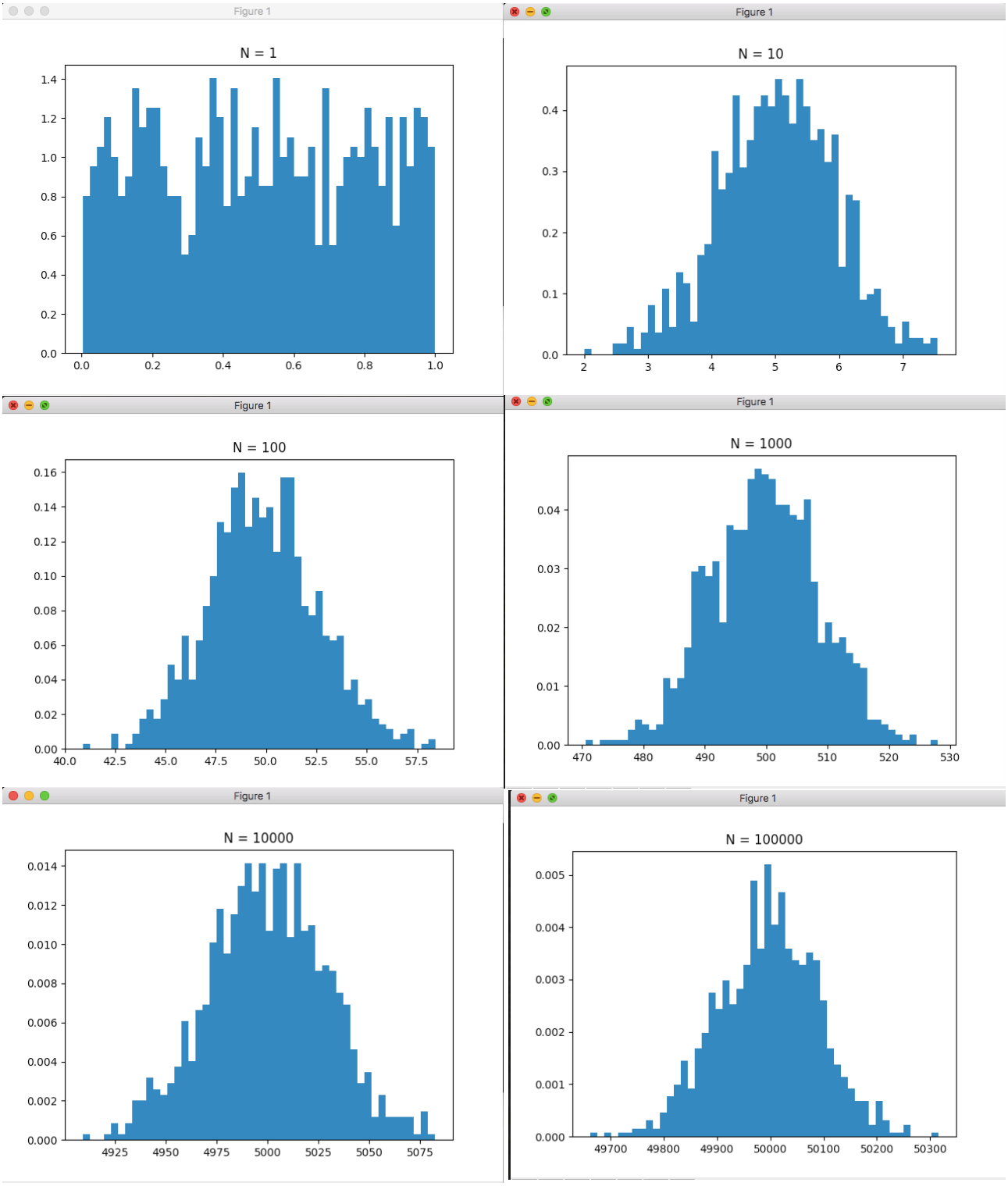
PR2017_HW1

106753027戎諒

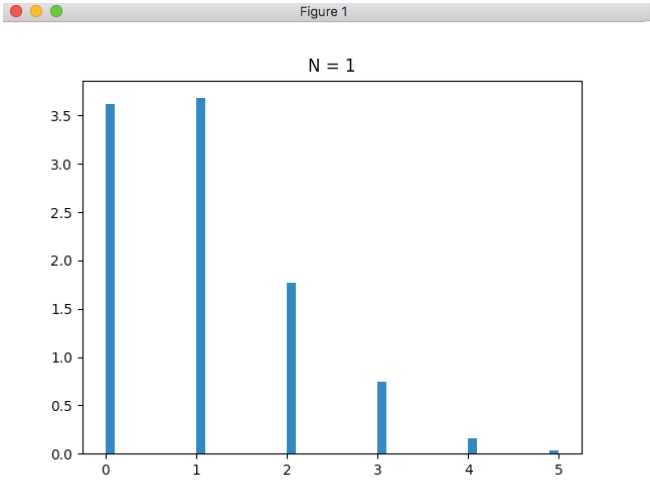
1、Simulation of Central Limit Theorem

