Inference 1 on Exploratory Data

Dataset details:

Link: https://aqs.epa.gov/aqsweb/airdata/download_files.html (https://aqs.epa.gov/aqsweb/airdata/download_files.html)

We have used a combination 2 types of Air Quality datasets for exploratory inferences:

1. County and State-wise AQI dataset for the years 2020 and 2021

   https://aqs.epa.gov/aqsweb/airdata/daily_aqi_by_county_2021.zip (https://aqs.epa.gov/aqsweb/airdata/daily_aqi_by_county_2021.zip)

   https://aqs.epa.gov/aqsweb/airdata/daily_aqi_by_county_2020.zip (https://aqs.epa.gov/aqsweb/airdata/daily_aqi_by_county_2020.zip)
2. County and State-wise AQI specific for Carbon Monoxide (CO) indexes for the years 2020 and 2021

   https://aqs.epa.gov/aqsweb/airdata/daily_42101_2021.zip (https://aqs.epa.gov/aqsweb/airdata/daily_42101_2021.zip)

   https://aqs.epa.gov/aqsweb/airdata/daily_42101_2020.zip (https://aqs.epa.gov/aqsweb/airdata/daily_42101_2020.zip)

Motivation

Air Quality Index (AQI) is an accurate measure for identifying the pollutants present in the air. During the time of Covid, lockdowns were imposed all around the US, which is bound to impact the overall AQI of the country. We wish to explore the relation between this measure and the cases/vaccines data provided in the question document. In addition, some particular pollutants (like Carbon Monoxide, etc) can also be changed in some way during and after the months of Covid, we wish to understand the overall trend of that.

In [47]:

```
import pandas as pd
import numpy as np
import math
```

In [48]:

```
vaccines = pd.read_csv("/Users/meet/Desktop/544_project/Mandatory_data
vaccines.head()
```

Out[48]:

	Date	MMWR_week	Location	Distributed	Distributed_Janssen	Distributed_Moderna
0	05/15/2022	20	PR	7552350	215000	2662120
1	05/15/2022	20	KS	6121515	256400	2354940
2	05/15/2022	20	VA	19949085	785300	7108700
3	05/15/2022	20	MT	2004895	105200	825200
4	05/15/2022	20	IH2	2965895	108400	1311680

5 rows × 82 columns

In [49]:

```
# filtering the vaccines data to get the location, date and distribute
vaccines_filtered = vaccines[["Date", "Location", "Distributed", "Dist

# sorting the data in ascending order by date
vaccines_filtered["Date"] = pd.to_datetime(vaccines_filtered["Date"])
vaccines_filtered.sort_values(by="Date", inplace=True)
vaccines_filtered.head()
```

Out[49]:

	Date	Location	Distributed	Distributed_Janssen	Distributed_Moderna	Distributed_Pfize
33431	2020-12-13	GU	3900	0	0	
33426	2020-12-13	LTC	0	0	0	
33427	2020-12-13	AS	3900	0	0	
33430	2020-12-13	US	13650	0	0	
33429	2020-12-13	VI	975	0	0	

In [50]:

```
vaccines_filtered.isnull().sum()
```

Out[50]:

```
Date                    0
Location                0
Distributed             0
Distributed_Janssen     0
Distributed_Moderna     0
Distributed_Pfizer      0
Administered            0
Administered_Janssen    0
Administered_Moderna    0
Administered_Pfizer     0
dtype: int64
```

Check the Dependency of Location (State) on AQI

Identify if locations where vaccines are manufactured and distributed more have a higher AQI or not

Motivation

A lot of pharma companies initiated their research on finding a vaccine for Covid and started mass manufacturing of the same. Hence, ideally the AQI should have gone low (got better) during the months of Covid due to lockdowns, but we would like to make a hypothesis that the states where vaccine were being manufactured and getting distributed the most, should still have bad AQI (worse than expected) due to the pollution from drug manufacturing factories, transport vehicles, and other factors included in vaccine administration.

In [292]:

```
# helper function to gerenrate estimate CDF
def generate_eCDF(X):
    n = len(X)
    Srt = sorted(X)
    delta = .1
    X = [min(Srt)-delta]
    Y = [0]
    for i in range(0, n):
        X = X + [Srt[i], Srt[i]]
        Y = Y + [Y[len(Y)-1], Y[len(Y)-1]+(1/n)]
    X = X + [max(Srt)+delta]
    # print(X)
    Y = Y + [1]
    return X, Y
```

In [293]:

```
# helper function to perform a 2 sample KS-Test
def ks_test_2_sample(X1, Y1, X2, Y2):
    tot_max = -1
    ks_table = np.zeros((len(X1),6))
    for i in range(len(ks_table)-1):
        ks_table[i,0] = Y1[i]
        ks_table[i,1] = Y1[i+1]
        ks_table[i,2]=0
        ks_table[i,3]=0
        for j in X2:
            if j<X1[i]:
                ks_table[i,2]+=1
            if j<=X1[i]:
                ks_table[i,3]+=1

        ks_table[i,3]/=len(X2)
        ks_table[i,2]/=len(X2)

        ks_table[i,4] = abs(ks_table[i,0] - ks_table[i,2])
        ks_table[i,5] = abs(ks_table[i,1] - ks_table[i,3])
        cmax = max(ks_table[i,4], ks_table[i,5])
        if cmax > tot_max:
            tot_max = cmax
            x1_max = X1[i]
            y1_max = ks_table[i,0]
            y2_max = ks_table[i,2]
    return tot_max
```

In [302]:

```
# helper function to perform Pearson Correlation
def pearson_corr(df):
    num = np.sum((df['Distributed'] - df['Distributed'].mean()) * (df[
    den = np.sqrt(np.sum(pow(df['Distributed'] - df['Distributed'].mea
    coeff = num/den

    return coeff
```

In [303]:

```
upby(["Location"]).agg({"Distributed_Pfizer": "max"}).sort_values(by="
```

Out[303]:

```
          Distributed_Pfizer
Location
      US           432073335
     LTC            73720035
      CA            54867275
      TX            37383255
      FL            28281365
      NY            28267955
      PA            17641365
      IL            16964305
      OH            13683215
      NJ            13345775
```

In [304]:

```
# Function to get the daily distributed vaccine data for particular st
def get_state_wise_daily(df, state):
    df_state = df.loc[df["Location"].str.startswith(state, na=False)].
    df_state.reset_index(drop=True, inplace=True)
    df_state_updated = pd.DataFrame(columns=["Date", "Location", "Dist
    df_state_updated = df_state_updated.append({"Date": df_state.Date[
    for i in range(1, len(df_state)):
        state_date = df_state.Date[i]
        state = df_state.Location[i]
        state_dis = df_state.Distributed[i] - df_state.Distributed[i -
        state_dis_jj = df_state.Distributed_Janssen[i] - df_state.Dist
        state_dis_md = df_state.Distributed_Moderna[i] - df_state.Dist
        state_dis_pf = df_state.Distributed_Pfizer[i] - df_state.Distr
        df_state_updated = df_state_updated.append({"Date": state_date

    return df_state_updated
```

In [155]:

```
# load aqi 2020 and 2021 dataset
aqi_2020 = pd.read_csv("/Users/meet/Desktop/544_project/X_dataset/dail
aqi_2021 = pd.read_csv("/Users/meet/Desktop/544_project/X_dataset/dail
```

In [156]:
```
# combining the AQI 2020 and 2021 dataset
aqi = pd.concat([aqi_2020, aqi_2021]).reset_index(drop=True)
print(len(aqi_2020) + len(aqi_2021), len(aqi))
```

```
556391 556391
```

In [157]:
```
aqi.head()
```

Out[157]:

	State Name	county Name	State Code	County Code	Date	AQI	Category	Defining Parameter	Defining Site	Number of Sites Reporting
0	Alabama	Baldwin	1	3	2020-01-01	48	Good	PM2.5	01-003-0010	1
1	Alabama	Baldwin	1	3	2020-01-04	13	Good	PM2.5	01-003-0010	1
2	Alabama	Baldwin	1	3	2020-01-07	14	Good	PM2.5	01-003-0010	1
3	Alabama	Baldwin	1	3	2020-01-10	39	Good	PM2.5	01-003-0010	1
4	Alabama	Baldwin	1	3	2020-01-13	29	Good	PM2.5	01-003-0010	1

In [158]:
```
aqi.rename(columns={"State Name": "state", "county Name": "county", "D
```

California

California has the highest number of vaccines distributed, so we tried to perform our inference on the state-specific data for this. **Null Hypothesis H0:** The AQI should be dependent on the vaccines distributed **Alternate Hypothesis H1:** The AQI should not be dependent on the vaccines distributed

