

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as pp
```

```
In [2]: d=pd.read_csv(r"C:\Users\user\Downloads\uber - uber.csv")
d
```

		Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_1
0	24238194	2015-05-07 19:52:06		7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-
1	27835199	2009-07-17 20:04:56		7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-
2	44984355	2009-08-24 21:45:00		12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-
3	25894730	2009-06-26 8:22:21		5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-
4	17610152	2014-08-28 17:47:00		16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-
...
199995	42598914	2012-10-28 10:49:00		3.0	2012-10-28 10:49:00 UTC	-73.987042	40.739367	-
199996	16382965	2014-03-14 1:09:00		7.5	2014-03-14 01:09:00 UTC	-73.984722	40.736837	-
199997	27804658	2009-06-29 0:42:00		30.9	2009-06-29 00:42:00 UTC	-73.986017	40.756487	-
199998	20259894	2015-05-20 14:56:25		14.5	2015-05-20 14:56:25 UTC	-73.997124	40.725452	-
199999	11951496	2010-05-15 4:08:00		14.1	2010-05-15 04:08:00 UTC	-73.984395	40.720077	-

200000 rows × 9 columns

```
In [3]: d.head()
```

Out[3]:

		key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude
		Unnamed: 0					
0	24238194	2015-05-07 19:52:06	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-73.9991
1	27835199	2009-07-17 20:04:56	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-73.9941
2	44984355	2009-08-24 21:45:00	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-73.9621
3	25894730	2009-06-26 8:22:21	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-73.9651
4	17610152	2014-08-28 17:47:00	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-73.9730

◀ ▶

In [4]:

```
d.tail()
```

		key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude
		Unnamed: 0					
199995	42598914	2012-10-28 10:49:00	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.739367	-73.987042
199996	16382965	2014-03-14 1:09:00	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.736837	-73.984722
199997	27804658	2009-06-29 0:42:00	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.756487	-73.986017
199998	20259894	2015-05-20 14:56:25	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.725452	-73.997124
199999	11951496	2010-05-15 4:08:00	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.720077	-73.984395

◀ ▶

In [5]:

```
d.describe()
```

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
count	2.000000e+05	200000.000000	200000.000000	200000.000000	199999.000000	199999.000000
mean	2.771250e+07	11.359955	-72.527638	39.935885	-72.525292	39.923811
std	1.601382e+07	9.901776	11.437787	7.720539	13.117408	6.794811

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
min	1.000000e+00	-52.000000	-1340.648410	-74.015515	-3356.666300	-881.9855
25%	1.382535e+07	6.000000	-73.992065	40.734796	-73.991407	40.7338
50%	2.774550e+07	8.500000	-73.981823	40.752592	-73.980093	40.7530
75%	4.155530e+07	12.500000	-73.967153	40.767158	-73.963659	40.7680
max	5.542357e+07	499.000000	57.418457	1644.421482	1153.572603	872.6976

In [6]: `np.shape(d)`

Out[6]: (200000, 9)

In [7]: `np.size(d)`

Out[7]: 1800000

In [8]: `d.isnull()`

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

200000 rows × 9 columns

In [9]: `d.fillna(value=5)`

Out[9]:

		Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_l
0	24238194	2015-05-07 19:52:06		7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-
1	27835199	2009-07-17 20:04:56		7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-
2	44984355	2009-08-24 21:45:00		12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-
3	25894730	2009-06-26 8:22:21		5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-
4	17610152	2014-08-28 17:47:00		16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-
...
199995	42598914	2012-10-28 10:49:00		3.0	2012-10-28 10:49:00 UTC	-73.987042	40.739367	-
199996	16382965	2014-03-14 1:09:00		7.5	2014-03-14 01:09:00 UTC	-73.984722	40.736837	-
199997	27804658	2009-06-29 0:42:00		30.9	2009-06-29 00:42:00 UTC	-73.986017	40.756487	-
199998	20259894	2015-05-20 14:56:25		14.5	2015-05-20 14:56:25 UTC	-73.997124	40.725452	-
199999	11951496	2010-05-15 4:08:00		14.1	2010-05-15 04:08:00 UTC	-73.984395	40.720077	-

200000 rows × 9 columns



In [10]:

d.dropna()

Out[10]:

		Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_l
0	24238194	2015-05-07 19:52:06		7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-
1	27835199	2009-07-17 20:04:56		7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_l
2	44984355	2009-08-24 21:45:00	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-
3	25894730	2009-06-26 8:22:21	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-
4	17610152	2014-08-28 17:47:00	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-
...
199995	42598914	2012-10-28 10:49:00	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.739367	-
199996	16382965	2014-03-14 1:09:00	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.736837	-
199997	27804658	2009-06-29 0:42:00	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.756487	-
199998	20259894	2015-05-20 14:56:25	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.725452	-
199999	11951496	2010-05-15 4:08:00	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.720077	-

199999 rows × 9 columns

picking any two columns