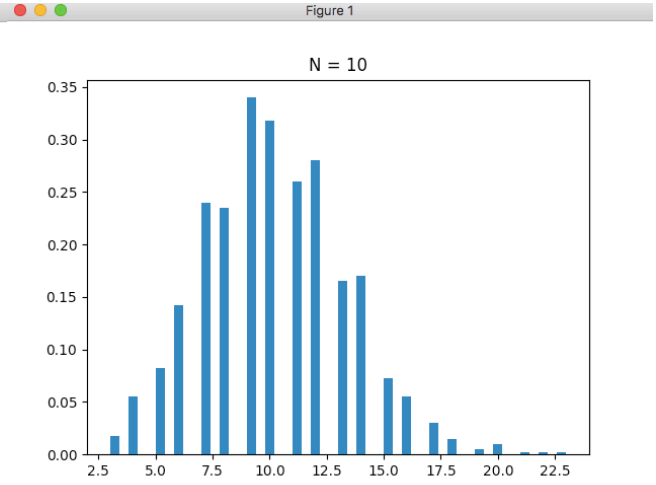
Screenshots(bin size is fixed to 50):

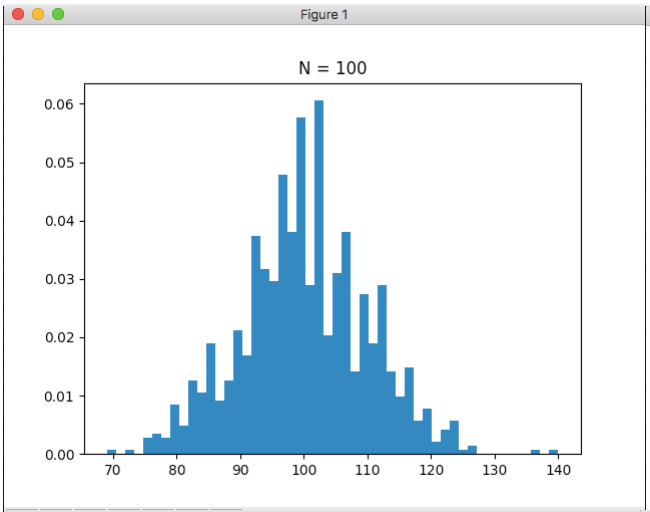
A.Uniform distribution, each N has 1k samples