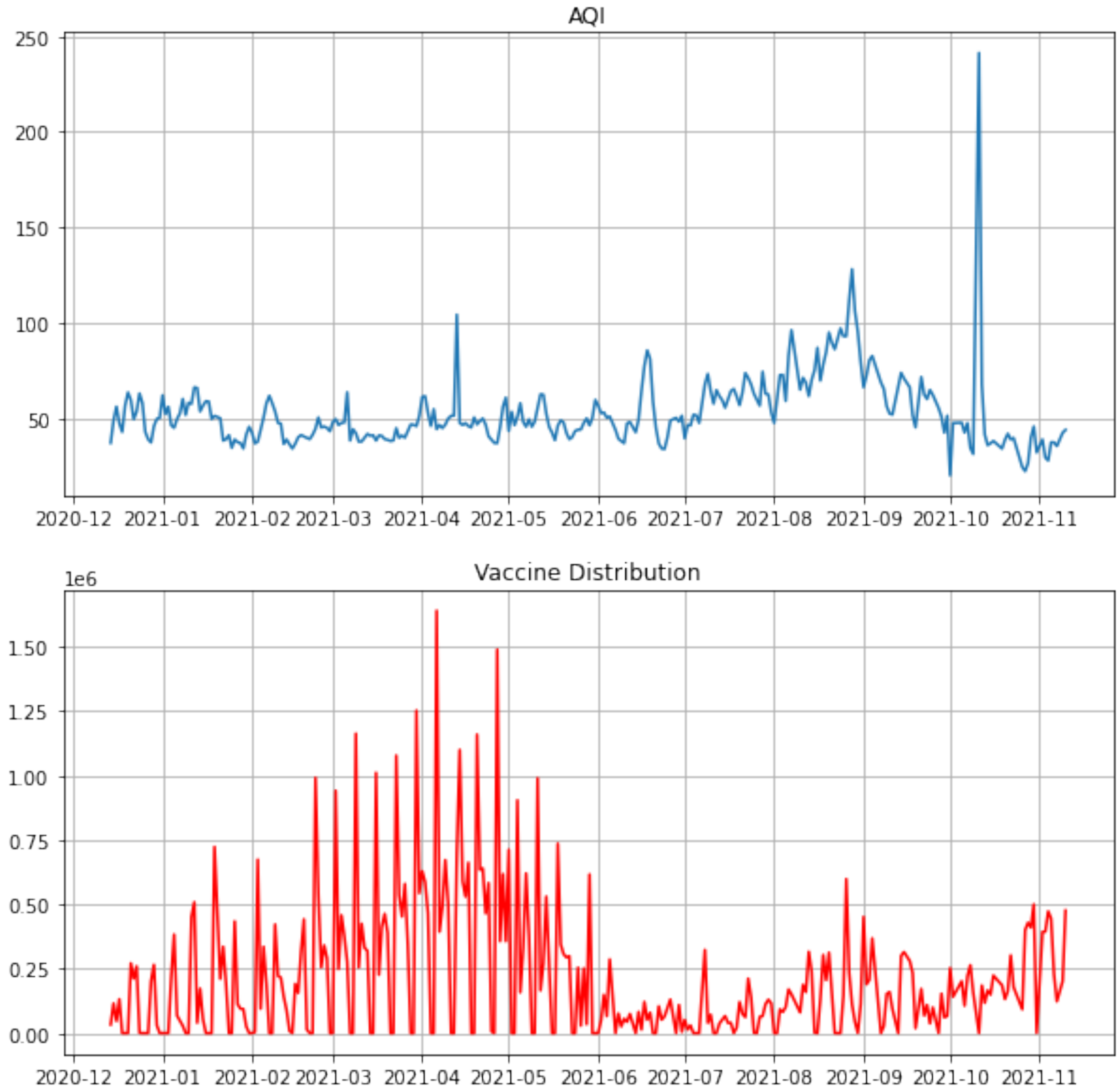
In [296]:

```
vaccines_ca = get_state_wise_daily(vaccines_filtered, "CA")
vaccines_ca = vaccines_ca[vaccines_ca["Distributed"] >= 0]
vaccines_ca.head()

aqi_ca = aqi[["state", "Date", "AQI", "Category", "defining_param"]][a

aqi_ca["Date"] = pd.to_datetime(aqi_ca["Date"])
aqi_ca.sort_values(by="Date", inplace=True)
aqi_ca.reset_index(drop=True, inplace=True)
aqi_ca.head()

aqi_ca_mean = aqi_ca[["state", "Date", "AQI", "Category", "defining_pa
aqi_ca_mean.reset_index(inplace=True)
aqi_ca_mean.head()

vaccines_aqi_ca = pd.merge(vaccines_ca, aqi_ca_mean, on=("Date"))
vaccines_aqi_ca.head()

vaccines_ca_x, vaccines_ca_y = generate_eCDF(vaccines_aqi_ca.Distribut
aqi_ca_x, aqi_ca_y = generate_eCDF(vaccines_aqi_ca.AQI.to_list())
print("KS Statistic", ks_test_2_sample(vaccines_ca_x, vaccines_ca_y, a

print("Pearson Correlation Coefficient", pearson_corr(vaccines_aqi_ca)
```

```
KS Statistic 0.7750965654034869
Pearson Correlation Coefficient -0.12324445671111955
```

In [263]:

```
f, axs = plt.subplots(2, 1, figsize=(10, 10))
# axs[0].subplot(2, 1, 1)
axs[0].grid()
axs[0].plot(vaccines_aqi_ca["Date"], vaccines_aqi_ca["AQI"])
axs[0].title.set_text("AQI")
# plt.subplot(2, 1, 2)
axs[1].grid()
axs[1].plot(vaccines_aqi_ca["Date"], vaccines_aqi_ca["Distributed"], c
axs[1].title.set_text("Vaccine Distribution")
# plt.show()
```

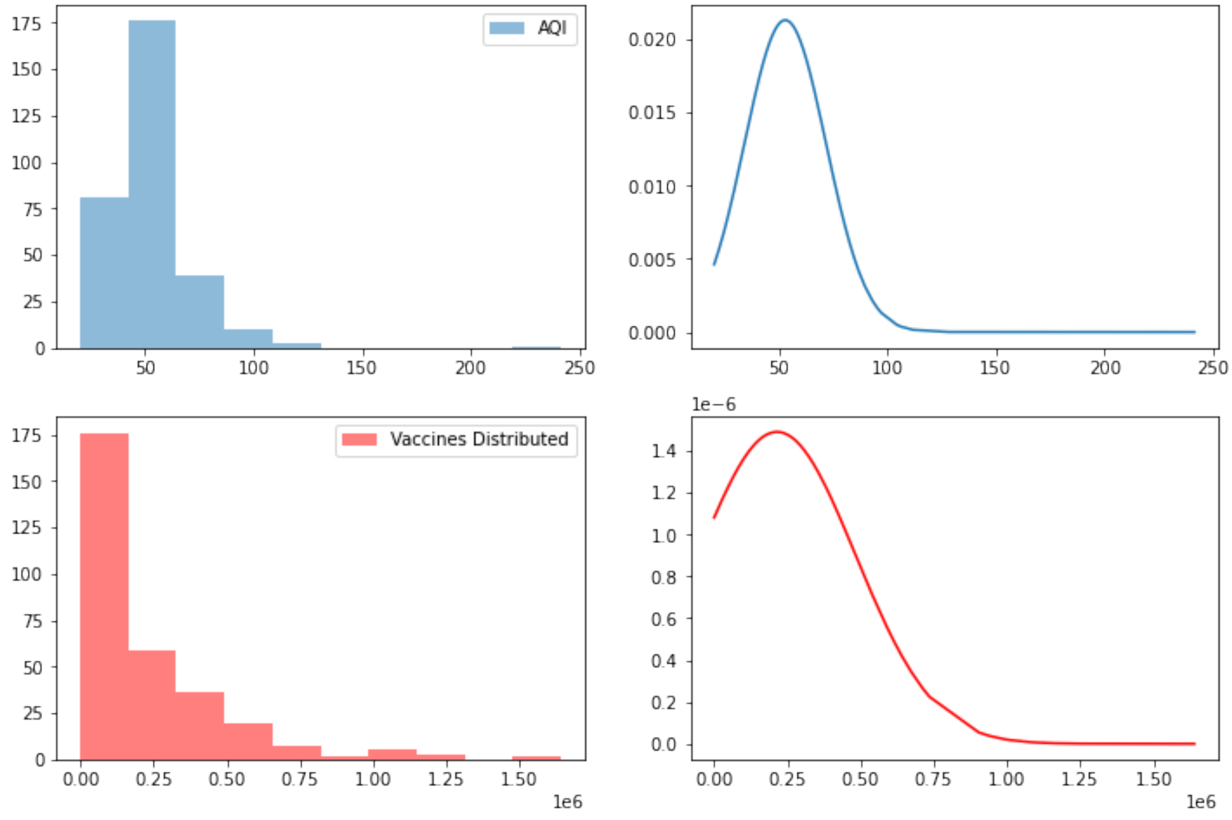


In [290]:

```
import numpy as np
import matplotlib.pyplot as plt
import statistics

x_axis = sorted(vaccines_aqi_ca.AQI.to_list())
x_axis_2 = sorted(vaccines_aqi_ca.Distributed.to_list())

mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)

mean2 = statistics.mean(x_axis_2)
sd2 = statistics.stdev(x_axis_2)

f, axs = plt.subplots(2, 2, figsize=(12, 8))
axs[0][0].hist(x_axis, alpha=0.5, label="AQI")
axs[0][1].plot(x_axis, norm.pdf(x_axis, mean, sd))
axs[0][0].legend(loc="upper right")
# axs[0][1].legend(loc="upper right")

axs[1][0].hist(x_axis_2, alpha=0.5, label="Vaccines Distributed", colo
axs[1][1].plot(x_axis_2, norm.pdf(x_axis_2, mean2, sd2), color="red")
axs[1][0].legend(loc="upper right")
# axs[1][1].legend(loc="upper right")
```

Out[290]:

```
<matplotlib.legend.Legend at 0x7f7e09efbe20>
```



Results for California

From the Pearson Correlation, KS-Statistic value, and the plots above, it can be inferred that the null hypothesis H0 will be rejected as the correlation coefficient is less than 0.5 and KS-stat is much larger than 0.05. It signifies that the number of vaccines distributed in a state does not provide dependency to the AQI of the state.

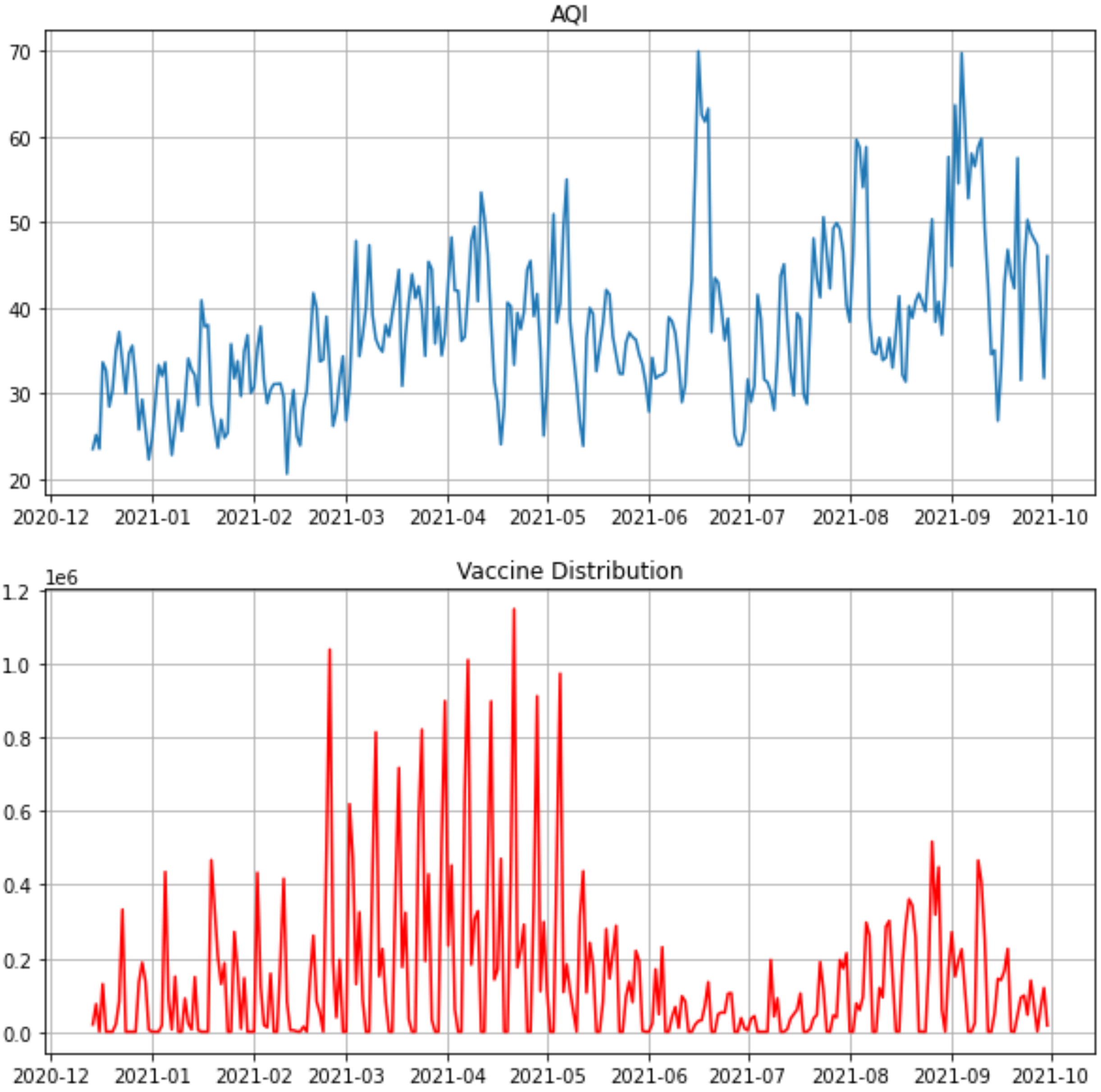
Texas

In [297]:
```
s_tx = get_state_wise_daily(vaccines_filtered, "TX")
s_tx = vaccines_tx[vaccines_tx["Distributed"] >= 0]
s_tx.head()

= aqi[["state", "Date", "AQI", "Category", "defining_param"]][aqi["sta

"Date"] = pd.to_datetime(aqi_tx["Date"])
sort_values(by="Date", inplace=True)
reset_index(drop=True, inplace=True)
head()

mean = aqi_tx[["state", "Date", "AQI", "Category", "defining_param"]].g
mean.reset_index(inplace=True)
mean.head()

s_aqi_tx = pd.merge(vaccines_tx, aqi_tx_mean, on=("Date"))
s_aqi_tx.head()

s_tx_x, vaccines_tx_y = generate_eCDF(vaccines_aqi_tx.Distributed.to_l
x, aqi_tx_y = generate_eCDF(vaccines_aqi_tx.AQI.to_list())
KS Statistic", ks_test_2_sample(vaccines_tx_x, vaccines_tx_y, aqi_tx_x

Pearson Correlation Coefficient", pearson_corr(vaccines_aqi_tx))
```

```
KS Statistic 0.7272727272727275
Pearson Correlation Coefficient 0.0733431010113787
```

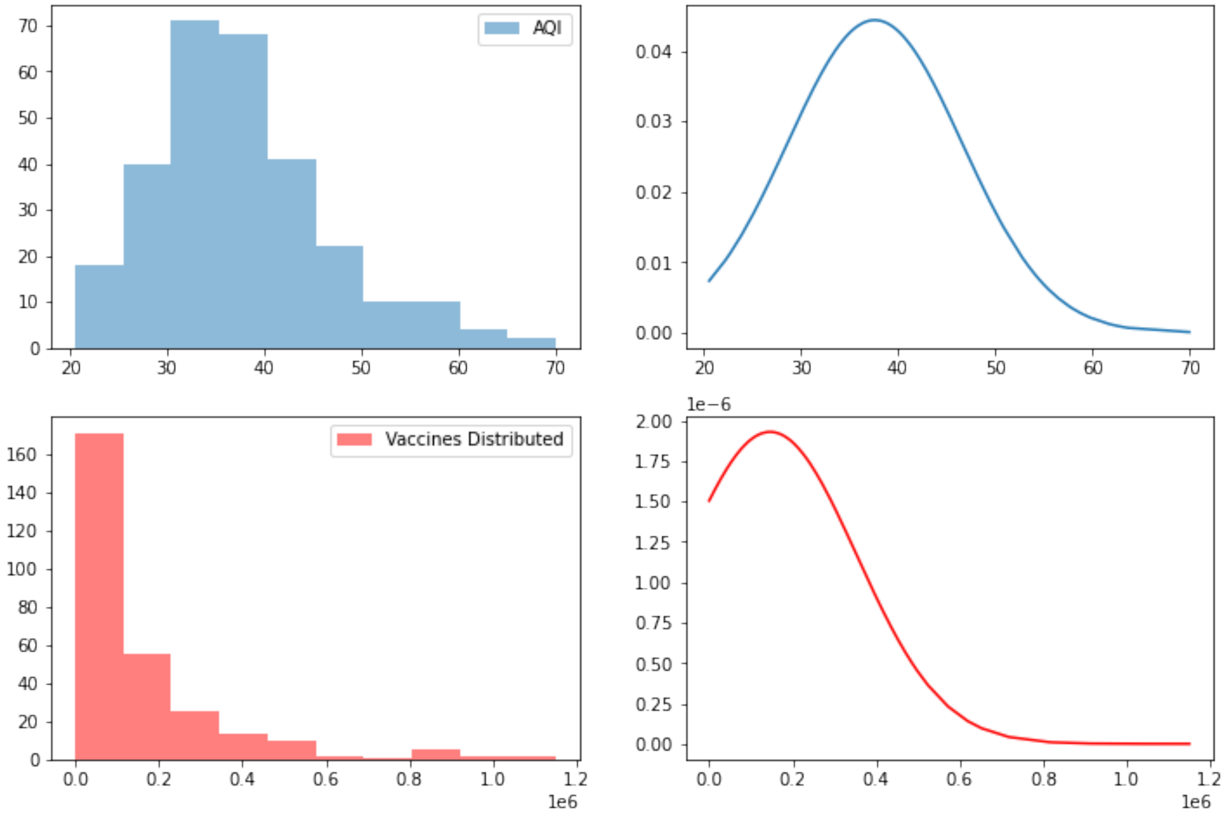
In [265]:

```
f, axs = plt.subplots(2, 1, figsize=(10, 10))
# axs[0].subplot(2, 1, 1)
axs[0].grid()
axs[0].plot(vaccines_aqi_tx["Date"], vaccines_aqi_tx["AQI"])
axs[0].title.set_text("AQI")
# plt.subplot(2, 1, 2)
axs[1].grid()
axs[1].plot(vaccines_aqi_tx["Date"], vaccines_aqi_tx["Distributed"], c
axs[1].title.set_text("Vaccine Distribution")
# plt.show()
```



