In [1]:

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

Import the dataset

In [2]:

```
data = pd.read_csv(r"C:\Users\user\Desktop\7_uber.csv")
```

Display top 7 and last 6 rows and print the output

In [3]:

```
data.head(7)
```

Out[3]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitud
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.73835
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.72822
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.74077
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.79084
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.74408
5	44470845	2011-02-12 02:27:09.0000006	4.9	2011-02-12 02:27:09 UTC	-73.969019	40.75591
6	48725865	2014-10-12 07:04:00.0000002	24.5	2014-10-12 07:04:00 UTC	-73.961447	40.69396

In [4]:
```
data.tail(6)
```

Out[4]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_l
199994	3189201	2014-01-31 14:42:00.000000181	12.0	2014-01-31 14:42:00 UTC	-73.983070	40.
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.

Fill with a constant value and print the output

```
In [5]: data.fillna('8')
```

```
Out[5]:
```

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_l
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.
...	...	...	...	...	...	
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.

200000 rows × 9 columns

Drop the column with missing values and print the output

In [6]:
```
data.dropna(axis=1)
```

Out[6]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_l
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.
...	...	...	...	...	...	
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.

200000 rows × 7 columns

Drop the row with missing values and print the output

In [7]:

```
data.dropna()
```

Out[7]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_l
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.
...	...	...	...	...	...	
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.

199999 rows × 9 columns

To check the presence of missing values in your dataframe

In [8]:

```
data.isna()
```

Out[8]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropc
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
...	...	...	...	...	...	...	
199995	False	False	False	False	False	False	
199996	False	False	False	False	False	False	
199997	False	False	False	False	False	False	
199998	False	False	False	False	False	False	
199999	False	False	False	False	False	False	

200000 rows × 9 columns

Use operators and check the condition and print the output

In [9]:
```
data[data['fare_amount']>15]
```

Out[9]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_l
4	17610152	2014-08-28 17:47:00.000000188	16.00	2014-08-28 17:47:00 UTC	-73.925023	40.
6	48725865	2014-10-12 07:04:00.0000002	24.50	2014-10-12 07:04:00 UTC	-73.961447	40.
30	31945670	2011-05-21 09:00:00.00000031	25.70	2011-05-21 09:00:00 UTC	-73.944815	40.
34	19277743	2014-06-04 06:49:00.000000102	39.50	2014-06-04 06:49:00 UTC	-73.788080	40.
39	38703737	2014-02-13 17:57:00.000000102	29.00	2014-02-13 17:57:00 UTC	-73.992600	40.
...	...	...	...	...	...	
199977	21117828	2012-11-20 21:04:30.0000001	43.50	2012-11-20 21:04:30 UTC	-73.996671	40.
199982	13096190	2014-08-06 11:06:06.0000001	57.33	2014-08-06 11:06:06 UTC	-73.969204	40.
199985	25830754	2015-04-18 15:16:06.0000005	24.00	2015-04-18 15:16:06 UTC	-74.005089	40.
199991	13512837	2015-06-08 10:49:14.0000001	17.50	2015-06-08 10:49:14 UTC	-73.981453	40.
199997	27804658	2009-06-29 00:42:00.00000078	30.90	2009-06-29 00:42:00 UTC	-73.986017	40.

35251 rows × 9 columns

Display your output using loc and iloc, row and column heading

In [10]:
```
data.iloc[199800:199900]
```

Out[10]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_l
199800	2101940	2011-02-27 22:13:10.0000001	8.90	2011-02-27 22:13:10 UTC	-73.965632	40.
199801	31241987	2013-05-02 07:33:00.000000248	23.50	2013-05-02 07:33:00 UTC	-73.965587	40.
199802	20588739	2013-06-20 23:57:54.0000002	9.50	2013-06-20 23:57:54 UTC	-73.989029	40.
199803	38936922	2012-11-22 09:07:00.00000096	13.00	2012-11-22 09:07:00 UTC	-73.998835	40.
199804	15237188	2015-03-15 21:29:53.0000006	8.50	2015-03-15 21:29:53 UTC	-73.950699	40.
...	...	...	...	...	...	
199895	55154831	2010-02-24 14:28:29.0000004	12.10	2010-02-24 14:28:29 UTC	-73.994908	40.
199896	33507841	2009-12-25 09:49:00.00000039	8.10	2009-12-25 09:49:00 UTC	-73.986803	40.
199897	39011824	2009-07-31 12:04:35.0000003	33.87	2009-07-31 12:04:35 UTC	-73.870864	40.
199898	32579637	2010-06-06 04:03:00.00000051	5.30	2010-06-06 04:03:00 UTC	-73.998410	40.
199899	39792005	2011-07-16 11:35:03.0000004	6.90	2011-07-16 11:35:03 UTC	-73.967501	40.

