

Van Gestel's Simulation Model in Python Simpy for Presentation

Here is the main Program for the simulation model.

It is OO so most of operations are done outside of this file.

Number of Iterations: 2

Subjects (patients) per iteration: 3000

QALY is calculated based on VFQ score (VFQ-25 questionnaire)

In [1]:

```
#error checking measures
import csv
import matplotlib.pyplot as plt
numberOfGraphs = 15
def csv_dict_writer(path, fieldnames, data):
    with open(path, "wb") as out_file:
        writer = csv.DictWriter(out_file, delimiter=',', fieldnames=fieldnames)
        writer.writeheader()
        for row in data:
            writer.writerow(row)
masterListforReplications = []
field_names = "QALY,TotalCost".split(",")
from PlottingSystemClass import PlottingSystem
from SimulationSystemClass import SimulationSystem
plottingsystem = PlottingSystem(plt)
order = 1
```

The main code is here

It is run for 20 rounds 7 plots are produced for each iteration

In [3]:

```
%matplotlib inline
for i in range(2):
    sysSimulation = SimulationSystem(3000,"PatientList/Patients_list_{}.csv".format(i))
    sysSimulation.SystemSimulation()
    plottingsystem.plot(sysSimulation,order,i,masterListforReplications)
    order += (numberOfGraphs*1)
del sysSimulation
print order
csv_dict_writer("MList.csv",field_names,masterListforReplications)
```

CURRENT ITERATION: 0

Average QALY: 12.6025879228

Average Medical Cost: 26246.4198333

Average MD: -12.753006932

[illegible]

Patient 0: List of Medication Progression is [0, 1, 1, 6, 6, 6, 6, 6, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 10, 10, 30, 30, 30, 30, 30, 30, 30, 30, 3
0, 30, 30]

Patient 0: List of Final Medication Amount is [8, 6, 4, 0, 2]

Patient 1: List of Medication Progression is [0, 1, 2, 2, 2, 16, 16, 18, 18, 18, 18,
18, 18, 18, 18, 18, 18, 18, 18, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19,
19, 19, 19, 19, 19]

Patient 1: List of Final Medication Amount is [1, 7, 3, 5, 1]

Patient 2: List of Medication Progression is [0, 1, 2, 2, 2, 2, 16, 16, 16, 16, 16,
16, 18, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 30, 30, 30, 30, 30, 30, 3
0, 30, 30, 30, 30, 30, 30, 30]

Patient 2: List of Final Medication Amount is [1, 5, 1, 3, 2]

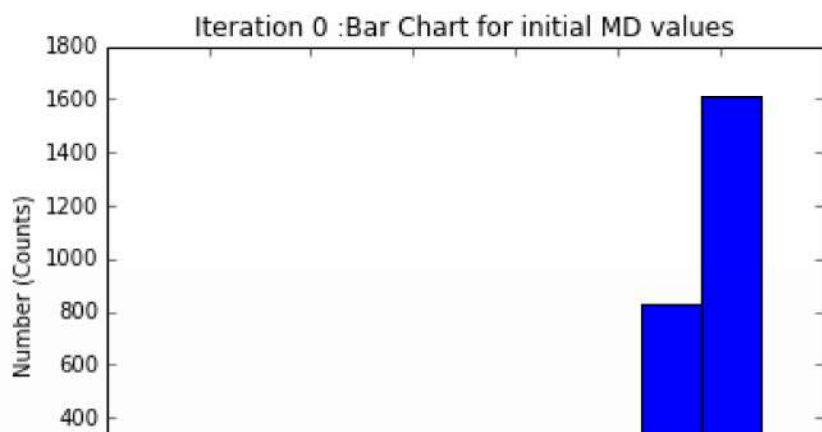
Patient 3: List of Medication Progression is [0, 1, 1, 6, 7, 7, 7, 13, 13, 14, 14, 1
4, 14, 14, 14, 14, 14, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 3
0, 30, 30, 30, 30, 30, 30, 30, 30]