In [289]:
```
import numpy as np
import matplotlib.pyplot as plt
import statistics

x_axis = sorted(vaccines_aqi_tx.AQI.to_list())
x_axis_2 = sorted(vaccines_aqi_tx.Distributed.to_list())

mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)

mean2 = statistics.mean(x_axis_2)
sd2 = statistics.stdev(x_axis_2)

f, axs = plt.subplots(2, 2, figsize=(12, 8))
axs[0][0].hist(x_axis, alpha=0.5, label="AQI")
axs[0][1].plot(x_axis, norm.pdf(x_axis, mean, sd))
axs[0][0].legend(loc="upper right")
# axs[0][1].legend(loc="upper right")

axs[1][0].hist(x_axis_2, alpha=0.5, label="Vaccines Distributed", colo
axs[1][1].plot(x_axis_2, norm.pdf(x_axis_2, mean2, sd2), color="red")
axs[1][0].legend(loc="upper right")
# axs[1][1].legend(loc="upper right")
```

Out[289]:
```
<matplotlib.legend.Legend at 0x7f7da8359ac0>
```



Results for Texas

The results are again similar to that of California. From the Pearson Correlation, KS-Statistic value, and the plots above, it can be inferred that the null hypothesis H0 will be rejected as the correlation coefficient is less than 0.5 and KS-stat is much larger than 0.05. It signifies that the number of vaccines distributed in a state does not provide dependency to the AQI of the state.

Delaware

Now, here we try to identify if the size of state has any impact on the results or not. For that, we try to perform the same hypothesis on Delaware and Rhode Island which are the 2 smallest states of US based on their size.

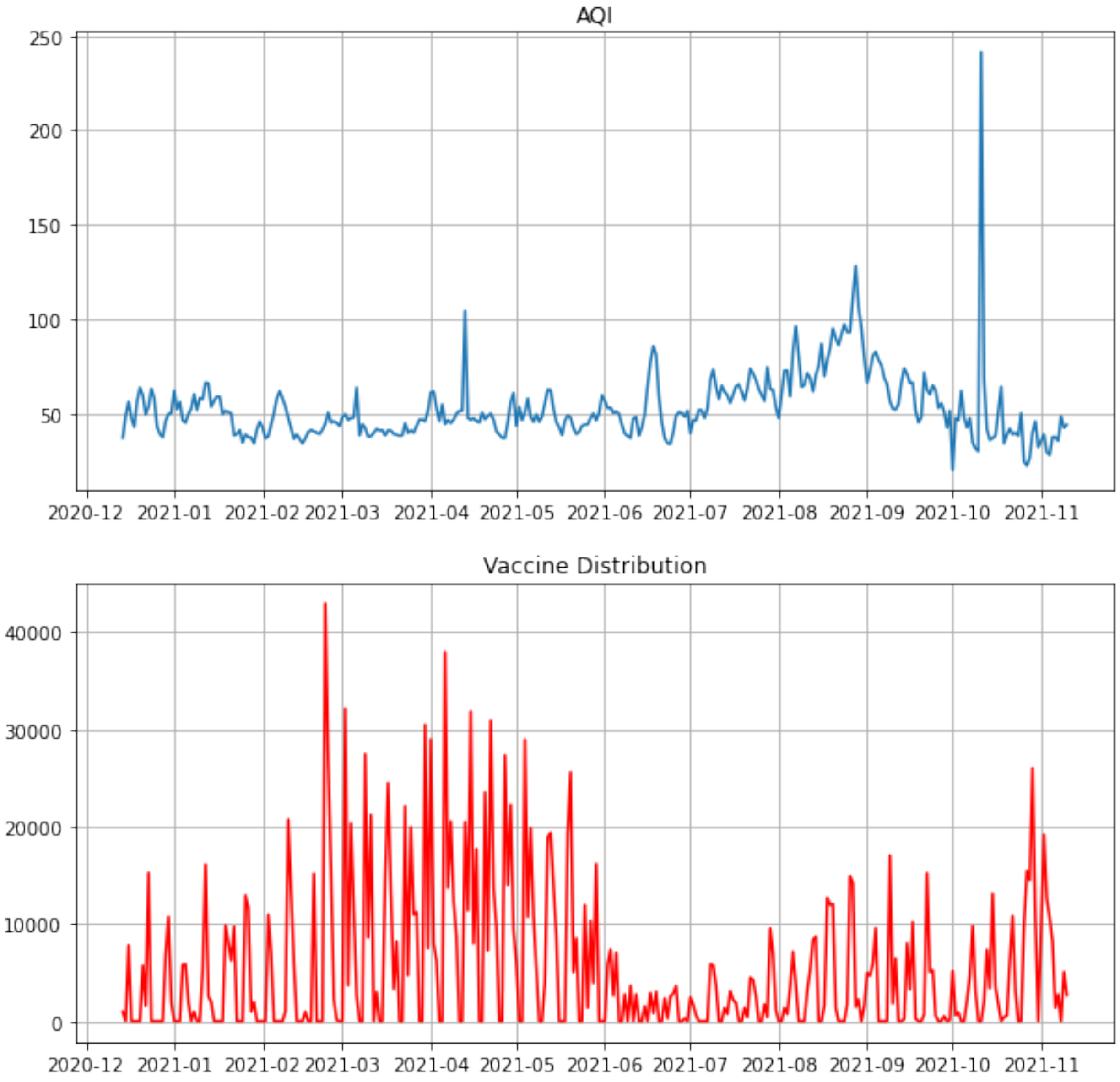
In [298]:

```
vaccines_de = get_state_wise_daily(vaccines_filtered, "DE")
vaccines_de = vaccines_de[vaccines_de["Distributed"] >= 0]
vaccines_de.head()

aqi_de = aqi[["state", "Date", "AQI", "Category", "defining_param"]][a

aqi_de["Date"] = pd.to_datetime(aqi_de["Date"])
aqi_de.sort_values(by="Date", inplace=True)
aqi_de.reset_index(drop=True, inplace=True)
aqi_de.head()

aqi_de_mean = aqi_de[["state", "Date", "AQI", "Category", "defining_pa
aqi_de_mean.reset_index(inplace=True)
aqi_de_mean.head()

vaccines_aqi_de = pd.merge(vaccines_de, aqi_de_mean, on=("Date"))
vaccines_aqi_de.head()

vaccines_de_x, vaccines_de_y = generate_eCDF(vaccines_aqi_de.Distribut
aqi_de_x, aqi_de_y = generate_eCDF(vaccines_aqi_de.AQI.to_list())
print("KS Statistic", ks_test_2_sample(vaccines_de_x, vaccines_de_y, a

print("Pearson Correlation Coefficient", pearson_corr(vaccines_aqi_de)
```

```
KS Statistic 0.6143524353176659
Pearson Correlation Coefficient -0.0903180969589416
```