```
In [20]: d=d[["fare_amount","pickup_latitude"]]
d
```

```
-----
KeyError                                                 Traceback (most recent call last)
<ipython-input-20-ead2789366ea> in <module>
----> 1 d=d[["fare_amount","pickup_latitude"]]
      2 d

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    3028         if is_iterator(key):
    3029             key = list(key)
-> 3030         indexer = self.loc._get_listlike_indexer(key, axis=1, raise_missing
=True)[1]
    3031
    3032         # take() does not accept boolean indexers
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py in _get_listlike_indexer(self, key, axis, raise_missing)
```

```

1264             keyarr, indexer, new_indexer = ax._reindex_non_unique(keyarr)
1265
-> 1266         self._validate_read_indexer(keyarr, indexer, axis, raise_missing=raise_
missing)
1267         return keyarr, indexer
1268

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py in _validate_read_indexer(self, key, indexer, axis, raise_missing)
    1314         if raise_missing:
    1315             not_found = list(set(key) - set(ax))
-> 1316             raise KeyError(f"{not_found} not in index")
    1317
    1318         not_found = key[missing_mask]

KeyError: "['fare_amount'] not in index"

```

visualization

In [12]:

```
d=d.head(1000)
d
```

Out[12]:

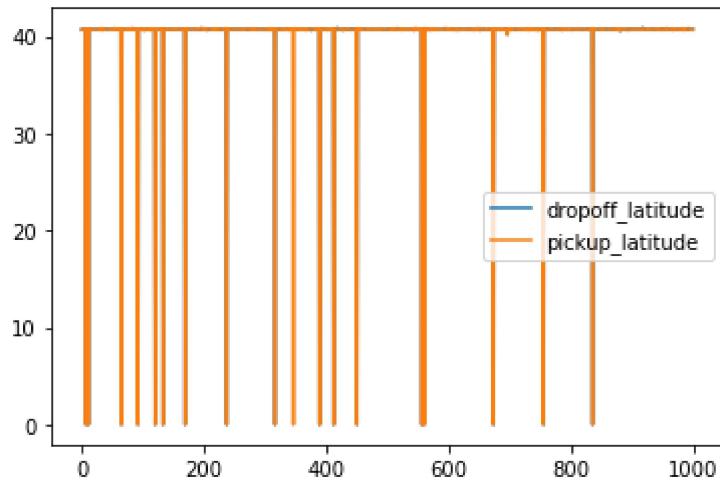
	dropoff_latitude	pickup_latitude
0	40.723217	40.738354
1	40.750325	40.728225
2	40.772647	40.740770
3	40.803349	40.790844
4	40.761247	40.744085
...
995	40.766960	40.757577
996	40.775632	40.757054
997	40.783833	40.765078
998	40.775048	40.784730
999	40.758208	40.729944

1000 rows × 2 columns

In [13]:

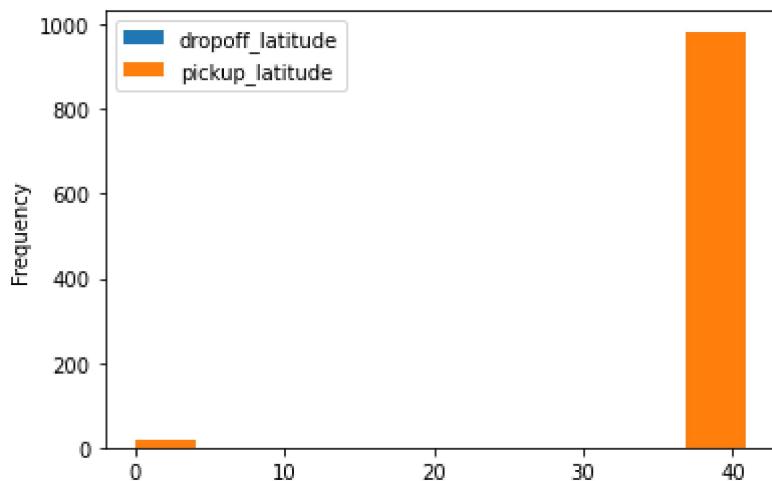
```
d.plot.line()
```

Out[13]: <AxesSubplot:>



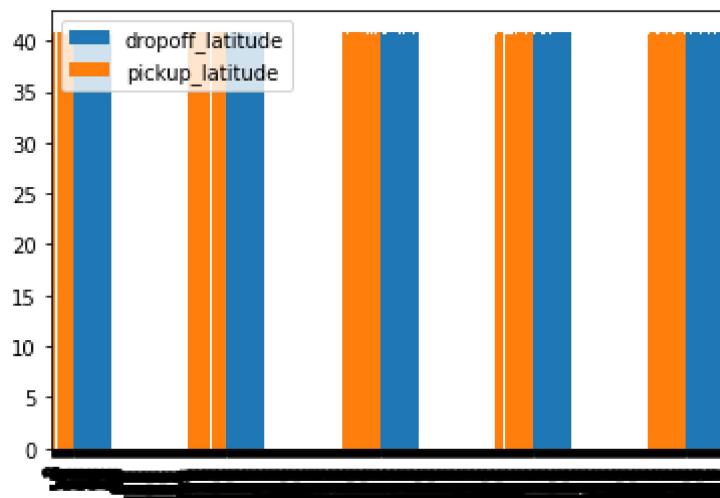
```
In [14]: d.plot.hist()
```