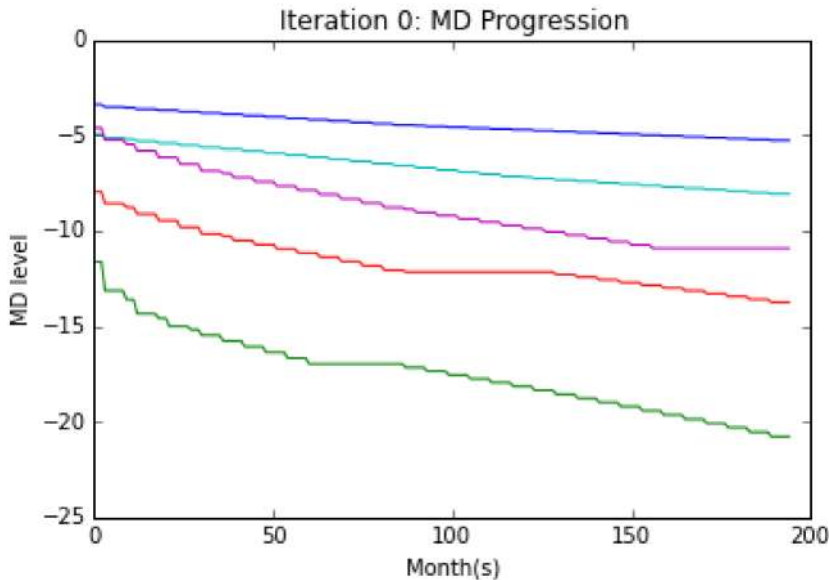
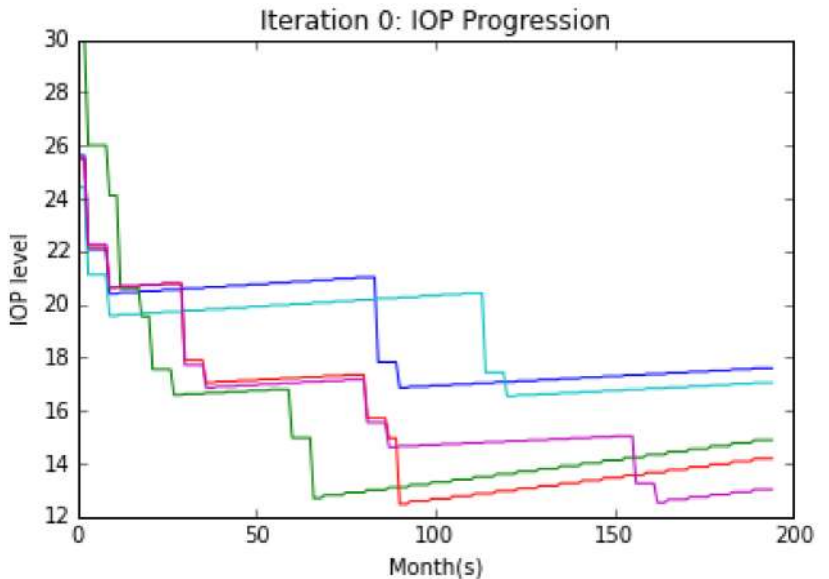
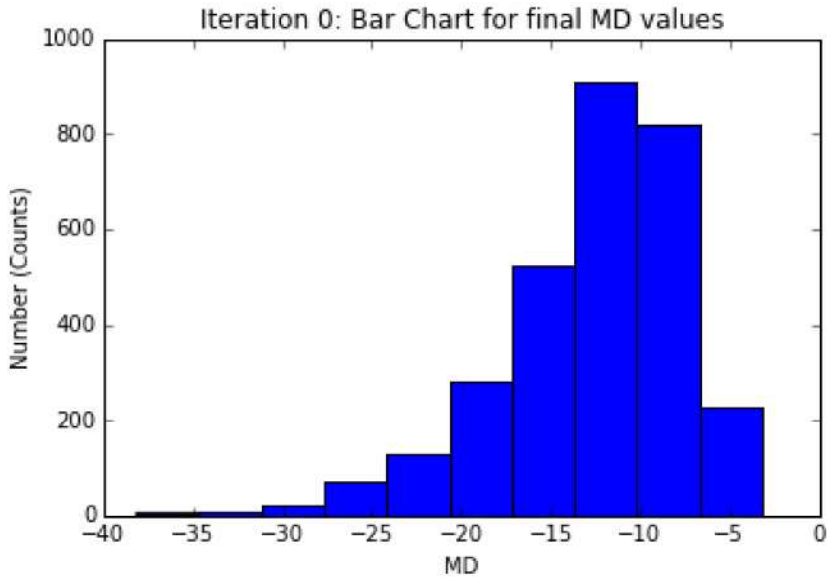
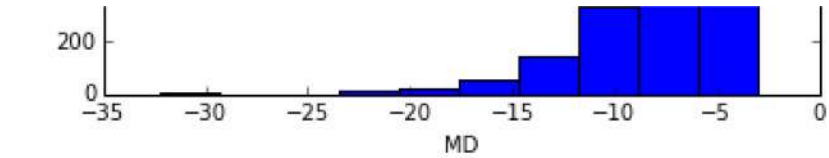
Patient 3: List of Final Medication Amount is [7, 3, 4, 4, 2]

Patient 4: List of Medication Progression is [0, 1, 1, 6, 6, 6, 8, 8, 8, 8, 8, 10, 1
0, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 30, 30, 30, 30, 30, 30, 3
0, 30, 30, 30, 30, 30, 30, 30]

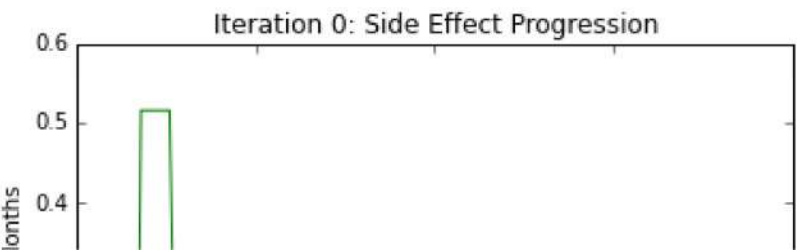
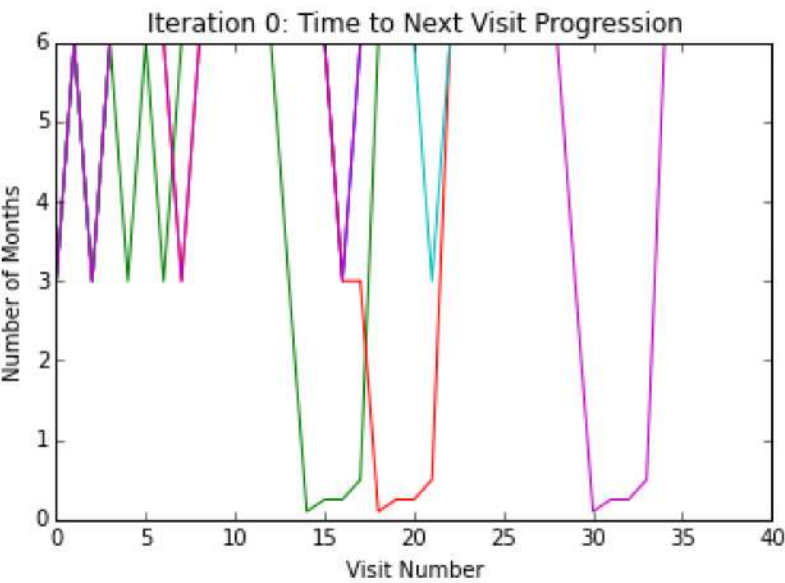
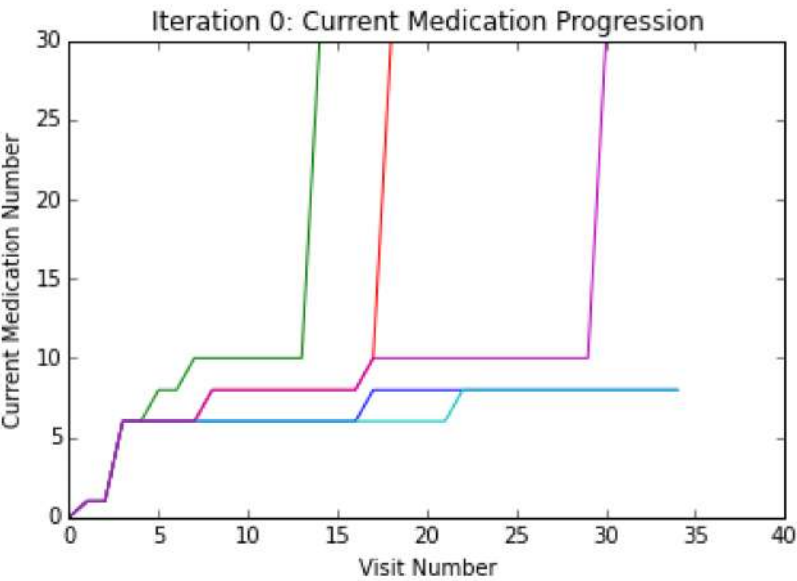
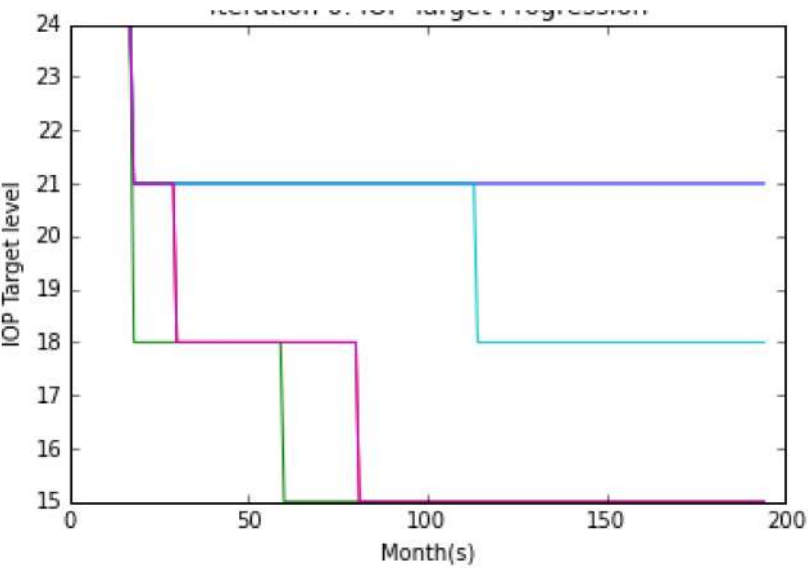
Patient 4: List of Final Medication Amount is [8, 6, 4, 0, 2]