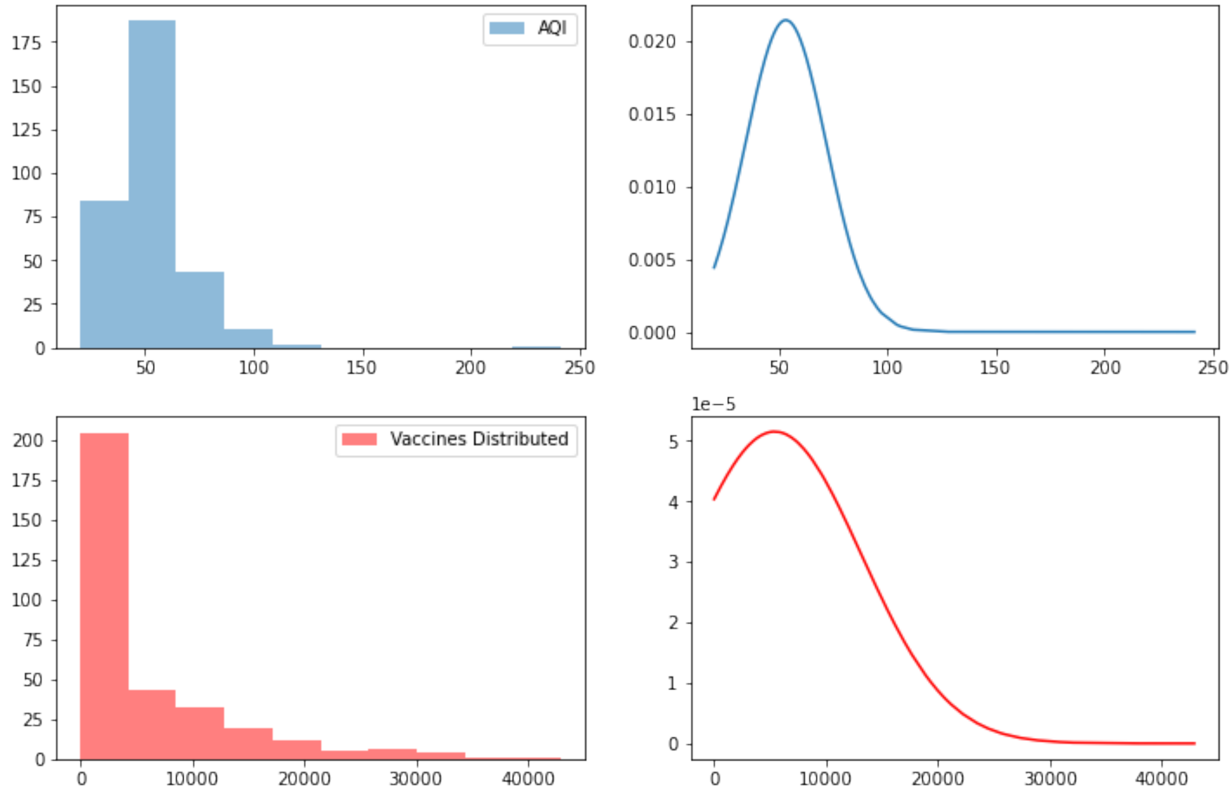
In [266]:

```
f, axs = plt.subplots(2, 1, figsize=(10, 10))
# axs[0].subplot(2, 1, 1)
axs[0].grid()
axs[0].plot(vaccines_aqi_de["Date"], vaccines_aqi_de["AQI"])
axs[0].title.set_text("AQI")
# plt.subplot(2, 1, 2)
axs[1].grid()
axs[1].plot(vaccines_aqi_de["Date"], vaccines_aqi_de["Distributed"], c
axs[1].title.set_text("Vaccine Distribution")
# plt.show()
```



In [288]:
```python
import numpy as np
import matplotlib.pyplot as plt
import statistics

x_axis = sorted(vaccines_aqi_de.AQI.to_list())
x_axis_2 = sorted(vaccines_aqi_de.Distributed.to_list())

mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)

mean2 = statistics.mean(x_axis_2)
sd2 = statistics.stdev(x_axis_2)

f, axs = plt.subplots(2, 2, figsize=(12, 8))
axs[0][0].hist(x_axis, alpha=0.5, label="AQI")
axs[0][1].plot(x_axis, norm.pdf(x_axis, mean, sd))
axs[0][0].legend(loc="upper right")
# axs[0][1].legend(loc="upper right")

axs[1][0].hist(x_axis_2, alpha=0.5, label="Vaccines Distributed", colc
axs[1][1].plot(x_axis_2, norm.pdf(x_axis_2, mean2, sd2), color="red")
axs[1][0].legend(loc="upper right")
# axs[1][1].legend(loc="upper right")
```

Out[288]:
```
<matplotlib.legend.Legend at 0x7f7e09c64580>
```



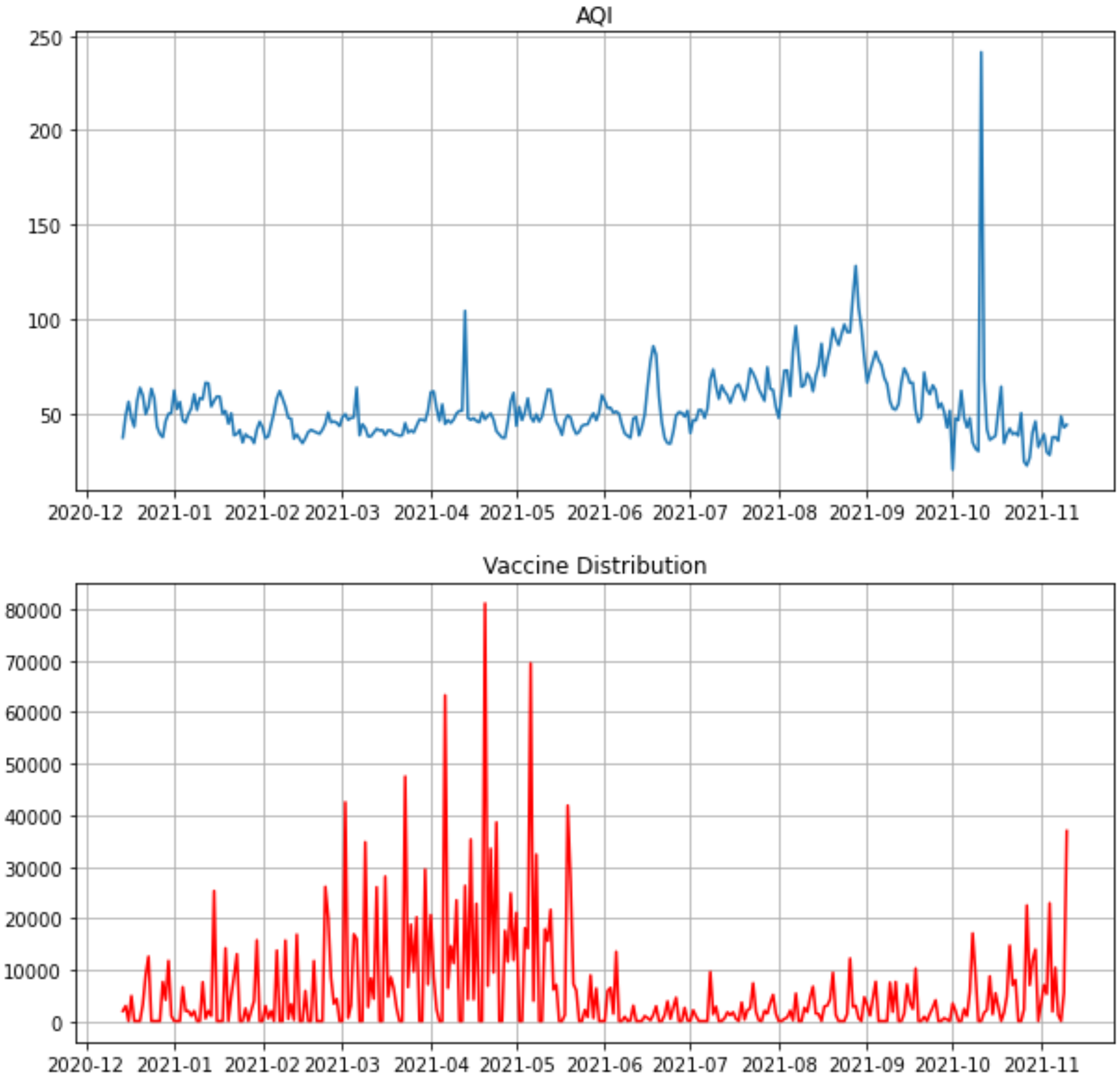
Rhode Island

In [299]:

```
vaccines_ri = get_state_wise_daily(vaccines_filtered, "RI")
vaccines_ri = vaccines_ri[vaccines_ri["Distributed"] >= 0]
vaccines_ri.head()

aqi_ri = aqi[["state", "Date", "AQI", "Category", "defining_param"]][a

aqi_ri["Date"] = pd.to_datetime(aqi_ri["Date"])
aqi_ri.sort_values(by="Date", inplace=True)
aqi_ri.reset_index(drop=True, inplace=True)
aqi_ri.head()

aqi_ri_mean = aqi_ca[["state", "Date", "AQI", "Category", "defining_pa
aqi_ri_mean.reset_index(inplace=True)
aqi_ri_mean.head()

vaccines_aqi_ri = pd.merge(vaccines_ri, aqi_ri_mean, on=("Date"))
vaccines_aqi_ri.head()

vaccines_ri_x, vaccines_ri_y = generate_eCDF(vaccines_aqi_ri.Distribut
aqi_ri_x, aqi_ri_y = generate_eCDF(vaccines_aqi_ri.AQI.to_list())
print("KS Statistic", ks_test_2_sample(vaccines_ri_x, vaccines_ri_y, a

print("Pearson Correlation Coefficient", pearson_corr(vaccines_aqi_ri)
```

```
KS Statistic 0.6611080408952756
Pearson Correlation Coefficient -0.11396235481327478
```