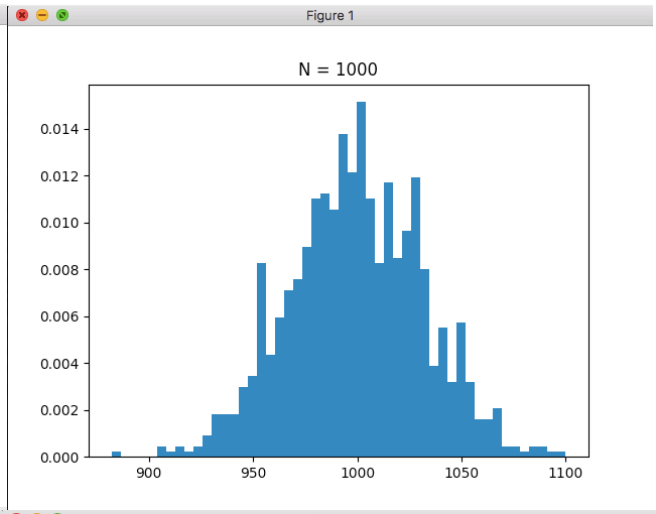


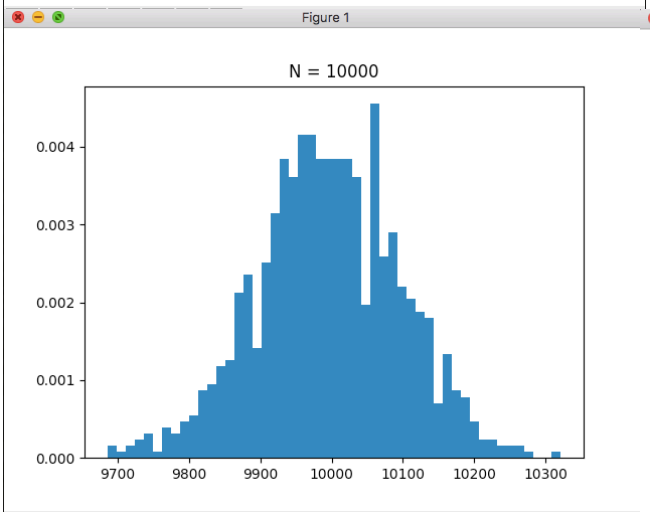
B.Poisson distribution with Lamda = 0.5, each N has 1k samples

