#CSC 521 Assignment 3 by Zhenyang Lu

Consider a small shipment company that every day ships N packages where N is random variable following the Poisson distribution with average of 100 packages per day. Each package is automatically insured for a random value following the Pareto distribution with an alpha of 3.0 and xm of $200. Each package has a 2% probability of being lost. How much capital should the shipping company set aside at the beginning of every year to cover the costs of lost packages, so that the probability of default is less than 3%?

0. Analysis of the solution

STEP 1: To simulate how many parcels shipped by day, you should write a code generating a Poisson random number using code:

```
def poisson(lamb):      #define a poisson PRNG
  u = random.random()
  prob= exp(-lamb)
  k=0
  while 1:
    if u <=prob:return k
    else:u=u-prob
    prob = float(lamb)/(k+1)*prob
    k+=1
```

STEP 2: To simulate how much each parcel is insured, you should define a Pareto random number generator:

```
def pareto(xm,pareto_a):    #define a pareto PRNG
  u = random.random()
  return xm*(1-u)**(-1.0/pareto_a)
```

STEP 3: Since 2% is every parcel likely to be lost, you generate a random number between 0 and 1, if it is below 0.02, then it indicates that the parcel has been lost. Then you redeem the parcel by generate a Pareto random number as the compensation (lost to the shipping firm) using the code in STEP 2. This is the process for the simulation of one parcel. Then you using Poisson random number generator to get a random number with a parameter of 100 to assume how many parcels you ship everyday. Finally, extend your shipping business from one day to a year (250 days).

STEP 4: You sum up the total redeem for the lost parcels, then compute how much you should deposit every year in cash you are default with a possibility of higher than 3% by simulating many times of yearly losses and get the 3% percentile of them. Combined with STEP 3, STEP 4 is defined as simulate_once( ) by the following code:

```
import random

from math import *

working_days = 250

amount_list= []

amount_list2 = []

def pareto(xm,pareto_a):    #define a pareto PRNG

def poisson(lamb):       #define a poisson PRNG


package_loss = []          # define a list that contains pareto random numbers,

for i in range(10000):     # which is used to resample the possible losses for lost parcel

    package_loss.append(pareto(200, 3.0))


package_shipped_perday = []     # define a list that contains poisson random numbers,

for i in range(10000):      # which is used to resample the number of parcels shipped daily

    package_shipped_perday.append(poisson(100))


def simulate_once():

    sum = 0.0

    amount = 0.0

    amount_list = []

    for i in range(0,working_days):

        amount = .0
        for p in range(0,package_shipped_perday[random.randint(0,len(package_shipped_perday)-1)]):
            if 0.02>random.random() :
                amount = amount + package_loss[random.randint(0, len(package_loss)-1)]
        amount_list.append(amount)
    amount_list.sort()
    for i in range(int(working_days)):
        sum = sum + amount_list[i]
    return sum
```

STEP 5: simulate the above code for N times and get the 3 % percentile as your final result. Code is like:

```
def simulate_many(n):
    fp = open('simulate_many.txt','w')
    amount = 0.0
    for i in range(n):
        amount_list2.append(simulate_once())
    for i in range(n):
        print >>fp,amount_list2[i]
    fp.close()
    print amount_list2[int(n*0.97)]    # return the 97% percentile
```

1. Approximation of the result

E[Parcels shipped per day] = 100 dollars

E[insurance per parcel] = alpha*Xm/(alpha-1) = 3*200/2 = 300 dollars

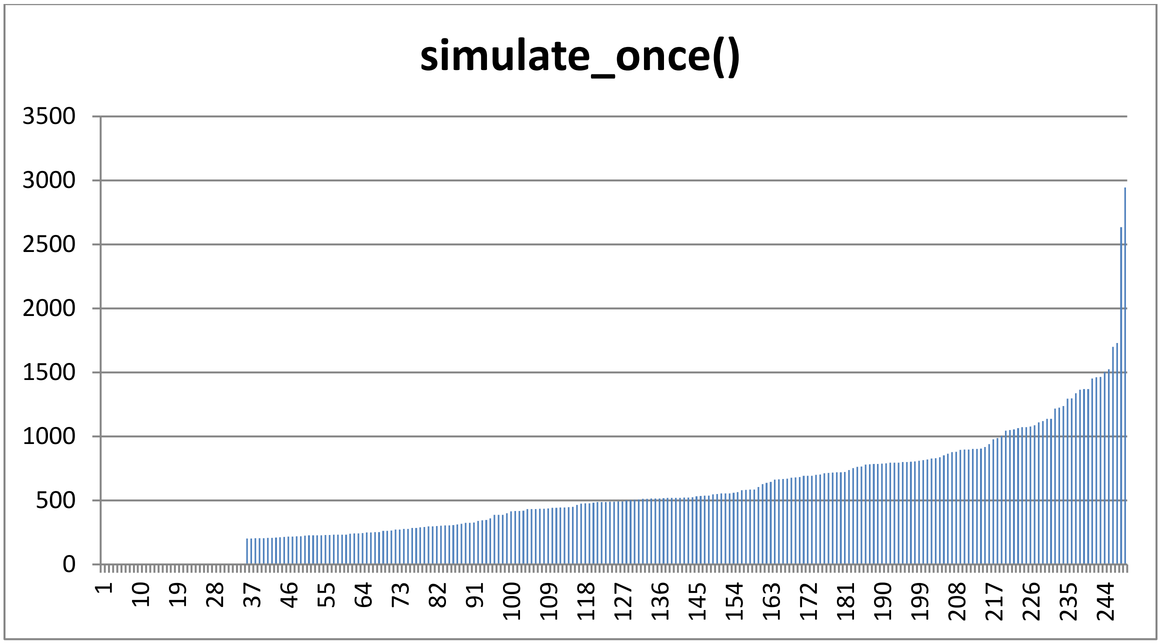
Since there is 2% likely losing the parcel,

