In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("Weather Data.csv")
df.head()
```

Out[2]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog
3	1/1/2012 3:00	-1.5	-3.2	88	6	4.0	101.27	Freezing Drizzle,Fog
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog

In [3]:
```
df.tail()
```

Out[3]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
8779	12/31/2012 19:00	0.1	-2.7	81	30	9.7	100.13	Snow
8780	12/31/2012 20:00	0.2	-2.4	83	24	9.7	100.03	Snow
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Snow
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Snow
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Snow

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Date/Time           0
Temp_C              0
Dew Point Temp_C    0
Rel Hum_%           0
Wind Speed_km/h     0
Visibility_km       0
Press_kPa           0
Weather             0
dtype: int64
```

In [5]:
```
df.skew()
```

```
/tmp/ipykernel_2960/1665899112.py:1: FutureWarning: The default value of numeric_only in DataFrame.skew is deprecate
d. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select
only valid columns or specify the value of numeric_only to silence this warning.
  df.skew()
```

Out[5]:
```
Temp_C             -0.177666
Dew Point Temp_C   -0.318433
Rel Hum_%          -0.323830
Wind Speed_km/h     0.871374
Visibility_km       0.413362
Press_kPa          -0.229925
dtype: float64
```

In [6]:
```
df.describe()
```

Out[6]:

	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
count	8784.000000	8784.000000	8784.000000	8784.000000	8784.000000	8784.000000
mean	8.798144	2.555294	67.431694	14.945469	27.664447	101.051623
std	11.687883	10.883072	16.918881	8.688696	12.622688	0.844005
min	-23.300000	-28.500000	18.000000	0.000000	0.200000	97.520000
25%	0.100000	-5.900000	56.000000	9.000000	24.100000	100.560000
50%	9.300000	3.300000	68.000000	13.000000	25.000000	101.070000
75%	18.800000	11.800000	81.000000	20.000000	25.000000	101.590000
max	33.000000	24.400000	100.000000	83.000000	48.300000	103.650000

In [7]:
```
corr = df.corr()
corr
```

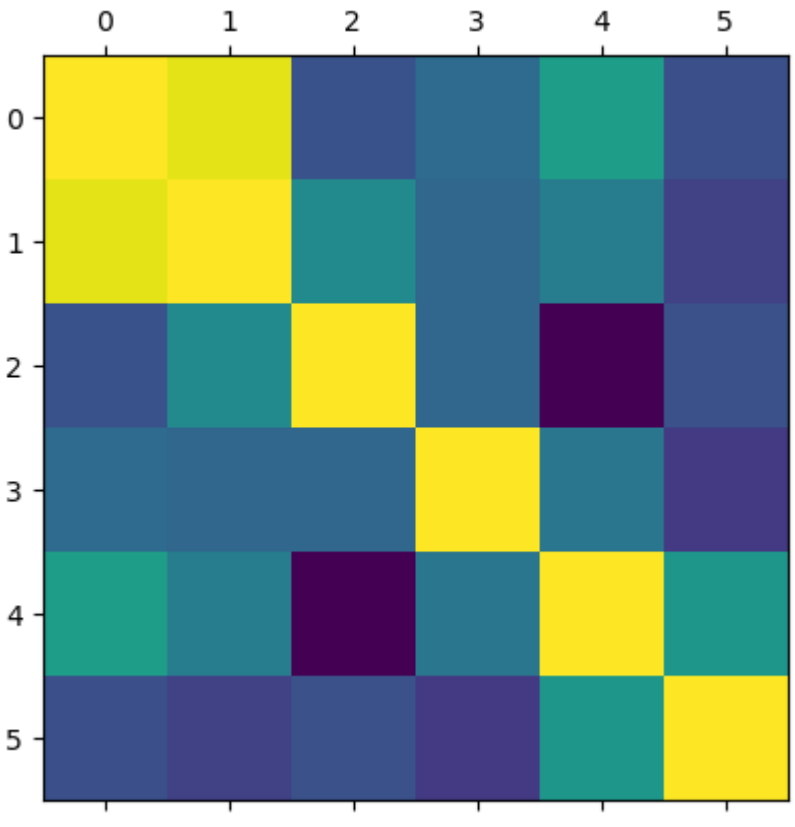
```
/tmp/ipykernel_2960/2438084875.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecate
d. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to
silence this warning.
  corr = df.corr()
```

Out[7]:

	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Temp_C	1.000000	0.932714	-0.220182	-0.061876	0.273455	-0.236389
Dew Point Temp_C	0.932714	1.000000	0.139494	-0.095685	0.050813	-0.320616
Rel Hum_%	-0.220182	0.139494	1.000000	-0.092743	-0.633683	-0.231424
Wind Speed_km/h	-0.061876	-0.095685	-0.092743	1.000000	0.004883	-0.356613
Visibility_km	0.273455	0.050813	-0.633683	0.004883	1.000000	0.231847
Press_kPa	-0.236389	-0.320616	-0.231424	-0.356613	0.231847	1.000000

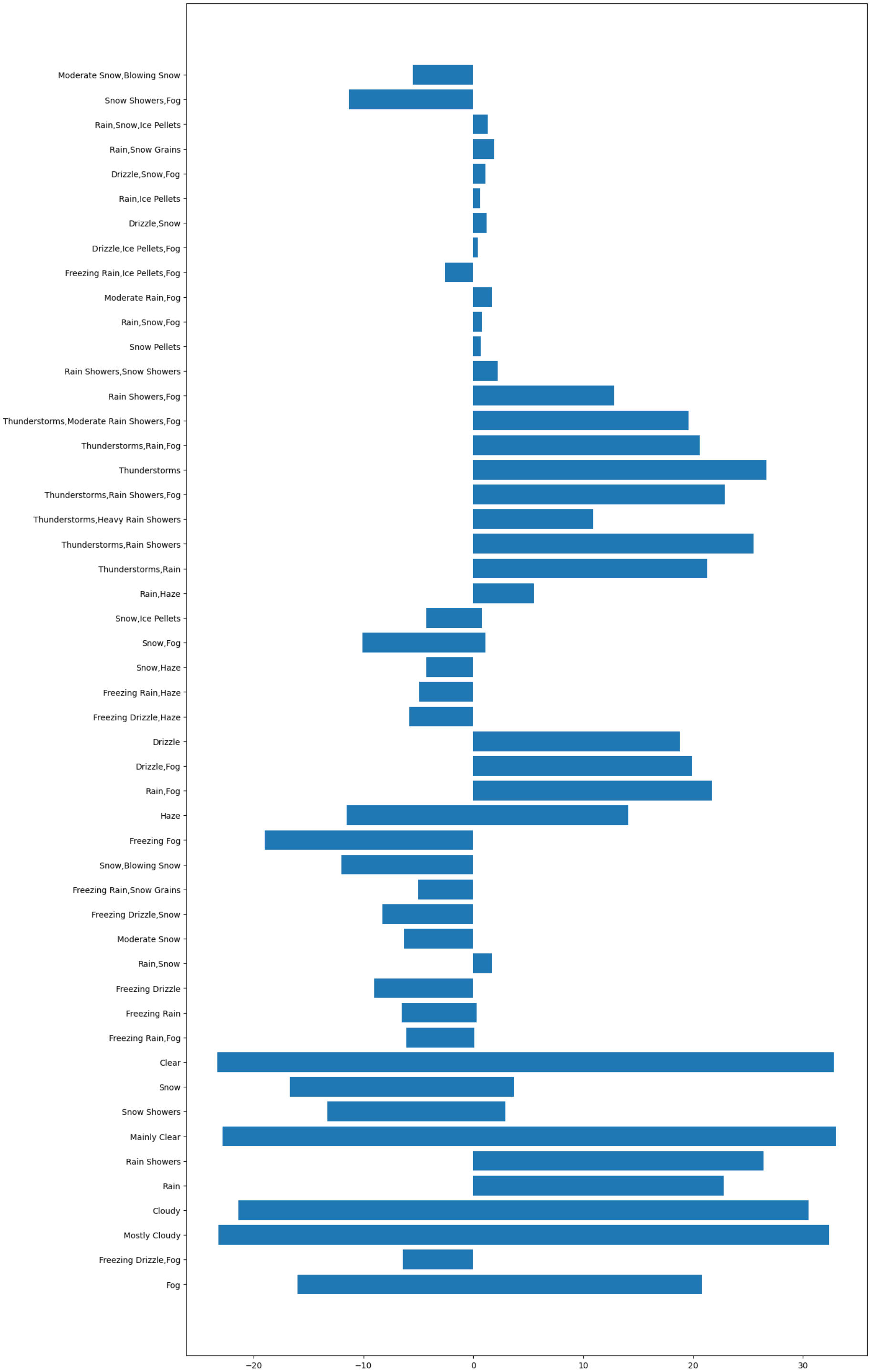
In [8]:
```
plt.matshow(corr)
```

Out[8]:
```
<matplotlib.image.AxesImage at 0x7f57bdee8b10>
```



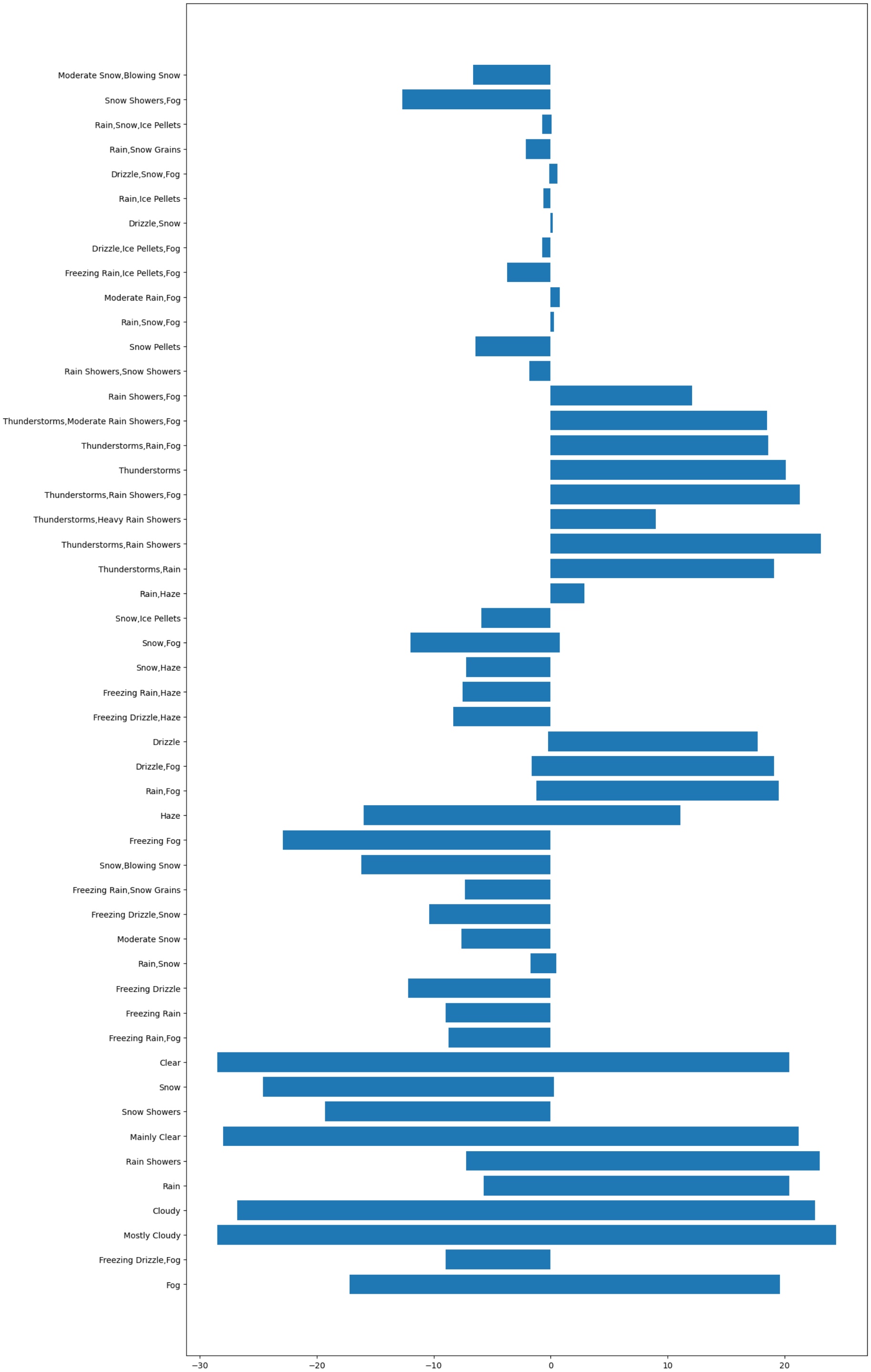
In [9]:
```
plt.figure(figsize=(15,30))
plt.barh(df["Weather"],df["Temp_C"])
```

Out[9]:
```
<BarContainer object of 8784 artists>
```


Moderate Snow,Blowing Snow
Snow Showers,Fog
Rain,Snow,Ice Pellets
Rain,Snow Grains
Drizzle,Snow,Fog
Rain,Ice Pellets
Drizzle,Snow
Drizzle,Ice Pellets,Fog
Freezing Rain,Ice Pellets,Fog
Moderate Rain,Fog
Rain,Snow,Fog
Snow Pellets
Rain Showers,Snow Showers
Rain Showers,Fog
Thunderstorms,Moderate Rain Showers,Fog
Thunderstorms,Rain,Fog
Thunderstorms
Thunderstorms,Rain Showers,Fog
Thunderstorms,Heavy Rain Showers
Thunderstorms,Rain Showers
Thunderstorms,Rain
Rain,Haze
Snow,Ice Pellets
Snow,Fog
Snow,Haze
Freezing Rain,Haze
Freezing Drizzle,Haze
Drizzle
Drizzle,Fog
Rain,Fog
Haze
Freezing Fog
Snow,Blowing Snow
Freezing Rain,Snow Grains
Freezing Drizzle,Snow
Moderate Snow
Rain,Snow
Freezing Drizzle
Freezing Rain
Freezing Rain,Fog
Clear
Snow
Snow Showers
Mainly Clear
Rain Showers
Rain
Cloudy
Mostly Cloudy
Freezing Drizzle,Fog
Fog
−20
−10
0
10
20
30

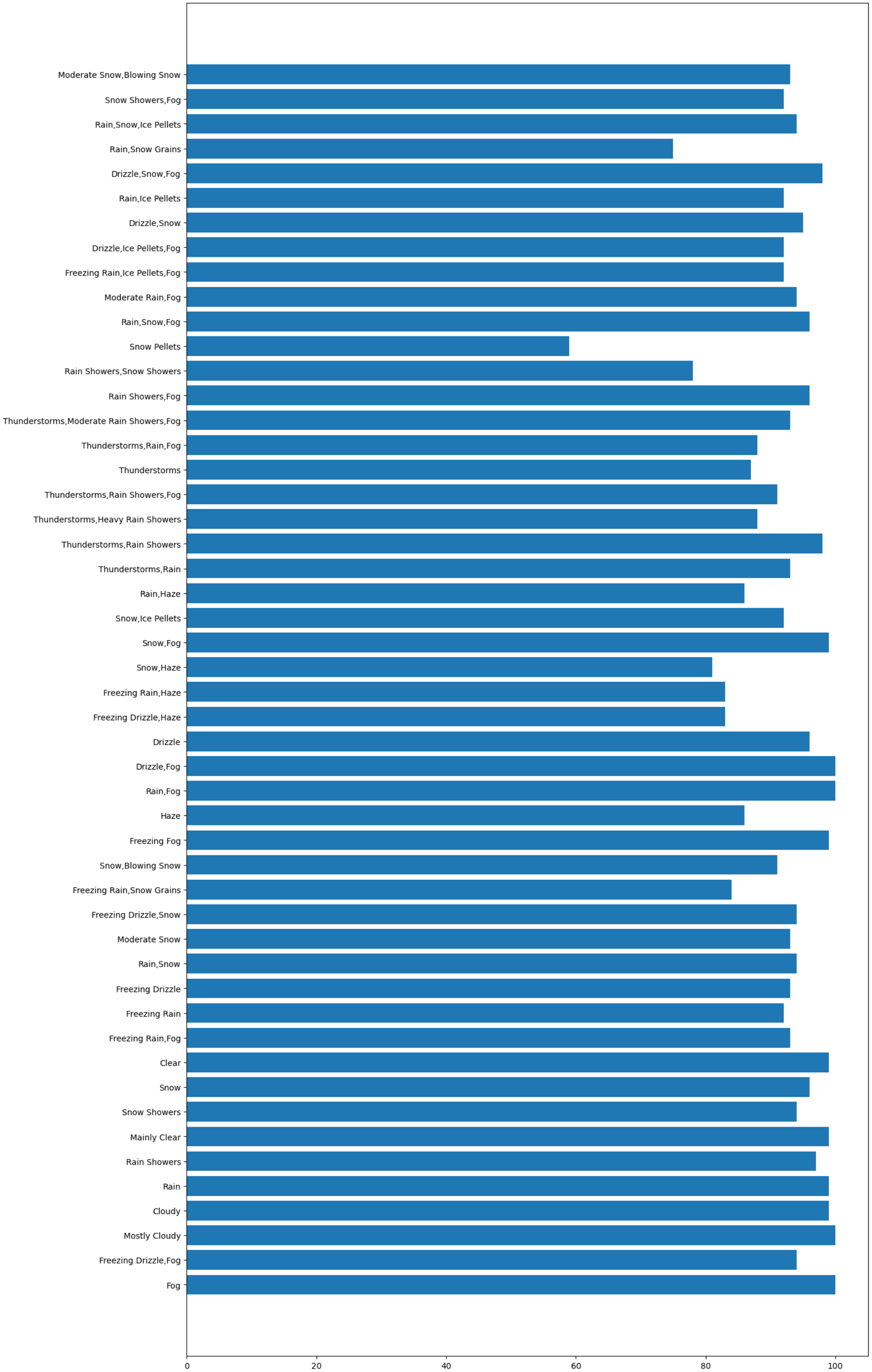
In [10]:
```
plt.figure(figsize=(15,30))
plt.barh(df["Weather"],df["Dew Point Temp_C"])
```