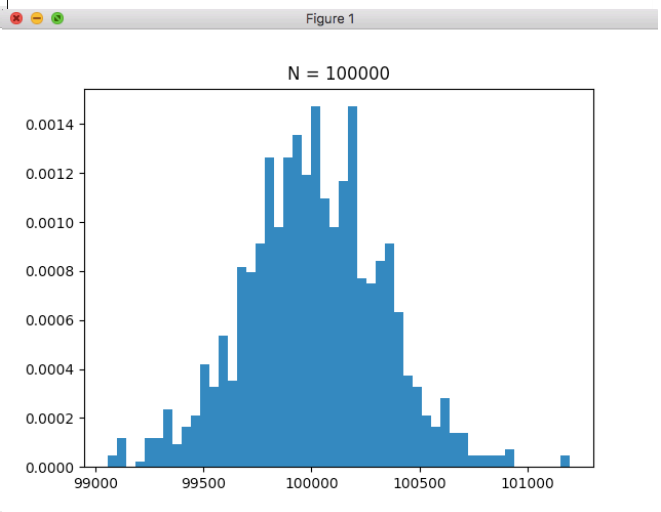




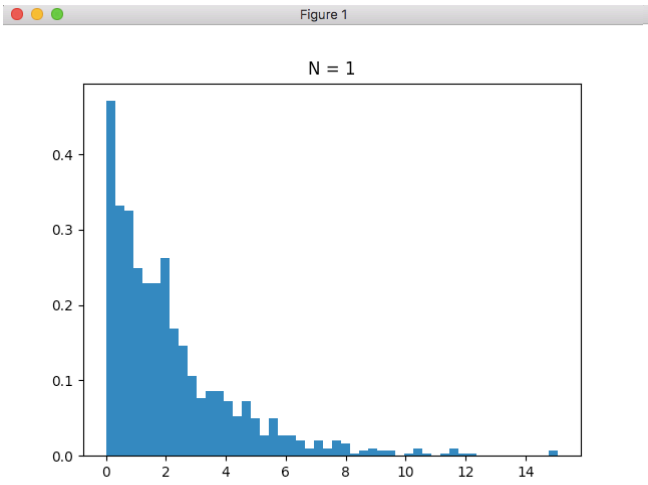


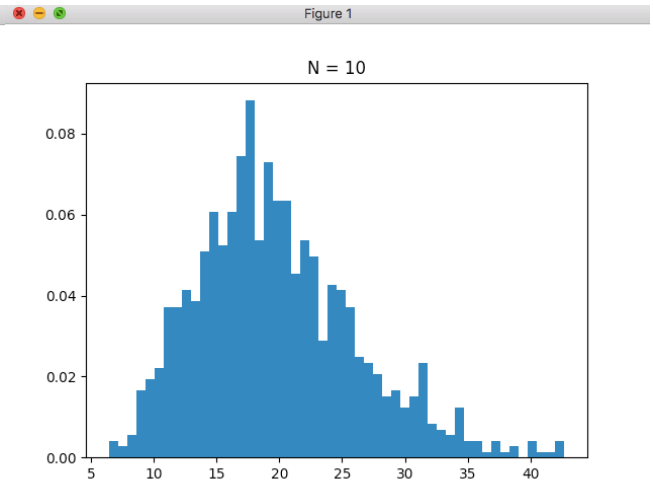


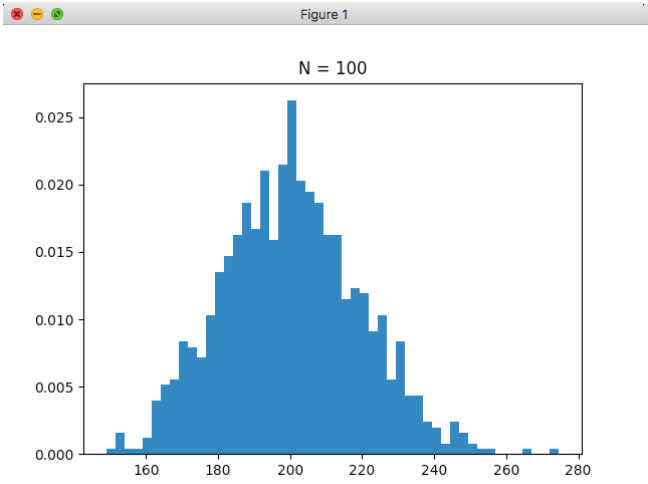


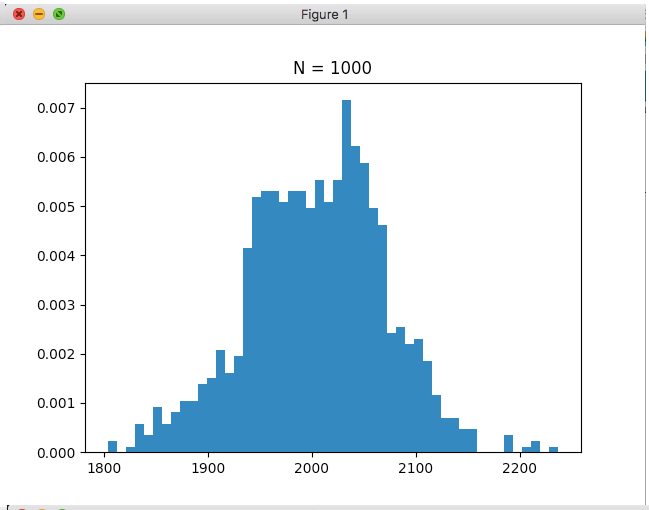


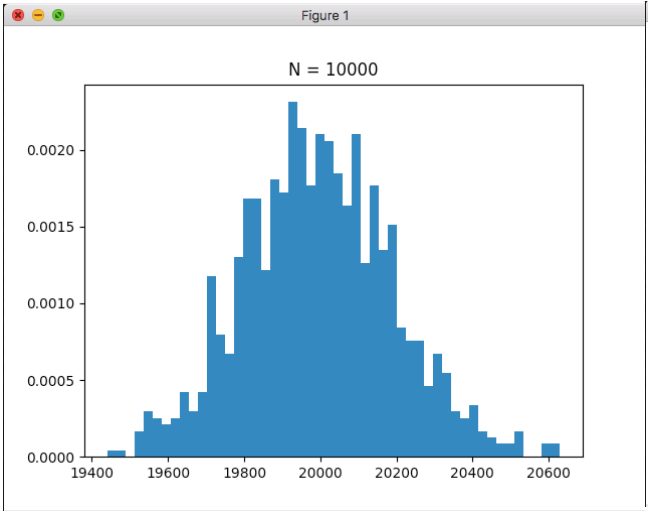
C.Chi-square with degree of freedom = 2, each N has 1k samples