E[redeem per day] = 100*300*2% = 600

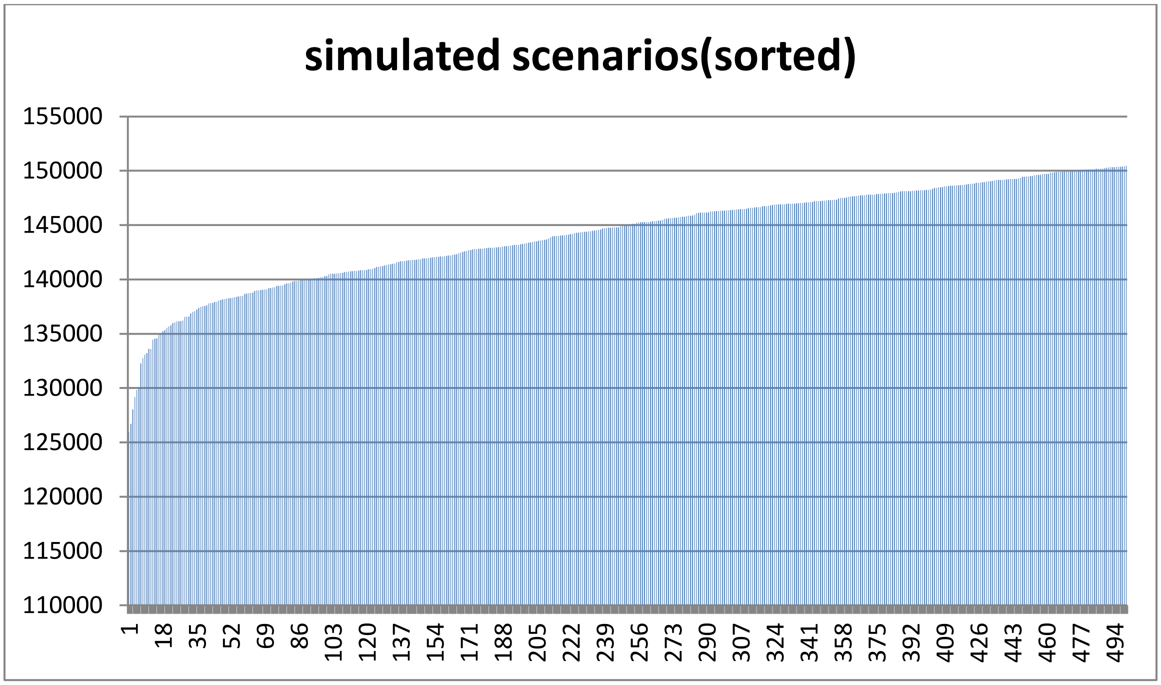
E[redeem per year] = 600 * 250 = 150000

2. Graphics and python result

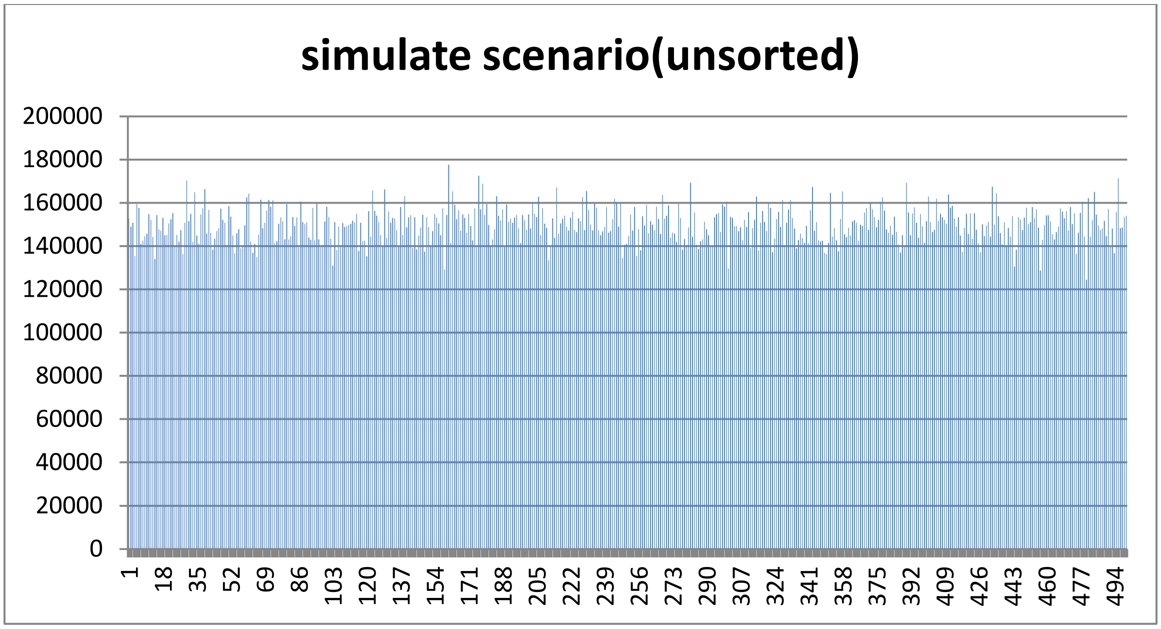