Out[10]:
```
<BarContainer object of 8784 artists>
```


Moderate Snow,Blowing Snow
Snow Showers,Fog
Rain,Snow,Ice Pellets
Rain,Snow Grains
Drizzle,Snow,Fog
Rain,Ice Pellets
Drizzle,Snow
Drizzle,Ice Pellets,Fog
Freezing Rain,Ice Pellets,Fog
Moderate Rain,Fog
Rain,Snow,Fog
Snow Pellets
Rain Showers,Snow Showers
Rain Showers,Fog
Thunderstorms,Moderate Rain Showers,Fog
Thunderstorms,Rain,Fog
Thunderstorms
Thunderstorms,Rain Showers,Fog
Thunderstorms,Heavy Rain Showers
Thunderstorms,Rain Showers
Thunderstorms,Rain
Rain,Haze
Snow,Ice Pellets
Snow,Fog
Snow,Haze
Freezing Rain,Haze
Freezing Drizzle,Haze
Drizzle
Drizzle,Fog
Rain,Fog
Haze
Freezing Fog
Snow,Blowing Snow
Freezing Rain,Snow Grains
Freezing Drizzle,Snow
Moderate Snow
Rain,Snow
Freezing Drizzle
Freezing Rain
Freezing Rain,Fog
Clear
Snow
Snow Showers
Mainly Clear
Rain Showers
Rain
Cloudy
Mostly Cloudy
Freezing Drizzle,Fog
Fog
−30
−20
−10
0
10
20

In [11]:
```
plt.figure(figsize=(15,30))
plt.barh(df["Weather"],df["Rel Hum_%"])
```

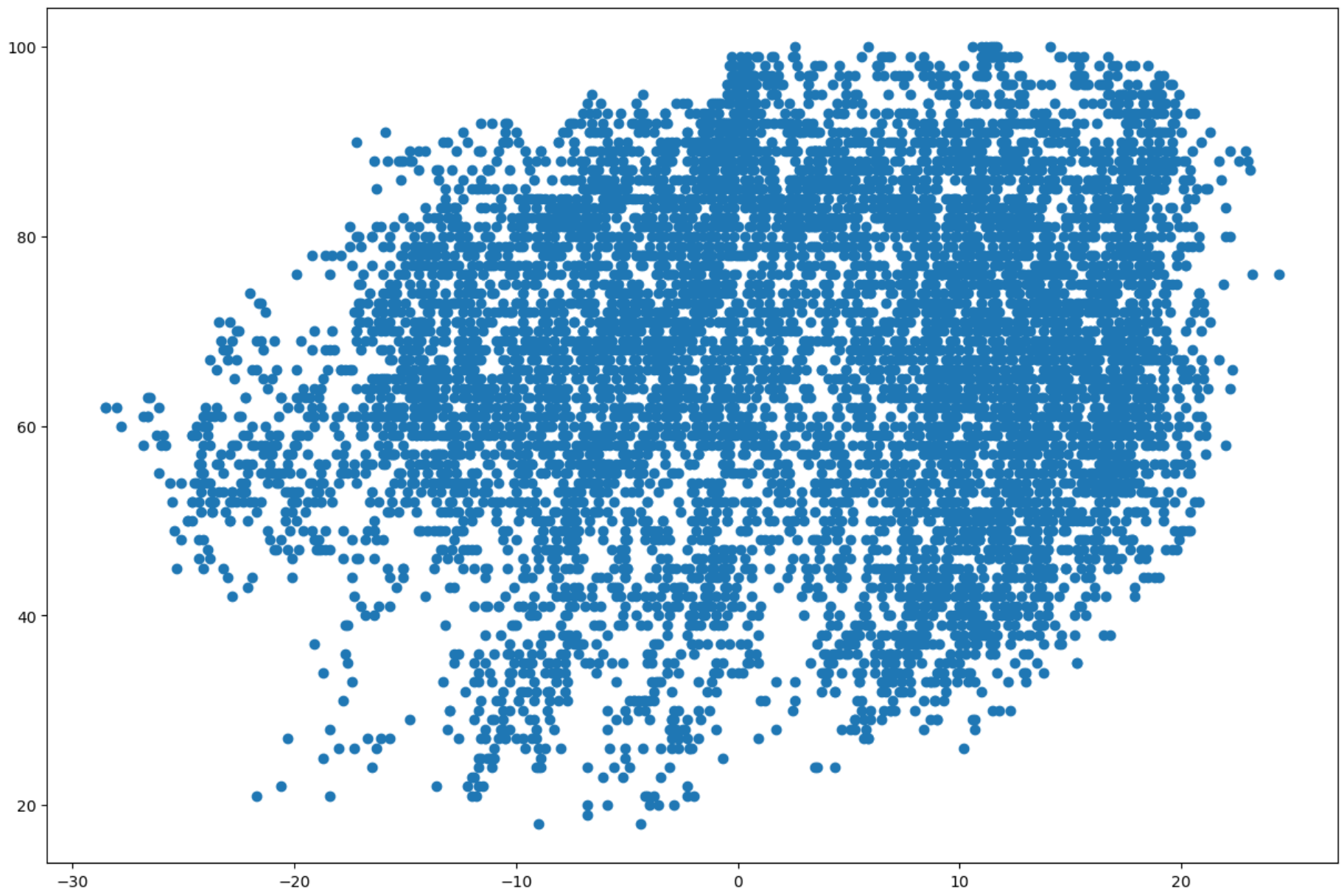
Out[11]:
```
<BarContainer object of 8784 artists>
```


Moderate Snow,Blowing Snow
Snow Showers,Fog
Rain,Snow,Ice Pellets
Rain,Snow Grains
Drizzle,Snow,Fog
Rain,Ice Pellets
Drizzle,Snow
Drizzle,Ice Pellets,Fog
Freezing Rain,Ice Pellets,Fog
Moderate Rain,Fog
Rain,Snow,Fog
Snow Pellets
Rain Showers,Snow Showers
Rain Showers,Fog
Thunderstorms,Moderate Rain Showers,Fog
Thunderstorms,Rain,Fog
Thunderstorms
Thunderstorms,Rain Showers,Fog
Thunderstorms,Heavy Rain Showers
Thunderstorms,Rain Showers
Thunderstorms,Rain
Rain,Haze
Snow,Ice Pellets
Snow,Fog
Snow,Haze
Freezing Rain,Haze
Freezing Drizzle,Haze
Drizzle
Drizzle,Fog
Rain,Fog
Haze
Freezing Fog
Snow,Blowing Snow
Freezing Rain,Snow Grains
Freezing Drizzle,Snow
Moderate Snow
Rain,Snow
Freezing Drizzle
Freezing Rain
Freezing Rain,Fog
Clear
Snow
Snow Showers
Mainly Clear
Rain Showers
Rain
Cloudy
Mostly Cloudy
Freezing Drizzle,Fog
Fog
0
20
40
60
80
100

Dew Point Vs Relative Humidity

In [12]:
```
plt.figure(figsize=(15,10))
plt.scatter(df["Dew Point Temp_C"],df["Rel Hum_%"])
```

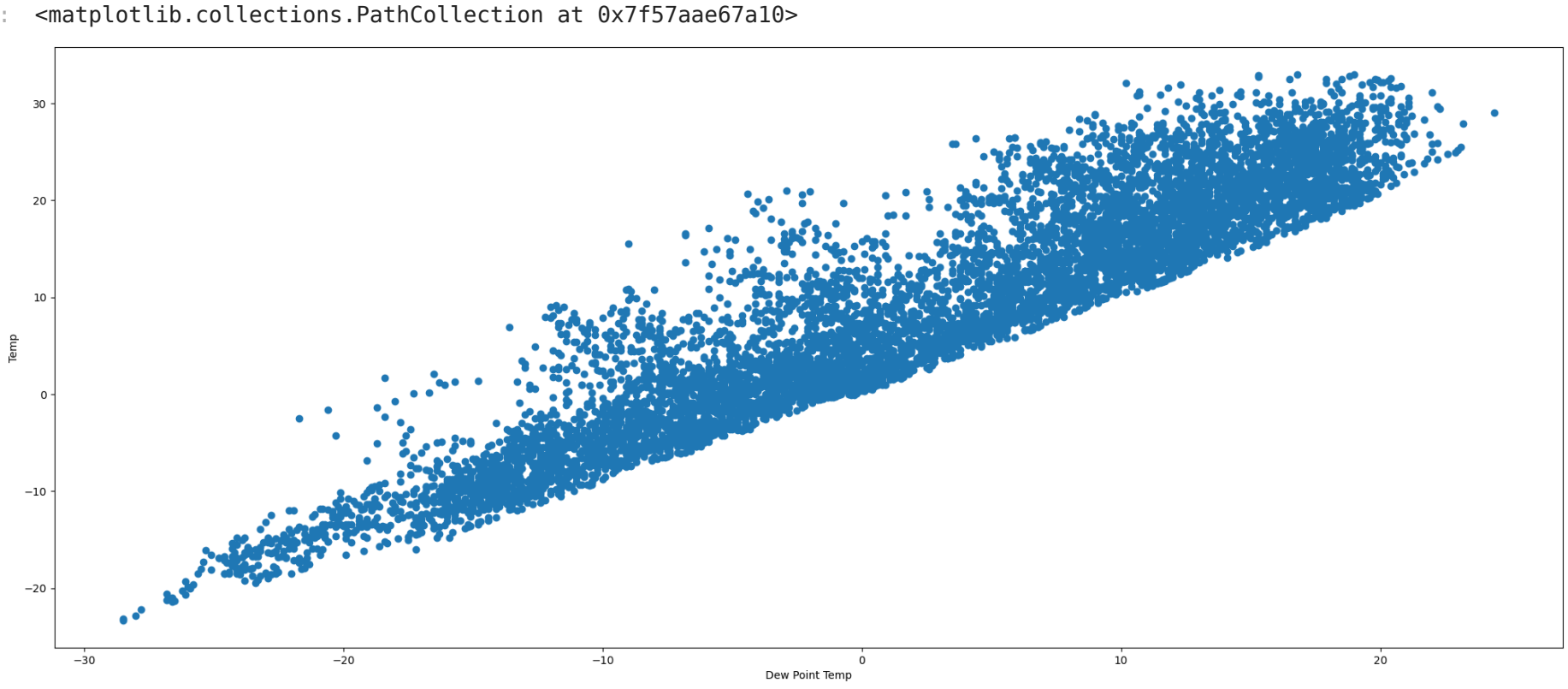
Out[12]: <matplotlib.collections.PathCollection at 0x7f57aaf4da90>



Dew Point Vs Temp_C

In [13]:
```
plt.figure(figsize=(25,10))
plt.xlabel("Dew Point Temp")
plt.ylabel("Temp")
plt.scatter(df["Dew Point Temp_C"],df["Temp_C"])
```

Out[13]: <matplotlib.collections.PathCollection at 0x7f57aae67a10>



Analyzing Data When Temprature Was -10 to 0

In [14]:
```
result_df = df.loc[df["Temp_C"]> -10]
result_df2 = result_df.loc[result_df["Temp_C"]<=0]
```

In [15]:
```
result_df2
```

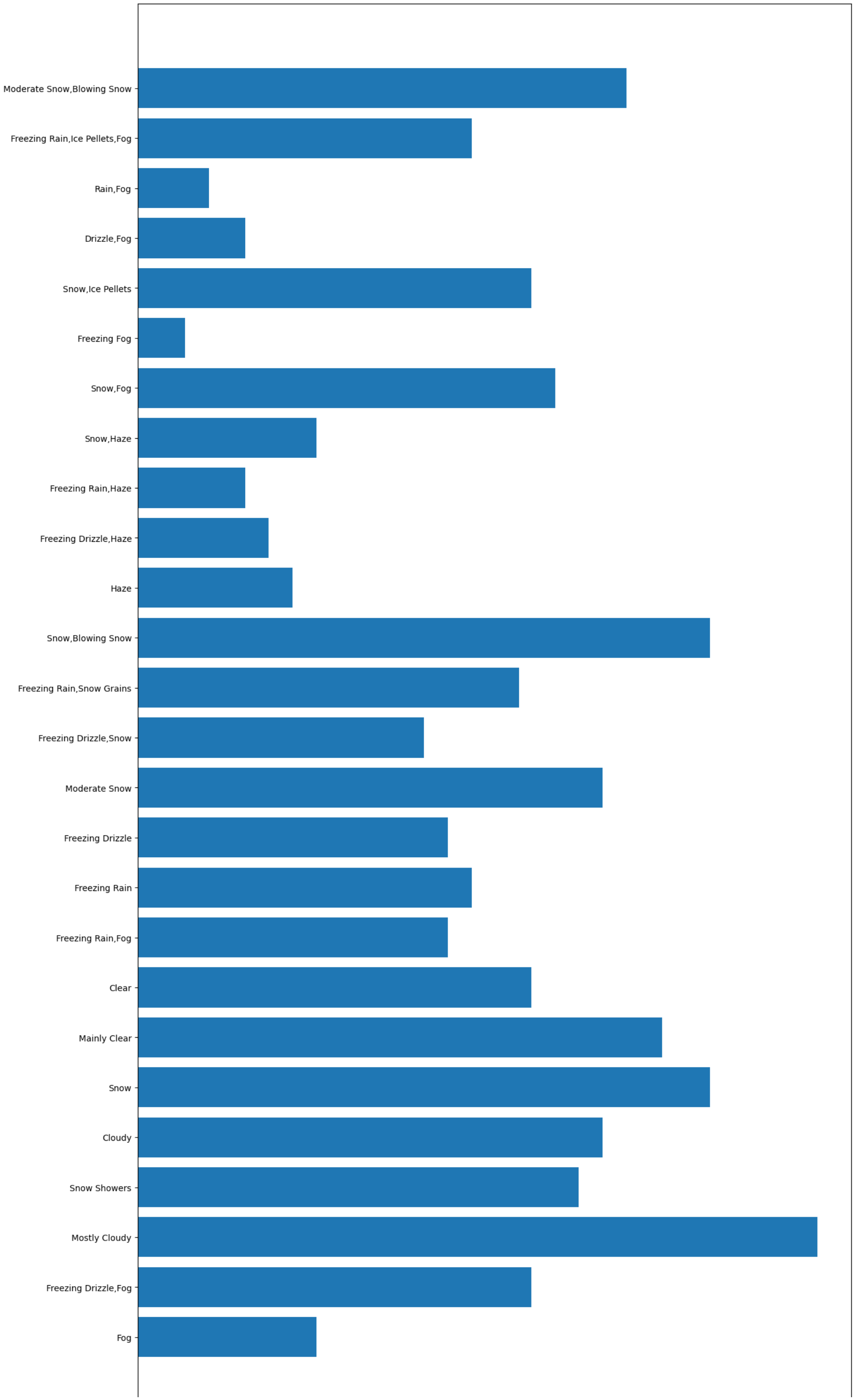
Out[15]:

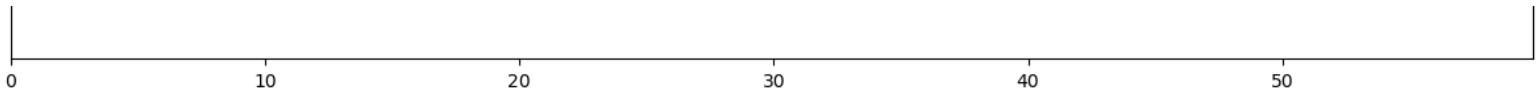
	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog
3	1/1/2012 3:00	-1.5	-3.2	88	6	4.0	101.27	Freezing Drizzle,Fog
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog
...	...	...	...	...	...	...	...	...
8777	12/31/2012 17:00	-1.1	-3.3	85	19	9.7	100.30	Snow
8778	12/31/2012 18:00	-1.3	-3.1	88	17	9.7	100.19	Snow
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Snow
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Snow
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Snow

1675 rows × 8 columns

In [16]:
```
plt.figure(figsize=(15,30))
plt.barh(result_df2["Weather"],result_df2["Wind Speed_km/h"])
```

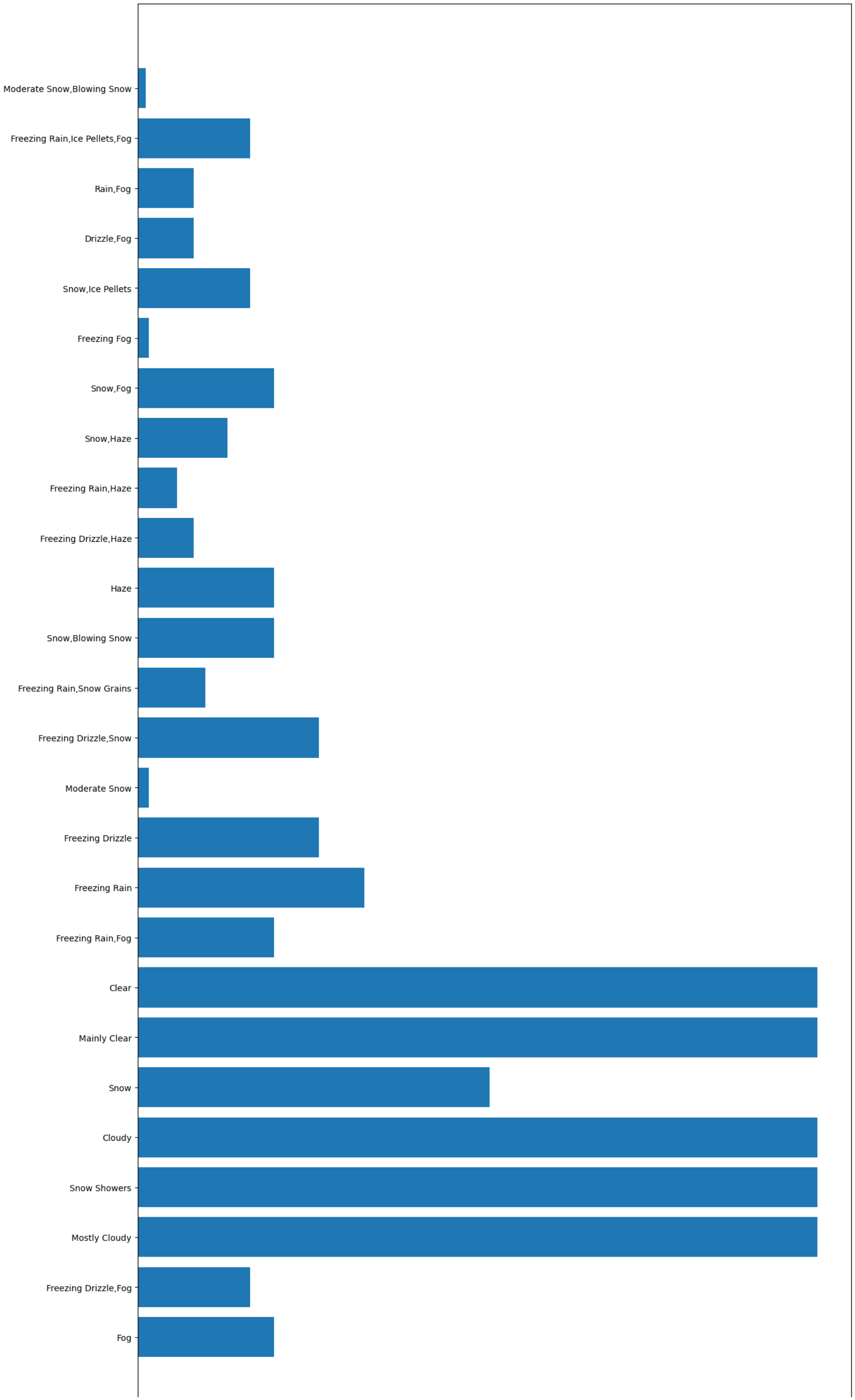
Out[16]: <BarContainer object of 1675 artists>


Moderate Snow,Blowing Snow
Freezing Rain,Ice Pellets,Fog
Rain,Fog
Drizzle,Fog
Snow,Ice Pellets
Freezing Fog
Snow,Fog
Snow,Haze
Freezing Rain,Haze
Freezing Drizzle,Haze
Haze
Snow,Blowing Snow
Freezing Rain,Snow Grains
Freezing Drizzle,Snow
Moderate Snow
Freezing Drizzle
Freezing Rain
Freezing Rain,Fog
Clear
Mainly Clear
Snow
Cloudy
Snow Showers
Mostly Cloudy
Freezing Drizzle,Fog
Fog



In [17]:
```
plt.figure(figsize=(15,30))
plt.barh(result_df2["Weather"],result_df2["Visibility_km"])
```

Out[17]: <BarContainer object of 1675 artists>

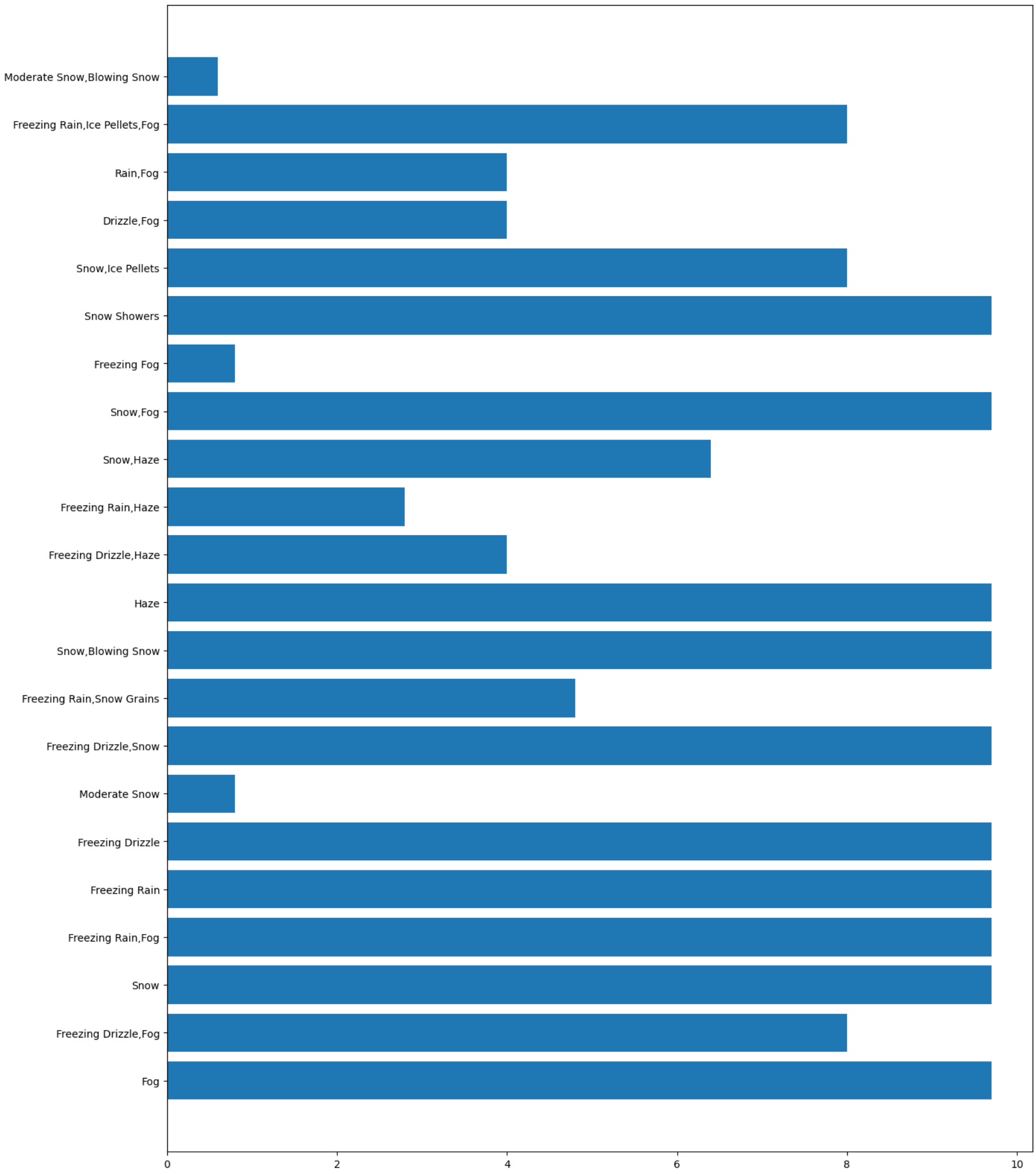

Moderate Snow,Blowing Snow
Freezing Rain,Ice Pellets,Fog
Rain,Fog
Drizzle,Fog
Snow,Ice Pellets
Freezing Fog
Snow,Fog
Snow,Haze
Freezing Rain,Haze
Freezing Drizzle,Haze
Haze
Snow,Blowing Snow
Freezing Rain,Snow Grains
Freezing Drizzle,Snow
Moderate Snow
Freezing Drizzle
Freezing Rain
Freezing Rain,Fog
Clear
Mainly Clear
Snow
Cloudy
Snow Showers
Mostly Cloudy
Freezing Drizzle,Fog
Fog

Visibility is said to be good when a person can see farther than about 10 kilometers.

In [18]:
```
result_df2_1 = result_df2.loc[result_df2["Visibility_km"]<=10]

plt.figure(figsize=(15,20))
plt.barh(result_df2_1["Weather"],result_df2_1["Visibility_km"])
```

Out[18]: <BarContainer object of 325 artists>

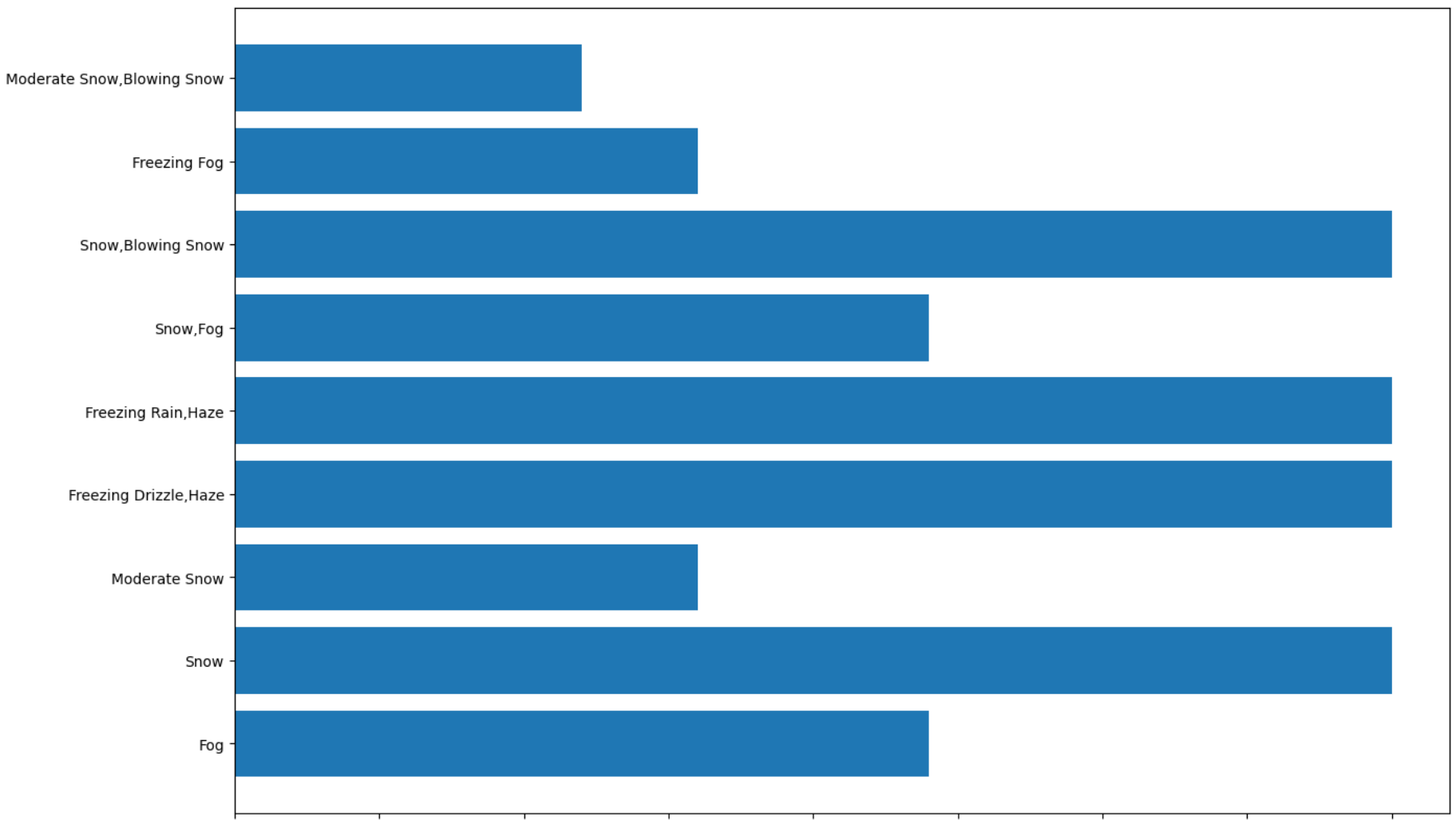


In these Weathers It was barely Visible Specially when it was Moderate Snow and Freezing Fog and Haze

In [19]:
```
result_df2_2 = result_df2.loc[result_df2["Visibility_km"]<=2]

plt.figure(figsize=(15,10))
plt.barh(result_df2_2["Weather"],result_df2_2["Visibility_km"])
```

