In [4]:

```python
import boto3
import pandas as pd
from sagemaker import get_execution_role

role = get_execution_role()
bucket='sagemaker-covid19data'
data_key = 'july09.csv'
data_location = 's3://{}/{}'.format(bucket, data_key)

churn = pd.read_csv(data_location)
```

In [5]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import io
import os
import sys
import time
import json
from IPython.display import display
from time import strftime, gmtime
import sagemaker
from sagemaker.predictor import csv_serializer
```

In [7]:

```python
churn.columns
```

Out[7]:

```
Index(['name', 'level', 'city', 'county', 'state', 'country', 'cases',
       'deaths', 'recovered', 'tested', 'active', 'population',
       'populationDensity', 'lat', 'long', 'url', 'hospitalized_current',
       'rating', 'tz', 'featureId', 'countryId', 'stateId', 'discharged',
       'countyId', 'aggregate', 'hospitalized', 'icu_current', 'icu',
       'publishedDate'],
      dtype='object')
```

In [5]:

```python
pd.set_option('display.max_columns', 500)
churn
```

Out[5]:

	name	level	city	county	state	country	cases	deaths	recovered	tested	active	
0	Lower Austria, Austria	state	NaN	NaN	Lower Austria	Austria	3047.0	106.0	2835.0	NaN	106.0	1
1	Vorarlberg, Austria	state	NaN	NaN	Vorarlberg	Austria	913.0	18.0	891.0	NaN	4.0	3
2	Upper Austria, Austria	state	NaN	NaN	Upper Austria	Austria	2848.0	58.0	2306.0	NaN	484.0	1
3	Styria, Austria	state	NaN	NaN	Styria	Austria	1921.0	141.0	1688.0	NaN	92.0	1
4	Burgenland, Austria	state	NaN	NaN	Burgenland	Austria	365.0	11.0	338.0	NaN	16.0	2
...	...	...	...	...	...	...	...	...	...	...	...	

In [7]:

```python
churn['country'].nunique()
```

Out[7]:

192

In [8]:

```python
churn['deaths'].plot(kind='hist')
```
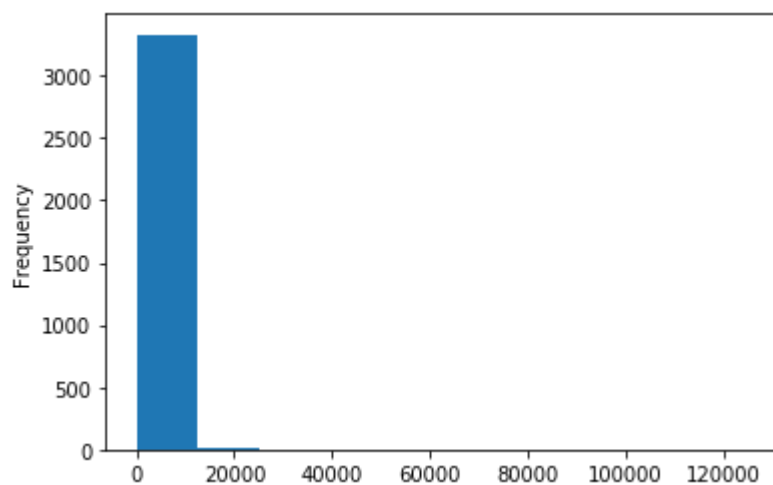
Out[8]:

<matplotlib.axes._subplots.AxesSubplot at 0x7fd1026968d0>

In [16]:

```
churn['deaths'].describe()
```

Out[16]:

```
count      3342.000000
mean        246.538300
std        2922.810966
min           0.000000
25%           0.000000
50%           2.000000
75%          17.000000
max      124723.000000
Name: deaths, dtype: float64
```

In [11]:

```
churn.describe().T
```

Out[11]:

	count	mean	std	min	25%	
city	0.0	NaN	NaN	NaN	NaN	
cases	3963.0	4.731101e+03	6.050708e+04	0.000000	18.000000	98.00(
deaths	3342.0	2.465383e+02	2.922811e+03	0.000000	0.000000	2.00(
recovered	685.0	8.175607e+03	5.428528e+04	0.000000	25.000000	108.00(
tested	587.0	1.643132e+05	1.595076e+06	0.000000	1023.000000	3158.00(
active	683.0	4.178505e+03	2.763800e+04	-3583.000000	6.000000	23.00(
population	4014.0	2.500440e+06	3.307690e+07	86.000000	10633.500000	29337.00(
populationDensity	3950.0	1.709854e+02	2.054800e+03	0.013715	8.233425	22.30{
lat	4022.0	3.869693e+01	1.046588e+01	-47.178000	34.782000	39.04(
long	4022.0	-6.773862e+01	5.467216e+01	-170.128000	-96.401375	-86.32'
hospitalized_current	59.0	8.976271e+01	2.839988e+02	0.000000	1.000000	5.00(
rating	4022.0	5.126608e-01	1.402721e-01	0.176471	0.470588	0.54{
discharged	30.0	2.079567e+03	3.523192e+03	1.000000	258.250000	847.50(
hospitalized	274.0	1.540255e+02	8.514175e+02	0.000000	3.000000	16.00(
icu_current	15.0	8.893333e+01	1.389108e+02	2.000000	18.000000	40.00(
icu	1.0	5.240000e+02	NaN	524.000000	524.000000	524.00(

In [ ]: