

DSC 430 Assignment 1002

2021 Winter

Name: Wanshu Wang

ID#: 1818536

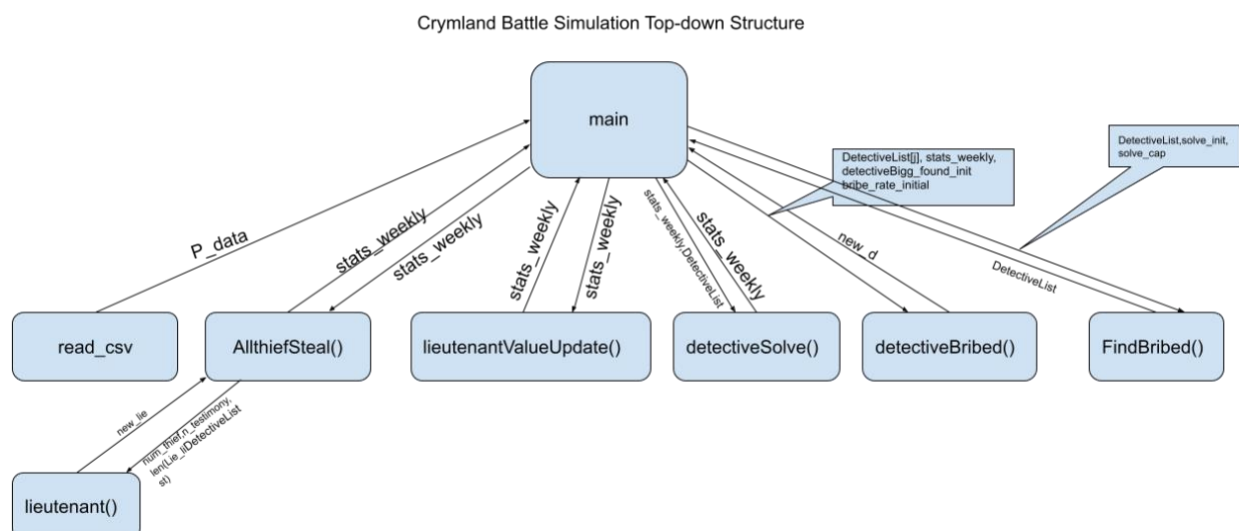
Honor Statement: I have not given or received any unauthorized assistance on this assignment

Video Link: <https://youtu.be/GZm87XcQUVA>

Crymland Stimulation Analysis Report

1. Top-down structure chart

The top-down chart of the simulation is as following. The main processes of every week actions of Bigg's criminal syndicate are 1. All thieves steal (function AllthiefSteal()), 2. Update the wealth of lieutenant, except Bigg by using function lieutenantValueUpdate()). 3. Detectives solve problem, and update the jailed status of Bigg's criminal syndicate, their wealth, and detectives' seize value. (function detectiveSolve()). 4. Bigg tries to bribe detectives. 5. Find out if any detective is bribed. The function lieutenant() in AllthiefSteal() is used to transfer a thief into a lieutenant if this thief gets a promotion. The assumption I made in the stimulation is once a lieutenant is arrested (not Bigg), all the thieves or other lieutenants who works for him/her, would be deleted in the Bigg's criminal syndicate. They would not be in jail, and not work for any other lieutenant either. It would be updated in detectiveSolve() function.



2. Describe the classes

The classes I created are:

- thief – can get a heist value of a week, and has wealth, jailed status, the information of the lieutenant he/she works for.
- lieutenant – expends thief, and has thief list, and can update wealth from getting heist from theives
- Bigg – expends lieutenant, has week number, bribed amount.
- Detective - has solve probability, seize value, and can update the solve probability
- detectiveBribed – expends detective. Bribed detective is with 0 as the probability to get a thief successfully, and has the probability of being discovered.
- StatsWeekly – save all the record for each week, including Bigg, the number of actors, the number of thieves, the numbers of thieves/lieutenants jailed, etc.

3. Data analysis

1) Time Series Charts

The following charts show Mr. Bigg's personal wealth and the amount of bribed, the number of actors, the number of thief/lieutenants jailed against week number. Fig 1 and Fig 2 are in one stimulation, and Fig 3 and Fig 4 are in another. They are all with the same initial parameters, which are showed in P1.csv in appendix.

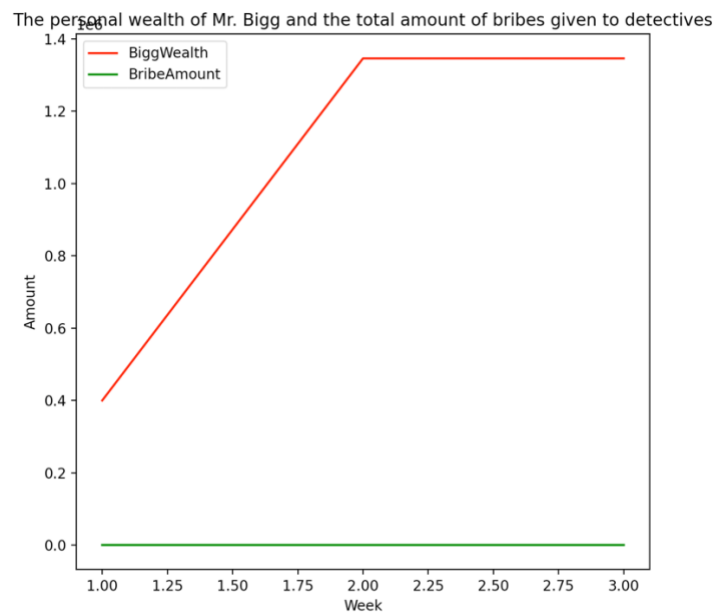


Fig 1

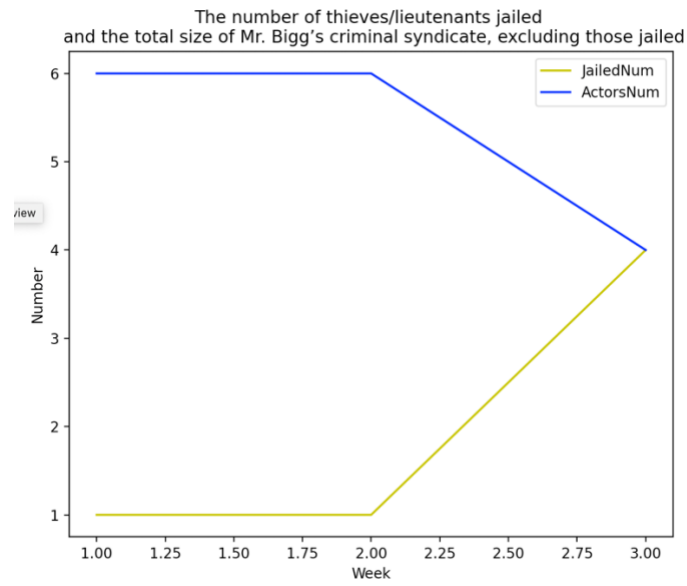


Fig 2

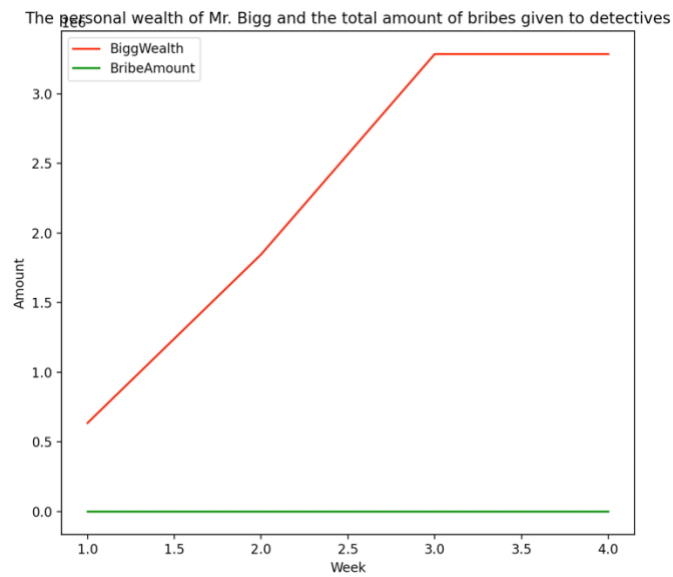


Fig 3

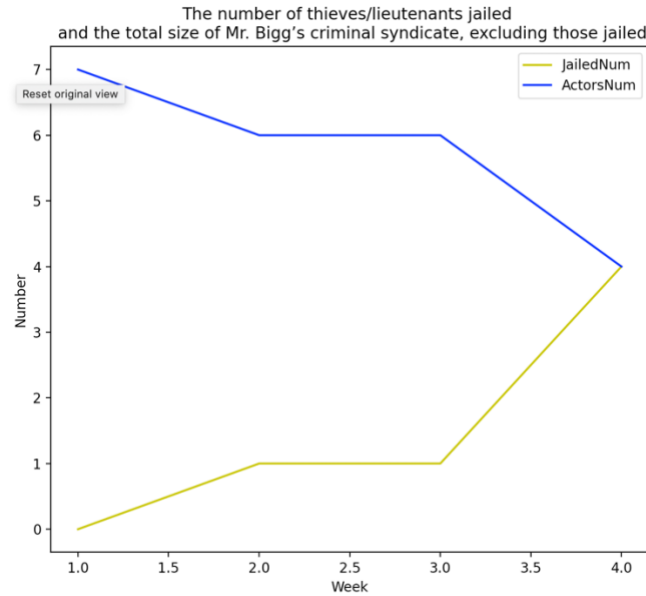


Fig 4

2) Discussion

We can see from Fig 1 and Fig 2 Mr. Bigg was arrested in Week 3. And his wealth increased quickly in the second week, when no thief was arrested in that week. Fig 3 and F4 show Mr. Bigg was arrested in Week 4. There are four in jail including him. His wealth still increased fast even when a thief was arrested in Week 2. It was up to \$ 1143500. Mr. Bigg would be arrested very fast with these parameters. (Appendix P1.csv). There is no chance that Bigg bribes detectives. Detectives would solve at least one thief in second week usually.

3) Other scenarios

I changed the initial number of thieves from 7 to 10, and the detectives from 3 to 2. (P2.csv) The results and the charts are showed below.

	Bigg_Jailed	BiggWealth	JailedNum	BribeAmount	ActorsNum
1	FALSE	1001500	0	0	10
2	FALSE	2489500	0	0	10
3	FALSE	4891000	0	0	10
4	FALSE	8125500	1	0	9
5	FALSE	12055000	1	0	9
6	FALSE	16605500	1	0	9
7	FALSE	21630000	2	0	8
8	TRUE	21630000	4	0	7

Fig 5

I run the simulation many times, and Mr. Bigg were still arrested with a couple of weeks. I changed the scenarios into P3.csv with very low solve probability. And I ran many time under this initials. Mr. Bigg were always free in Week 500. The charts are showing below.

weeks	num_thief	heist_coef	promotion_wealth	n_detectors	solve_init	solve_cap	n_testimony	bribe_init	bribe_rate_initial	detectiveBigg_found_init
500	10	10	100000000	3	0.001	0.005	10	100000		0.5

The personal wealth of Mr. Bigg and the total amount of bribes given to detectives

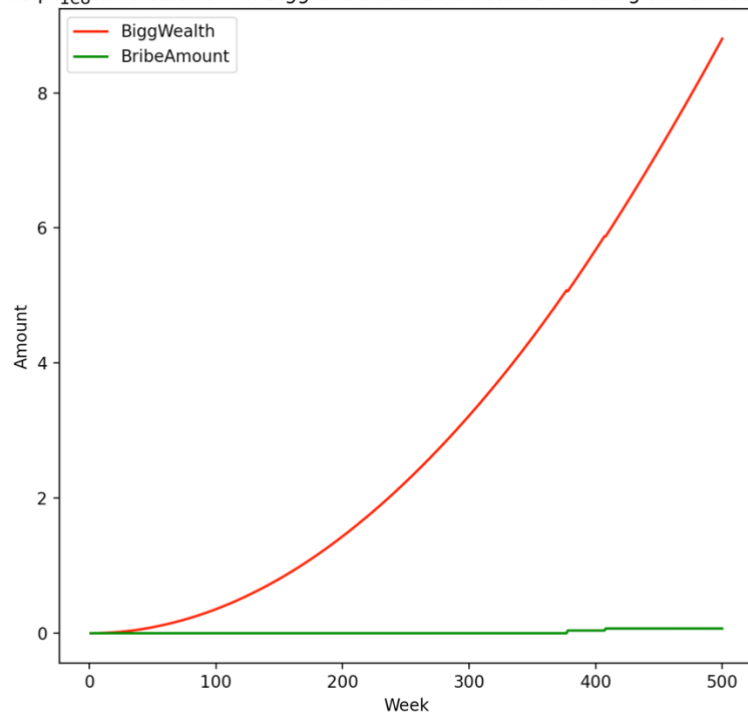


Fig 6

The number of thieves/lieutenants jailed and the total size of Mr. Bigg's criminal syndicate, excluding those jailed

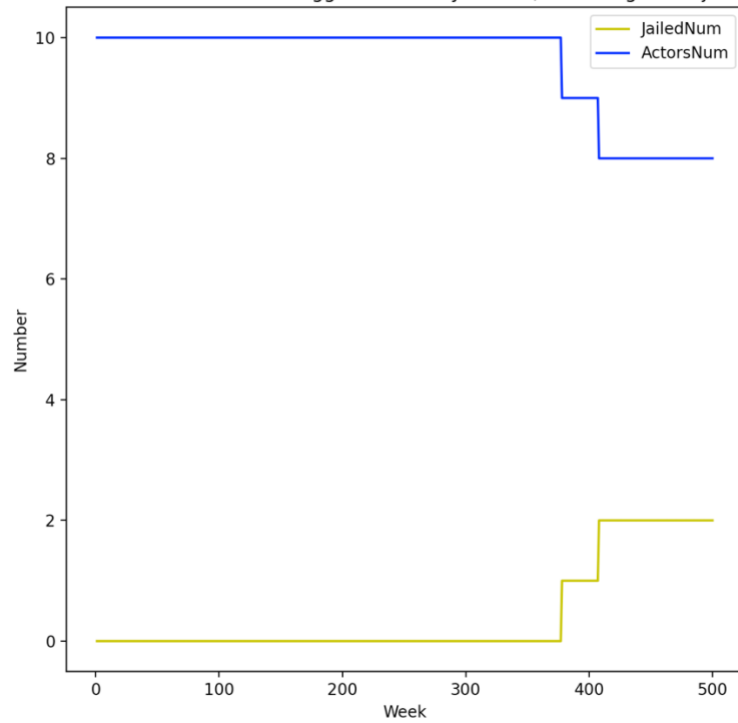


Fig 7

4. Stimulation Extension

There are three ways I'm thinking to extend this stimulation. First, the thief could have the probability to steal successfully. Similar to the experience gaining of detectives, the thief's steal probability could be increase and has a maximum. Second, the thief could have the probability to escape from the detective. This probability could decrease detectives solve the probability. Third, the thieves or lieutenants could also have chance to work for detectives after they arrested. They can make other thieves or lieutenants be arrested in order to get Bigg finally.

5. Appendix

P1.csv:

weeks	num_thief	heist_coef	promotion_wealth	n_detetectives	solve_init	solve_cap	n_testimony	bribe_init	bribe_rate_initial	detectiveBigg_found_init
500	7	1000	1000000	3	0.25	0.75	3	1000000	0.1	0.05

P2.csv

weeks	num_thief	heist_coef	promotion_wealth	n_detetectives	solve_init	solve_cap	n_testimony	bribe_init	bribe_rate_initial	detectiveBigg_found_init
500	10	1000	1000000	2	0.25	0.75	3	1000000	0.1	0.05

P3.csv

weeks	num_thief	heist_coef	promotion_wealth	n_detetectives	solve_init	solve_cap	n_testimony	bribe_init	bribe_rate_initial	detectiveBigg_found_init
500	10	10	100000000	3	0.001	0.005	10	100000	0.5	0

Code (plot)

```
d = pd.read_csv('BC_WeeklyData1.csv')
fig, ax = plt.subplots()
ax3 = ax.twinx()
rspine = ax3.spines['right']
rspine.set_position(('axes',1.15))
ax3.set_frame_on(True)
ax3.patch.set_visible(False)
fig.subplots_adjust(right=0.7)
d.BigWealth.plot(ax = ax, style = "r-", xlabel='Week', ylabel = 'Amount', legend = True, use_index = True)
d.BribeAmount.plot(ax = ax, style = "g-", xlabel='Week', ylabel = 'Amount', title='The personal wealth of Mr. Bigg and the total amount of bribes given to detectives', legend = True, use_index = True)

fig, ax = plt.subplots()
ax3 = ax.twinx()
rspine = ax3.spines['right']
rspine.set_position(('axes',1.15))
ax3.set_frame_on(True)
ax3.patch.set_visible(False)
fig.subplots_adjust(right=0.7)
d.JailedNum.plot(ax = ax, style = "y-", xlabel='Week', ylabel = 'Number', legend = True, use_index = True)
d.ActorsNum.plot(ax = ax, style = "b-", xlabel='Week', ylabel = 'Number', title='The number of thieves/lieutenants jailed \n and the total size of Mr. Bigg's criminal syndicate, excluding those jailed', legend = True, use_index = True)
plt.show()
```

Code (Class)

```
class thief():
    "class represent a thief"
    def __init__(self, heist_coef = 1000, LieID = 0):
        'initialize parameters'
        self.jailed_status = False
        self.wealth = 0
        self.value = 0 #heist of this week
        self.heist_coef = heist_coef
        self.LieID = LieID # ID of the lieutenant this thief is under, Bigg
        ID is 0, initial thief.LieID = Bigg.ID
    def UpdateWealth(self):
        'Update Wealth'
        if self.jailed_status == False: #if not jailed
            self.wealth = self.wealth + 0.5 * self.value
        else: self.wealth = 0
    def getWeeklyValue(self):
        'calculate the heist of this week'
        d = random.randint(1,20)
        self.value = self.heist_coef * (d**2)

class lieutenant(thief):
    "class represent a lieutenant, extends class thief"
    def __init__(self, num_thief=7, testimony=3, heist_coef=1000, LieID=0, ID =
    1):
        self.jailed_status = False
        self.wealth = 0
        self.value = 0 #heist of this week
        self.heist_coef = heist_coef
        self.LieID = LieID # ID of the lieutenant this thief is under, Bigg
        ID is 0, initial thief.LieID = Bigg.ID
        self.thievesList = []*num_thief #all the thief this lieutenant has,
        initial is 7
        self.lieutenantList = [] #all the lieutenant this lieutenant has
        self.num_thief = num_thief
        self.num_lie = 0
        self.n_testimony_jail = testimony
        self.n_testimony = 0
        self.ID = ID # the ID of this lieutenant, initial is 1 , Bigg.ID = 0
        self.heist_coef = heist_coef

    def init_thievesList(self):
        "initail this new lieutenant's thief list"
        for i in range(0, self.num_thief):
            self.thievesList.append(thief(self.heist_coef, self.ID))
        self.num_thief = len(self.thievesList)
```

```

def UpdateWealth(self):
    "Update the wealth by calculating the value of this week"
    if self.jailed_status == False: #if not jailed
        self.num_thief = self.getTheifNum()
        self.num_lie = self.getLieNum()

        if self.num_thief !=0:
            for i in range (0,self.num_thief ):
                self.value = self.value + self.thievesList[i].value * 0.5
        if self.num_lie !=0:
            for i in range (0,self.num_lie):
                self.value = self.value + self.lieutenantList[i].value *
                    0.5

        self.wealth = self.wealth + self.value

    else:
        self.wealth = 0

def UpdateJailedStatus(self):
    'Update Jailed Status'
    num_Jailed = 0
    n_thief = len(self.thievesList)
    n_lie = len(self.lieutenantList)
    i = 0
    j = 0
    while num_Jailed< self.n_testimony_jail and (i < n_thief):
        if self.thievesList[i].jailed_status == True: # if any thieves
            are arrested
            num_Jailed = num_Jailed + 1
        if j < n_lie and self.lieutenantList[j].jailed_status == True: #
            if any lieutenants are arrested
            num_Jailed = num_Jailed + 1
        i +=1
        j +=1
    self.n_testimony += num_Jailed
    if self.n_testimony >=self.n_testimony_jail: self.jailed_status =
    True # if n_testimony is more than n_testimony_jail, the rule number
    else: self.jailed_status = False