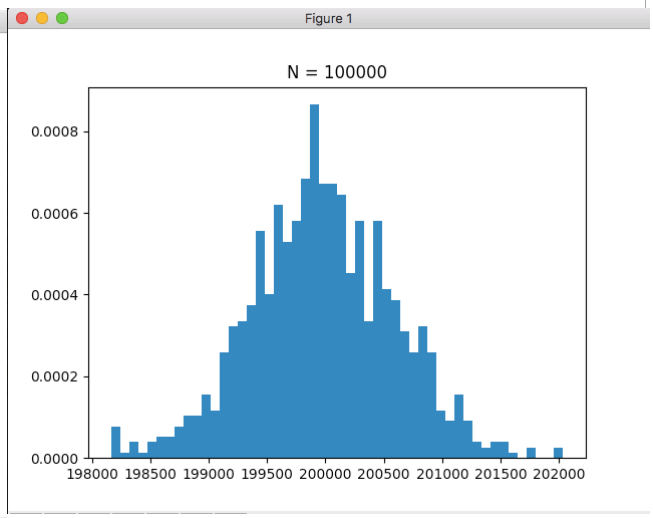








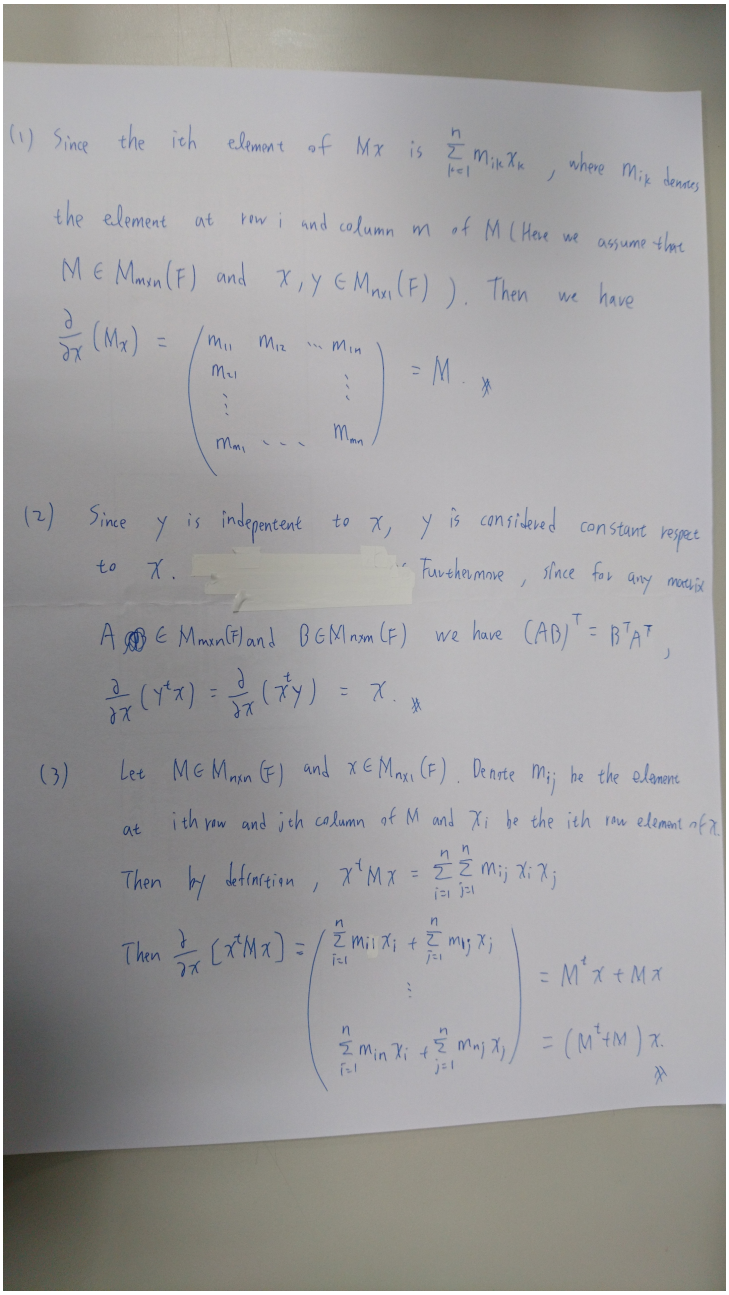




Code(in Python 3.6.2):

```
import numpy as np
import matplotlib.pyplot as plot
import sys, getopt
from enum import Enum

class Distribution(Enum):
    UNIFORM = 1
    CHISQUARE = 2
    POISSON = 3


def GenerateRandom( Type=Distribution.UNIFORM, Size=1000, Lamda=0.5 ):
    RetDistArr = np.zeros(1000)
    if Type == Distribution.UNIFORM:
        for iter in range(0, Size):
            RetDistArr = np.array(RetDistArr + np.random.uniform(0.0, 1.0,
1000))
    elif Type == Distribution.CHISQUARE:
        for iter in range(0, Size):
            RetDistArr = np.array(RetDistArr + np.random.chisquare(2, 1000))
    elif Type == Distribution.POISSON:
        for iter in range(0, Size):
            RetDistArr = np.array(RetDistArr + np.random.poisson(Lamda, 1000))
    return RetDistArr

def GetDistributionFromArgv():
    try:
        opts, args = getopt.getopt(sys.argv[1:], "hd:")
    except getopt.GetoptError:
        print("Usage : -h for help, -d[UNIFORM or POISSON]")
        sys.exit()

    if None == opts or 0 == len(opts):
        print("Usage : -h for help, -d[UNIFORM or POISSON]")
        sys.exit()

    for opt, arg in opts:
        if opt == "-h":
            print("Usage : -h for help, -d[UNIFORM or POISSON]")
            sys.exit()