Out[19]: <BarContainer object of 80 artists>



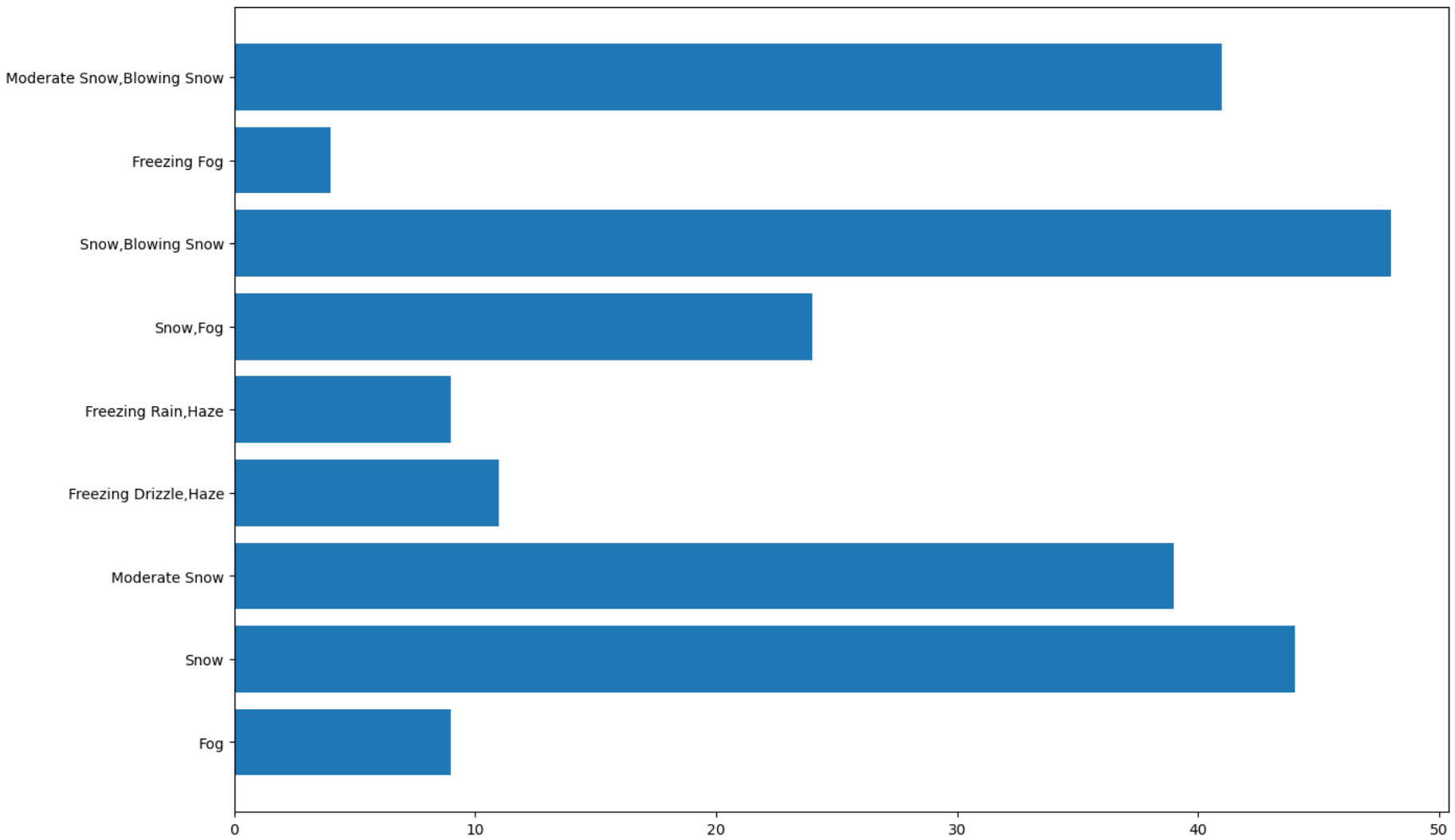
Barely Visble in these conditions Like Blowing Snow Freezing Fog and Moderate snow. These have visiblity less than 750 meters

In [ ]:

Looking At Wind Speed when Visiblity was too low

In [20]:
```
plt.figure(figsize=(15,10))
plt.barh(result_df2_2["Weather"],result_df2_2["Wind Speed_km/h"])
```

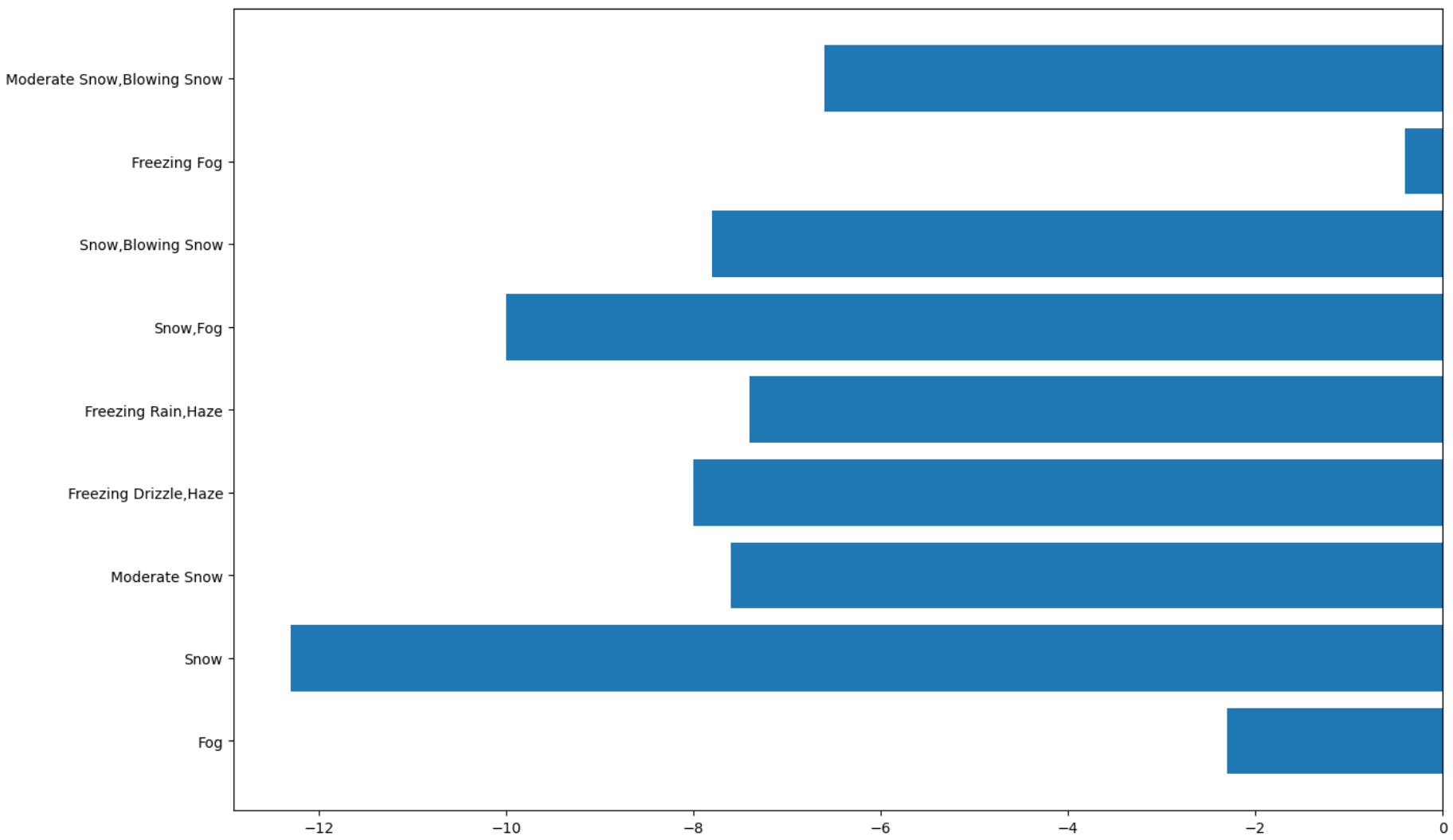
Out[20]: <BarContainer object of 80 artists>



When Weather is Blowing Snow/Moderate Snow Speed was above 40km/hr

In [21]:
```
plt.figure(figsize=(15,10))
plt.barh(result_df2_2["Weather"],result_df2_2["Dew Point Temp_C"])
```

Out[21]: <BarContainer object of 80 artists>



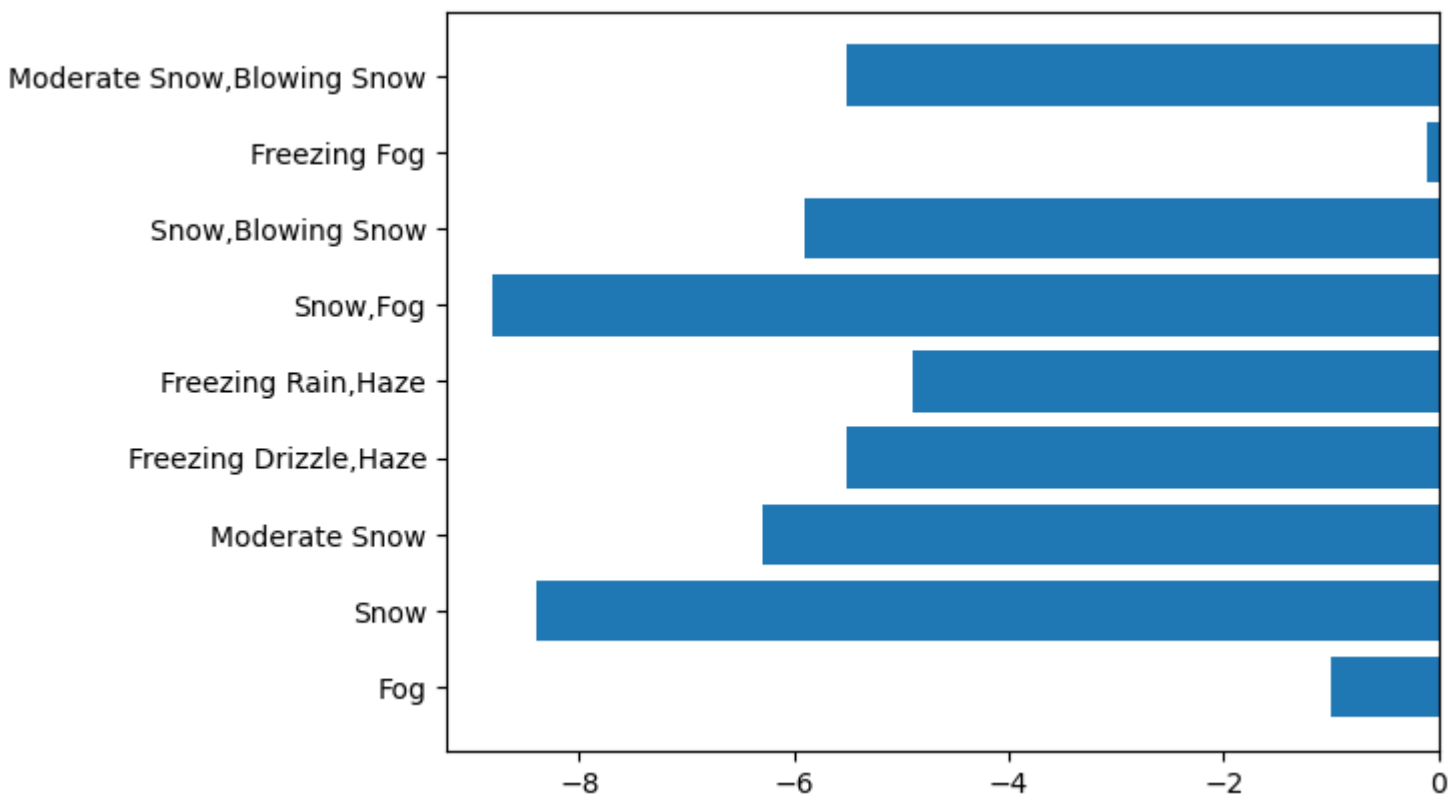
In Weather Conditions Like Moderate Snow,Blowing Snow, Frrexing Rain, Haze and Normal Snow the Dew point went bellow -12

Dew point temperature is a measure of atmospheric moisture. It is the temperature to which air must be cooled in order to reach saturation (assuming air pressure and moisture content are constant).

In [ ]:

In [22]:
```
plt.barh(result_df2_2["Weather"],result_df2_2["Temp_C"])
```

Out[22]: <BarContainer object of 80 artists>

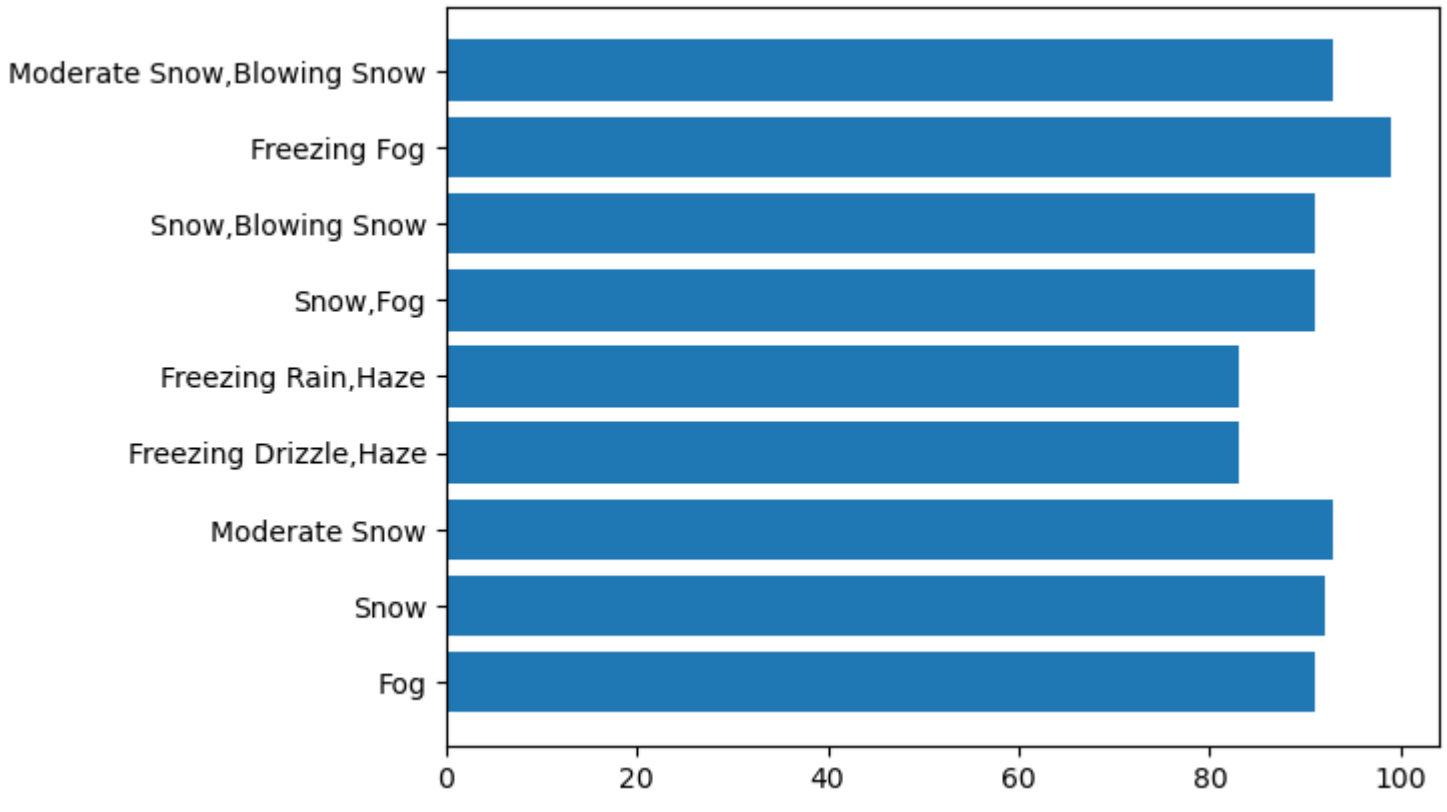


In These weather conditions where Visiblity is so low and Humidity is High the temprature goes bellow -5 Degree Celcius

In [ ]:

In [23]:
```
plt.barh(result_df2_2["Weather"],result_df2_2["Rel Hum_%"])
```

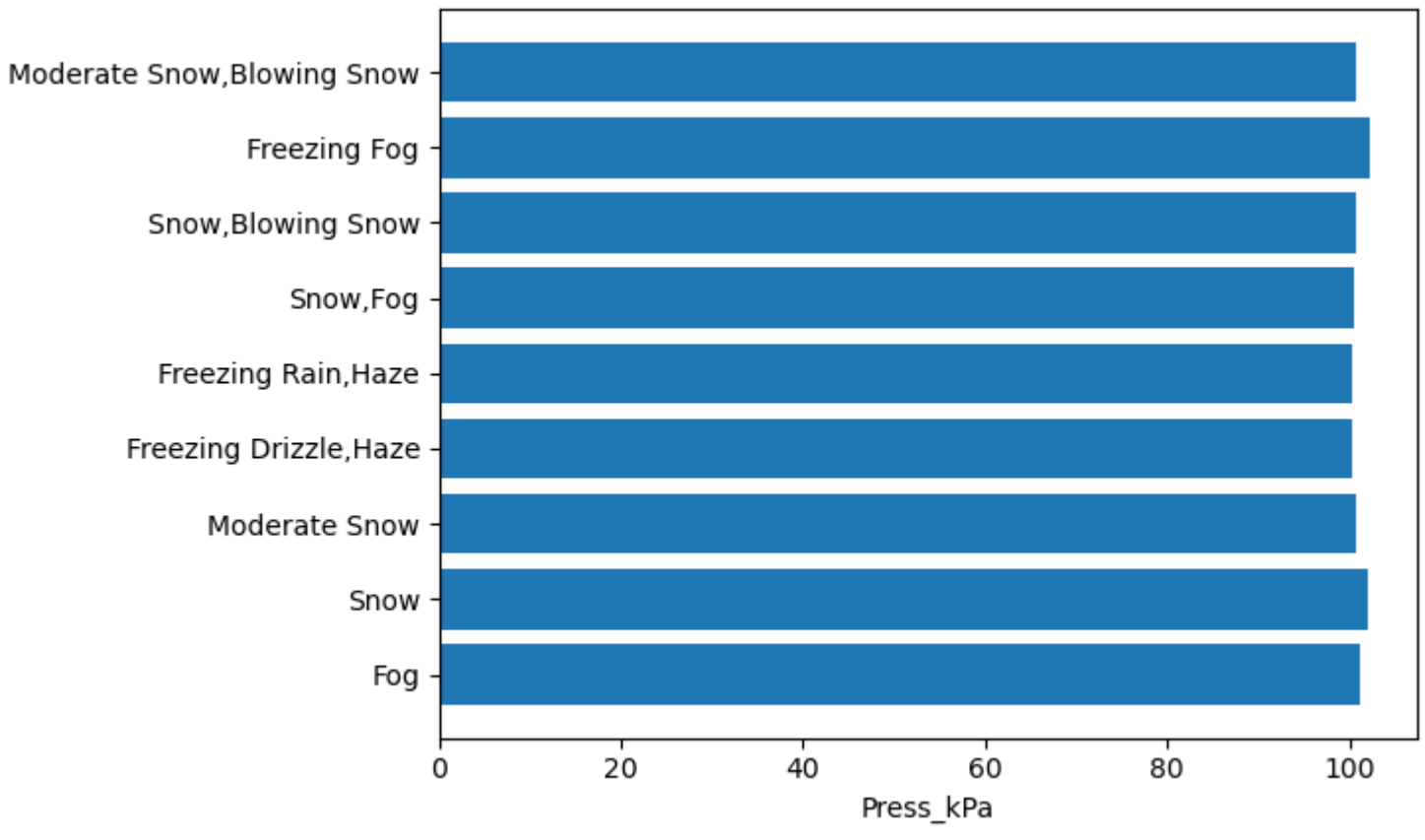
Out[23]: <BarContainer object of 80 artists>