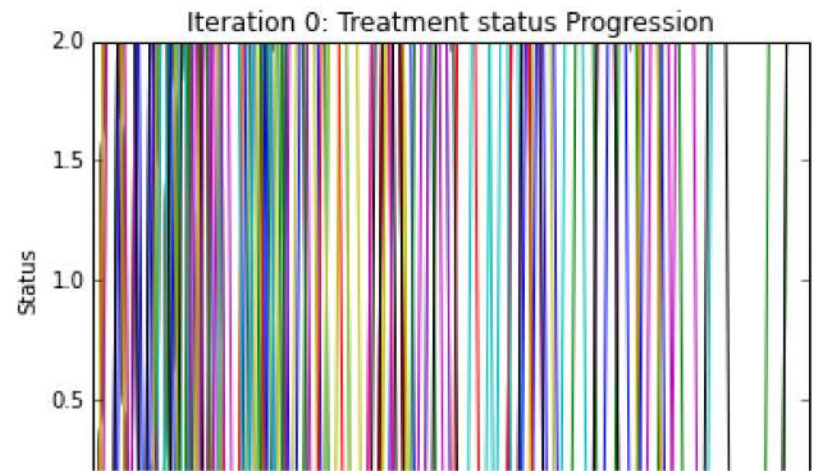
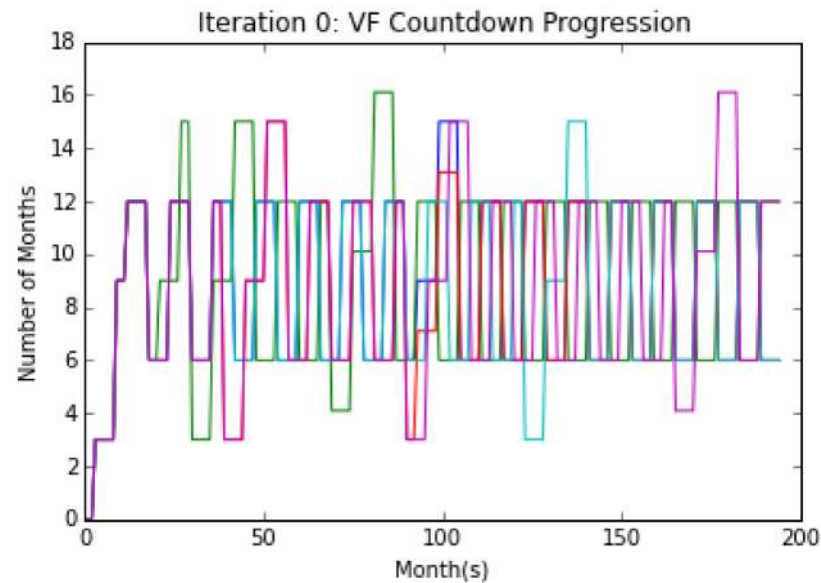
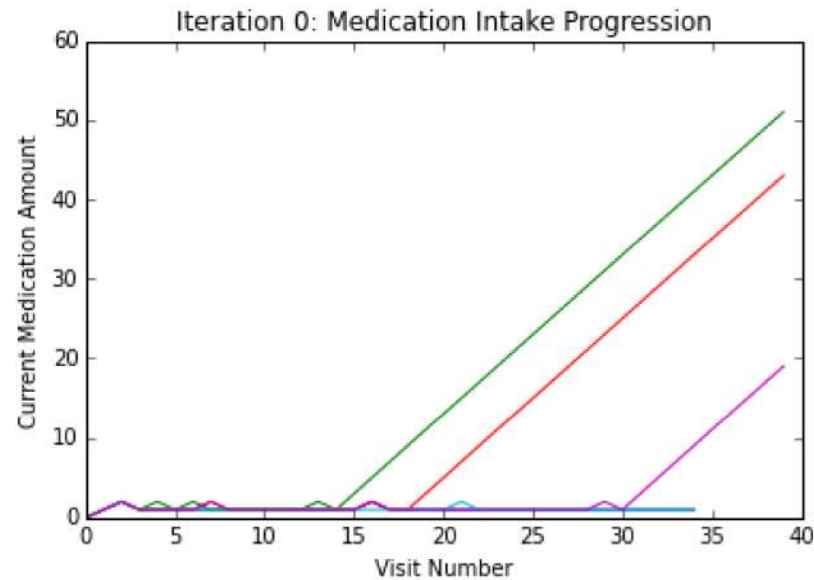
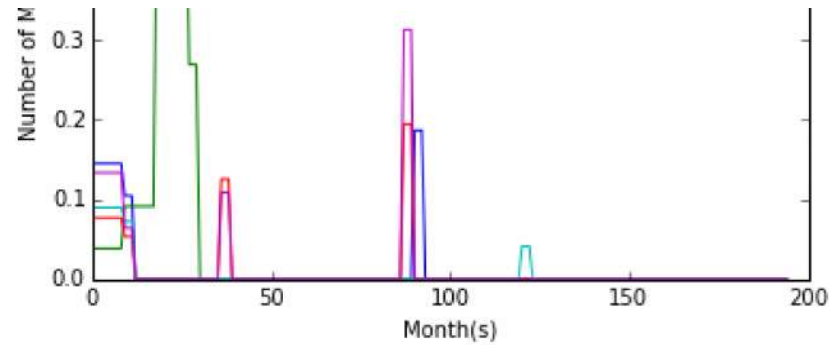
61