In [269]:
```
f, axs = plt.subplots(2, 1, figsize=(10, 10))
# axs[0].subplot(2, 1, 1)
axs[0].grid()
axs[0].plot(vaccines_aqi_ri["Date"], vaccines_aqi_ri["AQI"])
axs[0].title.set_text("AQI")
# plt.subplot(2, 1, 2)
axs[1].grid()
axs[1].plot(vaccines_aqi_ri["Date"], vaccines_aqi_ri["Distributed"], c
axs[1].title.set_text("Vaccine Distribution")
# plt.show()
```

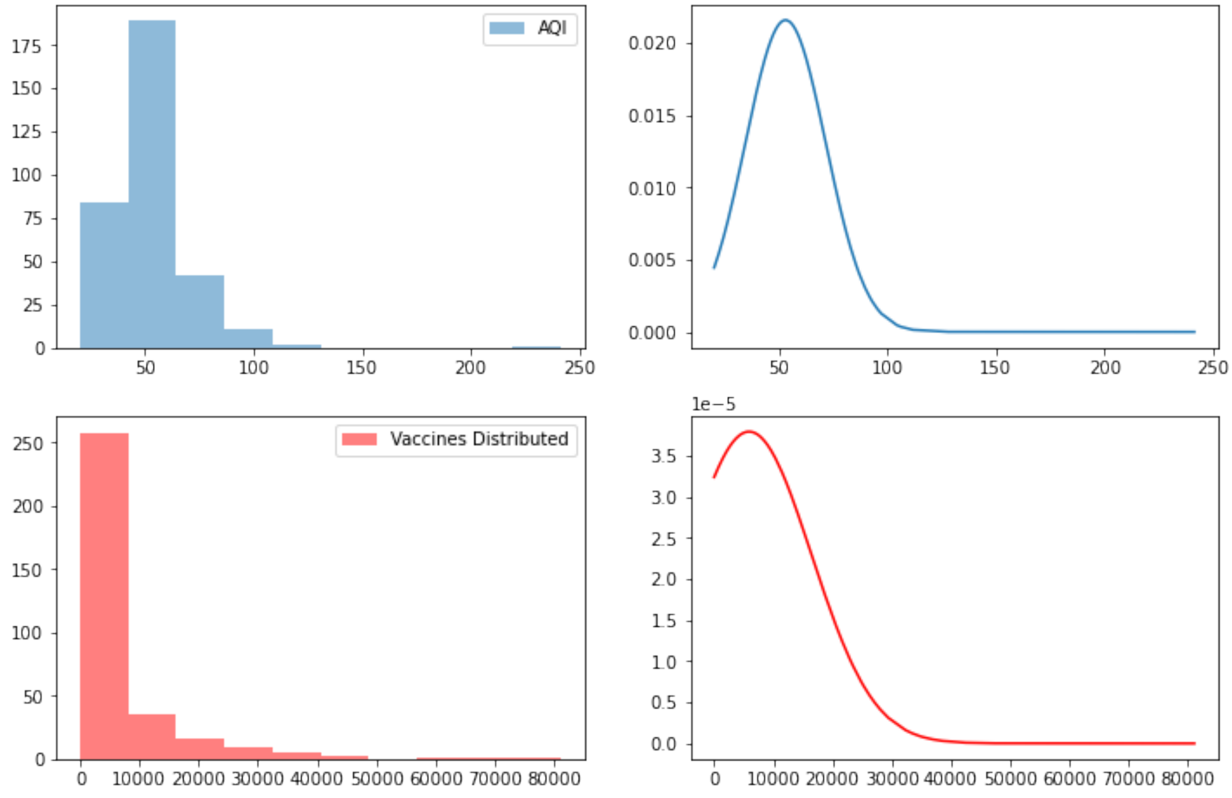


In [287]:

```
import numpy as np
import matplotlib.pyplot as plt
import statistics

x_axis = sorted(vaccines_aqi_ri.AQI.to_list())
x_axis_2 = sorted(vaccines_aqi_ri.Distributed.to_list())

mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)

mean2 = statistics.mean(x_axis_2)
sd2 = statistics.stdev(x_axis_2)

f, axs = plt.subplots(2, 2, figsize=(12, 8))
axs[0][0].hist(x_axis, alpha=0.5, label="AQI")
axs[0][1].plot(x_axis, norm.pdf(x_axis, mean, sd))
axs[0][0].legend(loc="upper right")
# axs[0][1].legend(loc="upper right")

axs[1][0].hist(x_axis_2, alpha=0.5, label="Vaccines Distributed", colo
axs[1][1].plot(x_axis_2, norm.pdf(x_axis_2, mean2, sd2), color="red")
axs[1][0].legend(loc="upper right")
# axs[1][1].legend(loc="upper right")
```

Out[287]:

```
<matplotlib.legend.Legend at 0x7f7e0831b7c0>
```



Results for Small States

Results above again signify similarity to that of states with most distribution of vaccines.

Massechusetts

Now, we try to perform the same hypothesis testing on the states where vaccines like Pfizer and Moderna are manufactured. i.e. Massechussetts and New York, as the factories emmit several pollutants during the process of manufacturing the vaccines.

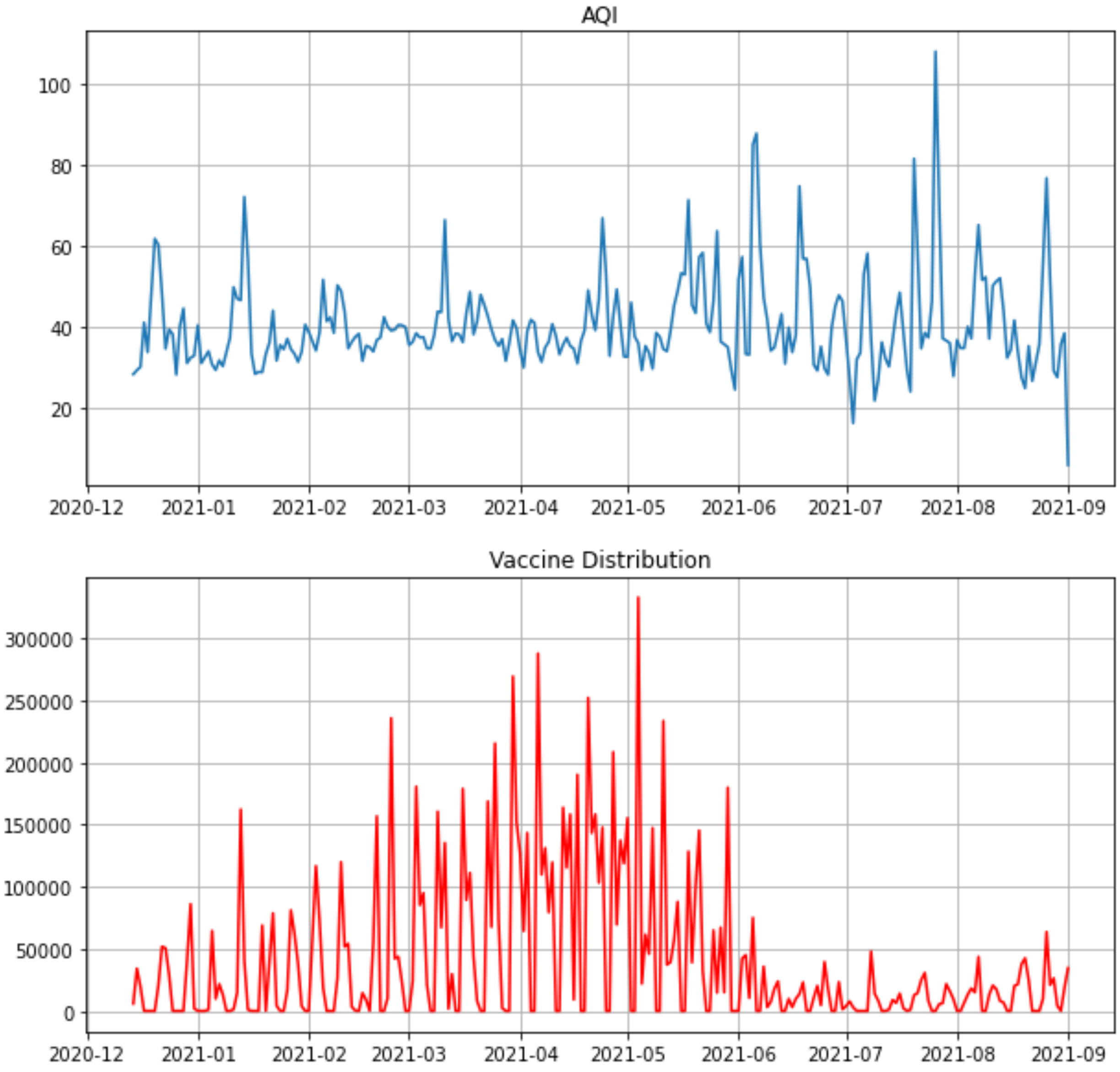
In [300]:

```
vaccines_ma = get_state_wise_daily(vaccines_filtered, "MA")
vaccines_ma = vaccines_ma[vaccines_ma["Distributed"] >= 0]
vaccines_ma.head()

aqi_ma = aqi[["state", "Date", "AQI", "Category", "defining_param"]][a

aqi_ma["Date"] = pd.to_datetime(aqi_ma["Date"])
aqi_ma.sort_values(by="Date", inplace=True)
aqi_ma.reset_index(drop=True, inplace=True)
aqi_ma.head()

aqi_ma_mean = aqi_ma[["state", "Date", "AQI", "Category", "defining_pa
aqi_ma_mean.reset_index(inplace=True)
aqi_ma_mean.head()

vaccines_aqi_ma = pd.merge(vaccines_ma, aqi_ma_mean, on=("Date"))
vaccines_aqi_ma.head()

vaccines_ma_x, vaccines_ma_y = generate_eCDF(vaccines_aqi_ma.Distribut
aqi_ma_x, aqi_ma_y = generate_eCDF(vaccines_aqi_ma.AQI.to_list())
print("KS Statistic", ks_test_2_sample(vaccines_ma_x, vaccines_ma_y, a

print("Pearson Correlation Coefficient", pearson_corr(vaccines_aqi_ma)
```

```
KS Statistic 0.6807692307692315
Pearson Correlation Coefficient 0.01600651989255773
```