        elif opt == "-d":
            if arg == "UNIFORM":
                return Distribution.UNIFORM
            elif arg == "CHI":
                return Distribution.CHISQUARE
            elif arg == "POISSON":
                return Distribution.POISSON
            else:
                print("Invalid distribution. Using UNIFORM")
                return Distribution.UNIFORM

RandomArr = []
Dist = GetDistributionFromArgv()

for iter in range(0, 6):
    RandomArr.append(np.array(GenerateRandom(Type=Dist, Size = 10**(iter))))
iter = 0
for Arr in RandomArr:
    count, bins, ignored = plot.hist(Arr, 50, normed=True)
    plot.title("N = " + str(10**iter))
    iter+=1
    plot.show()
```

2、



(1) Since the ith element of Mx is $\sum_{k=1}^{n} m_{ik}x_k$, where m_{ik} denotes the element at row i and column m of M (Here we assume that $M \in M_{m\times n}(F)$ and $x, y \in M_{n\times 1}(F)$). Then we have

$$\frac{\partial}{\partial x}(Mx) = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & & & \vdots \\ \vdots & & & \\ m_{m1} & \cdots & & m_{mn} \end{pmatrix} = M. \ \#$$

(2) Since y is indepentent to x, y is considered constant respect to x. Furthermore, since for any matrix $A \in M_{m\times n}(F)$ and $B \in M_{n\times m}(F)$ we have $(AB)^T = B^T A^T$,

$$\frac{\partial}{\partial x}(y^t x) = \frac{\partial}{\partial x}(x^t y) = x. \ \#$$