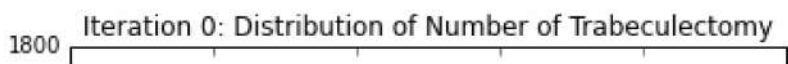
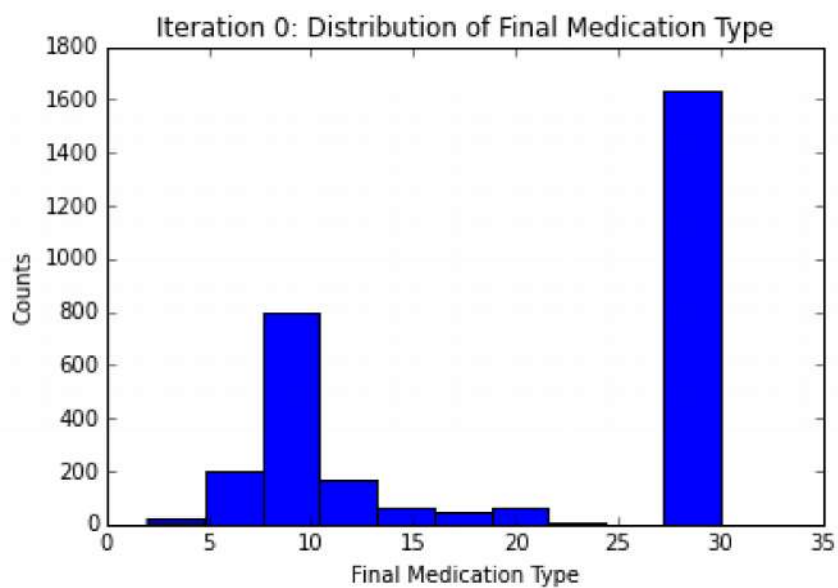
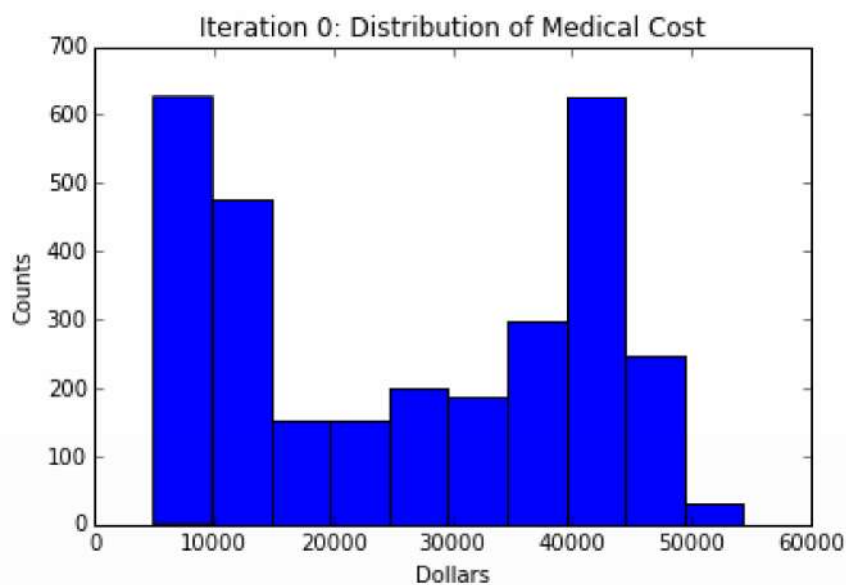
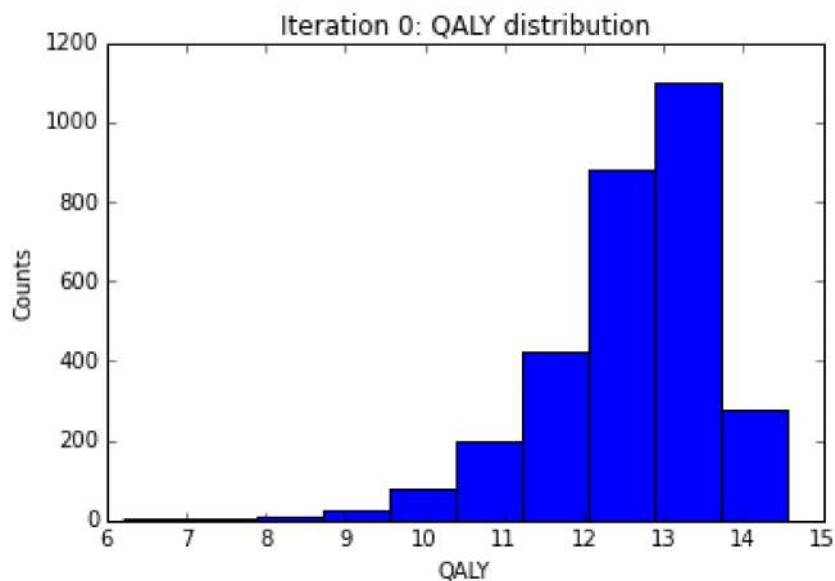