Simulate_once():



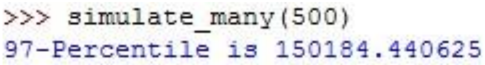
Simulate_many(500):



And

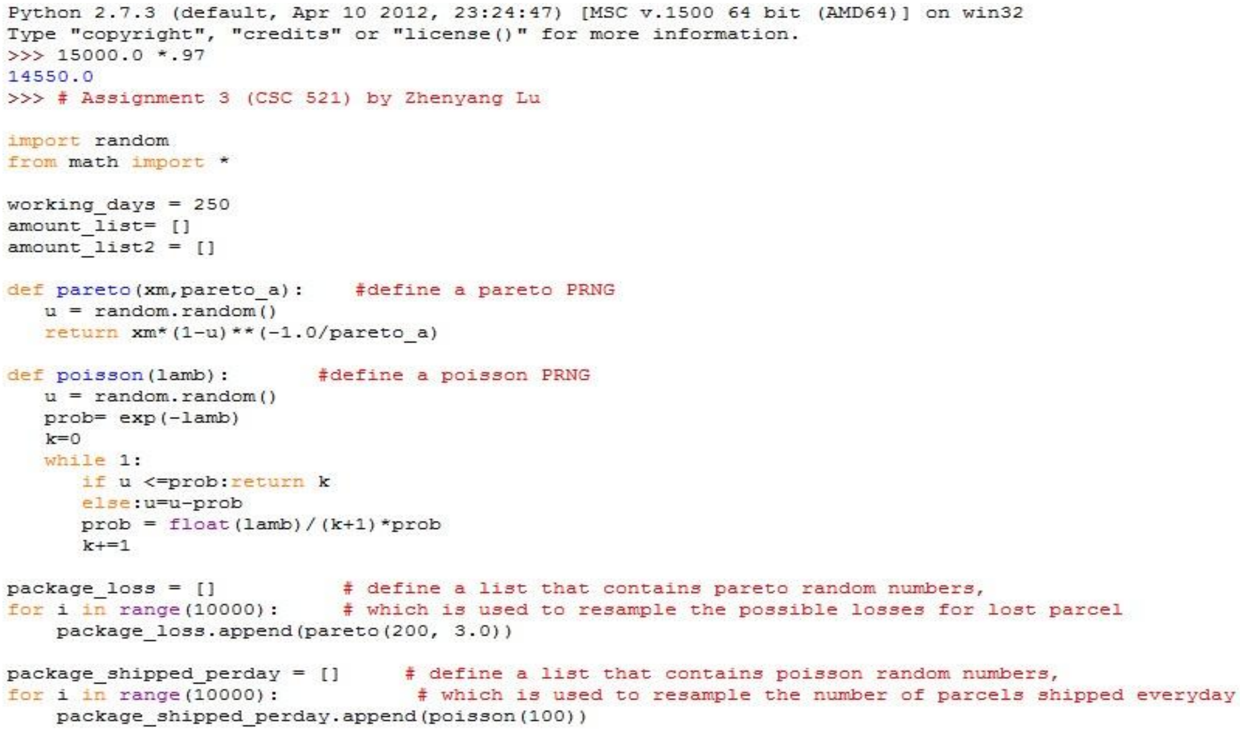


For the Python result:

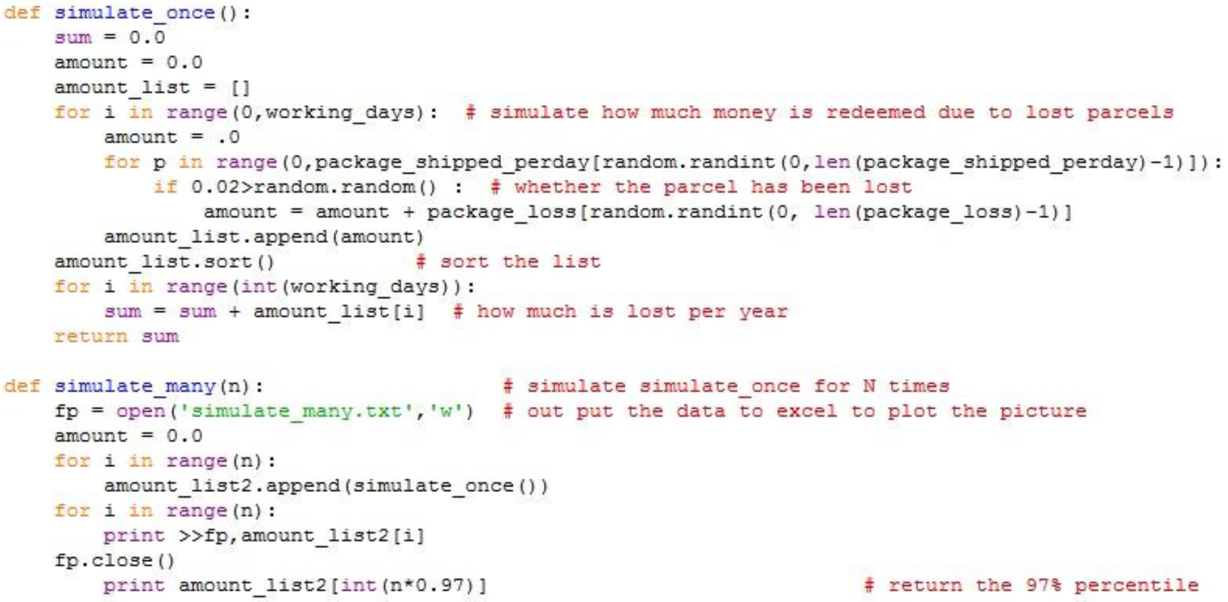


```
>>> simulate_many(500)
97-Percentile is 150184.440625
```

3. Full python code



```
Python 2.7.3 (default, Apr 10 2012, 23:24:47) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 15000.0 *.97
14550.0
>>> # Assignment 3 (CSC 521) by Zhenyang Lu

import random
from math import *

working_days = 250
amount_list= []
amount_list2 = []

def pareto(xm,pareto_a):     #define a pareto PRNG
   u = random.random()
   return xm*(1-u)**(-1.0/pareto_a)

def poisson(lamb):        #define a poisson PRNG
   u = random.random()
   prob= exp(-lamb)
   k=0
   while 1:
      if u <=prob:return k
      else:u=u-prob
      prob = float(lamb)/(k+1)*prob
      k+=1

package_loss = []          # define a list that contains pareto random numbers,
for i in range(10000):     # which is used to resample the possible losses for lost parcel
    package_loss.append(pareto(200, 3.0))

package_shipped_perday = []     # define a list that contains poisson random numbers,
for i in range(10000):           # which is used to resample the number of parcels shipped everyday
    package_shipped_perday.append(poisson(100))
```



```
def simulate_once():
    sum = 0.0
    amount = 0.0
    amount_list = []
    for i in range(0,working_days):  # simulate how much money is redeemed due to lost parcels
        amount = .0
        for p in range(0,package_shipped_perday[random.randint(0,len(package_shipped_perday)-1)]):
            if 0.02>random.random() :  # whether the parcel has been lost
                amount = amount + package_loss[random.randint(0, len(package_loss)-1)]
        amount_list.append(amount)
    amount_list.sort()           # sort the list
    for i in range(int(working_days)):
        sum = sum + amount_list[i]  # how much is lost per year
    return sum

def simulate_many(n):                   # simulate simulate_once for N times
    fp = open('simulate_many.txt','w')  # out put the data to excel to plot the picture
    amount = 0.0
    for i in range(n):
        amount_list2.append(simulate_once())
    for i in range(n):
        print >>fp,amount_list2[i]
    fp.close()
        print amount_list2[int(n*0.97)]                             # return the 97% percentile
```

*Plot is done by Excel