In All Of the Weather Conditions Humidity is above 80

In [24]:
```
plt.xlabel("Press_kPa")
plt.barh(result_df2_2["Weather"],result_df2_2["Press_kPa"])
```

Out[24]: <BarContainer object of 80 artists>



In [ ]:
```

```

Now That We analyzed when the Visiblity was low, Lets Analyze the Temprature, Wind Speed, Humidity When Visbility is High

In [25]:
```
result_df3 = result_df2.loc[result_df2["Visibility_km"]>=20]
result_df3
```

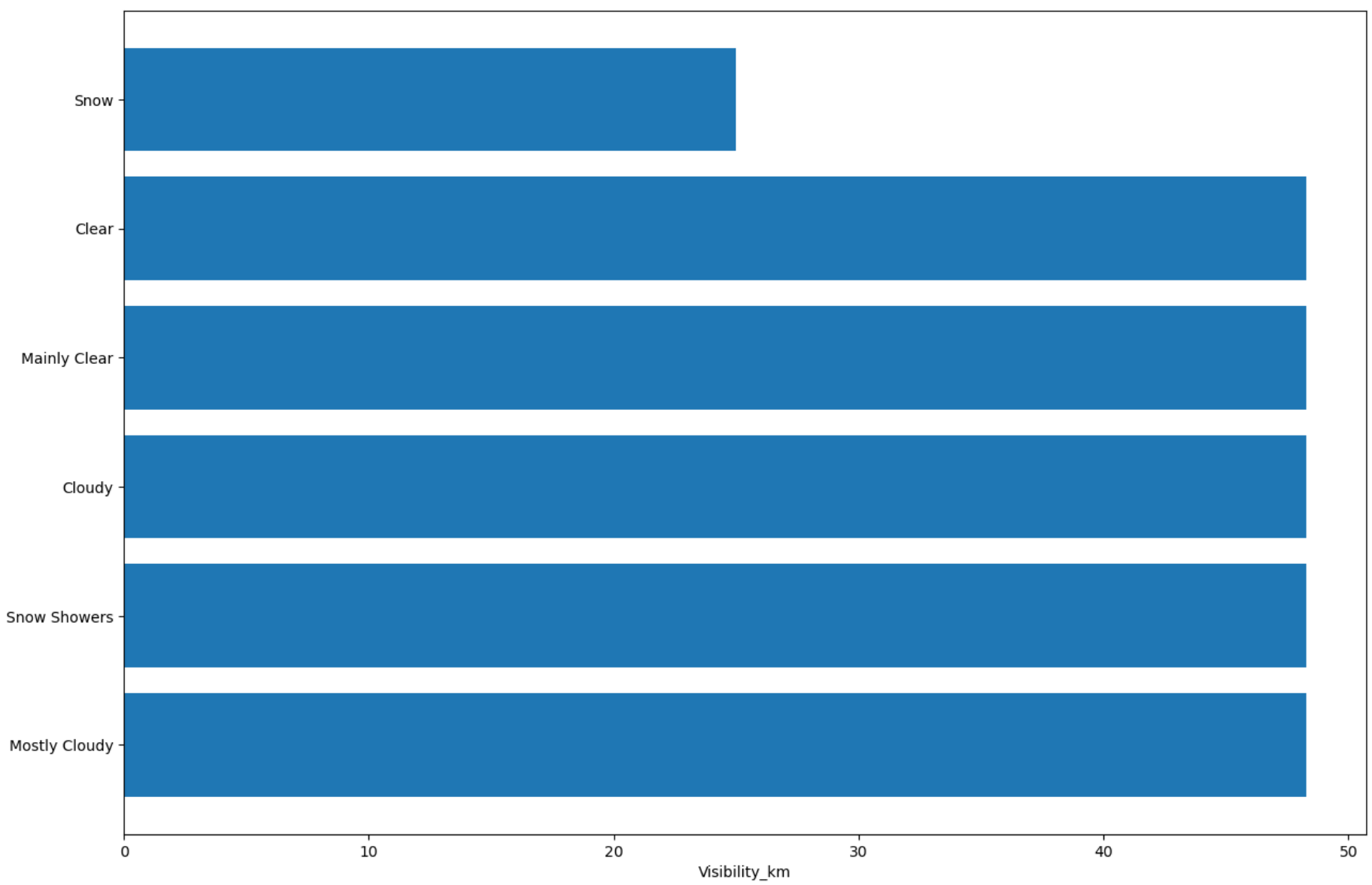
Out[25]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
39	1/2/2012 15:00	0.0	-7.0	59	33	24.1	99.41	Mostly Cloudy
40	1/2/2012 16:00	-0.7	-8.7	55	24	24.1	99.50	Mostly Cloudy
41	1/2/2012 17:00	-2.1	-9.5	57	22	25.0	99.66	Snow Showers
42	1/2/2012 18:00	-4.1	-11.4	57	28	25.0	99.86	Mostly Cloudy
43	1/2/2012 19:00	-4.8	-12.1	57	24	25.0	100.00	Mostly Cloudy
...	...	...	...	...	...	...	...	...
8708	12/28/2012 20:00	-9.8	-12.6	80	20	25.0	101.35	Mainly Clear
8709	12/28/2012 21:00	-9.8	-12.5	81	19	25.0	101.36	Mainly Clear
8710	12/28/2012 22:00	-9.9	-12.5	81	17	25.0	101.38	Mainly Clear
8743	12/30/2012 7:00	-9.9	-13.2	77	19	25.0	100.48	Mostly Cloudy
8766	12/31/2012 6:00	-9.7	-11.7	85	4	25.0	101.23	Cloudy

1182 rows × 8 columns

In [26]:
```
plt.figure(figsize=(15,10))
plt.xlabel("Visibility_km")
plt.barh(result_df3["Weather"],result_df3["Visibility_km"])
```

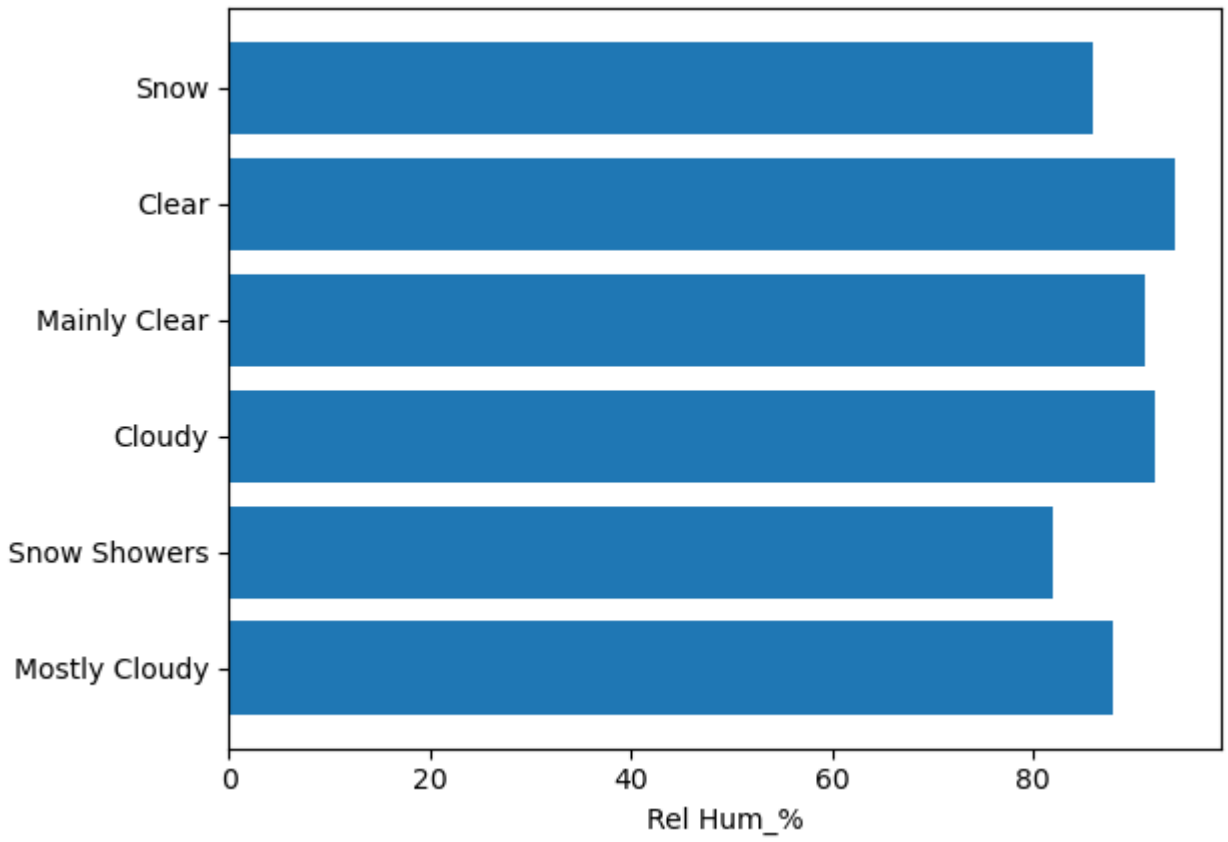
Out[26]: <BarContainer object of 1182 artists>



Lets Look At the Temperatures and Humidity

In [27]:
```
plt.xlabel("Rel Hum_%")
plt.barh(result_df3["Weather"],result_df3["Rel Hum_%"])
```

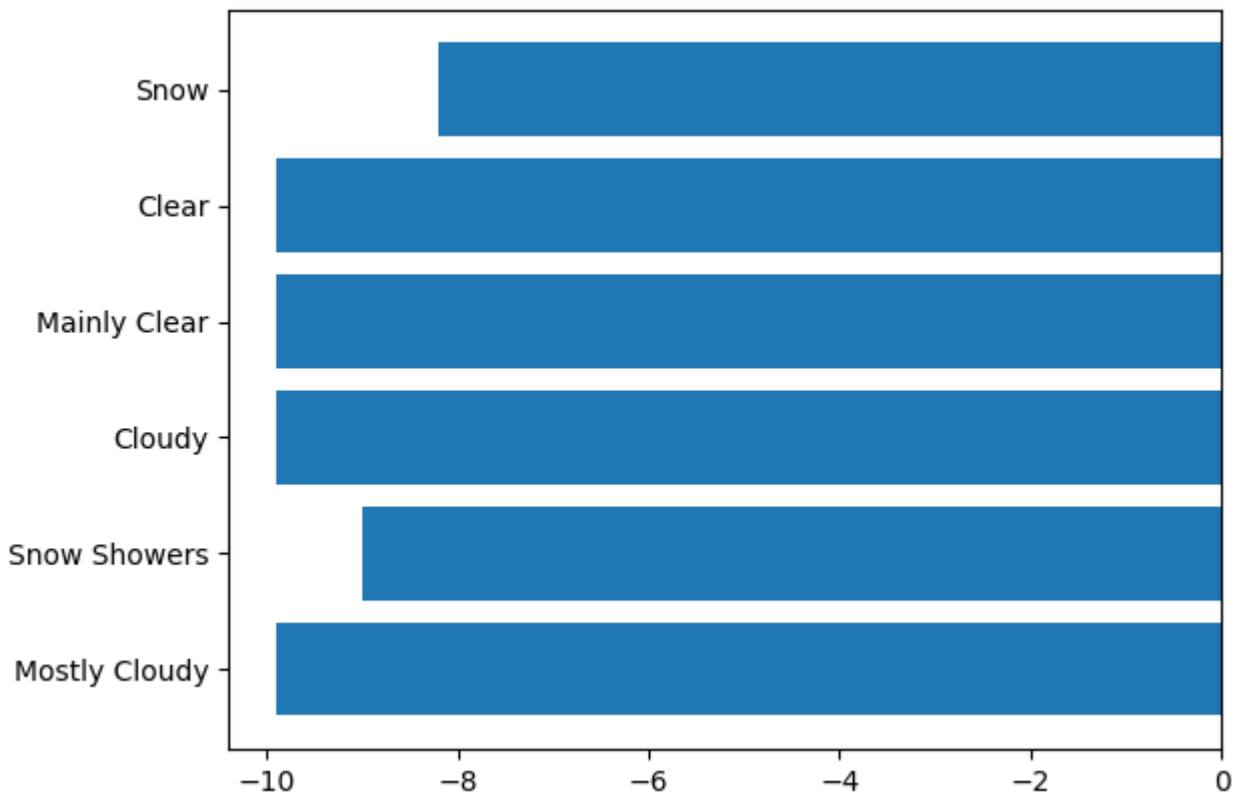
Out[27]: <BarContainer object of 1182 artists>



Though The Visiblity is Better Humidity Is still high

In [28]:
```
plt.barh(result_df3["Weather"],result_df3["Temp_C"])
```

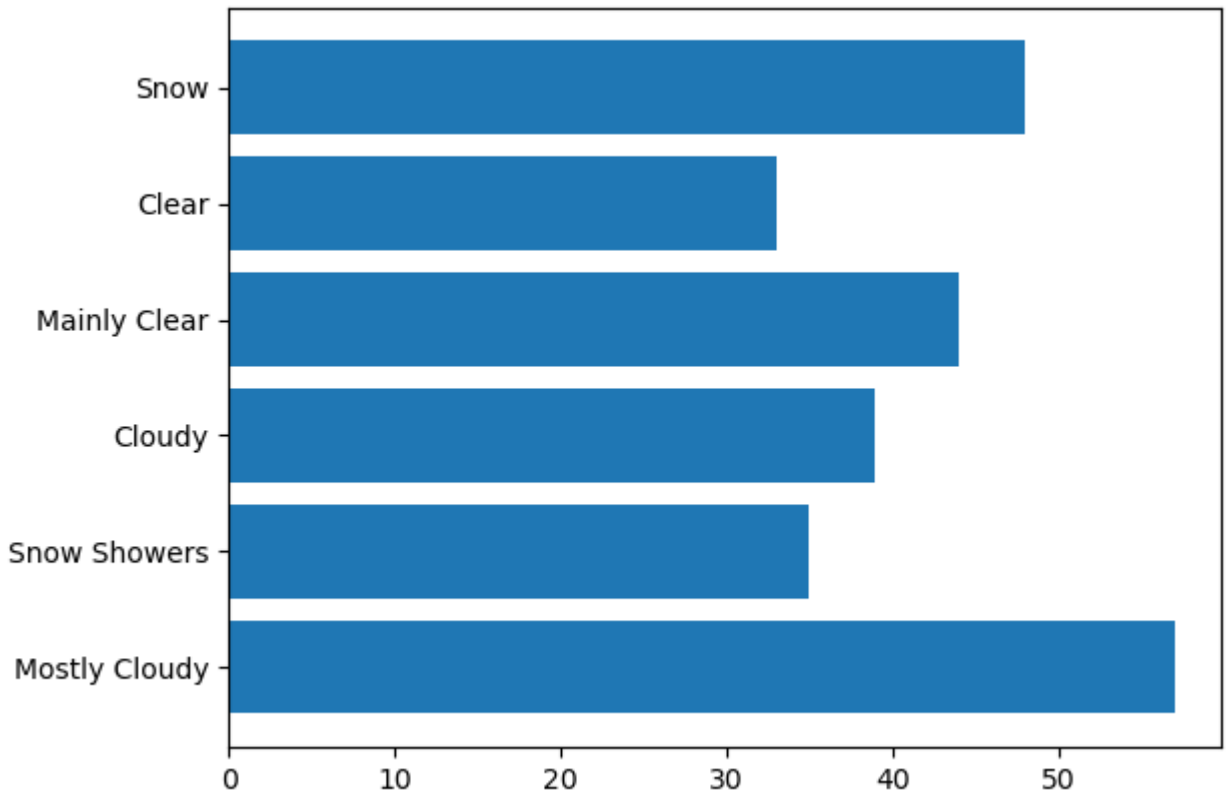
Out[28]: <BarContainer object of 1182 artists>



