Elements	New Sort - with NumPy Array			
	Start Time (s)	For Loop Start (s)	End Time (s)	For Loop Time (s)
1000	14.93	14.93	14.95	0.02
10000	19.95	19.95	20.24	0.30
100000	14.14	14.18	15.81	1.63
1000000	2.76	3.20	19.09	15.88
10000000	42.23	43.17		112.32

Python - Sort with Normal Array				
Elements	Start Time (s)	For Loop 1 start (s)	For Loop 1 end (s)	For Loop 2 start (s)
1000	14.50	14.50	14.51	14.51
10000	57.71	57.71	57.76	57.76
100000	44.10	44.10	44.41	44.41
1000000	53.90	53.90	56.17	56.17
10000000	0.02	0.02	0.02	0.02

Elements	Trials
1000	1
	2
	3
	AVG
10000	1
	2
	3
	AVG
100000	1
	2
	3
	AVG
1000000	1
	2
	3
	AVG
10000000	1
	2
	3
	AVG

For Loop 2 end (s)	Program End (s)	Time for For Loop 1	Time for For Loop 2
14.51	14.513059	0.015619	0
57.85	57.845783	0.046881	0.084669
44.96	44.959294	0.309251	0.544727
59.84	59.890061	2.266724	3.671183
0.02	22:53.4	21.23915	34.403685

Python - Sort with Normal Array		
Time for Stat Collection (s)	Time for Bucketing (s)	Total Time (s)
0.016	0.000	0.016
0.016	0.000	0.016
0.022	0.000	0.022
0.018	**0.000**	**0.018**
0.031	0.047	0.078
0.037	0.063	0.100
0.022	0.047	0.069
0.030	0.052	0.082
0.419	0.522	0.941
0.322	0.400	0.723
0.215	0.400	0.616
0.319	**0.441**	**0.760**
2.551	3.313	5.918
2.173	3.653	5.943
2.360	3.552	5.974
2.362	**3.506**	**5.945**
23.86	47.68	78.94
22.54	47.17	78.18
26.37	45.92	79.92
24.26	**46.93**	**79.01**

Time for Bucket Sorting	Total Time Taken
0	0.015619
0	0.13155
0.002507	0.856485
0.052904	5.990811
5.829744	61.472579

Python - Sort with Complete NumPy (S			
Elements	Trials	Time for NumPy Array (s)	Time for Stat Collection (s)
1000	1	0	0.000
	2	0	0.000
	3	0	0.000
	AVG	**0**	**0.000**
10000	1	0	0.000
	2	0	0.000
	3	0	0.000
	AVG	**0**	**0.000**
100000	1	0.031	0.000
	2	0.031	0.000
	3	0.031	0.000
	AVG	**0.031**	**0.000**
1000000	1	0.378	0.022
	2	0.199	0.017
	3	0.222	0.000
	AVG	**0.267**	**0.013**

		*could be because of the functions defined IN the for loop.	
		*It has to calculate values during the for loop so that maybe tak	
out of for loop, onl	1	0.265	0.016
	2	0.185	0.016
	3	0.216	0.016
	4	0.345	0.016
	5	0.100	0.000
	6	0.085	0.016
	7	0.118	0.009
	8	0.333	0.005
	9	0.253	0.016
	AVG	0.211	0.012

*creating the numpy array also takes some time and may increa
*see how fast it can go if it is fed a numpy array instead of havi
*sorting using the python sort is taking a long time , more than
*tried changing sort used to mergesort, but run time increases e
*can try to get stats using numpy and then keep the 2d array as

ort, Arrays)		
Time for Bucketing (s)	Total Time (s)	Sort
0.016	0.016	0.000
0.016	0.016	0.000
0.038	0.038	0.000
0.023	**0.023**	0.000
0.201	0.217	0.016
0.138	0.138	0.000
0.132	0.132	0.000
0.157	**0.162**	0.005
1.371	1.587	0.216
2.042	2.390	0.348
1.569	1.800	0.231
1.661	**1.926**	0.265
16.649	42.797	26.148
15.709	43.816	28.108
17.231	48.715	31.483
16.530	**45.109**	28.580

es time.		
13.705	38.121	24.136
13.218	39.578	26.159
15.086	40.735	25.417
16.341	44.983	28.280
14.046	33.954	19.808
0.590	30.136	29.446
9.481	36.310	26.702
15.714	44.104	28.051
13.878	38.367	24.220
12.451	38.476	25.802

se significantly after large data sets are inputted
ng to converting array to one
half of the time is dedicated to this
ven more
a normal one