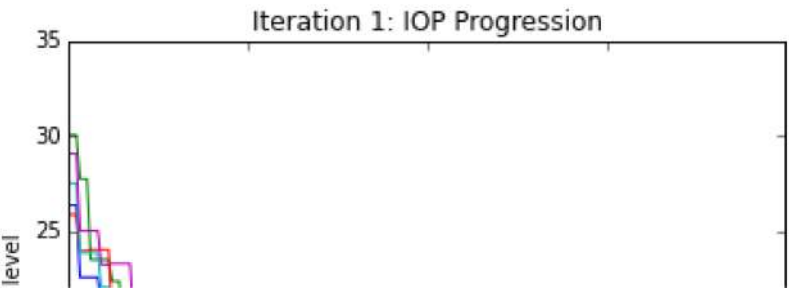
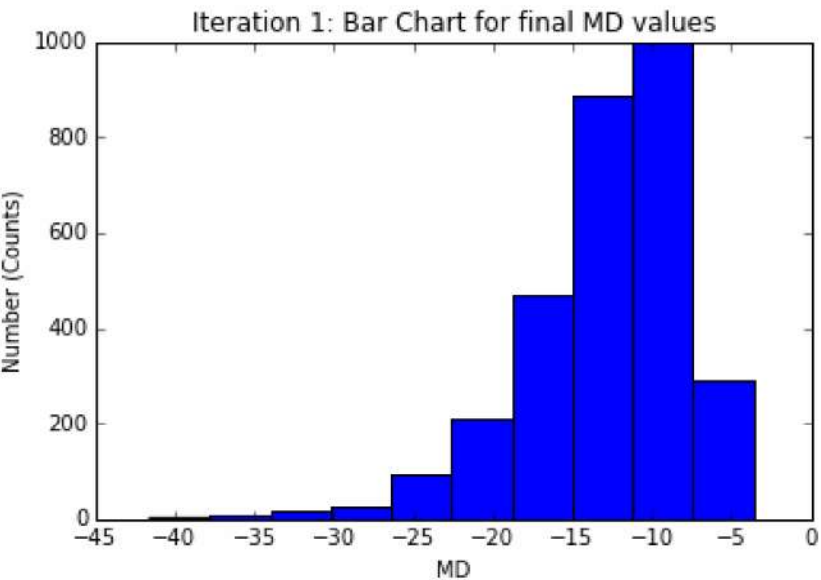
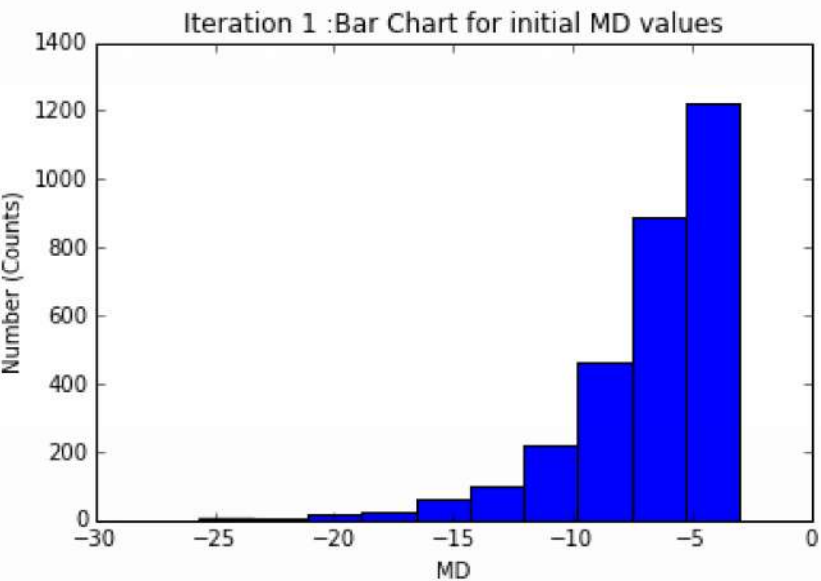
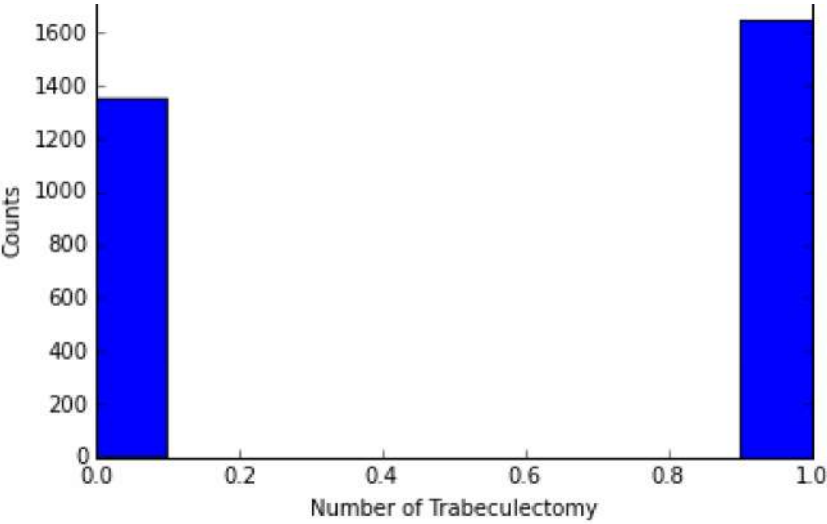


