ASSIGNMENT 1

Yasser Albert Corzo

CARNEGIE MELLON UNIVERSITY DATA, INFERENCE & APPLIED MACHINE LEARNING (COURSE 18-785)

Libraries Used

The following libraries were used to complete this assignment:

- Python math module
- Matplotlib
- Numpy
- Pandas
- Scipy

Assignment Report

Q1

ANSWER

27 folds is needed to exceed the height of Mount Everest.

APPROACH

Grabbed a normal 11'' x 8'' paper and folded once, measured thickness to be 0.2 mm. Kept on folding, recording thickness and found a relationship between the number of folds and thickness to be $0.1 \times (2^{number\ of\ folds})$. Iterated through the number of folds until thickness of folded paper exceeded the height of Mount Everest and recorded that number of folds.

Q2

ANSWER

It would take t = 7 time for the volume to decrease to less than half of its initial volume.

APPROACH

Set the initial volume to an arbitrary value, in this case 100 (can be any unit of volume). I then initialized the time counter t to 0. Using the formula given, iteratively increase t by 1 and calculate the volume of the water in the reservoir at time t until volume at time t is less than half of the initial volume.

Q3

ANSWER

After 1 year I'll have $105. After 2 years I'll have $110. After 3 years I'll have $116. After 4 years I'll have $122. After 5 years I'll have $128.

APPROACH

I used the compound interest formula provided in recitation:

$A = P(1 + \frac{r}{n})^{nt}$

Where A is the amount of money accumulated over n years, P is the initial amount, r is the annual interest rate, n is the number of times interest is applied per year, and t is the number of years the money is borrowed for. Since the problem is asking for how much money after one, two, three, four and five years, use t = 1, 2, 3, 4, 5 in the compound interest formula. I used the python round() function [1] to round to the nearest dollar.

Q4

ANSWER

Monthly payment to pay off debt in 1 year: $1777.

Monthly payment to pay off debt in 2 years: $941.

Monthly payment to pay off debt in 3 years: $664.

APPROACH

I used the loan payment formula provided in recitation:

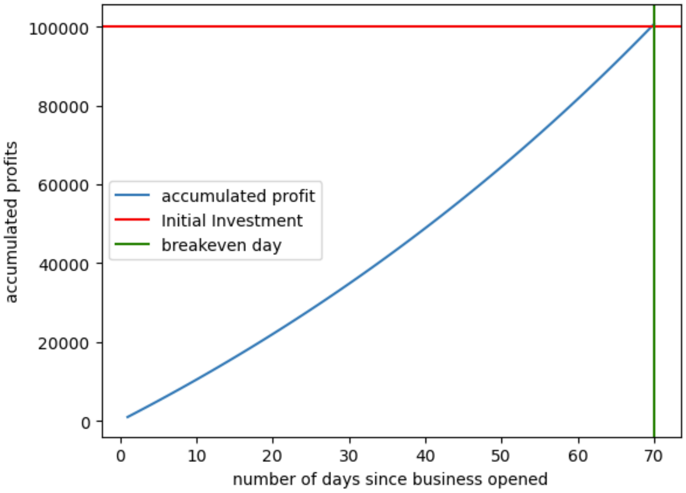
$$A = P\frac{r(1+r)^n}{(1+r)^n - 1}$$

Where A is the amount of payment, P is the principal loan amount, r is the interest rate and n is the number of monthly payments. Since we're asked to calculate the monthly payment to pay off debts in one, two, and three years, n must be 12, 24, and 36 respectively. I used the python round() function [1] to round to the nearest dollar.

Q5

ANSWER

It will take 70 days to repay the initial investment.



APPROACH

I set the initial number of customers to 100 (given). Initialized the total profit and days opened to 0. I iterated on the payments until the total profit was greater than or equal to the amount invested ($100000), keeping track of the number of days and accumulated profit. To keep track of when exactly this threshold was reached, I interpolated using scipy [2].