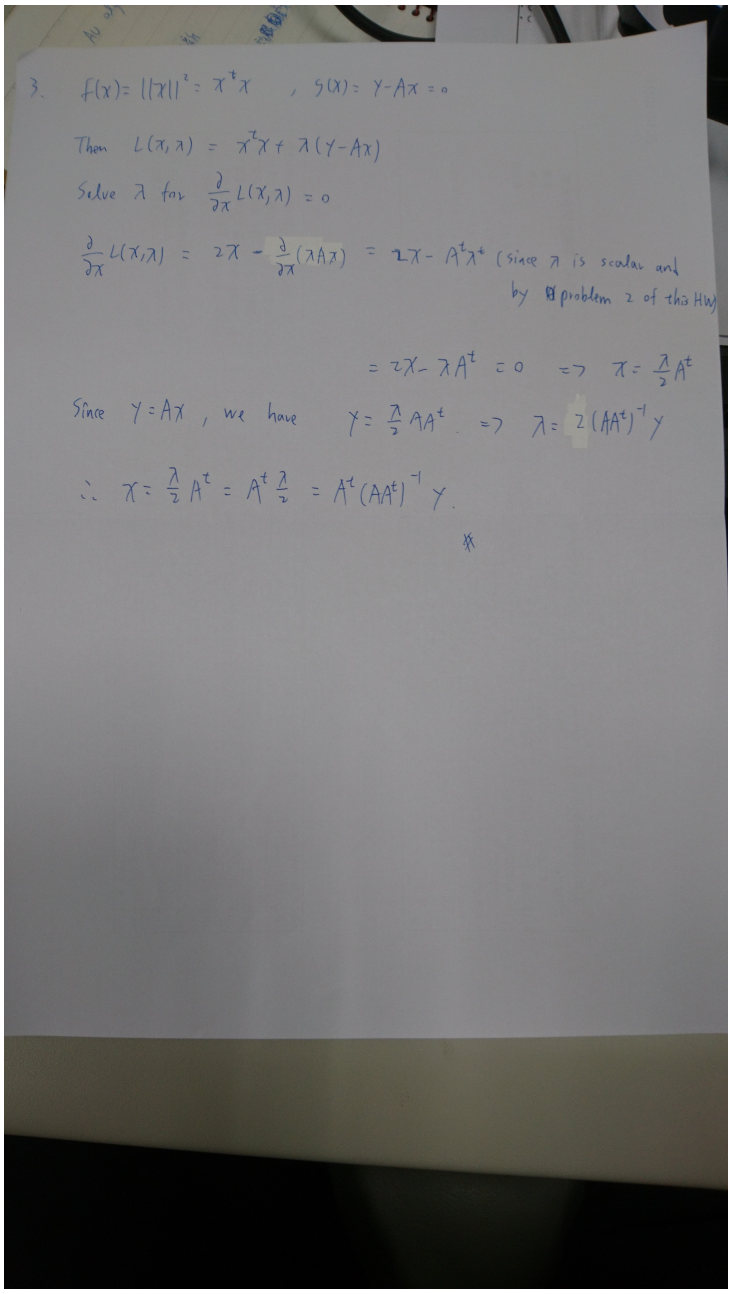
(3) Let $M \in M_{n\times n}(F)$ and $x \in M_{n\times 1}(F)$. Denote m_{ij} be the element at ith row and jth column of M and x_i be the ith row element of x.

Then by definition, $x^t M x = \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij} x_i x_j$

Then $\frac{\partial}{\partial x}[x^t M x] = \begin{pmatrix} \sum_{i=1}^{n} m_{i1}x_i + \sum_{j=1}^{n} m_{1j}x_j \\ \vdots \\ \sum_{i=1}^{n} m_{in}x_i + \sum_{j=1}^{n} m_{nj}x_j \end{pmatrix} = M^t x + Mx$

$= (M^t + M)x. \ \#$

3、



3. $f(x) = \|x\|^2 = x^t x$, $g(x) = y - Ax = 0$

Then $L(x, \lambda) = x^t x + \lambda(y - Ax)$

Solve λ for $\frac{\partial}{\partial x} L(x, \lambda) = 0$

$\frac{\partial}{\partial x} L(x, \lambda) = 2x - \frac{\partial}{\partial x}(\lambda A x) = 2x - A^t \lambda^t$ (since λ is scalar and by problem 2 of this HW)

$= 2x - \lambda A^t = 0 \Rightarrow x = \frac{\lambda}{2} A^t$

Since $y = Ax$, we have $y = \frac{\lambda}{2} A A^t \Rightarrow \lambda = 2(AA^t)^{-1} y$

$\therefore x = \frac{\lambda}{2} A^t = A^t \frac{\lambda}{2} = A^t (AA^t)^{-1} y$.

※