University of Toronto
Department of Electrical and Computer Engineering

ECE367 MATRIX ALGEBRA AND OPTIMIZATION

Problem Set #5

Autumn 2020

Prof. S. C. Draper **Due:** 5pm (Toronto time) Friday, 27 November 2020

Homework policy: Problem sets must be turned by the due date and time. Late problem sets will receive deductions for lateness. See the information sheet for futher details. The course text "Optimization Models" is abbreviated as "OptM". Also, see PS01 for details of the "Non-graded", "graded" and "optional" problem categories.

Problem Set #5 problem categories: A quick categorization by topic of the problems in this problem set is as follows:

- Linear and quadratic programs: 5.1-5.3
- Applications in control and finance: 5.4-5.7

NON-GRADED PROBLEMS

Problem 5.1 (Formulating problems as LPs and QPs)
OptM Problem 9.1. Do the problem for objective functions f_1, f_2, f_4, and f_5. (I.e., skip objective function f_3.) Also, you can ignore the part about putting your "problem in standard form", just state your formulations using equality and inequality constraints.

Problem 5.2 (Median versus average)
OptM Problem 9.8, parts 2-5 (i.e., skip the first problem part).

Problem 5.3 (Formulate as an LP), from a previous exam
Consider the problem of minimizing an objective function of the form $c^T x + f(d^T x)$ subject to the linear constraints $Gx \leq h$. The vectors c and $d \in \mathbb{R}^n$ are given and the function $f : \mathbb{R} \to \mathbb{R}$ is specified in Figure 1. (The function f has slope -1 for $z \leq 1$ and slope $+2$ for $z \geq 2$ and evaluates to $f(z) = 0$ for $1 \leq z \leq 2$.)

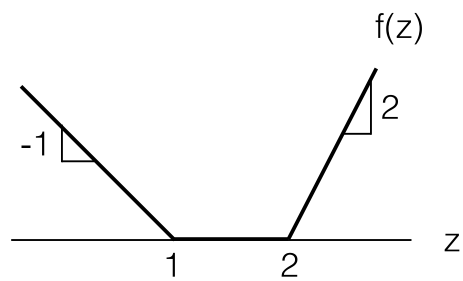


Figure 1: The function f.

Provide a linear programming formulation of this optimization problem.

GRADED PROBLEMS

Problem 5.4 (ℓ_1 regularized least squares), from a previous exam
This problem considers optimization problems of the form

$$\min_{x \in \mathbb{R}^2} \|Ax - y\|_2^2 + \gamma \|x\|_1 \tag{1}$$

where $\gamma \in \mathbb{R}$ and $\gamma \geq 0$. BE SURE TO NOTE THAT the second term in this problem is the ℓ_1 norm. In this problem we denote the x that minimizes (1) as x^*_γ.

(a) First consider the case where

$$A = \begin{bmatrix} 0 & 2 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}, \qquad y = \begin{bmatrix} -2 \\ 5 \\ 9 \end{bmatrix}, \qquad \gamma = 0.$$

Derive the result that, for the parameters given, $x_0^* = (5, -1)$. (Note, you must *derive* this result, not simply confirm it. We are providing the result to make sure you have the correct value of x_0^* for later parts of the problem.)

(b) In Figure 2 we plot $x_\gamma^* \in \mathbb{R}^2$ as a function of γ where $x_\gamma^* = (x_{\gamma,1}^*, x_{\gamma,2}^*)$ and $x_{\gamma,i}^*$, $i \in \{1, 2\}$, denotes the ith element of x_γ^*. As can be observed in the Figure 2 there is a range of values of γ such that $x_{\gamma,2}^* = 0$. (Note that this plot is not to scale.) What is the *smallest* value of γ, lets call it $\gamma_{\min}$, such that $x_{\gamma,2}^* = 0$? Also, what is the value of $x_{\gamma_{\min},1}^*$?

(c) It can be observed from the Figure 2 that for all $\gamma > \gamma_{\min}$ the optimizing x_γ^* has the form $(x_{\gamma,1}^*, 0)$. Explain why.

(d) Solve for x_γ^* for $\gamma \geq \gamma_{\min}$.