```
Out[14]: <AxesSubplot:ylabel='Frequency'>
```



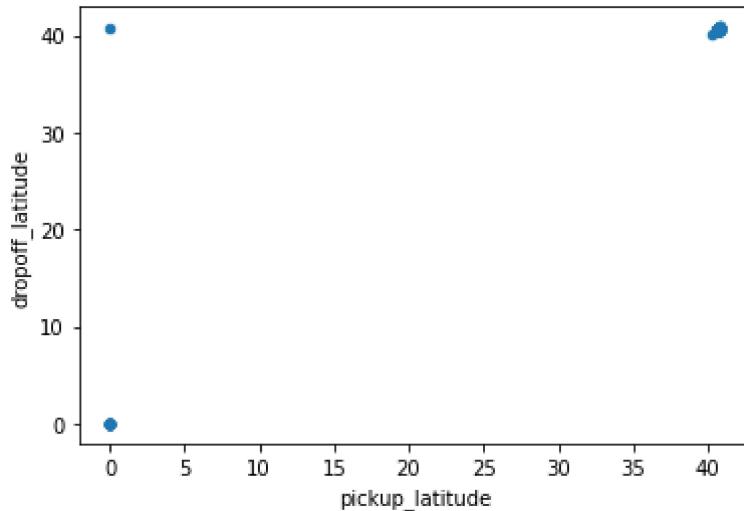
```
In [15]: d.plot.bar()
```

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



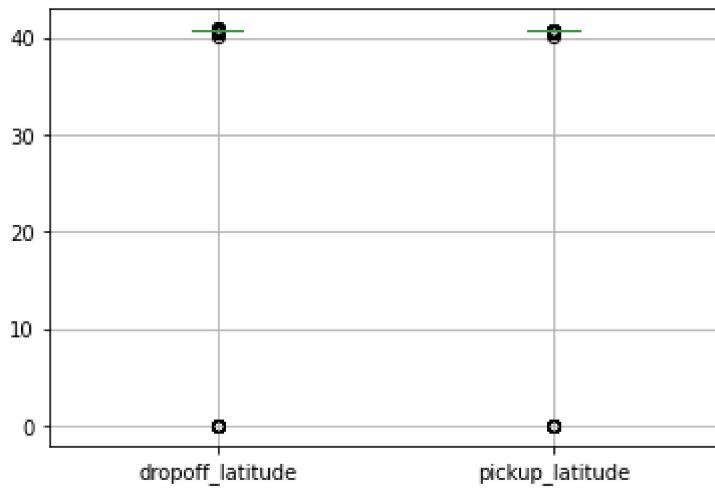
```
In [16]: d.plot.scatter(x="pickup_latitude",y="dropoff_latitude")
```