In [271]:

```
f, axs = plt.subplots(2, 1, figsize=(10, 10))
# axs[0].subplot(2, 1, 1)
axs[0].grid()
axs[0].plot(vaccines_aqi_ma["Date"], vaccines_aqi_ma["AQI"])
axs[0].title.set_text("AQI")
# plt.subplot(2, 1, 2)
axs[1].grid()
axs[1].plot(vaccines_aqi_ma["Date"], vaccines_aqi_ma["Distributed"], c
axs[1].title.set_text("Vaccine Distribution")
# plt.show()
```

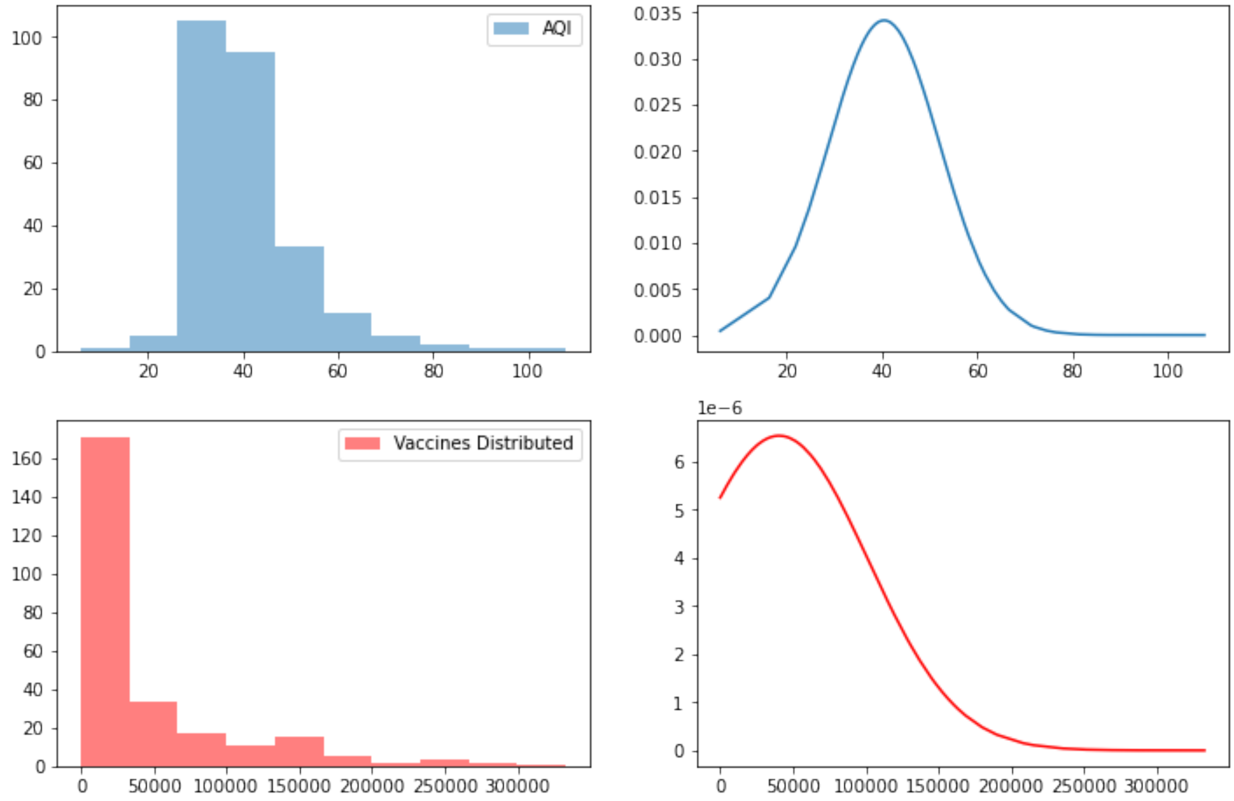


In [286]:

```
import numpy as np
import matplotlib.pyplot as plt
import statistics

x_axis = sorted(vaccines_aqi_ma.AQI.to_list())
x_axis_2 = sorted(vaccines_aqi_ma.Distributed.to_list())

mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)

mean2 = statistics.mean(x_axis_2)
sd2 = statistics.stdev(x_axis_2)

f, axs = plt.subplots(2, 2, figsize=(12, 8))
axs[0][0].hist(x_axis, alpha=0.5, label="AQI")
axs[0][1].plot(x_axis, norm.pdf(x_axis, mean, sd))
axs[0][0].legend(loc="upper right")
# axs[0][1].legend(loc="upper right")

axs[1][0].hist(x_axis_2, alpha=0.5, label="Vaccines Distributed", colo
axs[1][1].plot(x_axis_2, norm.pdf(x_axis_2, mean2, sd2), color="red")
axs[1][0].legend(loc="upper right")
# axs[1][1].legend(loc="upper right")
```