```



```

def getTheifNum(self):
    'get the number of thieves'
    self.num_thief = len(self.thievesList)
    return self.num_thief
def getLieNum(self):
    'get the number of lieutenants'
    self.num_lie = len(self.lieutenantList)
    return self.num_lie

```

```

class Bigg(lieutenant):
    "class repret the Bigg, extends class lieutenant"
    def __init__(self,num_thief=7,testimony=3,heist_coef=1000,LieID=0, ID =
0):
        self.jailed_status = False
        self.wealth = 0
        self.value = 0 #heist of this week
        self.heist_coef = heist_coef
        #self.LieID = LieID # ID of the lieutenant this thief is under, Bigg
        ID is 0, initial thief.LieID = Bigg.ID

        self.thievesList = []*num_thief #all the thieve this lieutenant has,
        inital is 7
        self.lieutenantList = [] #all the lieutenant this lieutenant has
        self.num_thief = num_thief
        self.num_lie = 0
        self.n_testimony_jail = testimony
        self.n_testimony = 0

        self.week_number = 0
        self.num_lie = 0 # number of lieutenants under Bigg this week,
        initial is 0
        self.bribe_amount = 0 #
        self.wealth_thisweek = 0
        self.ID = 0

```

```

class detective():
    'class detective'
    def __init__(self, solve_prob_init, solve_prob_cap):
        self.solve_prob = solve_prob_init
        self.solve_prob_cap = solve_prob_cap
        self.seizeValue = 0
    def solve(self):
        solve_succeed = False
        if random.random() <= self.solve_prob:
            solve_succeed = True
            self.solve_prob += random.randint(1,10)/100 # experience
            increases x%
            if self.solve_prob > self.solve_prob_cap: # self.solve_prob is
                more than the maximum, set it as the maximum
                self.solve_prob = self.solve_prob_cap

        return solve_succeed

class detectiveBribed(detective):
    'class detectiveBribed, the detective who works for Bigg, extends class
    detective'
    def __init__(self, detectiveBigg_found_init=0.05):
        'initial parameters'
        self.bribed_discover_prob = detectiveBigg_found_init
        self.discover_status = False
        self.solve_prob = 0
    def found(self):
        'check if he is found weekly'
        if random.random() <= self.bribed_discover_prob: # be found
            self.discover_status = True

        else: self.bribed_discover_prob += random.random(1,20)/100 #
            increase x%

        return self.discover_status

```

Code(delete jailed Lieutenant's thieves, in function DetectiveSolve())

```

else: # this lieutenant is not Bigg
    for j in range(1,LieID+1):
        if LieID == stats_weekly.LieList_tot[j].ID:
            stats_weekly.LieList_tot[j].n_testimony += 1
            if stats_weekly.LieList_tot[j].n_testimony>=stats_weekly.
Bigg_week.n_testimony_jail : # a lieutenant is arrested,
detective gets wealth, and remove it from lieList_tot,
remove all his lieutenants and thieves from list
                stats_weekly.LieList_tot[j].jailed_status = True
                DetectiveList[i].seizeValue += stats_weekly.
                LieList_tot[j].wealth
                stats_weekly.LieList_tot[j].wealth = 0 # the wealth
                of the jailed lieutenant is 0

            #remove all his lieutenants and thieves from list
            for l in range(0,len(stats_weekly.LieList_tot[j].
thievesList)): #remove
                temp_thiefList.remove(stats_weekly.LieList_tot
                [j].thievesList[l])
            # if he has lieutenant, remove it
            for l in range(0,len(stats_weekly.LieList_tot[j].
lieutenantList)):
                temp_lieList.remove(stats_weekly.LieList_tot[j].
                lieutenantList[l])

            stats_weekly.LieList_tot.remove(stats_weekly.
            LieList_tot[j])

            #update numbers of actors and arrested
            stats_weekly.tot_jailed += 1
            stats_weekly.tot_lie = stats_weekly.tot_lie - 1
            stats_weekly.n_actors = stats_weekly.n_actors - 1

```