Temprature Is also Quiet cold... Exeeding -8 Degree Celcius

In [29]:
```
plt.barh(result_df3["Weather"],result_df3["Wind Speed_km/h"])
```

Out[29]: <BarContainer object of 1182 artists>



Wind Speed Seems Better,,, Hovering between 30-40 km

Lets See Which season it was when Visiblity was High and Temps were within -10 degree Celcius

1,2,3,4,10,11,12

Jan,Feb,March,April,October,November And December 2012 had high visiblity

Jan,Feb,March,April Had Low Visiblity In 2012

In [ ]:

Analyzing the data when Temprature was 0 - 20 degree Celcius

In [32]:
```
result_df4 = df.loc[df["Temp_C"]> 0]
result_df5 = result_df4.loc[result_df4["Temp_C"]<=20]
```

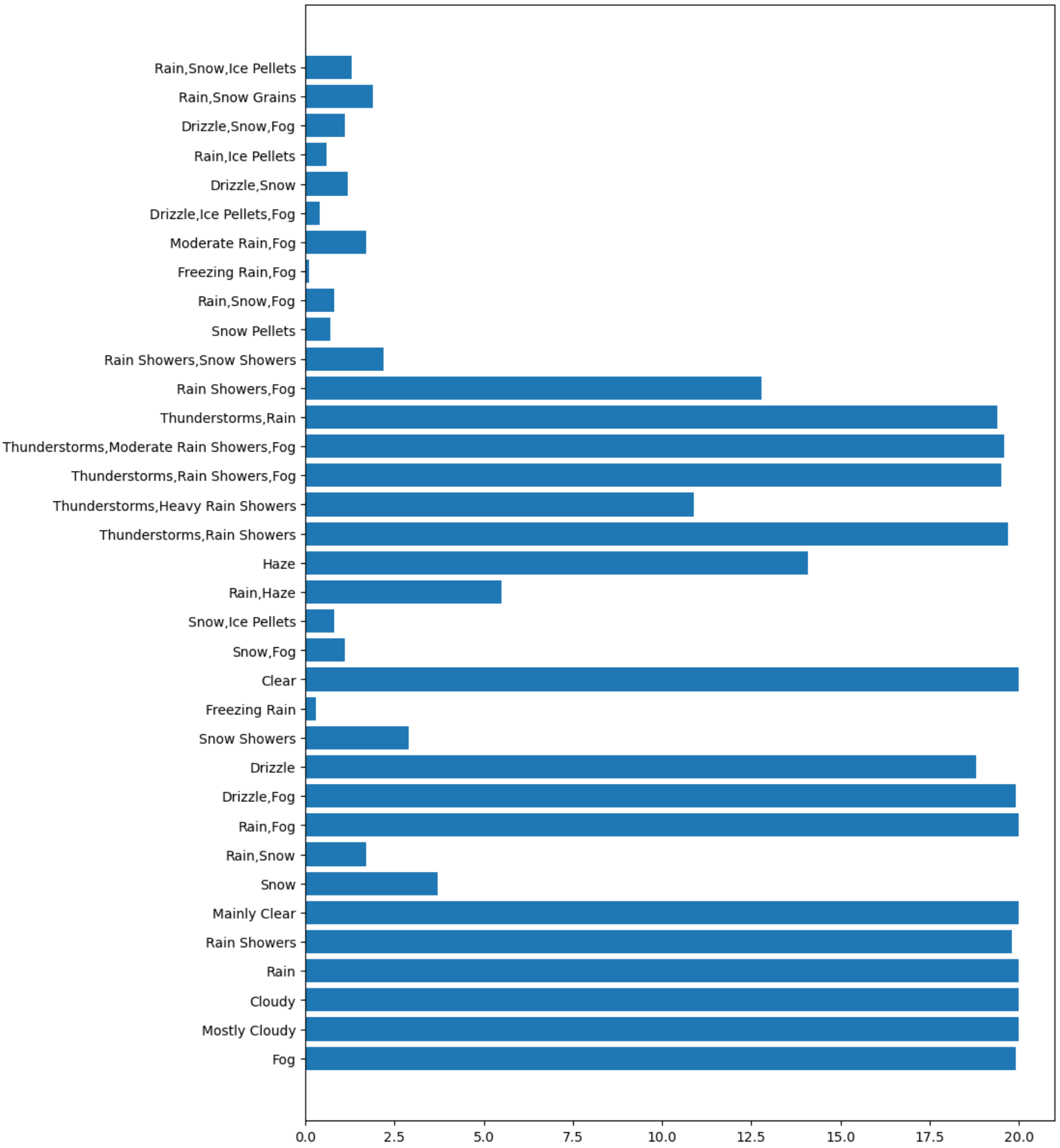
In [33]:
```
result_df5.head()
```

Out[33]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
13	1/1/2012 13:00	0.2	-1.7	87	13	4.8	100.58	Fog
14	1/1/2012 14:00	0.8	-1.1	87	20	4.8	100.31	Fog
15	1/1/2012 15:00	1.8	-0.4	85	22	6.4	100.07	Fog
16	1/1/2012 16:00	2.6	-0.2	82	13	12.9	99.93	Mostly Cloudy
17	1/1/2012 17:00	3.0	0.0	81	13	16.1	99.81	Cloudy

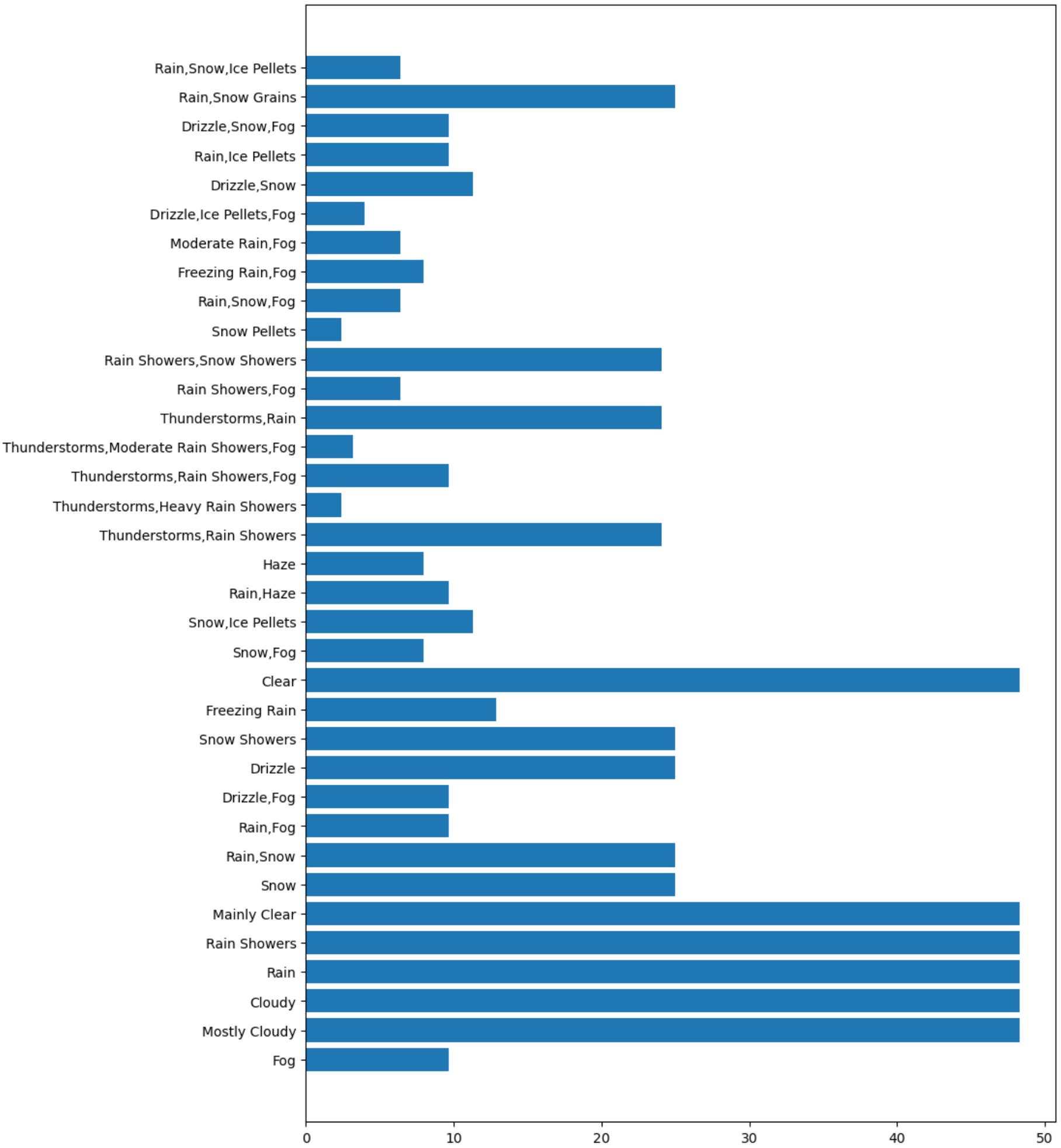
In [34]:
```
plt.figure(figsize=(10,15))
plt.barh(result_df5["Weather"],result_df5["Temp_C"])
```

Out[34]:
```
<BarContainer object of 4747 artists>
```



In [35]:
```
plt.figure(figsize=(10,15))
plt.barh(result_df5["Weather"],result_df5["Visibility_km"])
```

Out[35]: <BarContainer object of 4747 artists>



Analyzing the Data when Visiblity was more than 20km

In [36]:
```
result_df5_1 = result_df5.loc[result_df5["Visibility_km"]>20]
```

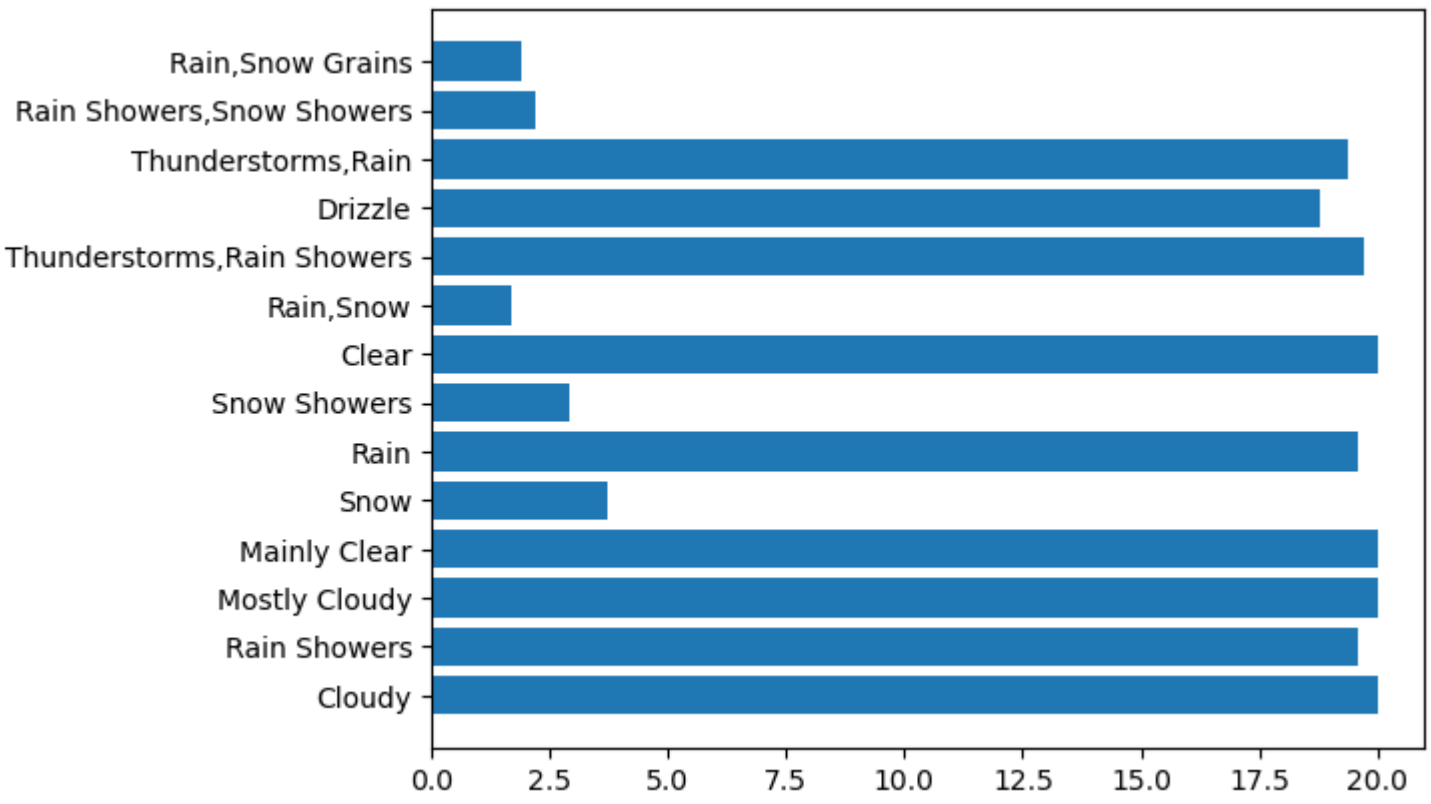
In [37]:
```
result_df5_1.head()
```

Out[37]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
20	1/1/2012 20:00	3.2	1.3	87	19	25.0	99.50	Cloudy
21	1/1/2012 21:00	4.0	1.7	85	20	25.0	99.39	Cloudy
23	1/1/2012 23:00	5.3	2.0	79	30	25.0	99.31	Cloudy
24	1/2/2012 0:00	5.2	1.5	77	35	25.0	99.26	Rain Showers
25	1/2/2012 1:00	4.6	0.0	72	39	25.0	99.26	Cloudy