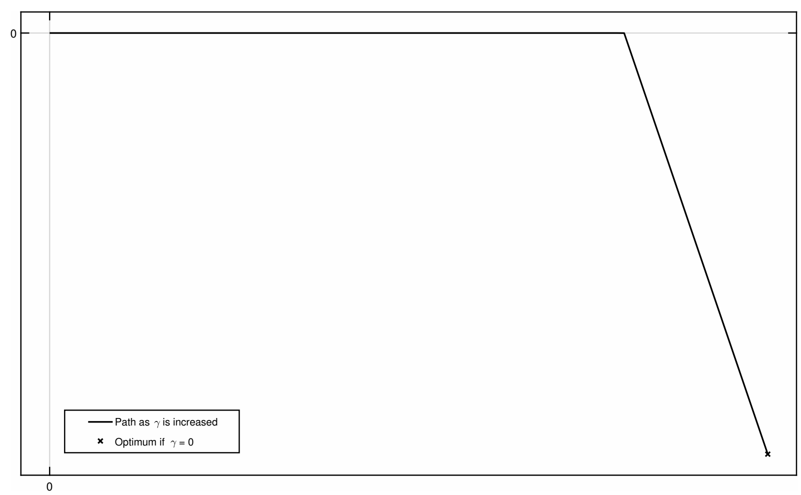


Figure 2: Path followed by x_γ^* as a function of γ.

Problem 5.5 (An optimal breakfast)

OptM Problem 9.5. In your solution both state the form of the optimization problem you want to solve and solve it, e.g., using `Matlab` and `CVX`. In your solution specify:

(a) The optimal variable x^*.

(b) The optimum value p^*.

(c) Remember to attach your code.

To help get you started we include below a snippet of CVX code in MATLAB that solves the "meat and potatoes" example from class. (See PS01 for instructions on getting the CVX toolbox.)

```
%%% Example done in class
%%% Two variables: pounds purchased of  meat or potatoes
%%% Cost vector c: $1 per pound mean and $0.25 per pound potatoes
%%% Data matrix D: grams carbs/fiber/protein (rows) per pound meat/potatoes (cols)
%%% Constraint vector R: daily reqirements grams carbs / fiber / protein

c = [1 0.25];
D = [40 200; 5 40; 100 20];
R = [400; 40; 200];

cvx_begin
variable x(2)
cost = c*x;
carbs = D(1,:)*x;
fibre = D(2,:)*x;
protein = D(3,:)*x;

minimize(cost)
subject to
carbs >= 400;
fibre >= 40;
protein >= 200;
x >= 0;
cvx_end
```

Problem 5.6 (Optimal control of a unit mass, new norms)

In an earlier assignment you solved an optimal control problem with a quadratic (i.e., $\|\cdot\|_2^2$) objective where the force applied was piece-wise constant, specified by a force vector $p = (p_1, \ldots, p_{10})$, and the goal was to move a mass from being at rest at the origin at time zero to being at rest at the unit position at time 10. In this problem we consider the same setup but with two different objectives, the ℓ_1 and ℓ_∞ norms, i.e.,

$$\|p\|_1 = \sum_i |p_i|, \quad \text{and} \quad \|p\|_\infty = \max_i |p_i|.$$

The ℓ_1 norm can serve as a proxy for fuel consumption, the ℓ_∞ norm tries to minimize the *peak* force used.

(a) First consider the problem of minimizing $\|p\|_1$ using the same setup as in the previous assignment. Find the optimal solution using, e.g., MATLAB and the CVX toolbox. (If using MATLAB without CVX you may find the MATLAB command linprog useful.) Plot the optimal force, position, and velocity. What do you observe about the form of the optimization variables at the optimum, how do they contrast to those of the ℓ_2 solution?

(b) Repeat part (a) for the $\|\cdot\|_\infty$ minimization problem. How does the ℓ_∞ solution compare to the ℓ_1 and ℓ_2 solutions? Does the solution make sense?

(c) Include your code with your assignment.

Further, you can also repeat the second part of the earlier problem—where the mass also is required to be at the origin at time 5—for the new norms. But this is optional.

Problem 5.7 (Portfolio Design)
In this problem we consider a classic approach to investment known as "Markowitz portfolio optimization." The idea is that there is often a risk/reward tradeoff in investing that must be managed. The QP that we discuss in this problem is one way to manage for that risk.

Say there are n stocks in which you can invest. We consider a single investment period (for convenience one year). Your must determine an investment strategy which boils down to an allocation of your funds p across the n stocks. Normalizing your wealth to one unit, $\sum_{i=1}^n p_i = 1$ (you must invest all your portfolio) and $p_i \geq 0$ for all i (you can't "short" stocks).