Out[286]:

```
<matplotlib.legend.Legend at 0x7f7db8878ca0>
```



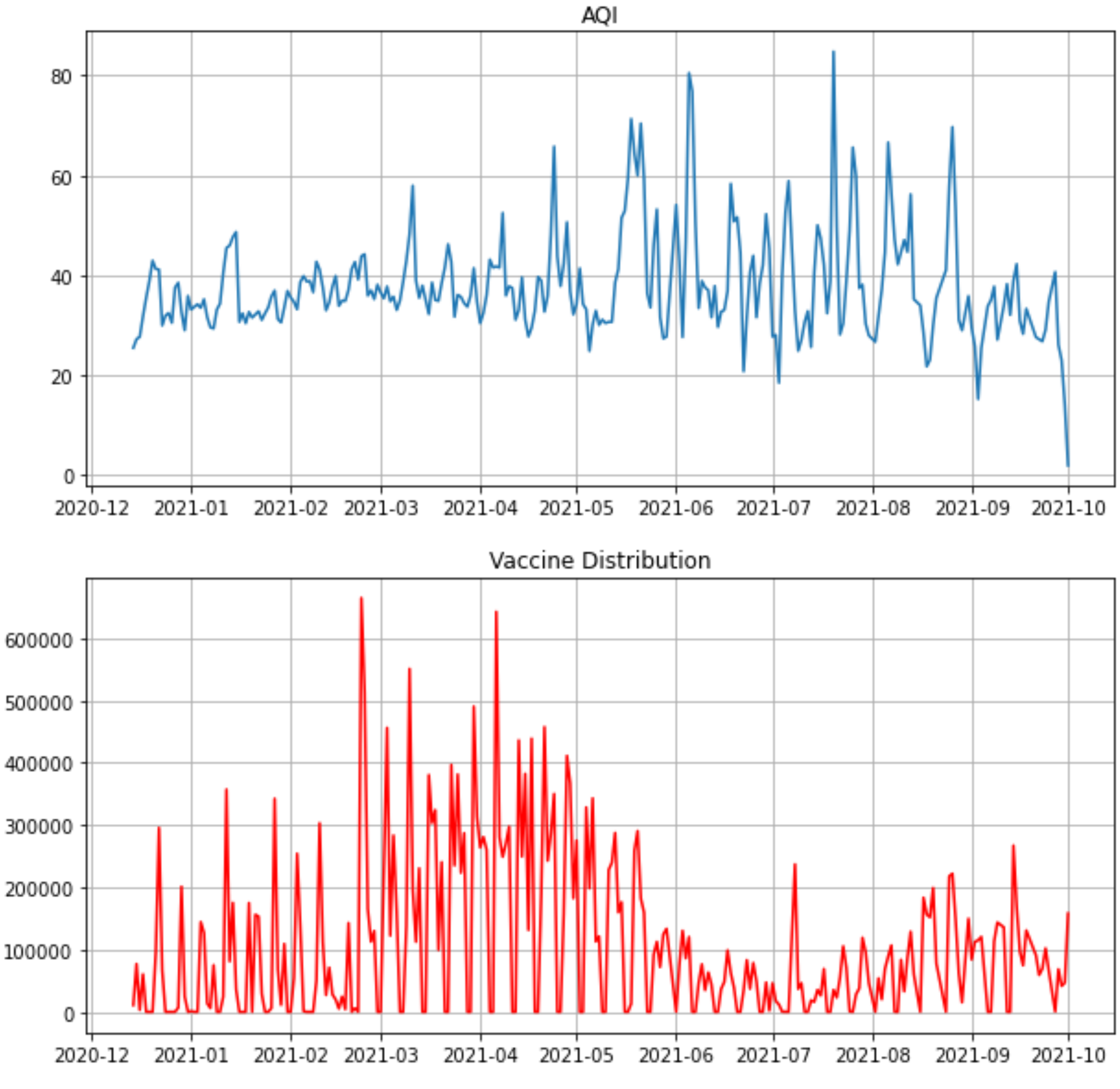
New York

In [301]:

```
vaccines_ny = get_state_wise_daily(vaccines_filtered, "NY")
vaccines_ny = vaccines_ny[vaccines_ny["Distributed"] >= 0]
vaccines_ny.head()

aqi_ny = aqi[["state", "Date", "AQI", "Category", "defining_param"]][a

aqi_ny["Date"] = pd.to_datetime(aqi_ny["Date"])
aqi_ny.sort_values(by="Date", inplace=True)
aqi_ny.reset_index(drop=True, inplace=True)
aqi_ny.head()

aqi_ny_mean = aqi_ny[["state", "Date", "AQI", "Category", "defining_pa
aqi_ny_mean.reset_index(inplace=True)
aqi_ny_mean.head()

vaccines_aqi_ny = pd.merge(vaccines_ny, aqi_ny_mean, on=("Date"))
vaccines_aqi_ny.head()

vaccines_ny_x, vaccines_ny_y = generate_eCDF(vaccines_aqi_ny.Distribut
aqi_ny_x, aqi_ny_y = generate_eCDF(vaccines_aqi_ny.AQI.to_list())
print("KS Statistic", ks_test_2_sample(vaccines_ny_x, vaccines_ny_y, a

print("Pearson Correlation Coefficient", pearson_corr(vaccines_aqi_ny)
```

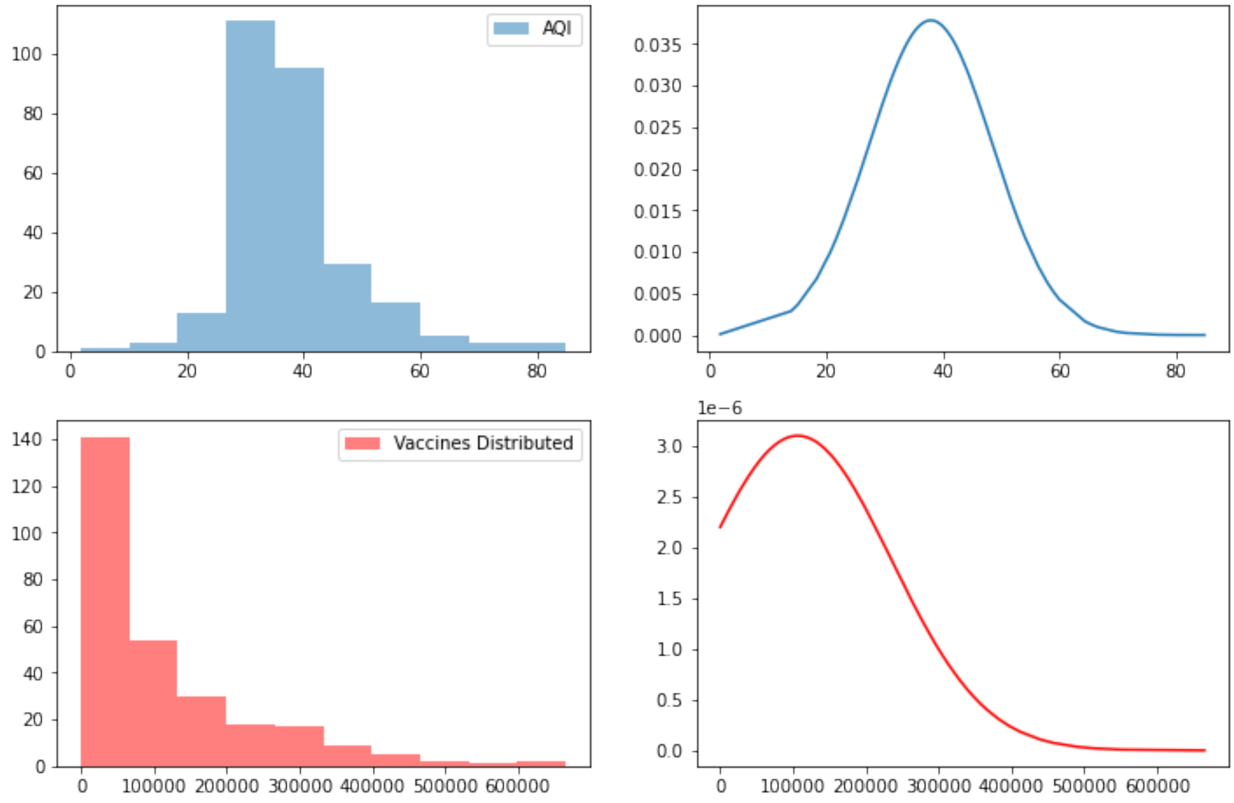
```
KS Statistic 0.7383512544802865
Pearson Correlation Coefficient 0.05423943373227678
```

```
In [273]: f, axs = plt.subplots(2, 1, figsize=(10, 10))
          # axs[0].subplot(2, 1, 1)
          axs[0].grid()
          axs[0].plot(vaccines_aqi_ny["Date"], vaccines_aqi_ny["AQI"])
          axs[0].title.set_text("AQI")
          # plt.subplot(2, 1, 2)
          axs[1].grid()
          axs[1].plot(vaccines_aqi_ny["Date"], vaccines_aqi_ny["Distributed"], c
          axs[1].title.set_text("Vaccine Distribution")
          # plt.show()
```



In [285]:
```
import numpy as np
import matplotlib.pyplot as plt
import statistics

x_axis = sorted(vaccines_aqi_ny.AQI.to_list())
x_axis_2 = sorted(vaccines_aqi_ny.Distributed.to_list())

mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)

mean2 = statistics.mean(x_axis_2)
sd2 = statistics.stdev(x_axis_2)

f, axs = plt.subplots(2, 2, figsize=(12, 8))
axs[0][0].hist(x_axis, alpha=0.5, label="AQI")
axs[0][1].plot(x_axis, norm.pdf(x_axis, mean, sd))
axs[0][0].legend(loc="upper right")
# axs[0][1].legend(loc="upper right")

axs[1][0].hist(x_axis_2, alpha=0.5, label="Vaccines Distributed", colc
axs[1][1].plot(x_axis_2, norm.pdf(x_axis_2, mean2, sd2), color="red")
axs[1][0].legend(loc="upper right")
# axs[1][1].legend(loc="upper right")
```

Out[285]:
```
<matplotlib.legend.Legend at 0x7f7e1ddd9280>
```



Results

The results of theses states are again the same as before, which signifies that the vaccine manufacturing and distribution does not provide dependency to the AQI of those states.