Problem1

1. $P_t = P_{t-1} + r_t$

$E(P_t) = E(P_{t-1}) + E(r_t) = P_{t-1}$

$std(P_t) = std(r_t) = 1$

2. $P_t = P_{t-1}(1+r_t)$

$E(P_t) = E(P_{t-1}) + E(P_{t-1}) \cdot E(r_t) = P_{t-1}$

$std(P_t) = P_{t-1} \cdot std(r_t) = P_{t-1}$

3. $P_t = P_{t-1} e^{r_t}$

$E(P_t) = E(P_{t-1}) \cdot e^{E(r_t)} = P_{t-1}$

$std(P_t) = P_{t-1} \cdot std(e^{r_t}) = P_{t-1} \cdot \sqrt{e^2 - e}$ (log normal)

	Classical Brownian Motion	Arithmetic Return System	Log Return or Geometric Brownian Motion
Numerical Mean	10.0019	10.0148	16.5752
Numerical Std	1.0013	10.0436	22.1062
Analytical Mean	10	10	16.4872
Analytical Std	1	10	21.6120

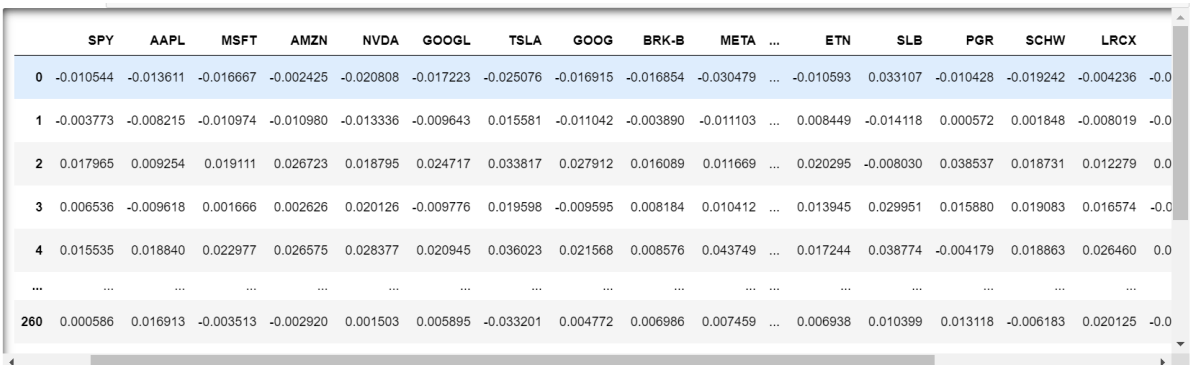
Assume rt ~ N(0, 1), Pt-1 = 10

It can be found that the numerical results and analytical results are almost the same.

Problem 2

By applying the Return calculate function and assigning the method to “DISCRETE”, the returns of

each asset are calculated:



	SPY	AAPL	MSFT	AMZN	NVDA	GOOGL	TSLA	GOOG	BRK-B	META	...	ETN	SLB	PGR	SCHW	LRCX	
0	-0.010544	-0.013611	-0.016667	-0.002425	-0.020808	-0.017223	-0.025076	-0.016915	-0.016854	-0.030479	...	-0.010593	0.033107	-0.010428	-0.019242	-0.004236	-0.0
1	-0.003773	-0.008215	-0.010974	-0.010980	-0.013336	-0.009643	0.015581	-0.011042	-0.003890	-0.011103	...	0.008449	-0.014118	0.000572	0.001848	-0.008019	-0.0
2	0.017965	0.009254	0.019111	0.026723	0.018795	0.024717	0.033817	0.027912	0.016089	0.011669	...	0.020295	-0.008030	0.038537	0.018731	0.012279	0.0
3	0.006536	-0.009618	0.001666	0.002626	0.020126	-0.009776	0.019598	-0.009595	0.008184	0.010412	...	0.013945	0.029951	0.015880	0.019083	0.016574	-0.0
4	0.015535	0.018840	0.022977	0.026575	0.028377	0.020945	0.036023	0.021568	0.008576	0.043749	...	0.017244	0.038774	-0.004179	0.018863	0.026460	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
260	0.000586	0.016913	-0.003513	-0.002920	0.001503	0.005895	-0.033201	0.004772	0.006986	0.007459	...	0.006938	0.010399	0.013118	-0.006183	0.020125	-0.0

	Normal	EW_Normal	MLE_T	AR_1	Historical
VaR $	**16.236135**	**8.966972**	**12.90073**	**16.101476**	**12.569656**
VaR %	**-0.054287**	**-0.029982**	**-0.043135**	**-0.053837**	**-0.042028**

We can find that the VaR results of Normal and AR_1 method is similar, which I believe is because they both assumes the Normal Distribution.

For the VaR calculated by Historical method and MLE_T, it is little smaller than the VaR calculated by Normal and VaR methods.

For the EW_Normal method, we can see that the result becomes much smaller by applying different weights to given time., which may because the recent data is less volatile than the past data,

Problem 3

I defined a function called VaR_port_data that accepts portfolio dataframe, price dataframe, and portfolio name as the arguments, and return the data needed for the VaR_cal function that calculate the VaR

For the first function, Firstly, it constructs holdings dictionary that records the information of stock name (key) and its holdings (value) from portfolio dataframe given the portfolio name passed in port. Secondly, current prices, returns (all stocks' mean of return equal to 0), PV (Present Value), and Asset_value (a list contains different assets' value) are calculated. Finally, all of these variables are returned.

Then, by using the for loop to iteratively generate the data given the portfolio name and pass the data

generated by this function into the VaR_cal function set in the problem2 (set the method to "EW_Normal") to calculate the VaR. The result is as following:

	A	**B**	**C**	**Total**
VaR	**15426.968017**	**8082.572402**	**18163.291619**	**38941.375729**

I choose log return to calculate VaR again because of the stability of logarithmic returns: Logarithmic returns can stably describe changes in asset prices, especially in the short term. Besides, in some cases, the log return can be approximated as a normal distribution, which makes it easier to apply the properties of the normal distribution when calculating VaR. It can also better meet the assumptions of portfolio theory, such as the normal distribution assumption, etc. The result is as following:

	A	B	C	Total
VaR	15433.515096	8089.616241	18081.612831	38904.842412

It can be found that the result doesn't change a lot compared to the arithmetic return result, so maybe for this dataset, the effect of applying two returns is similar.