There are two pieces of knowledge you have: the expected return of each of the n stocks, and the variability in those returns. As we next discuss, the former is parameterized by the vector $\bar{x}$ and the latter by the matrix Σ. If you research tells you that the annual mean return of stock i is 35% then, if you invest \$1 in stock i on 1 January, your expected investment on 31 December will be worth $\bar{x}_i = \$1.35$. Of course there is variability about this return and we denote by x_i the actual value of your investment, so $E[x_i] = \bar{x}_i$. The variability is denoted by the variance in the stock $E[(x-x_i)^2] = \Sigma_{ii}$. Stocks are correlated so their joint variability is encapsulated by their covariance $E[(x-x_i)(x-x_j)] = \Sigma_{ij}$. We stack the covariance into the $n \times n$ matrix Σ.

For this problem we consider four stocks, $n = 4$. The returns of the 4 stocks are shown on the left-hand table below, which the covariance in the stocks is shown in the right-hand table below. In other words, suppose that you invest \$1 in each stock at the start of the year. Then, $\mathbb{E}[x] = \overline{x} = [1.1 \;\; 1.35 \;\; 1.25 \;\; 1.05]^T$, and $\mathbb{E}[(x-\overline{x})(x-\overline{x})^T] = \Sigma$.

IBM	10%
Google	35%
Apple	25%
Intel	5%

	IBM	Google	Apple	Intel
IBM	0.2	−0.2	−0.12	0.02
Google	−0.2	1.4	0.02	0
Apple	−0.12	0.02	1	−0.4
Intel	0.02	0	−0.4	0.2

We wish to design a portfolio (i.e., the proportion of money invested in each company) to minimize the variance of the investment subject to some fixed minimum expected return $r_{\min}$. The variance of an investment allocation p is $p^T\Sigma p$. (Observe that this expression already arose in the course when we discussed the "sample variance" in the derivation of principal components analysis.)

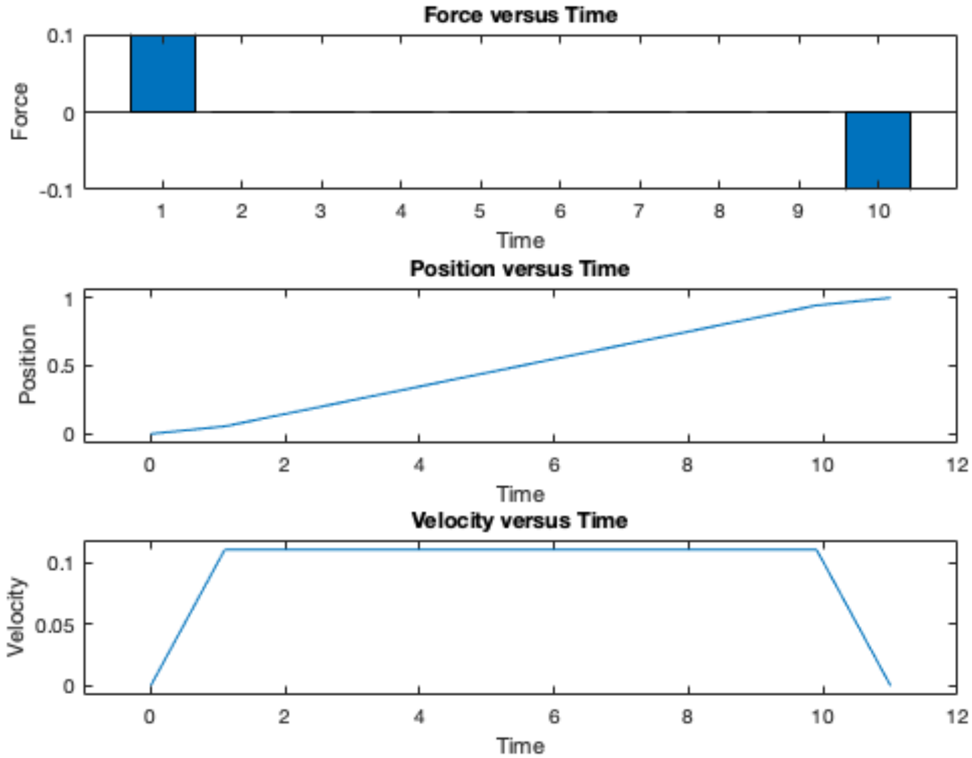
(a) Formulate the optimization problem as a quadratic programming problem. Plot the tradeoff curve between the variance and the expected return $r_{\min}$ as you vary the lower bound on expected return $r_{\min}$ to plot the risk-return tradeoff curve from one extreme to the other..