```
Out[16]: <AxesSubplot:xlabel='pickup_latitude', ylabel='dropoff_latitude'>
```



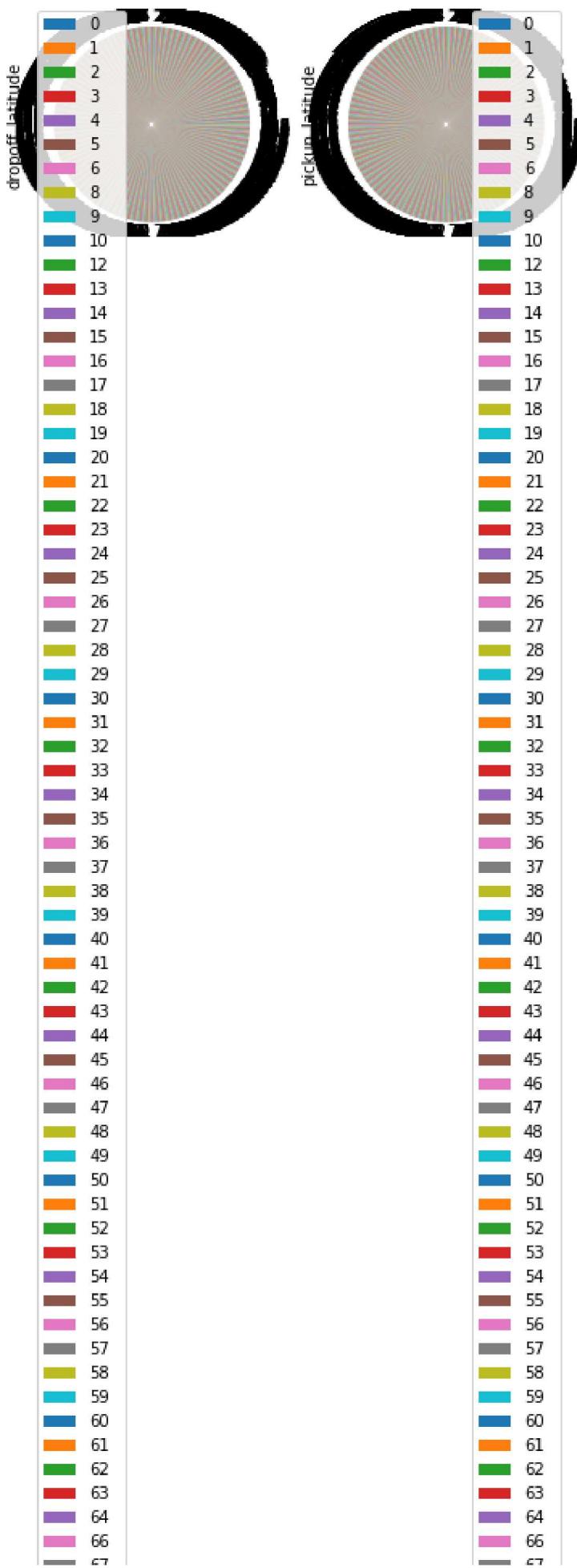
```
In [17]: d.boxplot()
```

```
Out[17]: <AxesSubplot:>
```



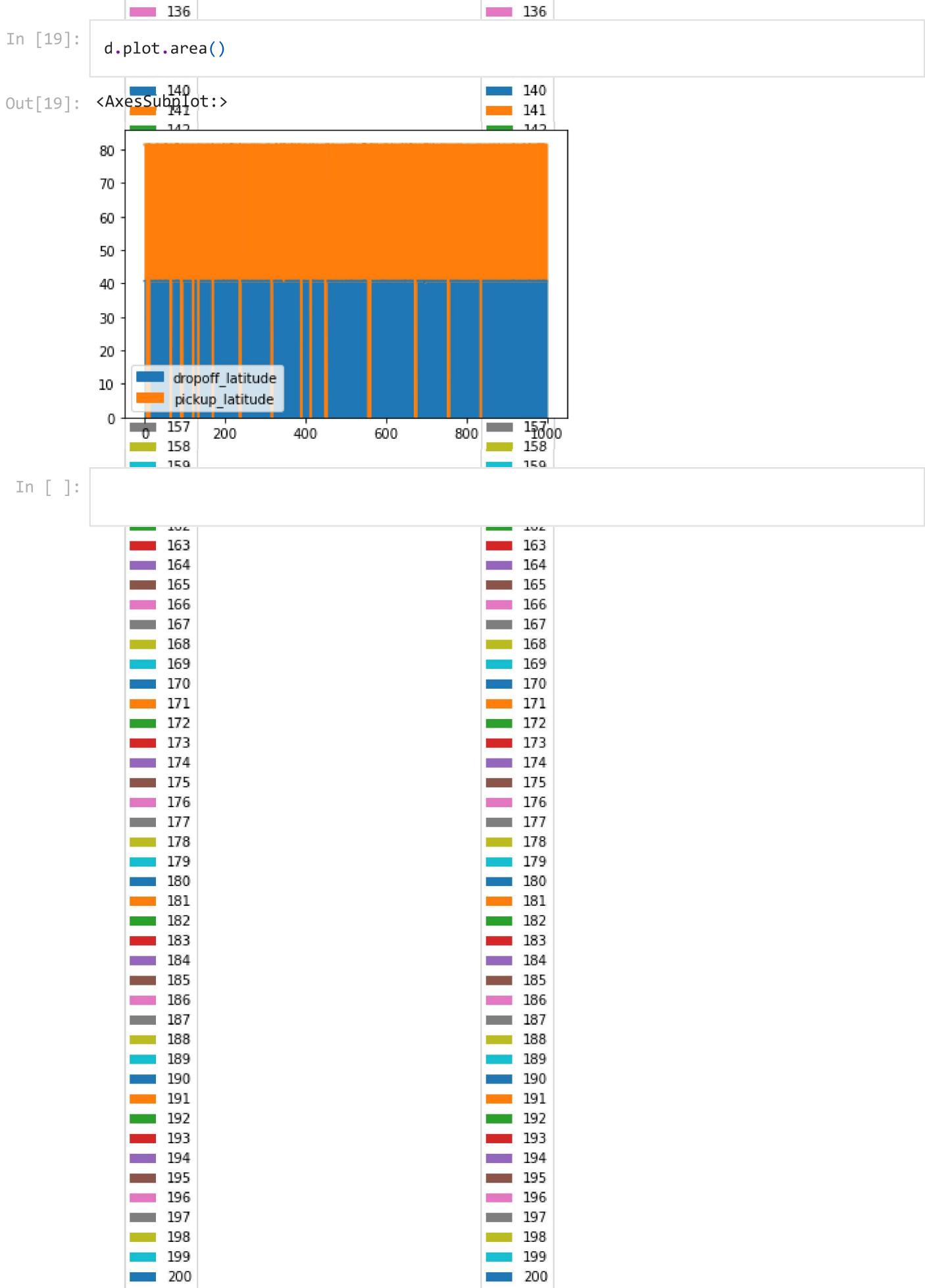
```
In [18]: d.plot.pie(subplots=True)
```

```
Out[18]: array([<AxesSubplot:ylabel='dropoff_latitude'>,
   <AxesSubplot:ylabel='pickup_latitude'>], dtype=object)
```



68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
121
122
123
124
125
126
127
128
129
130
131
132
134
135

68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
121
122
123
124
125
126
127
128
129
130
131
132
134
135



201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267

201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267

267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333

267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333

334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
390
391
392
393
394
395
396
397
398
399
334
335
336
337
338
339
340
341
342
343
344
345
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
390
391
392
393
394
395
396
397
398
399
400

400	401
401	402
402	403
403	404
404	405
405	406
406	407
407	408
408	409
409	410
410	411
411	413
413	414
414	415
415	416
416	417
417	418
418	419
419	420
420	421
421	422
422	423
423	424
424	425
425	426
426	427
427	428
428	429
429	430
430	431
431	432
432	433
433	434
434	435
435	436
436	437
437	438
438	439
439	440
440	441
441	442
442	443
443	444
444	445
445	446
446	447
447	448
448	449
449	451
451	452
452	453
453	454
454	455
455	456
456	457
457	458
458	459
459	460
460	461
461	462
462	463
463	464
464	465
465	466
466	467
	467

467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

533	534
534	535
535	536
536	537
537	538
538	539
539	540
540	541
541	542
542	543
543	544
544	545
545	546
546	547
547	548
548	549
549	550
550	551
551	552
552	553
553	554
554	555
555	556
556	557
557	558
558	559
559	560
560	561
561	562
562	563
563	564
564	565
565	566
566	567
567	568
568	569
569	570
570	571
571	572
572	573
573	574
574	575
575	576
576	577
577	578
578	579
579	580
580	581
581	582
582	583
583	584
584	585
585	586
586	587
587	588
588	589
589	590
590	591
591	592
592	593
593	594
594	595
595	596
596	597
597	598
598	599
599	600

600	601
601	602
602	603
603	604
604	605
605	606
606	607
607	608
608	609
609	610
610	611
611	612
612	613
613	614
614	615
615	616
616	617
617	618
618	619
619	620
620	621
621	622
622	623
623	624
624	625
625	626
626	627
627	628
628	629
629	630
630	631
631	632
632	633
633	634
634	635
635	636
636	637
637	638
638	639
639	640
640	641
641	642
642	643
643	644
644	645
645	646
646	647
647	648
648	649
649	650
650	651
651	652
652	653
653	654
654	655
655	656
656	657
657	658
658	659
659	660
660	661
661	662
662	663
663	664
664	665
665	666

665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

732	733
733	734
734	735
735	736
736	737
737	738
738	739
739	740
740	741
741	742
742	743
743	744
744	745
745	746
746	747
747	748
748	749
749	750
750	751
751	752
752	753
753	754
754	756
756	757
757	758
758	759
759	760
760	761
761	762
762	763
763	764
764	765
765	766
766	767
767	768
768	769
769	770
770	771
771	772
772	773
773	774
774	775
775	776
776	777
777	778
778	779
779	780
780	781
781	782
782	783
783	784
784	785
785	786
786	787
787	788
788	789
789	790
790	791
791	792
792	793
793	794
794	795
795	796
796	797
797	798

[Color Box]	798
[Color Box]	799
[Color Box]	800
[Color Box]	801
[Color Box]	802
[Color Box]	803
[Color Box]	804
[Color Box]	805
[Color Box]	806
[Color Box]	807
[Color Box]	808
[Color Box]	809
[Color Box]	810
[Color Box]	811
[Color Box]	812
[Color Box]	813
[Color Box]	814
[Color Box]	815
[Color Box]	816
[Color Box]	817
[Color Box]	818
[Color Box]	819
[Color Box]	820
[Color Box]	821
[Color Box]	822
[Color Box]	823
[Color Box]	824
[Color Box]	825
[Color Box]	826
[Color Box]	827
[Color Box]	828
[Color Box]	829
[Color Box]	830
[Color Box]	831
[Color Box]	832
[Color Box]	833
[Color Box]	834
[Color Box]	836
[Color Box]	837
[Color Box]	838
[Color Box]	839
[Color Box]	840
[Color Box]	841
[Color Box]	842
[Color Box]	843
[Color Box]	844
[Color Box]	845
[Color Box]	846
[Color Box]	847
[Color Box]	848
[Color Box]	849
[Color Box]	850
[Color Box]	851
[Color Box]	852
[Color Box]	853
[Color Box]	854
[Color Box]	855
[Color Box]	856
[Color Box]	857
[Color Box]	858
[Color Box]	859
[Color Box]	860
[Color Box]	861
[Color Box]	862
[Color Box]	863
[Color Box]	864
[Color Box]	865

804	865
865	866
866	867
867	868
868	869
869	870
870	871
871	872
872	873
873	874
874	875
875	876
876	877
877	878
878	879
879	880
880	881
881	882
882	883
883	884
884	885
885	886
886	887
887	888
888	889
889	890
890	891
891	892
892	893
893	894
894	895
895	896
896	897
897	898
898	899
899	900
900	901
901	902
902	903
903	904
904	905
905	906
906	907
907	908
908	909
909	910
910	911
911	912
912	913
913	914
914	915
915	916
916	917
917	918
918	919
919	920
920	921
921	922
922	923
923	924
924	925
925	926
926	927
927	928
928	929
929	930

930	931
931	932
932	933
933	934
934	935
935	936
936	937
937	938
938	939
939	940
940	941
941	942
942	943
943	944
944	945
945	946
946	947
947	948
948	949
949	950
950	951
951	952
952	953
953	954
954	955
955	956
956	957
957	958
958	959
959	960
960	961
961	962
962	963
963	964
964	965
965	966
966	967
967	968
968	969
969	970
970	971
971	972
972	973
973	974
974	975
975	976
976	977
977	978
978	979
979	980
980	981
981	982
982	983
983	984
984	985
985	986
986	987
987	988
988	989
989	990
990	991
991	992
992	993
993	994
994	995

■	995
■	996
■	997
■	998
■	999

■	996
■	997
■	998
■	999