100 rows × 9 columns

In [11]:
```
data.loc[100:200]
```

Out[11]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitu
100	29350780	2014-02-19 18:02:00.00000011	9.0	2014-02-19 18:02:00 UTC	-73.958280	40.768
101	30977645	2011-02-20 21:01:58.0000003	4.1	2011-02-20 21:01:58 UTC	-73.977633	40.753
102	7290820	2011-02-04 11:48:00.000000216	14.9	2011-02-04 11:48:00 UTC	-73.962218	40.640
103	47729464	2009-07-31 09:29:00.000000167	6.5	2009-07-31 09:29:00 UTC	-74.000545	40.737
104	46435788	2015-05-15 18:58:16.0000001	43.0	2015-05-15 18:58:16 UTC	-73.862701	40.768
...	...	...	...	...	...	
196	51452192	2009-05-12 10:32:00.000000154	24.0	2009-05-12 10:32:00 UTC	-73.981558	40.783
197	45317989	2012-08-07 20:53:18.0000001	10.5	2012-08-07 20:53:18 UTC	-73.965930	40.805
198	41858701	2009-09-24 16:21:42.0000001	8.9	2009-09-24 16:21:42 UTC	-73.952080	40.790
199	13472186	2011-04-03 00:01:40.0000002	14.1	2011-04-03 00:01:40 UTC	-74.000190	40.718
200	12508194	2012-03-22 07:14:24.0000002	6.1	2012-03-22 07:14:24 UTC	-73.965017	40.764

101 rows × 9 columns

Display the statistical summary of data

In [12]:
```
data.describe()
```

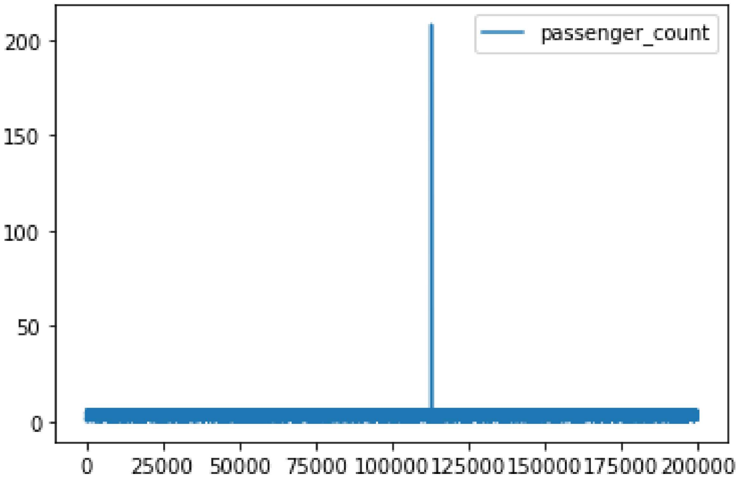
Out[12]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff
count	2.000000e+05	200000.000000	200000.000000	200000.000000	199999.000000	19999
mean	2.771250e+07	11.359955	-72.527638	39.935885	-72.525292	3
std	1.601382e+07	9.901776	11.437787	7.720539	13.117408	
min	1.000000e+00	-52.000000	-1340.648410	-74.015515	-3356.666300	-88
25%	1.382535e+07	6.000000	-73.992065	40.734796	-73.991407	4
50%	2.774550e+07	8.500000	-73.981823	40.752592	-73.980093	4
75%	4.155530e+07	12.500000	-73.967154	40.767158	-73.963658	4
max	5.542357e+07	499.000000	57.418457	1644.421482	1153.572603	87