(`MATLAB` hint: If you are not using `CVX` you may find the `MATLAB` routine `quadprog` useful.)

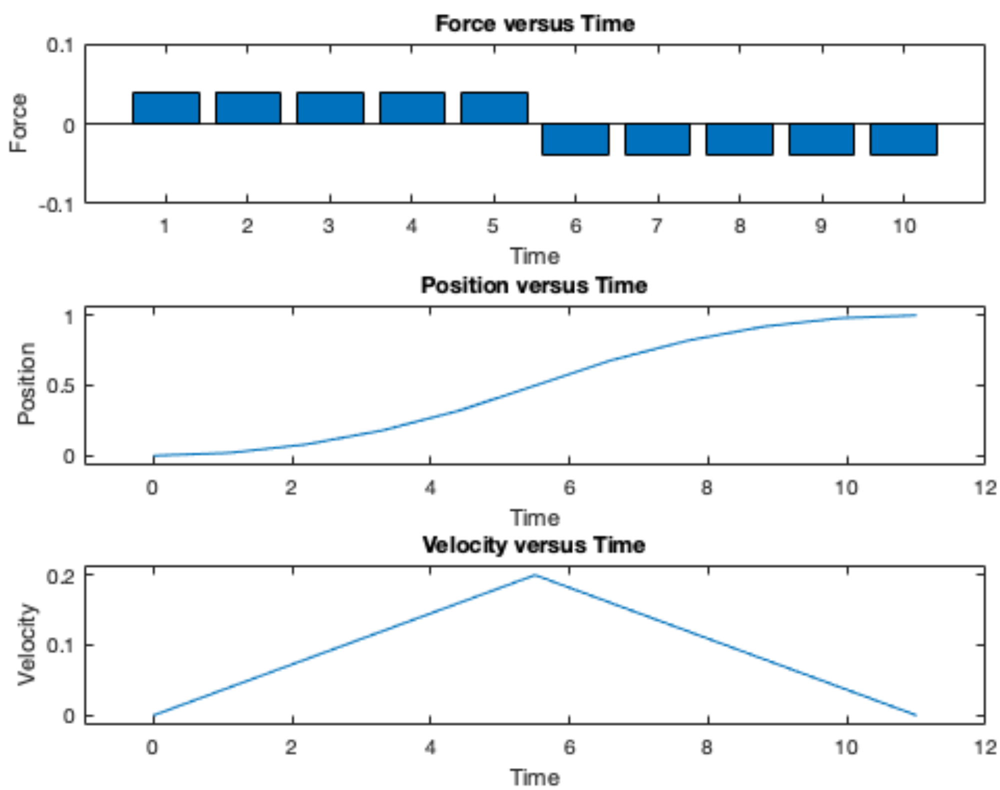
(b) Plot the composition of the portfolio as you move from one extreme of the risk-return tradeoff curve to the other extreme. Comment on the benefit of diversification.

