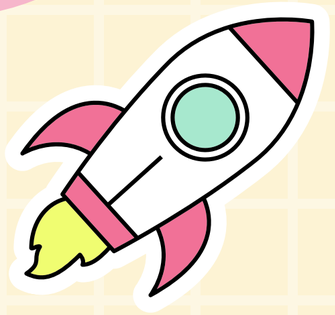
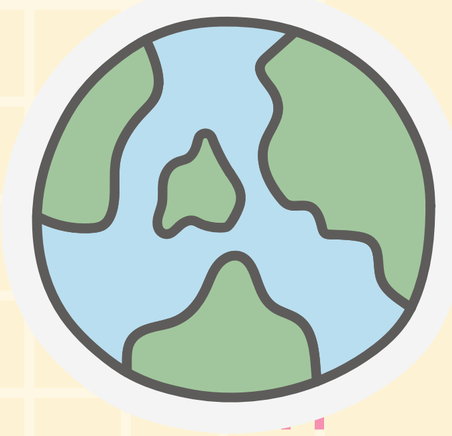




FINAL PROJECT PRESENTATION

Avaria, Bayu, Bernard, Darryl

L3BC

AGENDA

Background

Problem and Solution

Results and Findings

Conclusion

BACKGROUND

Timekeeping is integral to an advanced society. It is one of the most important inventions ever devised in human history. Without the ability to count seconds, minutes, hours, and days, humans will never be able to meet deadlines and keep track of events in the past effectively.

PROBLEMS

WHAT IS THE PROBLEM?

With most datasets in the world, most of them use a different time format versus each other. Some can come out of the gate with a 24-hour format ; the format that is the most readable format for computers, but some are also using an am / pm format. The problem here is that with an am / pm format, the computer cannot read it correctly for further analysis.

GOAL OF PROEJCT

We'll be making date and time easier to read for computers in order to make it easier to resample, analyze, and resample the data to get the exact analysis we need. Afterwards, we will be applying sorting algorithm to help users track sorted data that can be to their liking.

PROPOSED SOLUTION

STEP 1

We aim to make a program that can correctly format the time given in the dataset so it can further analyze said database

STEP 2

We then want to analyze and visualize the data for further analysis

STEP 3

We want to incorporate quicksort algorithm to then sort a certain category by either ascending or descending order for ease of use.

RESULTS AND FINDINGS

PRE SORTED DATA

Date	Symbol	Open	High	Low	Close	Volume
2020-03-13 08-PM	ETHUSD	129.94	131.82	126.87	128.71	1940673.93
2020-03-13 07-PM	ETHUSD	119.51	132.02	117.1	129.94	7579741.09
2020-03-13 06-PM	ETHUSD	124.47	124.85	115.5	119.51	4898735.81
2020-03-13 05-PM	ETHUSD	124.08	127.42	121.63	124.47	2753450.92
2020-03-13 04-PM	ETHUSD	124.85	129.51	120.17	124.08	4461424.71
2020-03-13 03-PM	ETHUSD	128.39	128.9	116.06	124.85	7378976.0
2020-03-13 02-PM	ETHUSD	134.03	137.9	125.5	128.39	3733916.89
2020-03-13 01-PM	ETHUSD	131.35	140.95	128.99	134.03	9582732.93

POST SORTED DATA

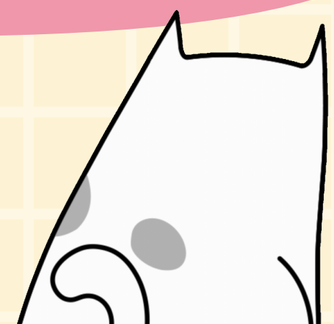
Date	Symbol	Open	High	Low	Close	Volume	Day Of Week
2018-01-13 18:00:00	ETHUSD	1388.99	1418.96	1388.99	1410.0	27959588.92	Saturday
2018-01-13 22:00:00	ETHUSD	1395.0	1404.0	1383.63	1392.24	9849624.19	Saturday
2018-01-13 20:00:00	ETHUSD	1382.56	1418.8	1382.5	1418.61	15617541.24	Saturday
2018-01-14 03:00:00	ETHUSD	1376.01	1385.75	1370.95	1383.44	4304573.35	Sunday
2018-01-13 19:00:00	ETHUSD	1410.0	1418.33	1365.88	1382.56	23061241.98	Saturday
2018-01-13 17:00:00	ETHUSD	1365.01	1393.0	1365.0	1388.99	19922526.55	Saturday
2018-01-13 23:00:00	ETHUSD	1392.24	1393.12	1364.0	1386.02	13550709.64	Saturday
2018-01-13 21:00:00	ETHUSD	1418.61	1419.96	1360.99	1395.0	11452917.63	Saturday

AVERAGE TIME

```
23.1 ns ± 2.24 ns per loop (mean ± std. dev. of 10 runs, 50 loops each)
75.4 ns ± 4.08 ns per loop (mean ± std. dev. of 10 runs, 50 loops each)
```

MEMORY SPACE

```
Memory Used: (1525777, 15414931)
```



TIME PROFILING

```
# Time Profiling (sorting)
%timeit -r10 -n50 sorted_df

# Time Profiling (exporting to .csv)
%timeit -r10 -n50 sorted_df.to_csv
```

MEMORY ALLOCATION

```
# Start tracing memory
tracemalloc.start()

## -- Start of Algorithm -- #

## the .sort_values method returns a new dataframe, so make sure to
# assign this to a new variable.
sorted_df = df.sort_values(by=["Low"], ascending=False)

# Index=False is a flag that tells pandas not to write
# the index of each row to a new column. If you'd like
# your rows to be numbered explicitly, leave this as
# the default, True
sorted_df.to_csv('testrun69.csv', index=True)

## -- End of Algorithm -- ##

# displaying the memory
print('Memory Used:', tracemalloc.get_traced_memory())

# stopping the library
tracemalloc.stop()
```

CONCLUSION

In Conclusion, we discovered that although the quicksort algorithm is quite memory intensive and still has a respectable variance in time complexity (in the nanosecond range), it does its job to sort data for better analysis.

MERCI DE VOTRE ATTENTION