Visualization

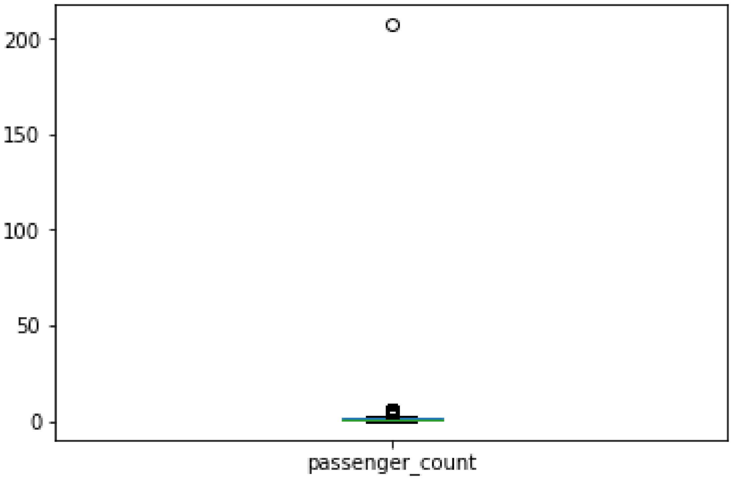
In [13]:
```
df=data[['key','passenger_count']]
df.plot.line()
```

Out[13]:
```
<AxesSubplot:>
```



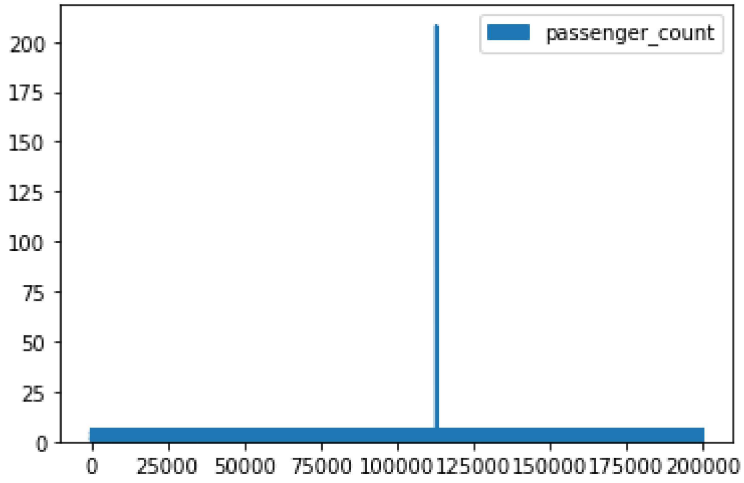
In [14]:
```
df.plot.box()
```

Out[14]:
```
<AxesSubplot:>
```



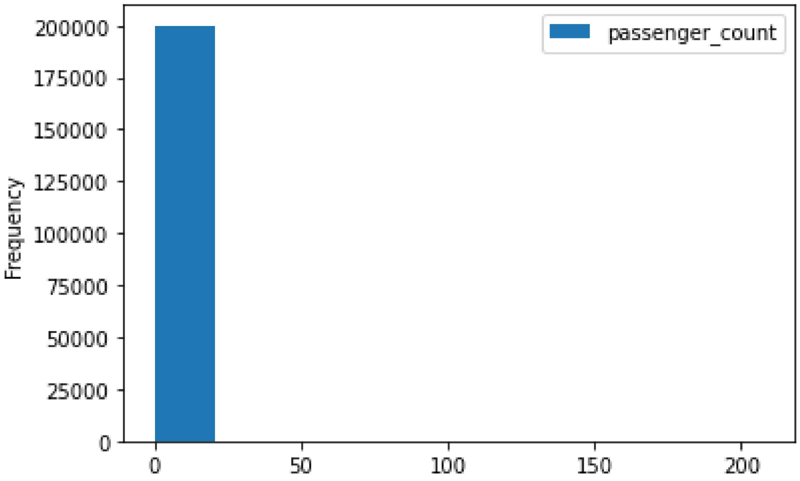
```
In [15]: df.plot.area()
```

```
Out[15]: <AxesSubplot:>
```



```
In [16]: df.plot.hist()
```

```
Out[16]: <AxesSubplot:ylabel='Frequency'>
```


passenger_count
Frequency