Problem 5.6

Norm = 1



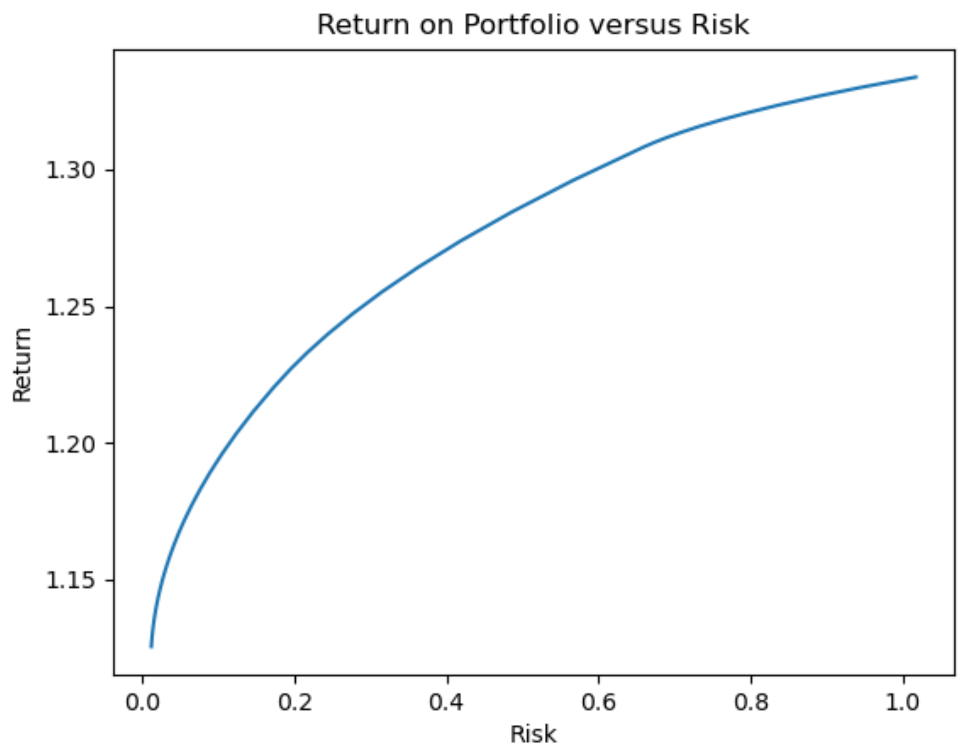
Norm = Inf



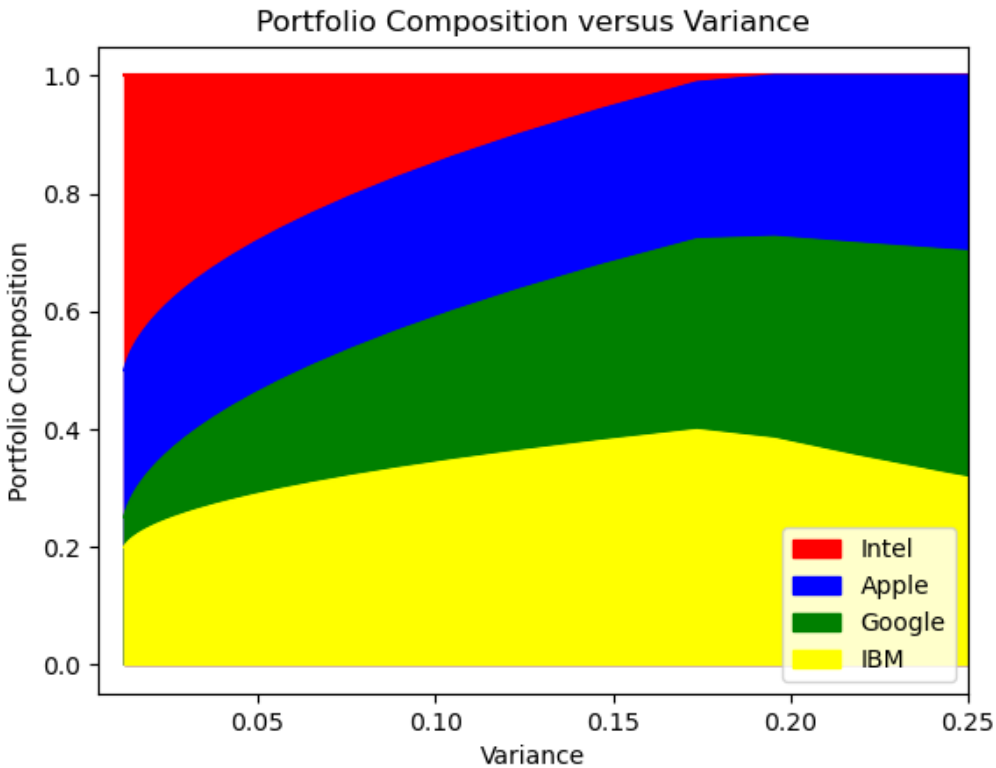
Based on the graph for the l1 norm, it becomes apparent that fuel consumption in this case is minimized as expected. The force/acceleration is indeed minimal when compared to the l2 norm. The l2 norm leads to constant changes in acceleration which leads to varying velocities over time in comparison to the l1 norm which has a much smoother overall motion.

Based on the graph for the l_inf norm, it is apparent that the peak fuel efficiency is indeed minimized in comparison to the l1 and l2 norm. As such, the speed of the mass changes gradually over time and has a peak velocity that only lasts for an instant. The l2 norm leads to changes in velocity at every time step while the l_inf norm leads to only one change in velocity over the entirety of the simulation. The solution in this case makes sense based on the properties of the l_inf norm in comparison to the behaviour of the l1 and l2 norm.