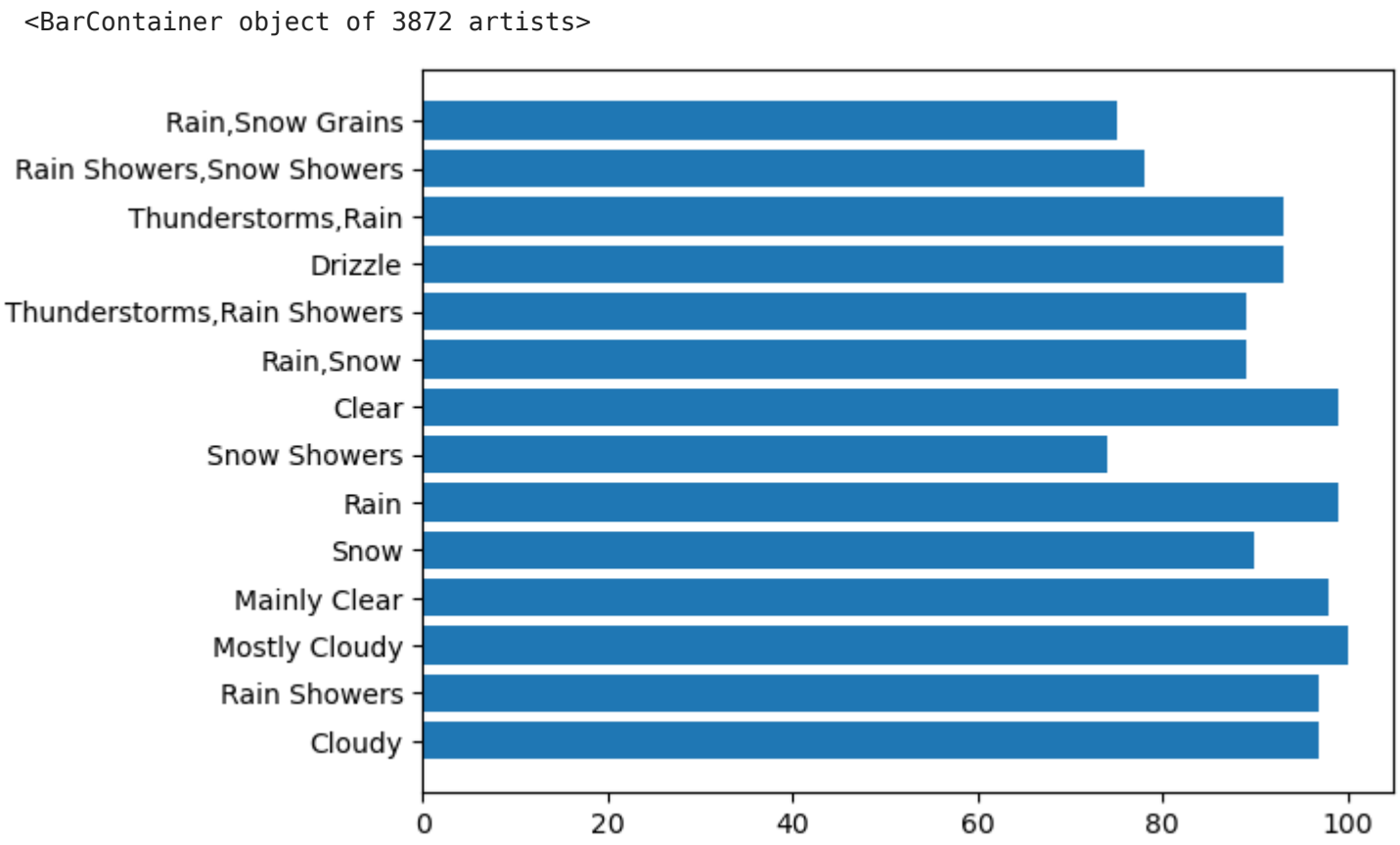
In [38]:
```
plt.barh(result_df5_1["Weather"],result_df5_1["Temp_C"])
```

Out[38]:
```
<BarContainer object of 3872 artists>
```



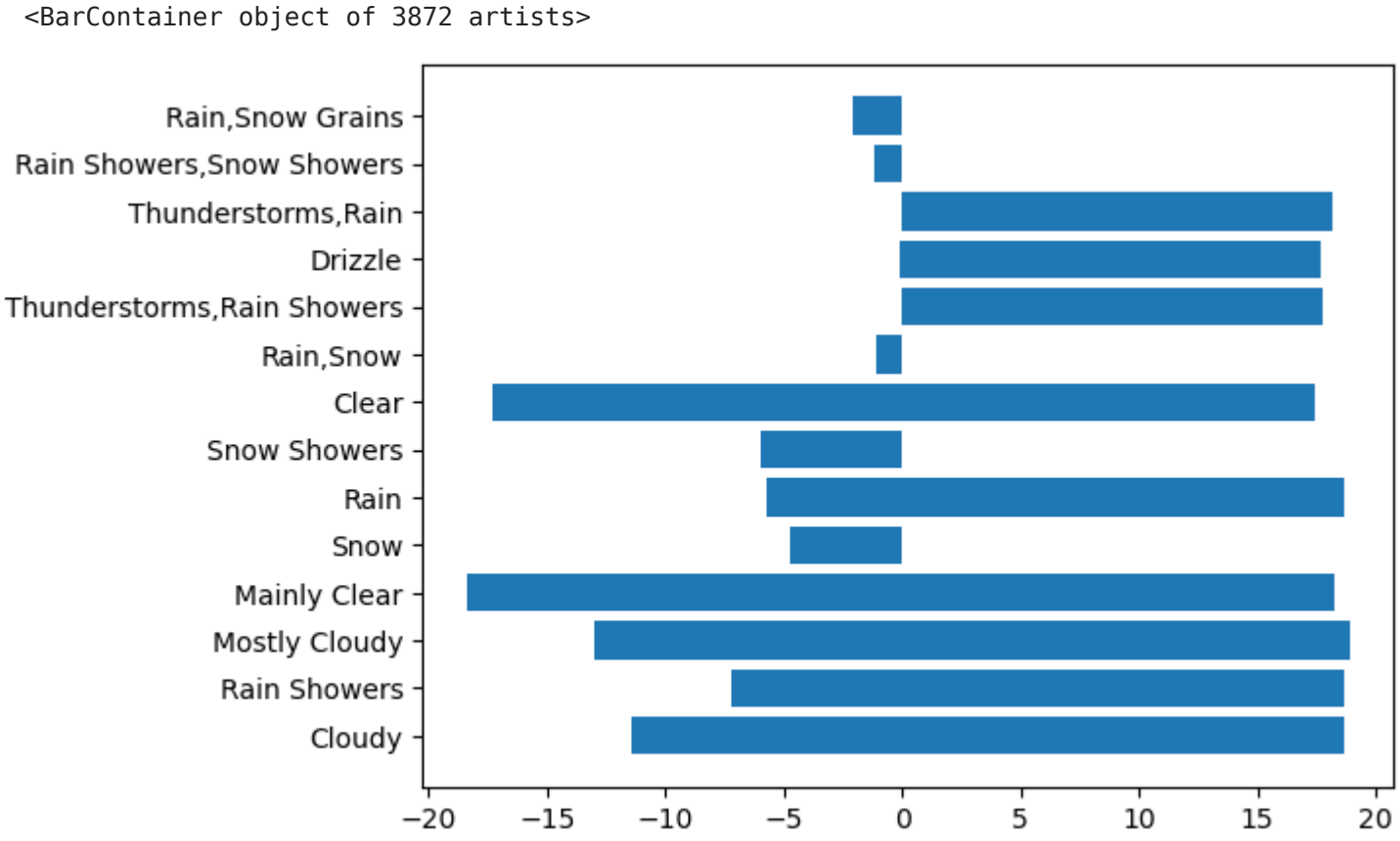
In [39]:
```
plt.barh(result_df5_1["Weather"],result_df5_1["Rel Hum_%"])
```

Out[39]: <BarContainer object of 3872 artists>



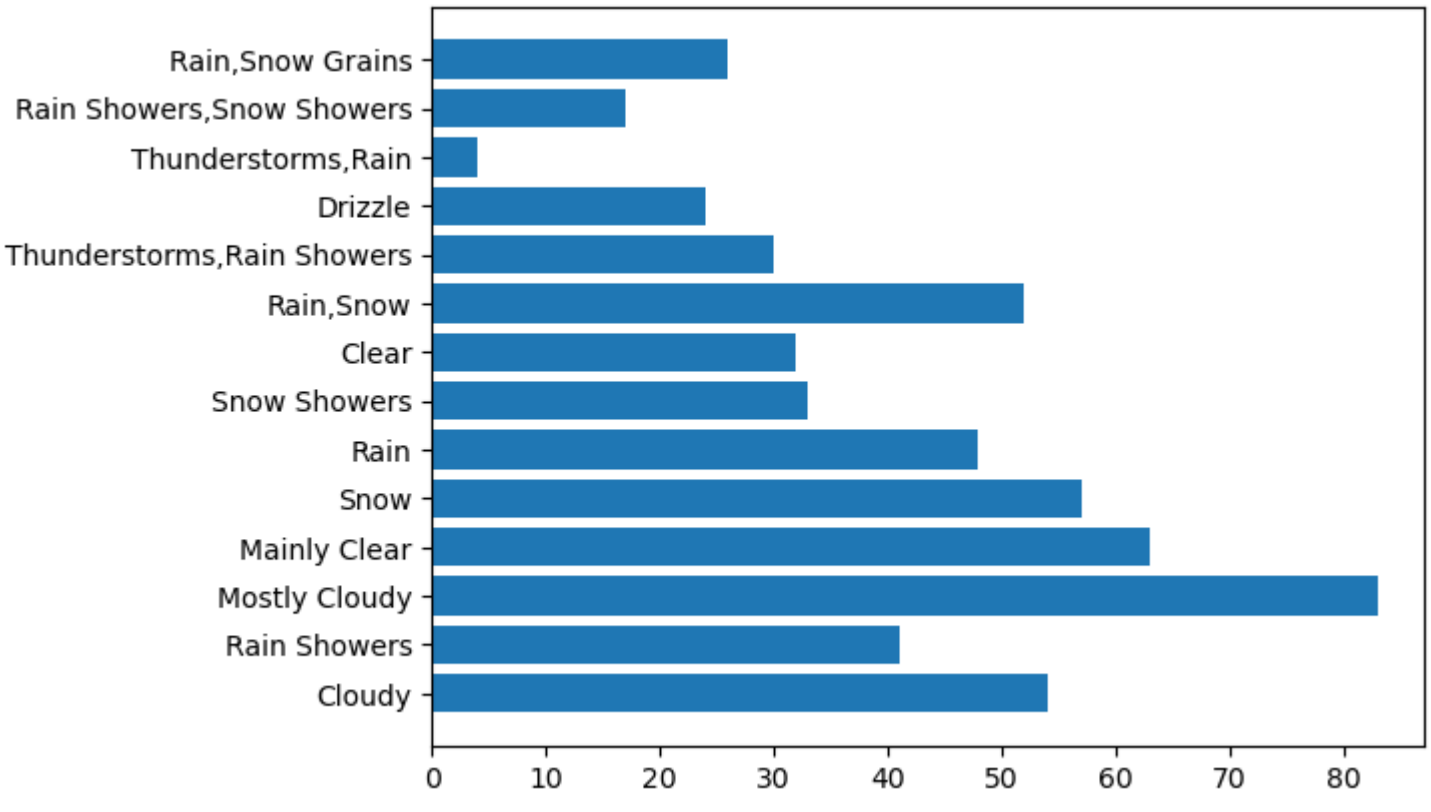
In [40]:
```
plt.barh(result_df5_1["Weather"],result_df5_1["Dew Point Temp_C"])
```

Out[40]: <BarContainer object of 3872 artists>



In [41]:
```
plt.barh(result_df5_1["Weather"],result_df5_1["Wind Speed_km/h"])
```

Out[41]: <BarContainer object of 3872 artists>



In [ ]:

Classifying Data

In [185…
```
X1 = df[["Temp_C","Dew Point Temp_C","Rel Hum_%","Wind Speed_km/h","Visibility_km","Press_kPa"]]
Y1 = df["Weather"]

from sklearn.model_selection import train_test_split
```

In [195…
```
x_train,x_test,y_train,y_test = train_test_split(X1,Y1,test_size=0.3,random_state=42,shuffle=True)
```

In [200…
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
```

In [248…
```
model = DecisionTreeClassifier(criterion="gini")
model.fit(x_train,y_train)
```

Out[248…
```
DecisionTreeClassifier()
```

In [249…
```
pred = model.predict(x_test)
pred
accuracy_score(y_test,pred)
```

Out[249… 0.4393019726858877

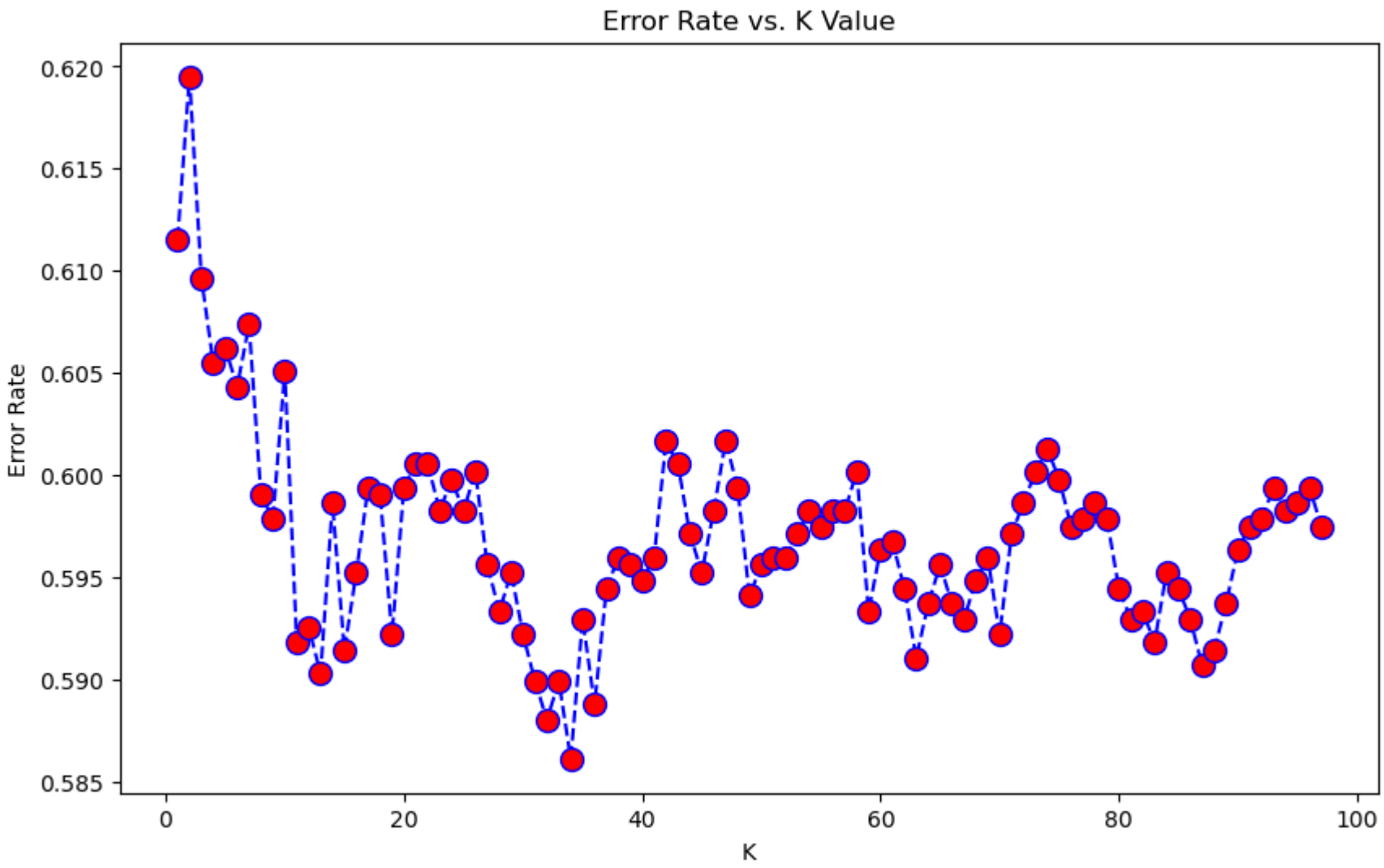
In [235…
```
import numpy as np
error_rate = []

# Will take some time
for i in range(1, 98):

    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train, y_train)
    pred_i = knn.predict(x_test)
    error_rate.append(np.mean(pred_i != y_test))

plt.figure(figsize=(10, 6))
plt.plot(range(1, 98), error_rate, color='blue',
         linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)

plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
plt.show()
```



In [229…
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report


knn = KNeighborsClassifier(n_neighbors = 34)

knn.fit(x_train, y_train)
pred = knn.predict(x_test)
accuracy_score(y_test,pred)
```

Out[229… 0.41388467374810317

Forcasting Timeseries Data

In [44]:
```
df[['Date', 'Time']] = df['Date/Time'].str.split(' ', 1, expand=True)
```

```
/tmp/ipykernel_2960/1918148834.py:1: FutureWarning: In a future version of pandas all arguments of StringMethods.spl
it except for the argument 'pat' will be keyword-only.
  df[['Date', 'Time']] = df['Date/Time'].str.split(' ', 1, expand=True)
```

In [45]:
```
df
```

Out[45]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather	Date	Time
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog	1/1/2012	0:00
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog	1/1/2012	1:00
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog	1/1/2012	2:00
3	1/1/2012 3:00	-1.5	-3.2	88	6	4.0	101.27	Freezing Drizzle,Fog	1/1/2012	3:00
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog	1/1/2012	4:00
...	...	...	...	...	...	...	...	...	...	...
8779	12/31/2012 19:00	0.1	-2.7	81	30	9.7	100.13	Snow	12/31/2012	19:00
8780	12/31/2012 20:00	0.2	-2.4	83	24	9.7	100.03	Snow	12/31/2012	20:00
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Snow	12/31/2012	21:00
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Snow	12/31/2012	22:00
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Snow	12/31/2012	23:00