Q6

ANSWER

date exceeded 100 cases: 2014-03-28

date exceeded 500 cases: 2014-06-14

date exceeded 1000 cases: 2014-07-16

date exceeded 2000 cases: 2014-08-12

date exceeded 5000 cases: 2014-09-12

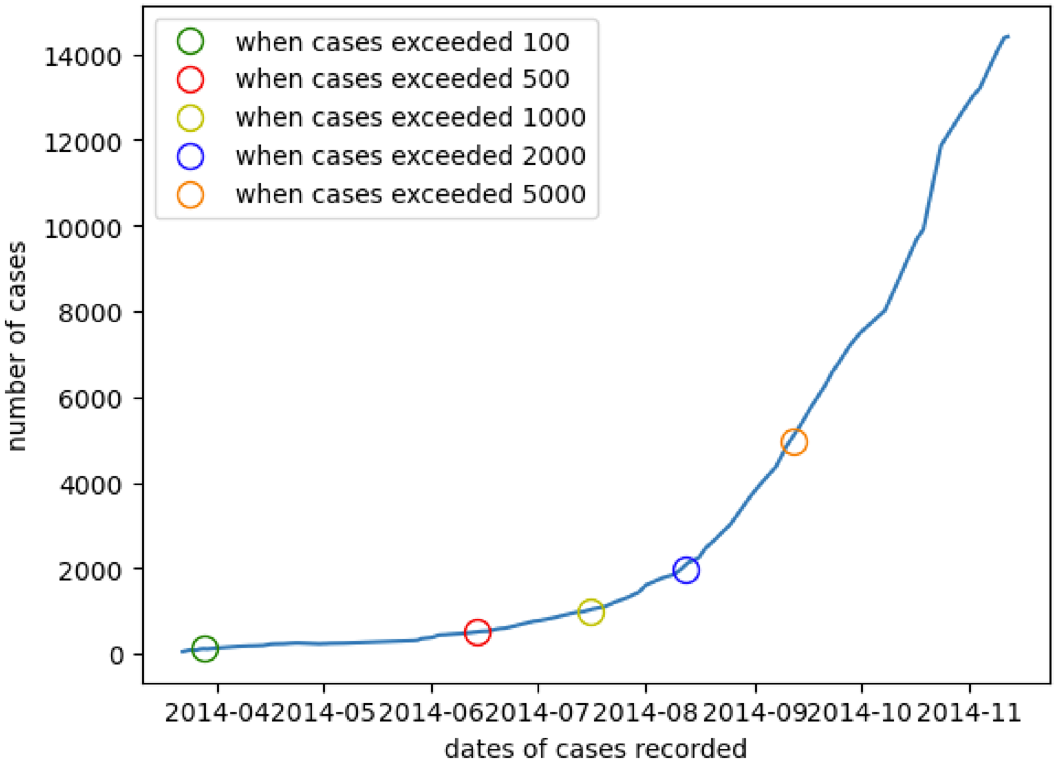
date exceeded 100 deaths: 2014-04-06

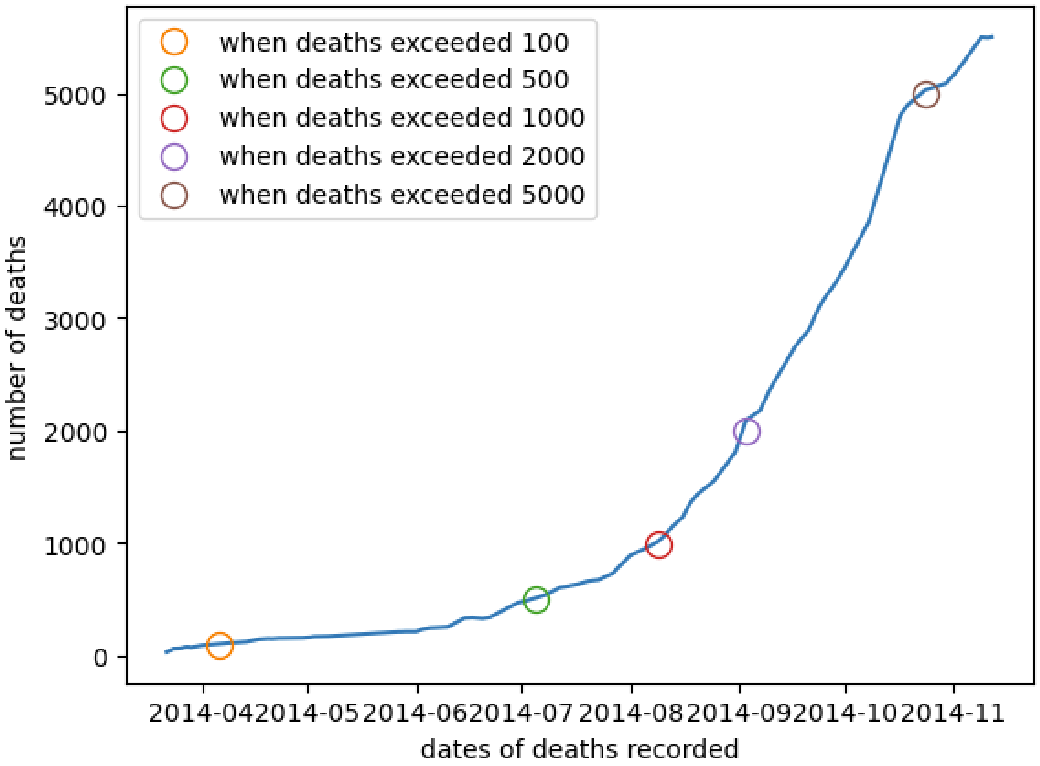
date exceeded 500 deaths: 2014-07-05

date exceeded 1000 deaths: 2014-08-09

date exceeded 2000 deaths: 2014-09-03

date excceeded 5000 deaths: 2014-10-24


when cases exceeded 100
when cases exceeded 500
when cases exceeded 1000
when cases exceeded 2000
when cases exceeded 5000
number of cases
0
2000
4000
6000
8000
10000
12000
14000
2014-04
2014-05
2014-06
2014-07
2014-08
2014-09
2014-10
2014-11
dates of cases recorded


when deaths exceeded 100
when deaths exceeded 500
when deaths exceeded 1000
when deaths exceeded 2000
when deaths exceeded 5000
number of deaths
0
1000
2000
3000
4000
5000
2014-04
2014-05
2014-06
2014-07
2014-08
2014-09
2014-10
2014-11
dates of deaths recorded

APPROACH

Extract the cases, deaths, and dates. Found the date when the thresholds were crossed by interpolating with scipy [2] as well as getting the missing values for cases/deaths on these missing dates. Plotted values on the graph using matplotlib [3].

Q7

ANSWER

average growth rate for ebola cases: 7.897609748950631 %

average growth rate for ebola deaths: 7.569168161668559 %

APPROACH

Used the interpolated missing dates and values from Q6 to add them to the original data at their corresponding location. Computed the growth rate for deaths and cases with formula:

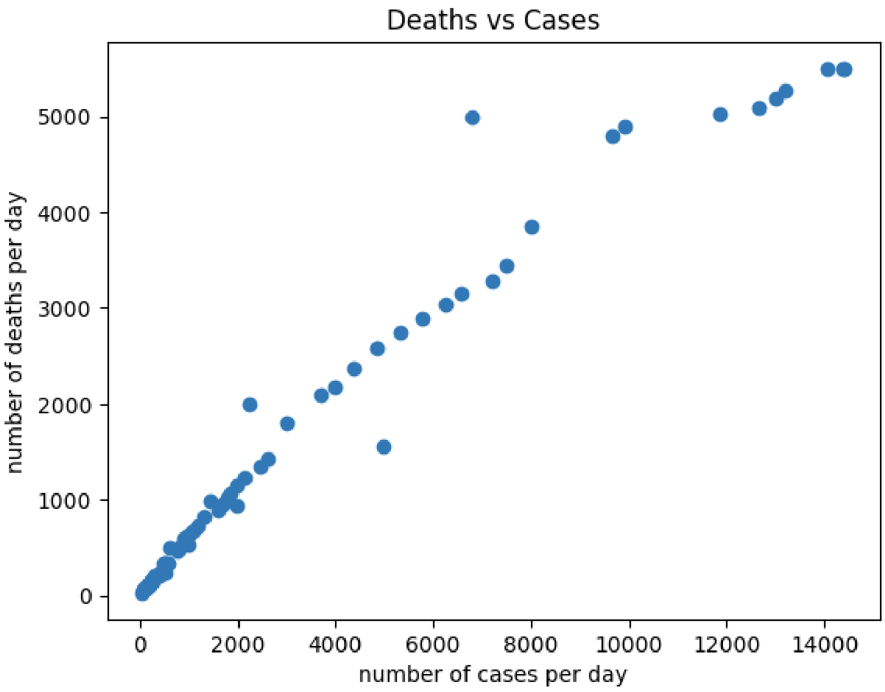
$$\frac{current\ value - prev\ value}{prev\ value} * 100$$

Computed the mean for the growth rates.