Problem 5.7



The portfolio composition shown below shows that the ideal portfolio composition that leads to low risk and high reward allocates some value of money to all of the stocks available. In the case of extrema (either high risk and variance or low risk and variance) there exists more uncertainty and imbalance in the composition, occurring when too much value is assigned to certain stocks over another.



Code

Problem 5.5

```
from scipy.optimize import linprog

obj = [0.15, 0.25, 0.05]
lh_ineq = [[70,121,65],[107,500,0],[45,40,60],[1,0,0],[0,1,0],[0,0,1],[-70,-121,-
65],[-107,-500,0]]
rh_ineq = [2250,10000,1000,10,10,10,-2000,-5000]

opt = linprog(c=obj,A_ub=lh_ineq,b_ub=rh_ineq,method='revised simplex')
print(opt)
```

Problem 5.6

```
function C = C(A,b,n)
    C = zeros(2,n);
    for i=1:n
        C(:,i) = (A^(n-i)) * b;
    end
end
```

```
function p = minP(norm,mat,equ)
    n = size(mat) * 2;
    cvx_begin
        variable p(n)
        minimize(norm(p,norm))
        subject to
            mat*p == equ;
    cvx_end
end


function v = vals(A,b,n_lim,p_min)
    x = zeros(2,n_lim+1);
    for n=1:n_lim
        c = C(A,b,n);
        x(:,n+1) = c*p_min(1:n,1);
    end
end
```

```
norm = inf;
A = [1 1;0 1];
b = [1/2; 1];
mat = C(A,b,10);
equ = [1;0];

p_min = minP(norm,mat,equ);
trajectory = vals(A,b,10,p_min);

subplot(3,1,1)
t = linspace(1,11,11);
bar(t,p_min)
axis([0 11 -0.1 0.1])
title("Force versus Time")
xlabel("Time")
ylabel("Force")

subplot(3,1,2)
plot(t,trajectory(2,:))
title("Velocity versus Time")
xlabel("Time")
ylabel("Velocity")

subplot(3,1,3)
t = linspace(0,11,11)
plot(t,trajectory(1,:))
title("Position versus Time")
xlabel("Time")
ylabel("Position")
```

Problem 5.7

```
import numpy as np
import matplotlib.pyplot as plt
import cvxopt
from cvxopt.blas import dot
# import pylab

stocks = 4
sigma = cvxopt.matrix([[0.2,-0.2,-0.12,0.02],[-0.2,1.4,0.02,0.0],[-0.12,0.02,1.0,-
0.4],[0.02,0.0,-0.4,0.2]])

c1 = cvxopt.matrix(1.0,(1,stocks))
c2 = cvxopt.matrix(1.0)
c3 = -cvxopt.matrix(np.eye(stocks))
c4 = cvxopt.matrix(0.0,(stocks,1))
expect = cvxopt.matrix([1.1,1.35,1.25,1.05])

r = []
p = []
returns = []
var = []
for t in range (50):
    r.append(10**(2*t/50-1))
for r_min in r:
    p.append(cvxopt.solvers.qp(r_min*sigma,-expect,c3,c4,c1,c2)['x'])

for p_val in p:
    returns.append(dot(expect,p_val))
    var.append(dot(p_val,sigma*p_val))

plt.plot(var, returns)
plt.title('Return on Portfolio versus Risk')
plt.ylabel('Return')
plt.xlabel('Risk')
plt.show()

stock0, stock1, stock2, stock3 = [], [], [], []
for p_val in p:
    stock0.append(p_val[0])
    stock1.append(p_val[0] + p_val[1])
    stock2.append(p_val[0] + p_val[1] + p_val[2])
    stock3.append(p_val[0] + p_val[1] + p_val[2] + p_val[3])

plt.plot(var, stock3, color='red')
plt.plot(var, stock2, color='blue')
plt.plot(var, stock1, color='green')
```

```
plt.plot(var, stock0, color='yellow')
plt.fill_between(var, stock3, label='Intel', color='red')
plt.fill_between(var, stock2, label='Apple', color = 'blue')
plt.fill_between(var, stock1, label='Google', color='green')
plt.fill_between(var, stock0, label='IBM', color='yellow')
plt.title('Portfolio Composition versus Variance')
plt.ylabel('Portfolio Composition')
plt.xlabel('Variance')
plt.legend()
plt.show()
```