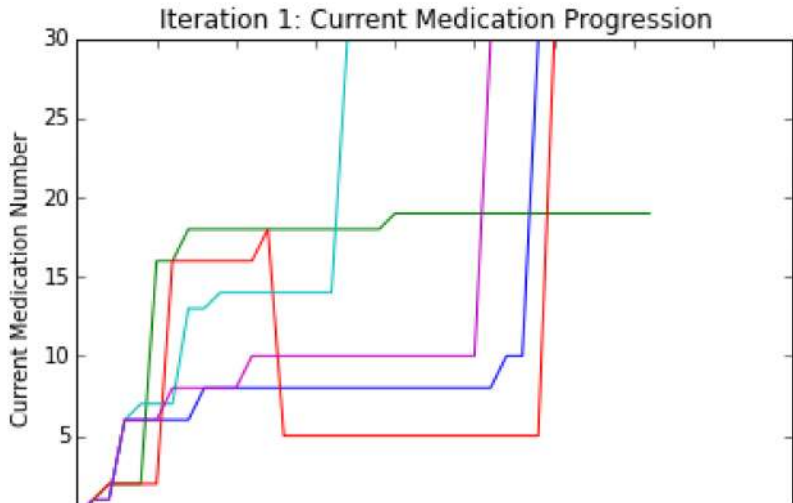
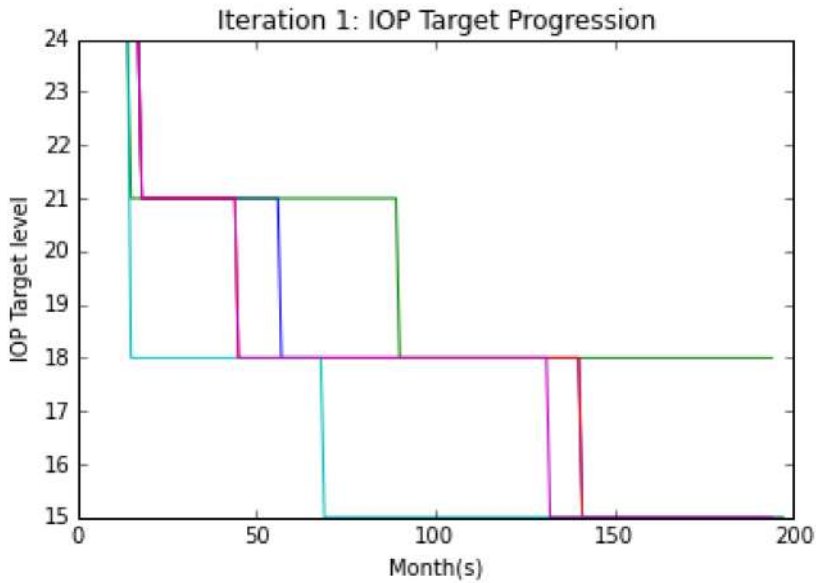
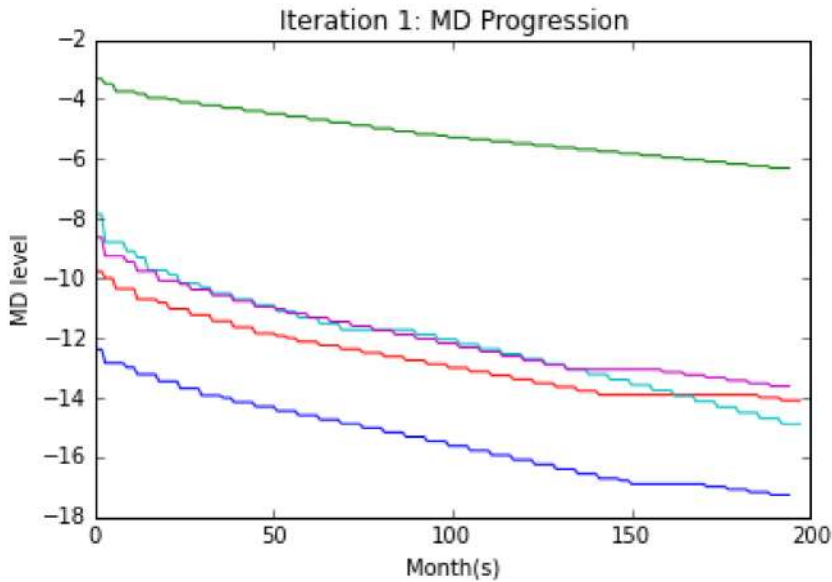
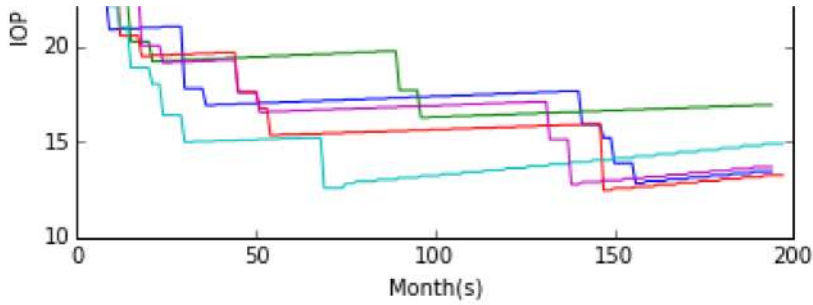
Iteration 0: IOP Target Progression

