip)
138          density, grid, bw = kdensityfft(endog, kernel=kernel, bw=bw,
139                                         adjust=adjust, weights=weights, gridsize=gridsize,
--> 140                                         clip=clip, cut=cut)
141      else:
142          density, grid, bw = kdensity(endog, kernel=kernel, bw=bw,
C:\anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py in kdensityfft(X, kernel, bw,
weights, gridsize, adjust, clip, cut, retgrid)
451      bw = float(bw)
452  except:
--> 453      bw = bandwidths.select_bandwidth(X, bw, kern) # will cross-val fit this pattern?
454  bw *= adjust
455
C:\anaconda3\lib\site-packages\statsmodels\nonparametric\bandwidths.py in select_bandwidth(x, bw,
kernel)
172      # eventually this can fall back on another selection criterion.
173      err = "Selected KDE bandwidth is 0. Cannot estimate density."
--> 174      raise RuntimeError(err)
175  else:
176      return bandwidth

```

**RuntimeError:** Selected KDE bandwidth is 0. Cannot estimate density.

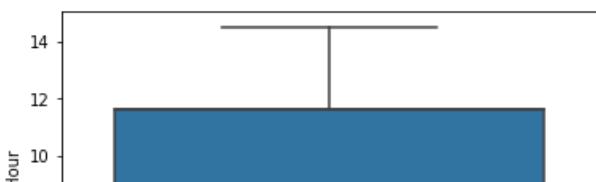


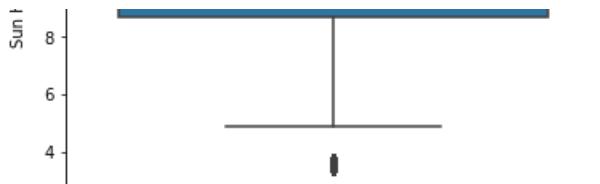
In [91]:

```
sns.boxplot(df["Sun Hour"], orient="v")
```

Out[91]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24096843b48>
```





In [92]:

```
df["Sun Hour"].describe()
```

Out[92]:

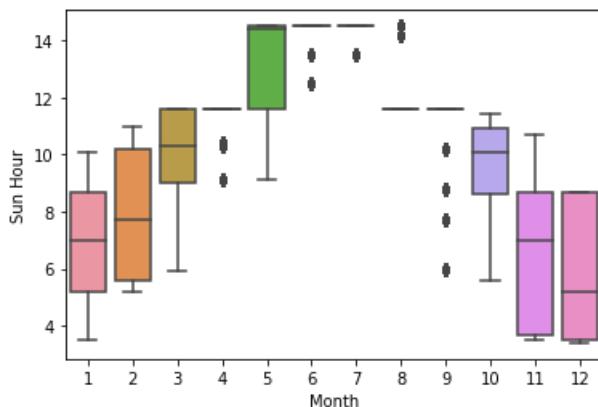
```
count      50530.000000
mean       10.394415
std        3.198427
min        3.400000
25%        8.700000
50%        11.600000
75%        11.600000
max        14.500000
Name: Sun Hour, dtype: float64
```

In [93]:

```
sns.boxplot(y=df["Sun Hour"], orient="v", x="Month", data=df)
```

Out[93]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24096894808>
```



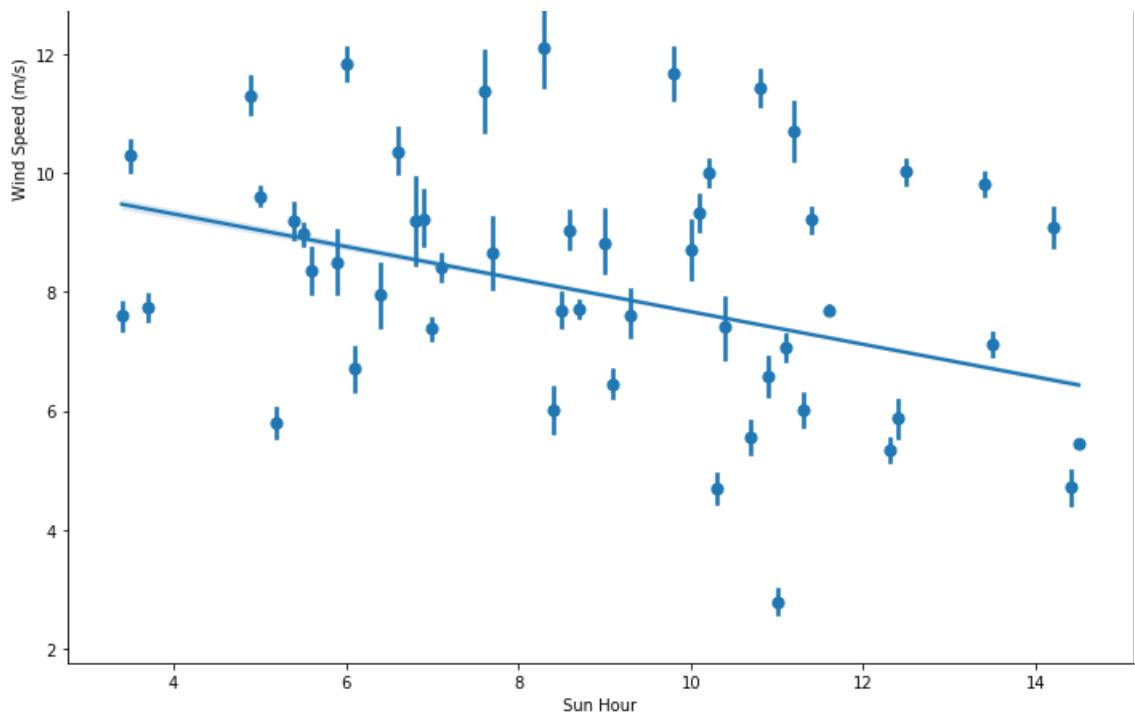
In [95]:

```
sns.lmplot(x="Sun Hour", y="Wind Speed (m/s)", data=df, height=10, x_bins=500)
```

Out[95]:

```
<seaborn.axisgrid.FacetGrid at 0x240969f7cc8>
```



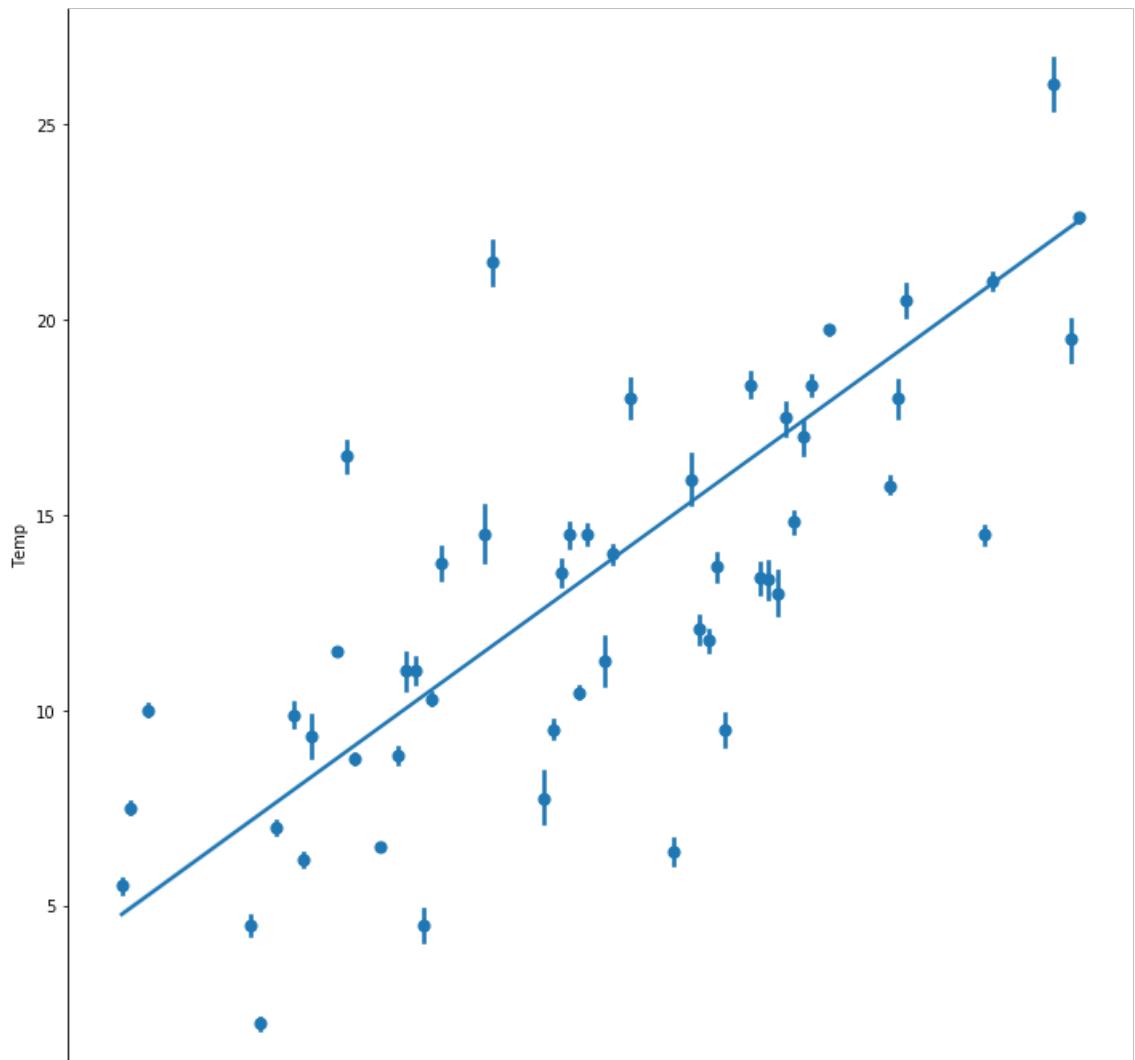


In [96]:

```
sns.lmplot(x="Sun Hour",y="Temp",data=df,height=10,x_bins=500)
#Seem to have a linear relationship
```

Out [96]:

```
<seaborn.axisgrid.FacetGrid at 0x24096b0ad88>
```



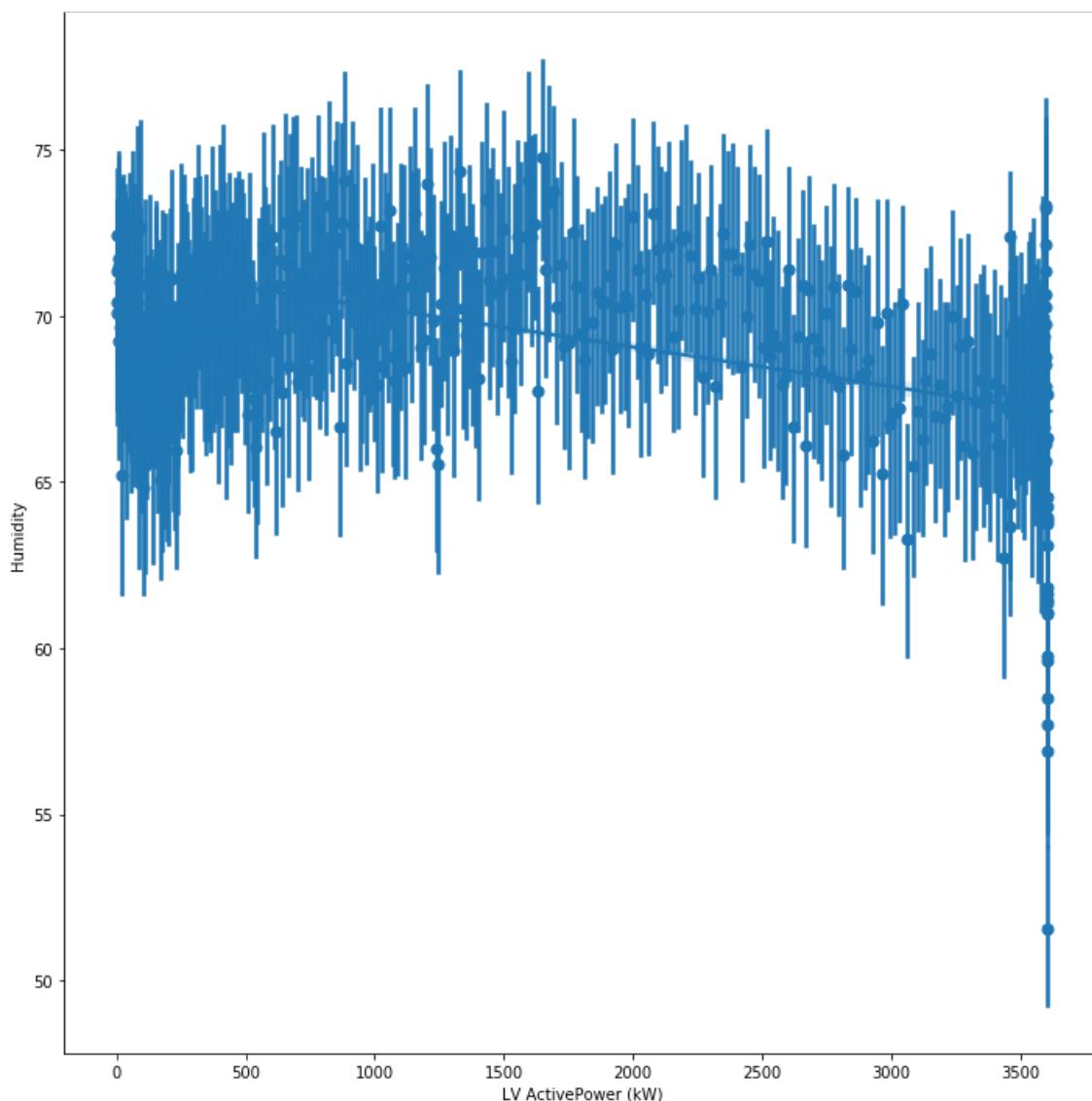


In [97]:

```
sns.lmplot(x="LV ActivePower (kW)", y="Humidity", data=df, height=10, x_bins=500)
```

Out[97]:

```
<seaborn.axisgrid.FacetGrid at 0x24096cde208>
```

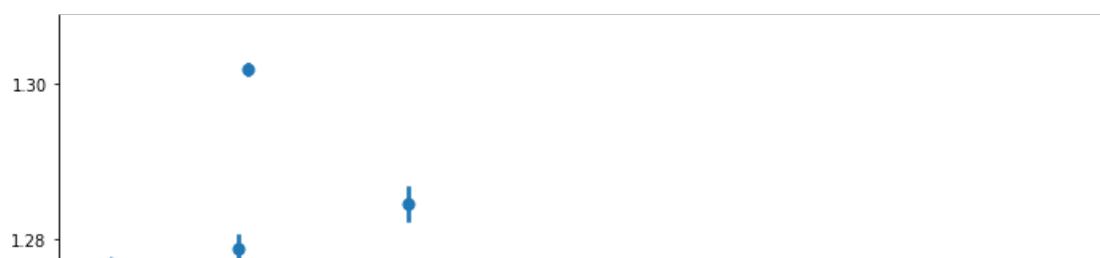


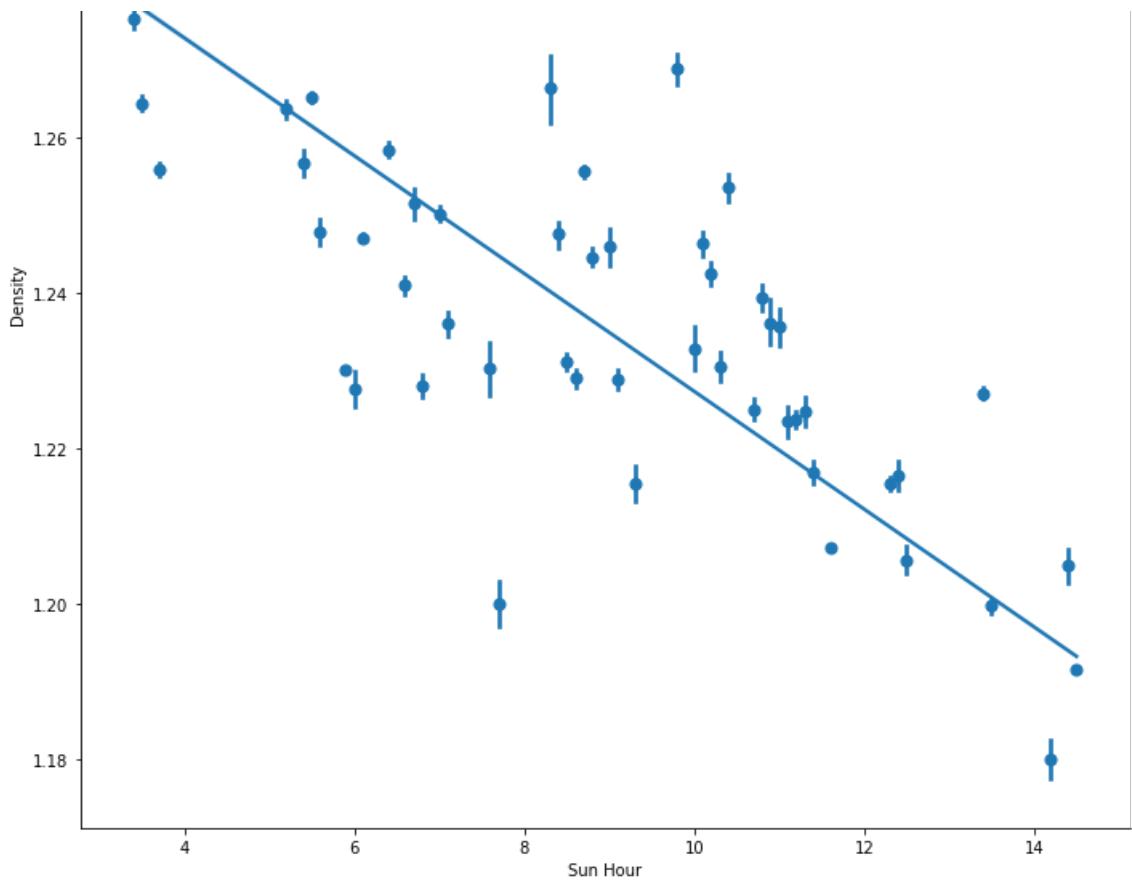
In [98]:

```
sns.lmplot(x="Sun Hour", y="Density", data=df, height=10, x_bins=500)  
#Density and Sun hour seem to have negative linear relationship
```

Out[98]:

```
<seaborn.axisgrid.FacetGrid at 0x24099d69d88>
```



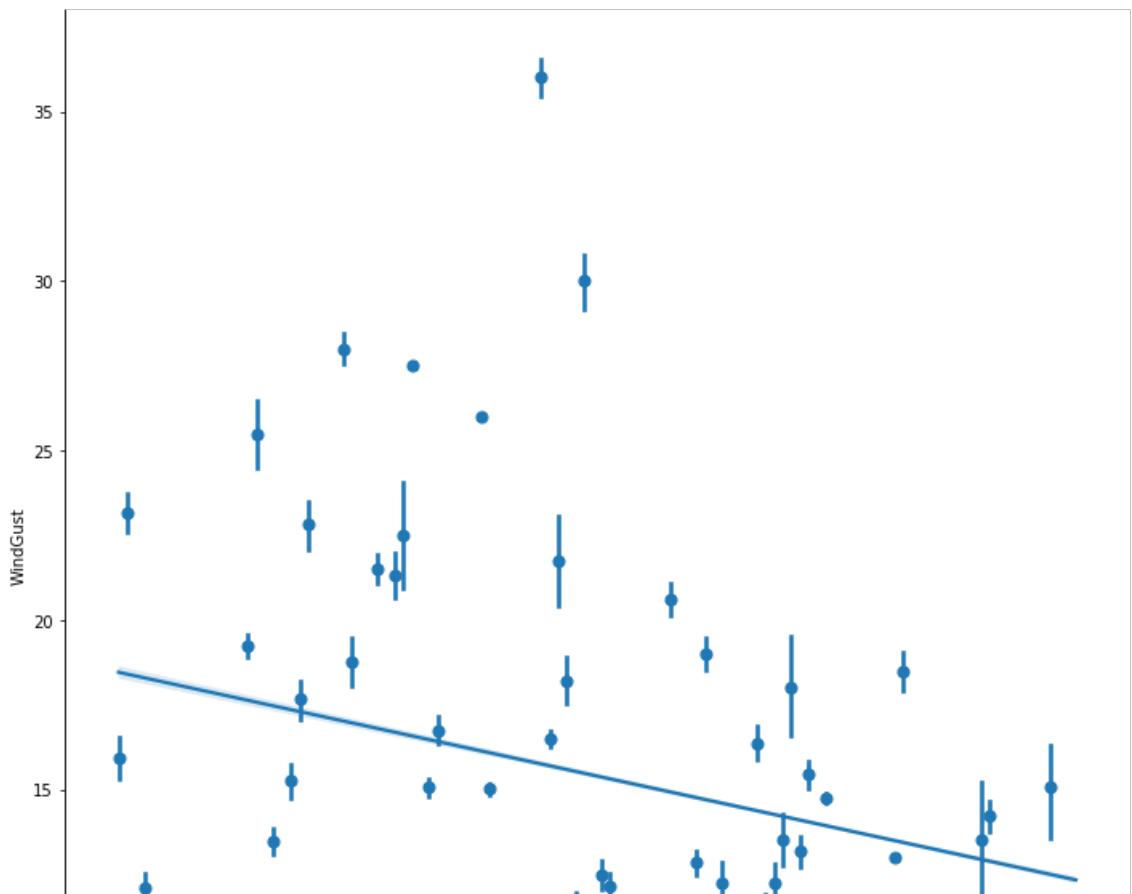


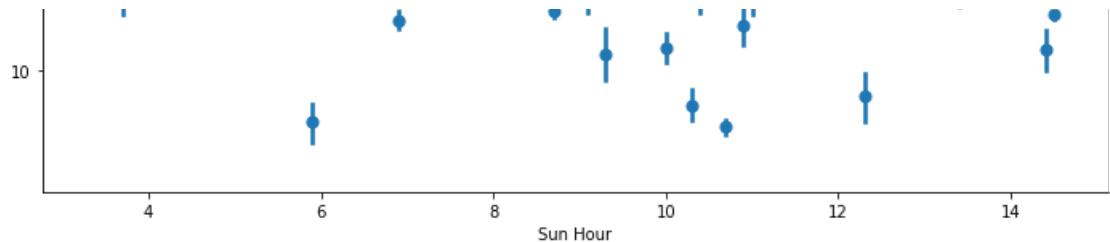
In [99]:

```
sns.lmplot(x="Sun Hour",y="WindGust",data=df,height=10,x_bins=500)
#Wind gust and Sun hour seem to have a negative linear relationship
```

Out [99]:

```
<seaborn.axisgrid.FacetGrid at 0x2409a27f5c8>
```



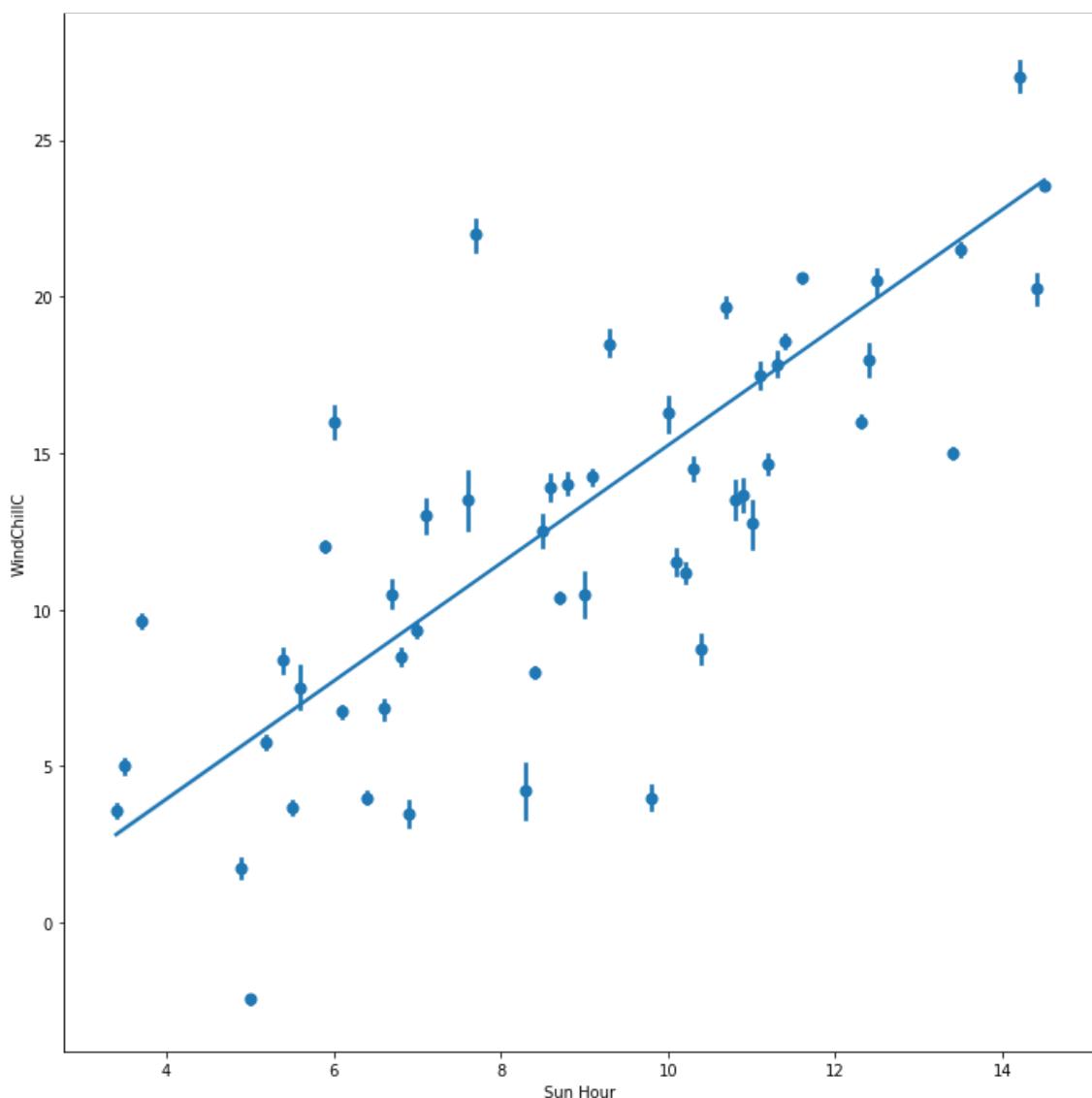


In [100]:

```
sns.lmplot(x="Sun Hour",y="WindChillC",data=df,height=10,x_bins=500)
```

Out[100]:

```
<seaborn.axisgrid.FacetGrid at 0x2409a791088>
```

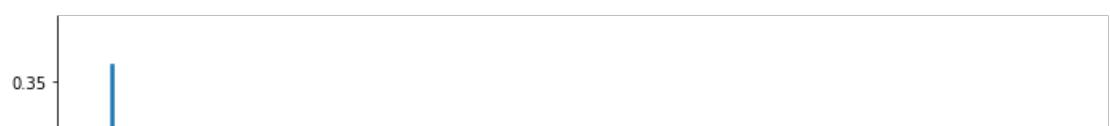


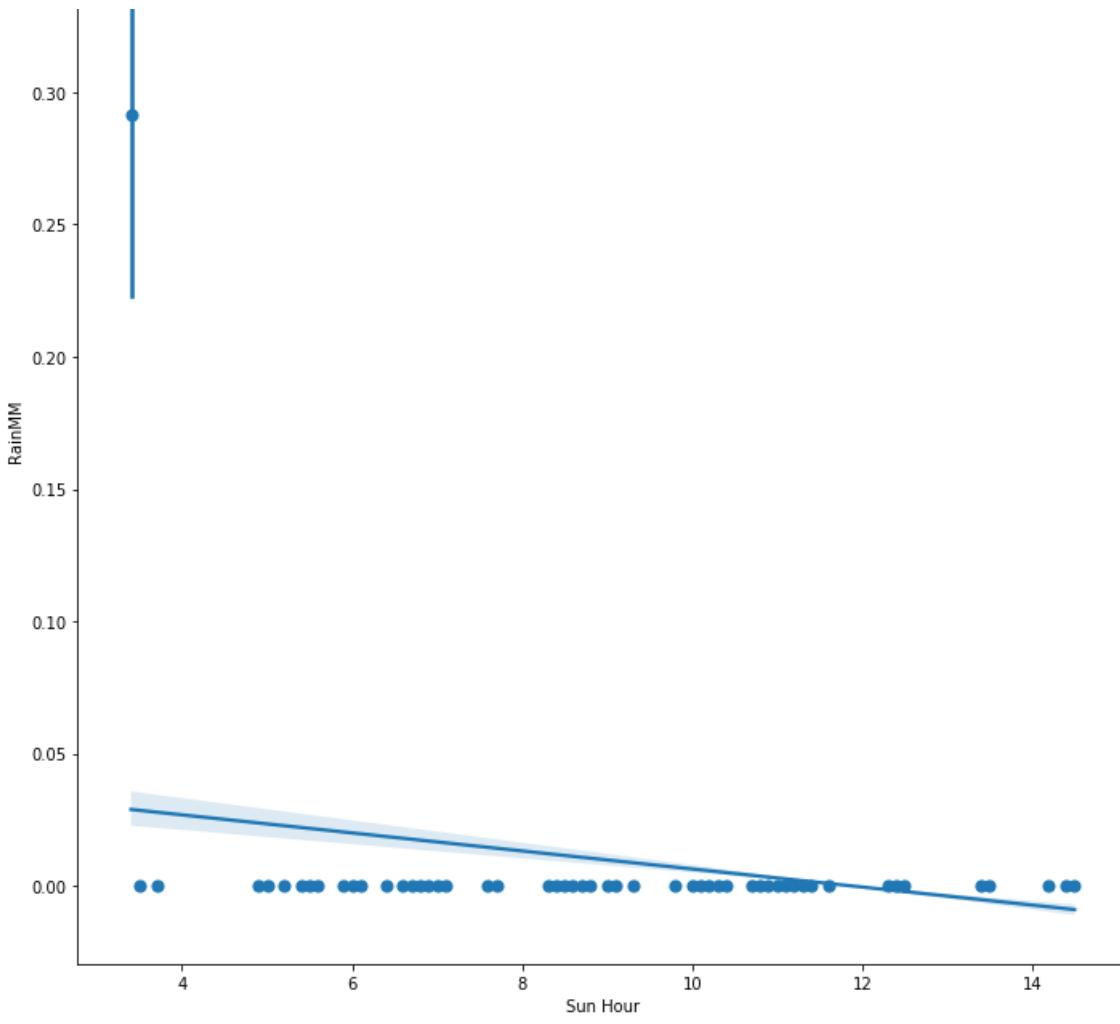
In [101]:

```
sns.lmplot(x="Sun Hour",y="RainMM",data=df,height=10,x_bins=500)
```

Out[101]:

```
<seaborn.axisgrid.FacetGrid at 0x2409acc9748>
```



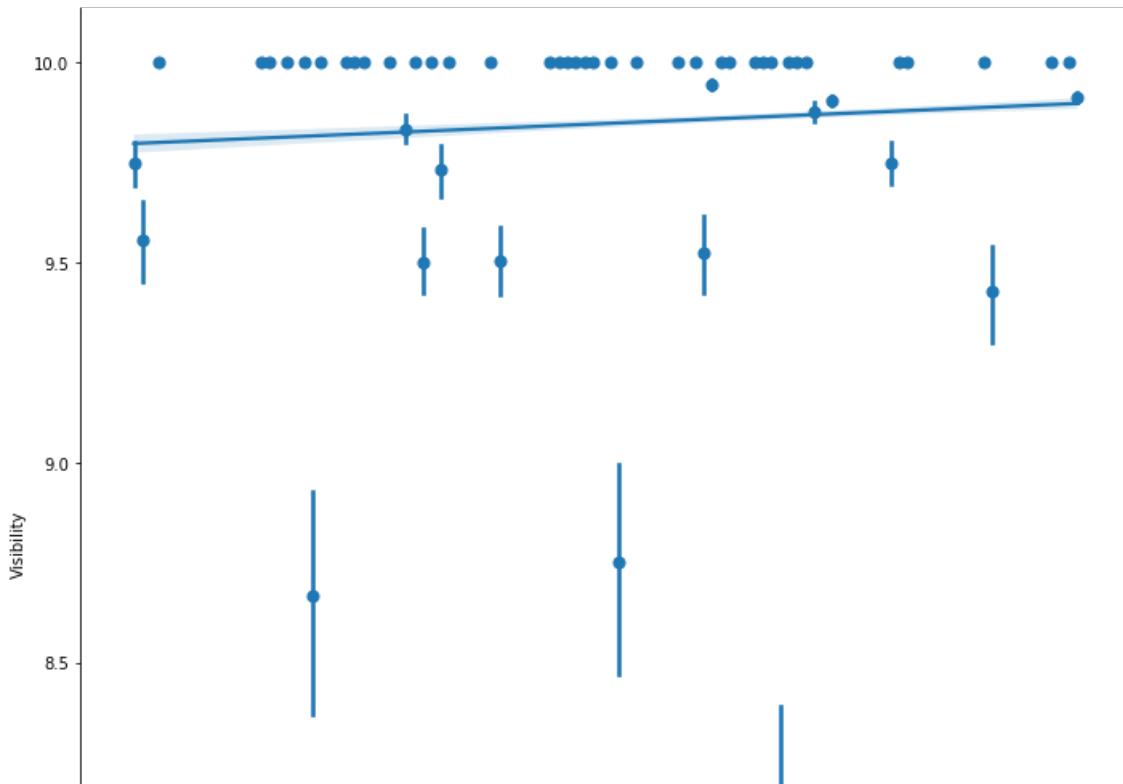


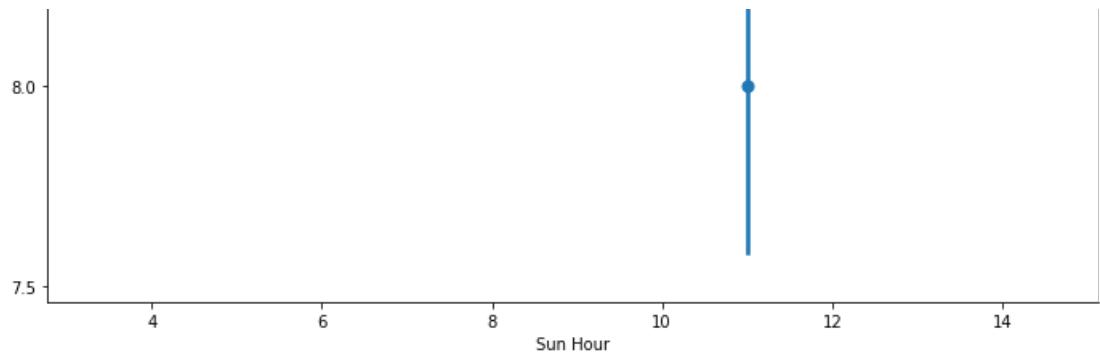
In [102]:

```
sns.lmplot(x="Sun Hour",y="Visibility",data=df,height=10,x_bins=500)
```

Out[102]:

```
<seaborn.axisgrid.FacetGrid at 0x2409b1b4608>
```



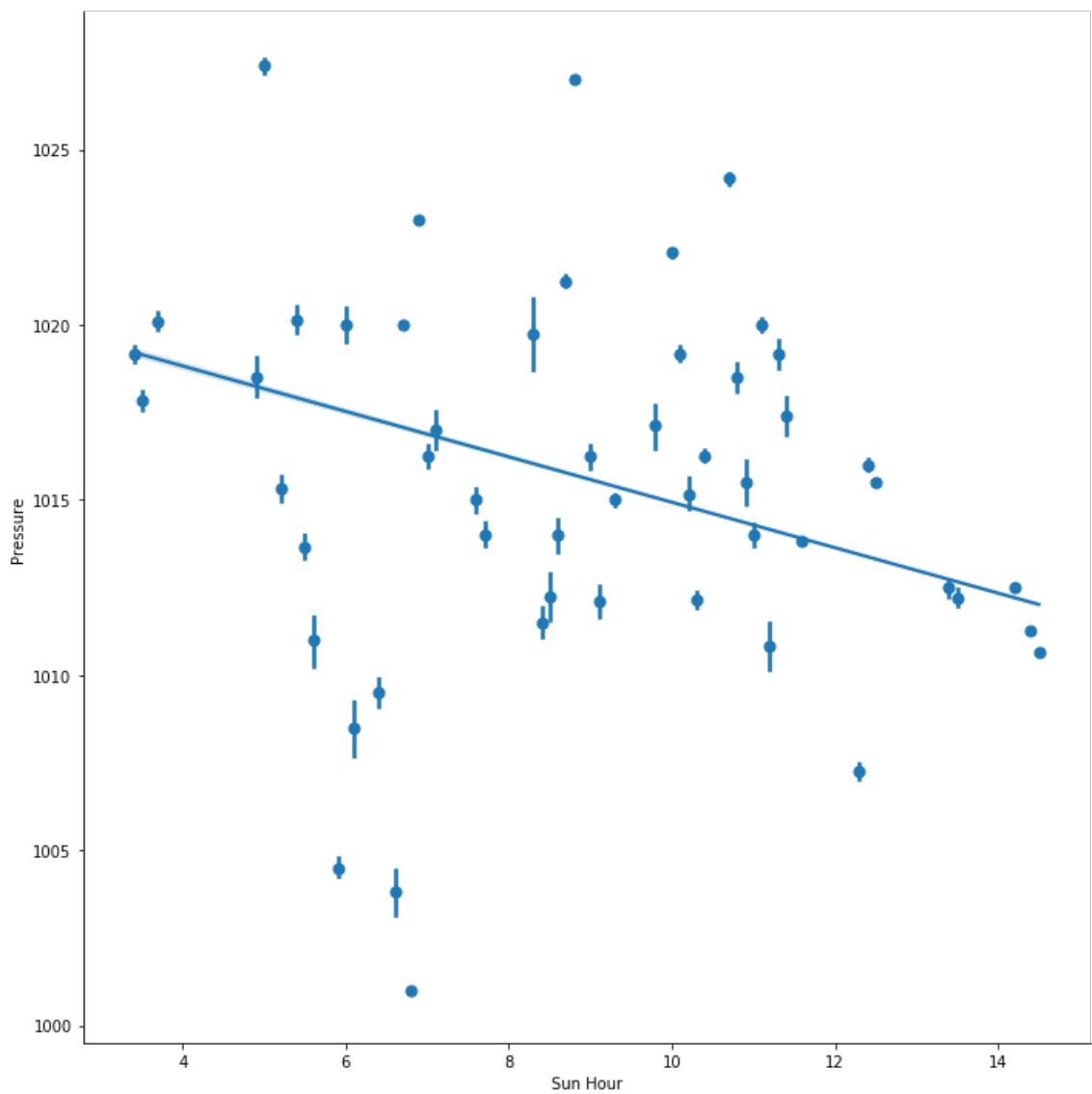


In [103]:

```
sns.lmplot(x="Sun Hour",y="Pressure",data=df,height=10,x_bins=500)
Sun Hour and Wind Chill seem to have a negative linear relationship
```

Out [103]:

```
<seaborn.axisgrid.FacetGrid at 0x2409b6ef8c8>
```



In [ ]:

