In [1]:
```
#import needed liberaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
%matplotlib inline
```

In [2]:
```
#import Data
data = pd.read_csv('DailyDelhiClimateTrain.csv')
data.head()
```

Out[2]:

	date	meantemp	humidity	wind_speed	meanpressure
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000

In [3]:
```
data.describe()
```

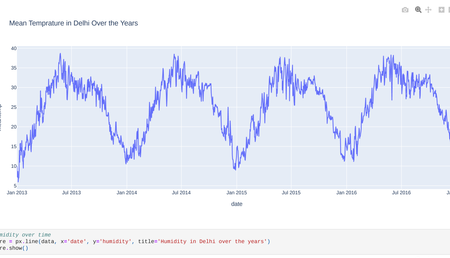
Out[3]:

	meantemp	humidity	wind_speed	meanpressure
count	1462.000000	1462.000000	1462.000000	1462.000000
mean	25.495521	60.771702	6.802209	1011.104548
std	7.348103	16.769652	4.561602	180.231668
min	6.000000	13.428571	0.000000	-3.041667
25%	18.857143	50.375000	3.475000	1001.580357
50%	27.714286	62.625000	6.221667	1008.563492
75%	31.305804	72.218750	9.238235	1014.944901
max	38.714286	100.000000	42.220000	7679.333333

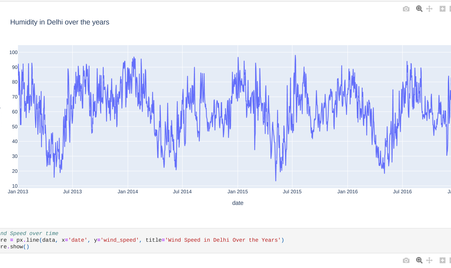
In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1462 entries, 0 to 1461
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   date          1462 non-null   object 
 1   meantemp      1462 non-null   float64
 2   humidity      1462 non-null   float64
 3   wind_speed    1462 non-null   float64
 4   meanpressure  1462 non-null   float64
dtypes: float64(4), object(1)
memory usage: 57.2+ KB
```

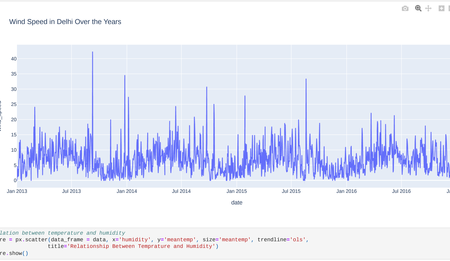
In [5]:
```
# Mean Temprature over time using plotly liberary
figure = px.line(data, x='date', y='meantemp', title='Mean Temprature in Delhi Over the Years')
figure.show()
```



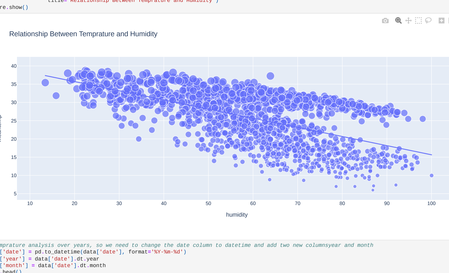
In [6]:
```
# Humidity over time
figure = px.line(data, x='date', y='humidity', title='Humidity in Delhi over the years')
figure.show()
```



In [7]:
```
# Wind Speed over time
figure = px.line(data, x='date', y='wind_speed', title='Wind Speed in Delhi Over the Years')
figure.show()
```



In [8]:
```
# Relation between temprature and humidity
figure = px.scatter(data_frame = data, x='humidity', y='meantemp', size='meantemp', trendline='ols',
                    title='Relationship Between Temprature and Humidity')
figure.show()
```

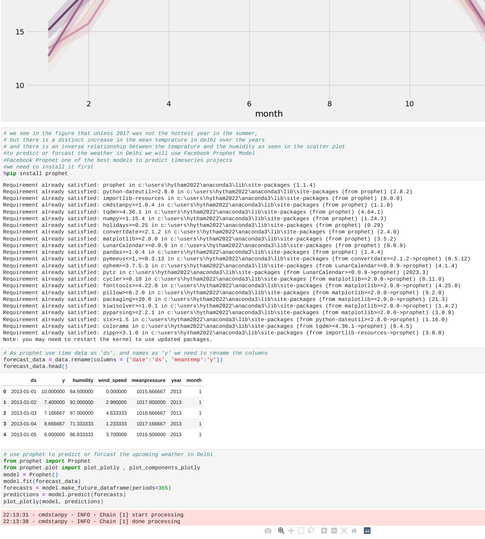


In [9]:
```
# temprature analysis over years, so we need to change the date column to datetime and add two new columnsyear and month
data['date'] = pd.to_datetime(data['date'], format='%Y-%m-%d')
data['year'] = data['date'].dt.year
data['month'] = data['date'].dt.month
data.head()
```

Out[9]:

	date	meantemp	humidity	wind_speed	meanpressure	year	month
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667	2013	1
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000	2013	1
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667	2013	1
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667	2013	1
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000	2013	1

In [10]:
```
# Now lets have a look on the change of the temprature over the years
plt.style.use('fivethirtyeight')
plt.figure(figsize=(15,10))
plt.title('Temprature Change in Delhi Over the Years')
sns.lineplot(data = data, x='month', y='meantemp', hue='year')
plt.show()
```



In [11]:
```
# we see in the figure that unless 2017 was not the hottest year in the summer,
# but there is a distinct increase in the mean temprature in delhi over the years
# and there is an inverse relationship between the temprature and the humidity as seen in the scatter plot
#to predict or forcast the weather in Delhi we will use Facebook Prophet Model
#Facebook Prophet one of the best models to predict timeseries projects
#we need to install it first
!pip install prophet
```

```
Requirement already satisfied: prophet in c:\users\hytham2022\anaconda3\lib\site-packages (1.1.4)
Requirement already satisfied: python-dateutil>=2.8.0 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (2.8.2)
Requirement already satisfied: importlib-resources in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (6.0.0)
Requirement already satisfied: cmdstanpy>=1.0.4 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (1.1.0)
Requirement already satisfied: tqdm>=4.36.1 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (4.64.1)
Requirement already satisfied: numpy>=1.15.4 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (1.24.3)
Requirement already satisfied: holidays>=0.25 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (0.29)
Requirement already satisfied: convertdate>=2.1.2 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (2.4.0)
Requirement already satisfied: matplotlib>=2.0.0 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (3.5.2)
Requirement already satisfied: LunarCalendar>=0.0.9 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (0.0.9)
Requirement already satisfied: pandas>=1.1.4 in c:\users\hytham2022\anaconda3\lib\site-packages (from prophet) (2.0.3)
Requirement already satisfied: pymeeus<=1,>=0.3.13 in c:\users\hytham2022\anaconda3\lib\site-packages (from convertdate>=2.1.2->prophet) (0.5.12)
Requirement already satisfied: ephem>=3.7.5.3 in c:\users\hytham2022\anaconda3\lib\site-packages (from LunarCalendar>=0.0.9->prophet) (4.1.4)
Requirement already satisfied: pytz in c:\users\hytham2022\anaconda3\lib\site-packages (from LunarCalendar>=0.0.9->prophet) (2023.3)
Requirement already satisfied: cycler>=0.10 in c:\users\hytham2022\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hytham2022\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (4.25.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hytham2022\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (9.2.0)
Requirement already satisfied: packaging>=20.0 in c:\users\hytham2022\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hytham2022\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (1.4.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\hytham2022\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (3.0.9)
Requirement already satisfied: six>=1.5 in c:\users\hytham2022\anaconda3\lib\site-packages (from python-dateutil>=2.8.0->prophet) (1.16.0)
Requirement already satisfied: colorama in c:\users\hytham2022\anaconda3\lib\site-packages (from tqdm>=4.36.1->prophet) (0.4.5)
Requirement already satisfied: zipp>=3.1.0 in c:\users\hytham2022\anaconda3\lib\site-packages (from importlib-resources->prophet) (3.8.0)
Note: you may need to restart the kernel to use updated packages.
```

In [12]:
```
# As prophet use time data as 'ds', and names as 'y' we need to rename the columns
forecast_data = data.rename(columns = {'date':'ds', 'meantemp':'y'})
forecast_data.head()
```

Out[12]:

	ds	y	humidity	wind_speed	meanpressure	year	month
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667	2013	1
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000	2013	1
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667	2013	1
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667	2013	1
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000	2013	1

In [13]:
```
# use prophet to predict or forcast the upcoming weather in Delhi
from prophet.prophet import Prophet
from prophet.plot import plot_plotly , plot_components_plotly
model = Prophet()
model.fit(forecast_data)
forecasts = model.make_future_dataframe(periods=365)
predictions = model.predict(forecasts)
plot_plotly(model, predictions)
```

```
22:11:31 - cmdstanpy - INFO - Chain [1] start processing
22:11:38 - cmdstanpy - INFO - Chain [1] done processing
```