Q8

ANSWER

average ratio of deaths to cases: 0.5865909041003434

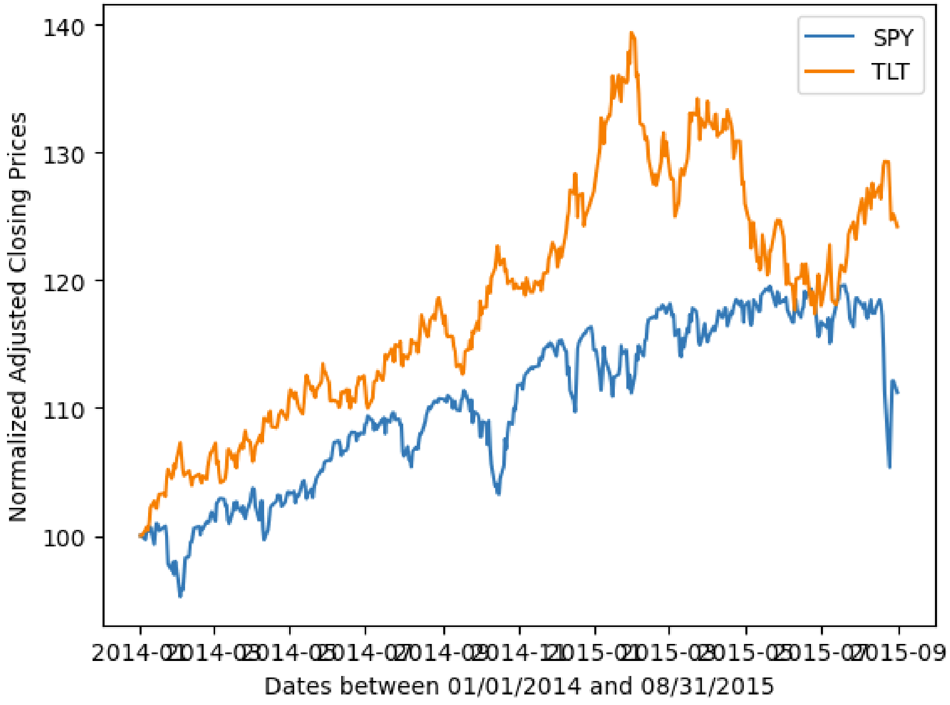


APPROACH

Used interpolated data from Q6 (containing missing dates and their values). Calculated the ratio of deaths to cases and computed the mean.

Q9

ANSWER



APPROACH

Extracted the dates and adjusted closing price from csv files on Yahoo for SPY and TLT. Normalized all the values of both stocks so that the first adjusted closing price value for SPY and TLT is 100. Afterwards, I plotted the normalized values with matplotlib [3].

Q10

ANSWER

max SPY daily return: 3.839375%

max TLT daily return: 2.646922%

min SPY daily return: -4.210687%

min TLT daily return: -2.43248%

average SPY daily return: 0.028612%

average TLT daily return: 0.055378%

APPROACH

I computed the daily returns for TLT and SPY using the formula given in the assignment and found the max, min, and mean values via panda methods [4].

References

[1] Python round() function, https://www.w3schools.com/python/ref_func_round.asp (accessed Sep. 4, 2023).

[2] “Scipy.interpolate.interp1d#,” scipy.interpolate.interp1d - SciPy v1.11.2 Manual, https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html (accessed Sep. 4, 2023).

[3] Matplotlib pyplot, https://www.w3schools.com/python/matplotlib_pyplot.asp (accessed Sep. 4, 2023).

[4] “Pandas.dataframe.max#,” pandas.DataFrame.max - pandas 2.1.0 documentation, https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.max.html (accessed Sep. 4, 2023).