8784 rows × 10 columns

In [46]:
```
resp = pd.read_csv("Data.csv")
```

In [47]:
```
resp = resp[["Date","Temp_C","Dew Point Temp_C"]]
resp
```

Out[47]:

	Date	Temp_C	Dew Point Temp_C
0	1/1/2012	1.8	-0.4
1	1/2/2012	0.0	-7.0
2	1/3/2012	-14.8	-22.2
3	1/4/2012	-10.2	-16.3
4	1/5/2012	-4.3	-12.0
...	...	...	...
361	12/27/2012	-5.0	-6.2
362	12/28/2012	-6.8	-10.3
363	12/29/2012	-8.8	-10.0
364	12/30/2012	-11.3	-15.6
365	12/31/2012	-2.3	-4.6

366 rows × 3 columns

In [48]:
```
resp.to_csv("Data.csv")
```

In [49]:
```
resp["Date"] = pd.to_datetime(resp["Date"])
```

In [50]:
```
resp.dtypes
```

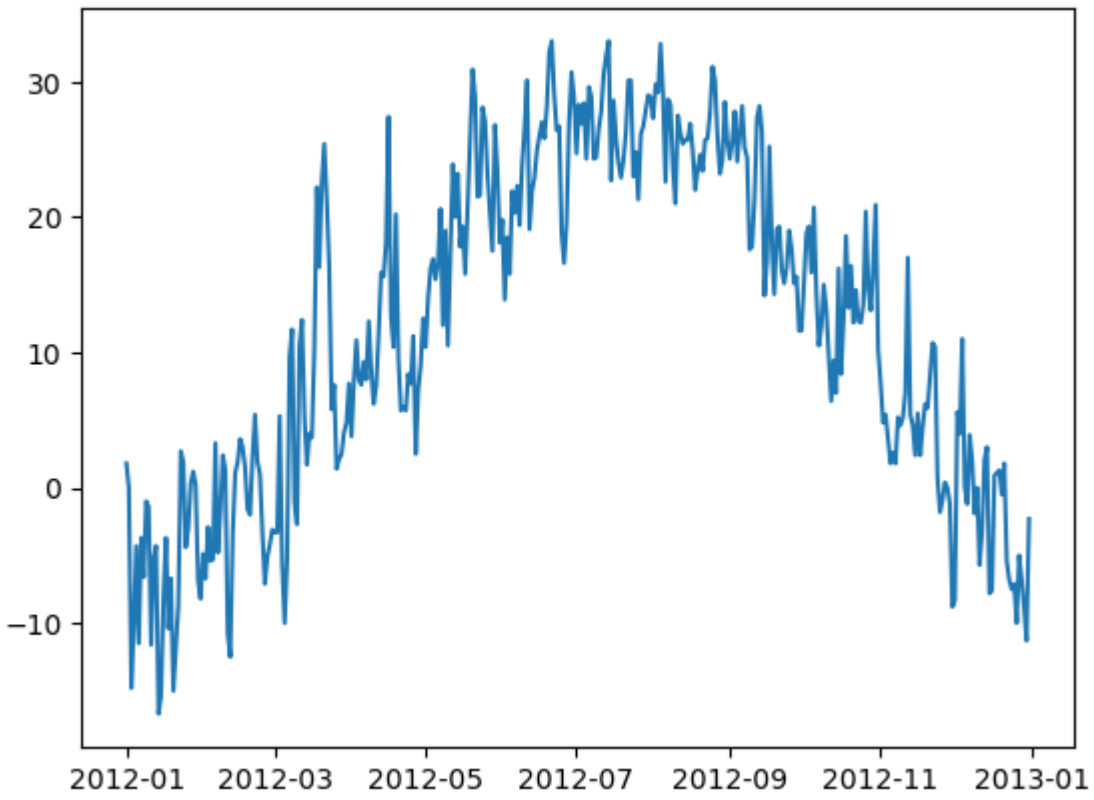
Out[50]:
```
Date                datetime64[ns]
Temp_C                     float64
Dew Point Temp_C           float64
dtype: object
```

In [51]:
```
X = resp["Date"]
Y = resp["Temp_C"]
```

In [52]:
```
from statsmodels.tsa.seasonal import seasonal_decompose
```

In [53]:
```
plt.plot(X,Y)
```

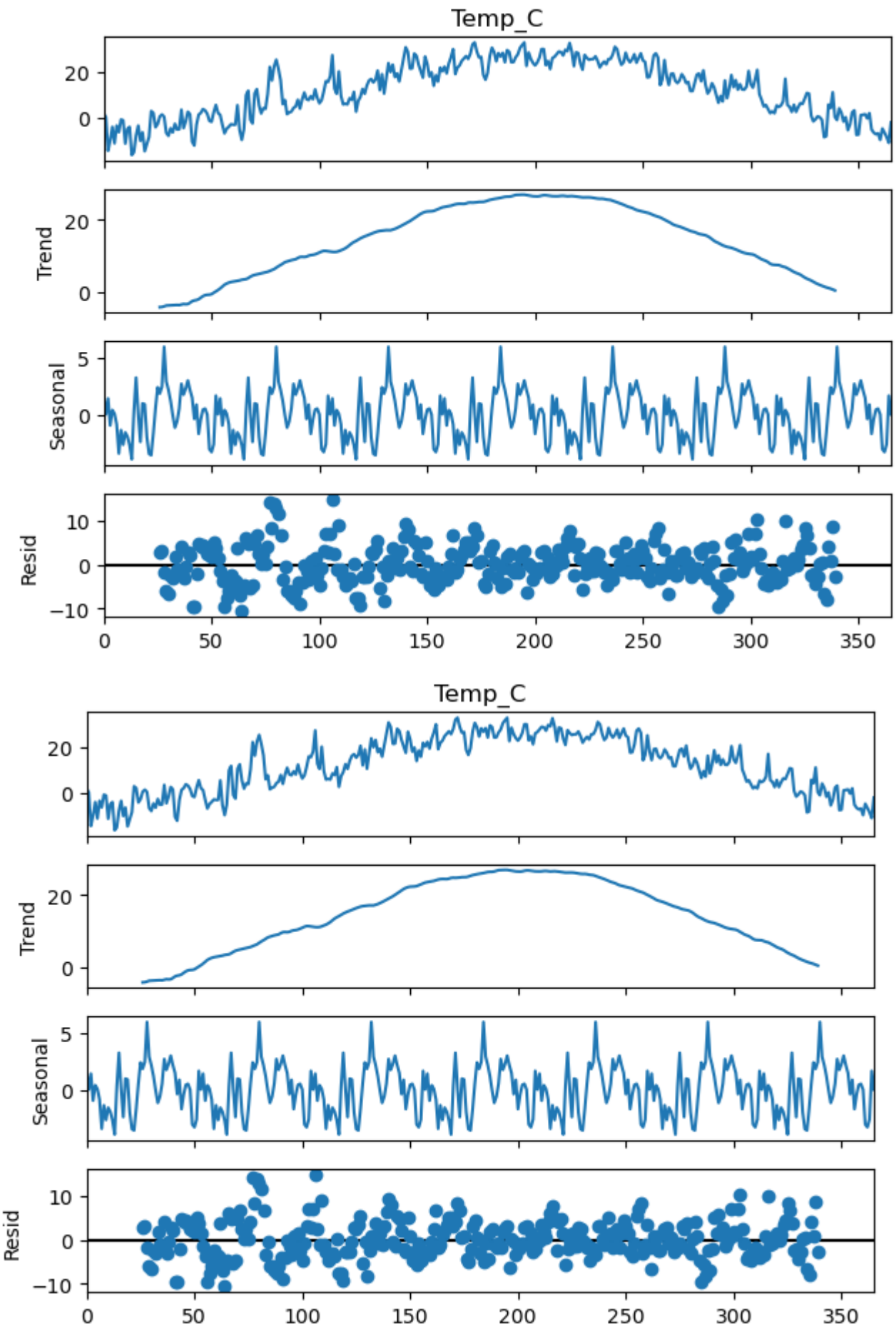
Out[53]:
```
[<matplotlib.lines.Line2D at 0x7f57a872da50>]
```



In [54]:
```
decomp = seasonal_decompose(Y,model="additive",period=52)
```

In [55]:
```
decomp.plot()
```

Out[55]:



In [56]:
```
train = resp.iloc[:int(resp.shape[0]*0.70)]
test = resp.iloc[int(resp.shape[0]*0.70):]
```

In [57]:
```
train.head()
```

Out[57]:

	Date	Temp_C	Dew Point Temp_C
0	2012-01-01	1.8	-0.4
1	2012-01-02	0.0	-7.0
2	2012-01-03	-14.8	-22.2
3	2012-01-04	-10.2	-16.3
4	2012-01-05	-4.3	-12.0

In [58]:
```
test.head()
```

Out[58]:

	Date	Temp_C	Dew Point Temp_C
256	2012-09-13	28.2	13.9
257	2012-09-14	26.3	16.4
258	2012-09-15	14.2	11.4
259	2012-09-16	16.7	3.6
260	2012-09-17	25.2	10.8

In [59]:
```
import statsmodels.api as sm
```

In [60]:
```
exponential_smoothing = sm.tsa.ExponentialSmoothing(train["Temp_C"],trend="additive",seasonal="additive",seasonal_p
```

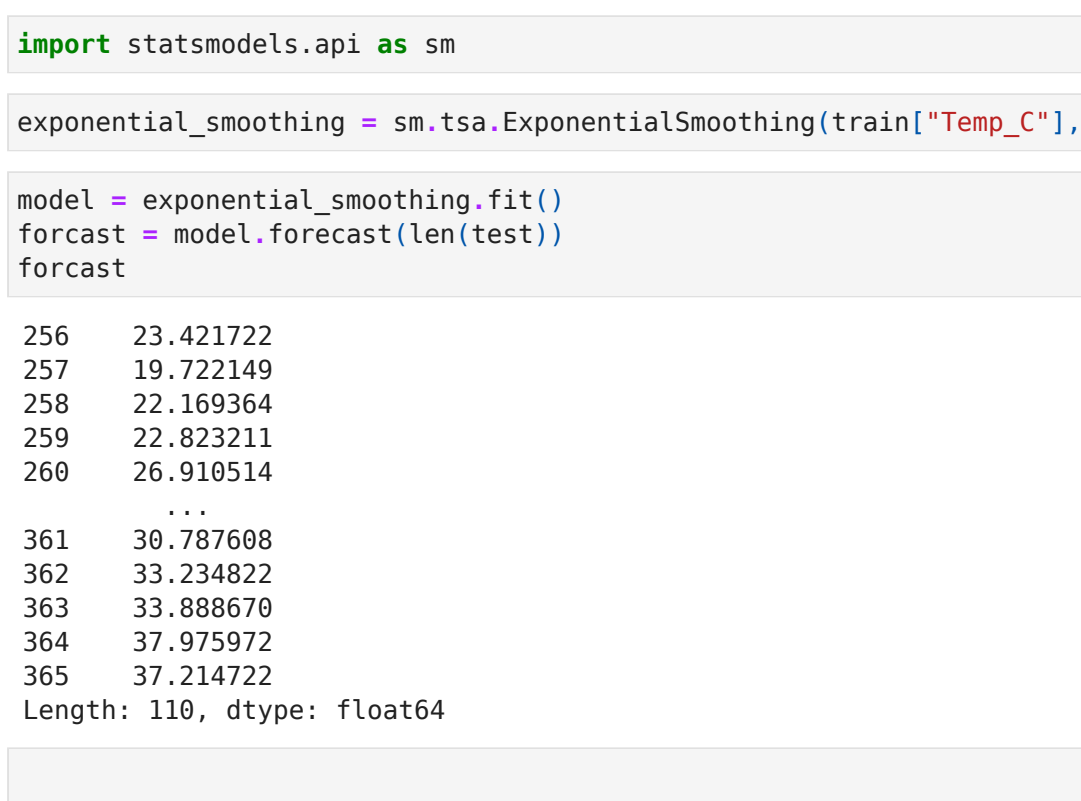
In [61]:
```
model = exponential_smoothing.fit()
forcast = model.forecast(len(test))
forcast
```

Out[61]:
```
256    23.421722
257    19.722149
258    22.169364
259    22.823211
260    26.910514
         ...
361    30.787608
362    33.234822
363    33.888670
364    37.975972
365    37.214722
Length: 110, dtype: float64
```

In [ ]:
```
```

In [62]:
```
plt.plot(resp['Temp_C'])
plt.plot(forcast)
plt.plot
```

Out[62]:
```
<function matplotlib.pyplot.plot(*args: 'float | ArrayLike | str', scalex: 'bool' = True, scaley: 'bool' = True, d
ata=None, **kwargs) -> 'list[Line2D]'>
```



In [63]:
```
model = sm.tsa.ExponentialSmoothing(test["Temp_C"],trend='additive',seasonal="additive",seasonal_periods=52)
tes_model = model.fit()
forcast1 = tes_model.forecast(8)
```

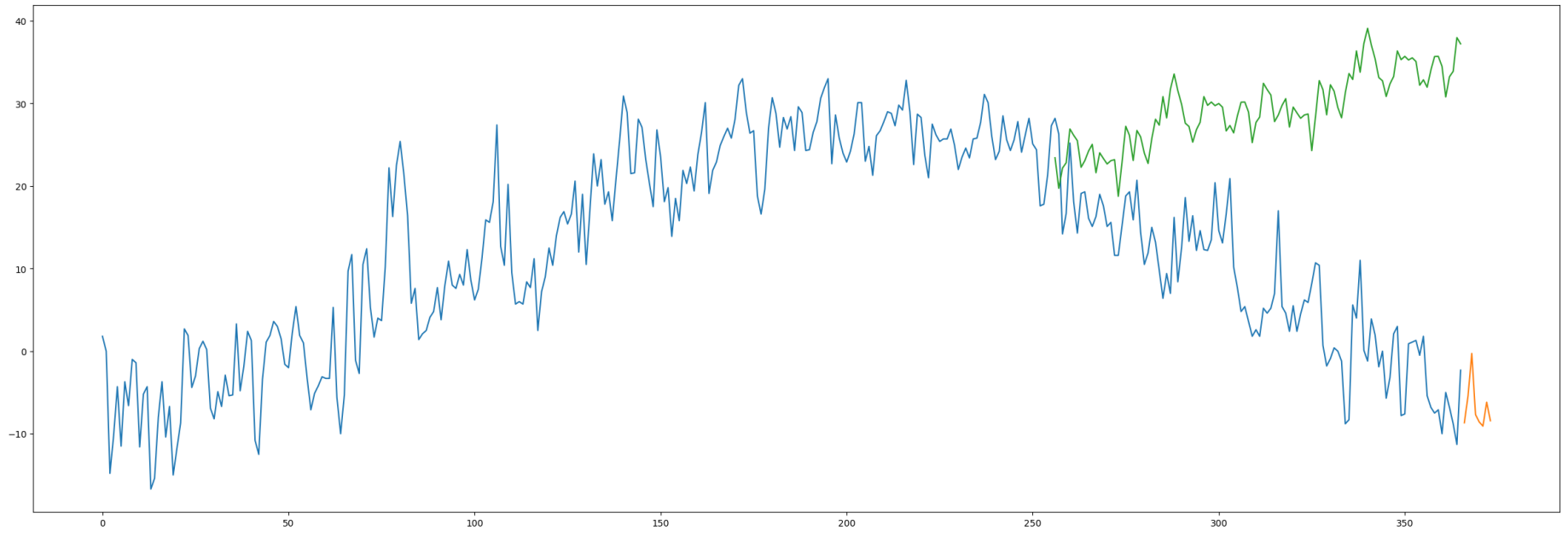
In [64]:
```
forcast1
```

Out[64]:
```
366   -8.677518
367   -5.377505
368   -0.277631
369   -7.677573
370   -8.577687
371   -9.077681
372   -6.177736
373   -8.427788
dtype: float64
```

In [65]:
```
plt.figure(figsize=(30,10))
plt.plot(resp['Temp_C'])
plt.plot(forcast1)
plt.plot(forcast)

plt.plot
```

Out[65]:
```
<function matplotlib.pyplot.plot(*args: 'float | ArrayLike | str', scalex: 'bool' = True, scaley: 'bool' = True, d
ata=None, **kwargs) -> 'list[Line2D]'>
```



In [ ]:
```
```