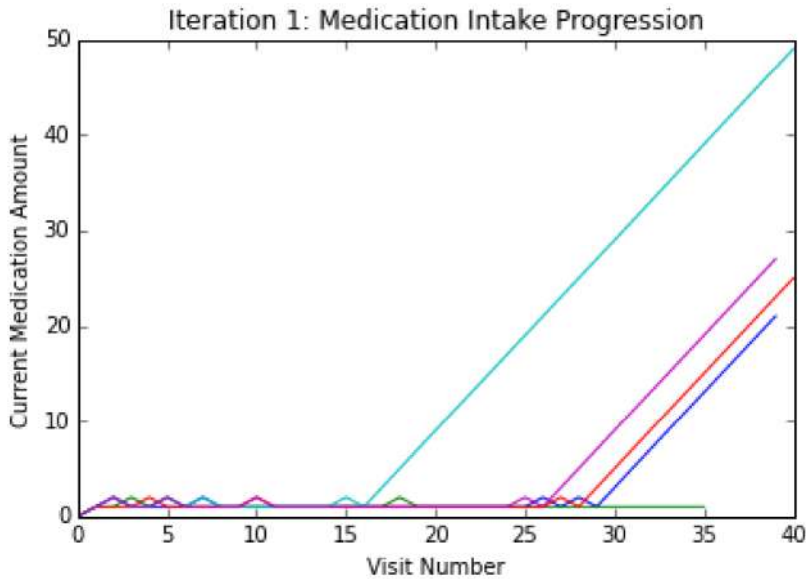
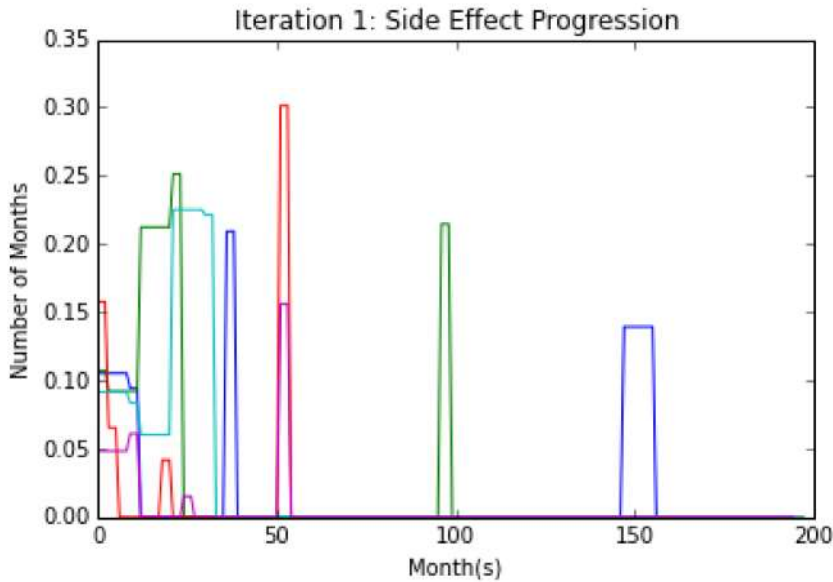
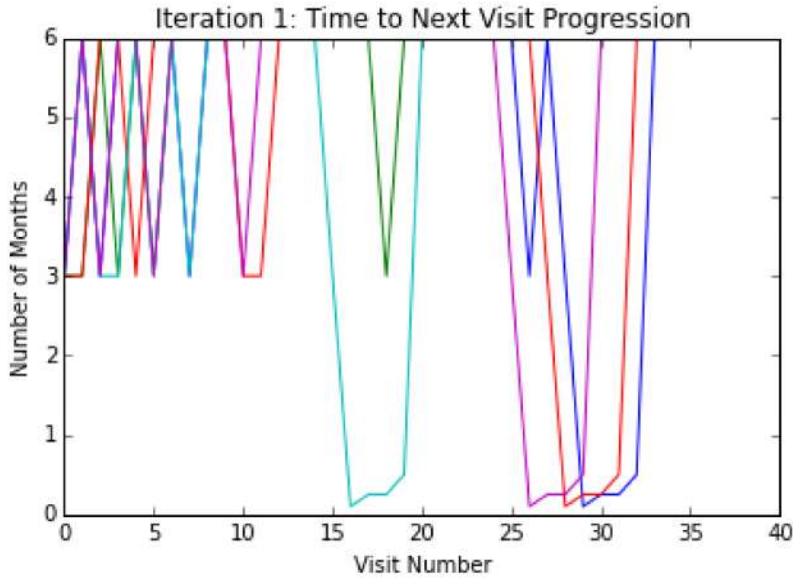
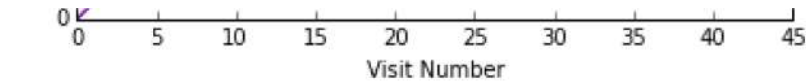


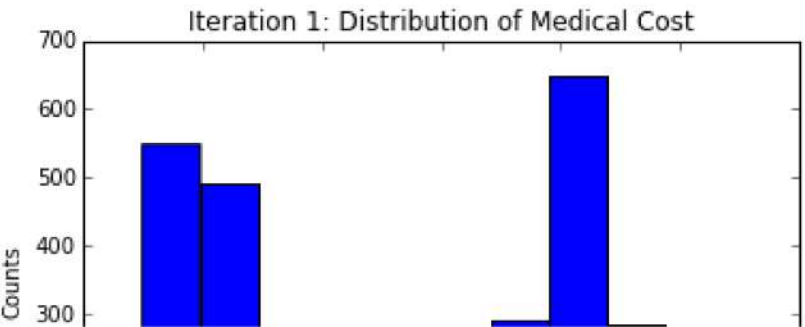
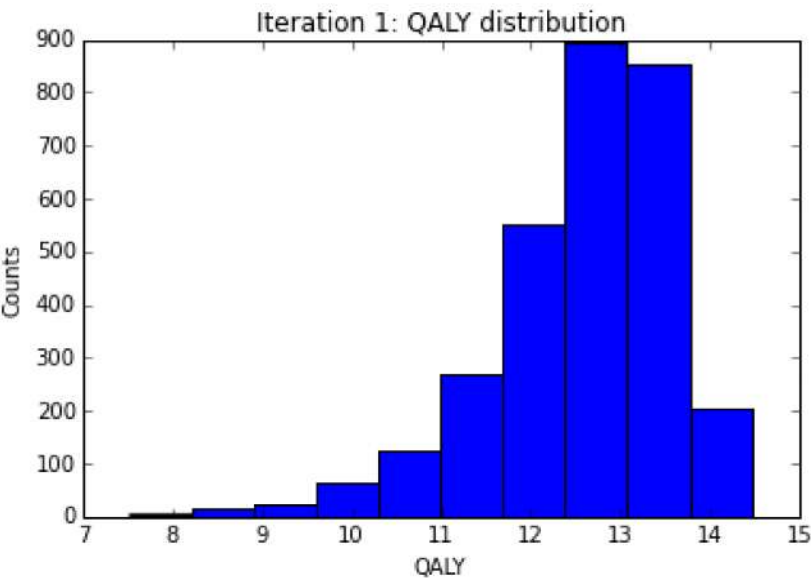
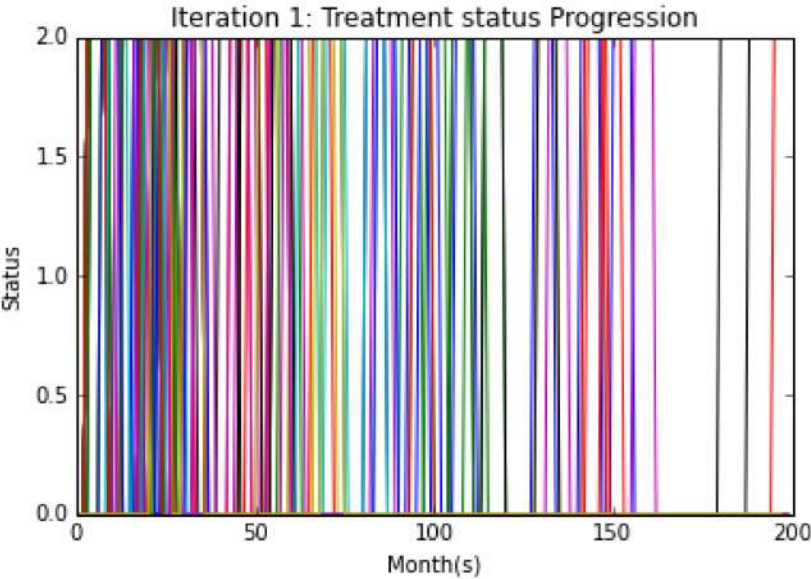
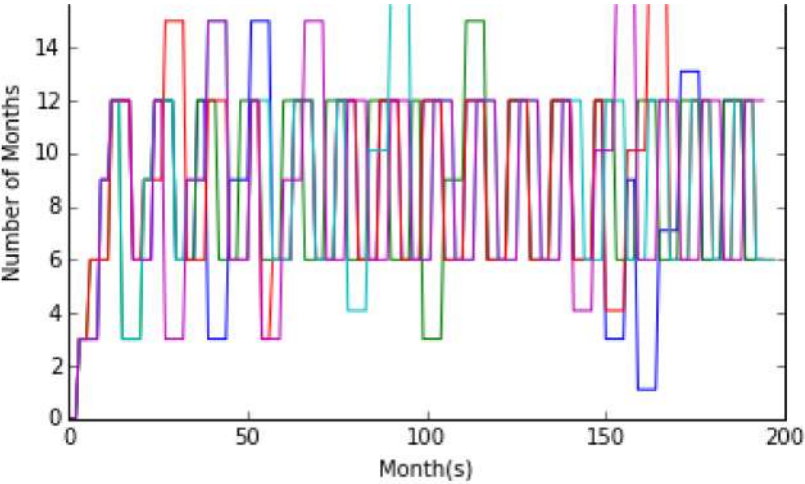


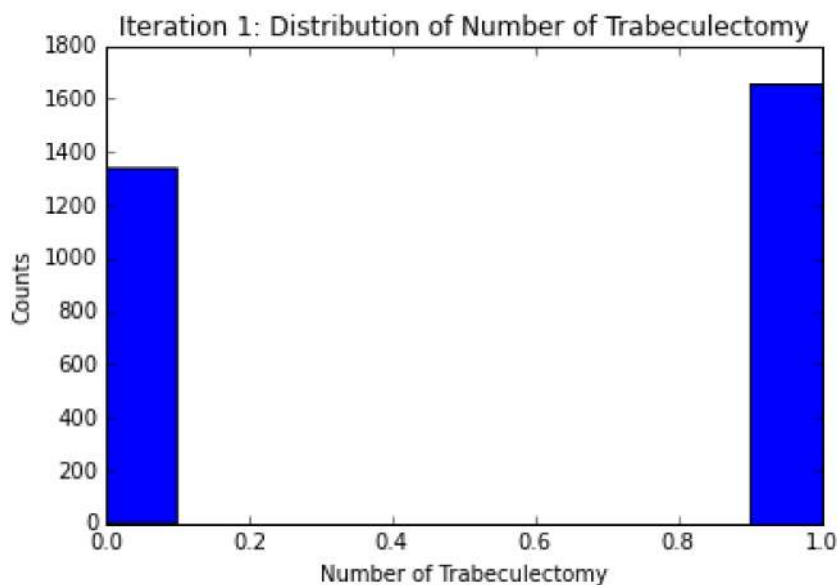
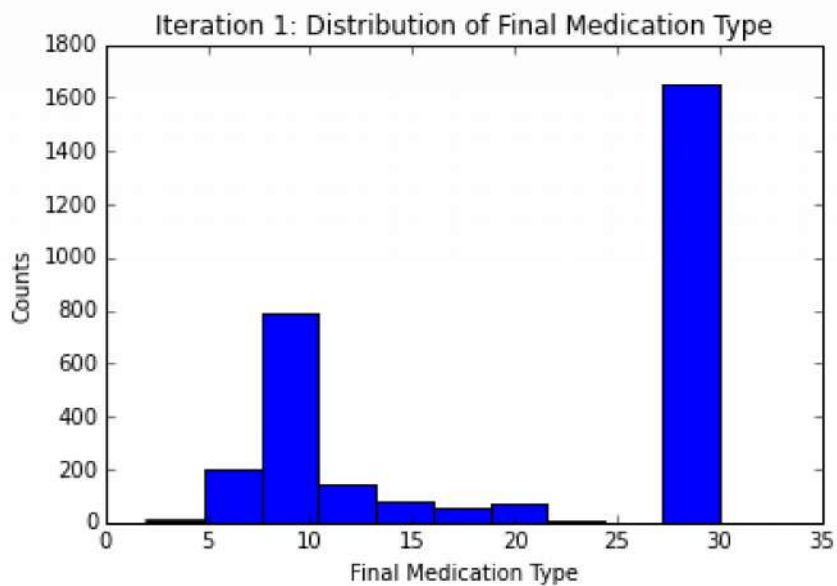
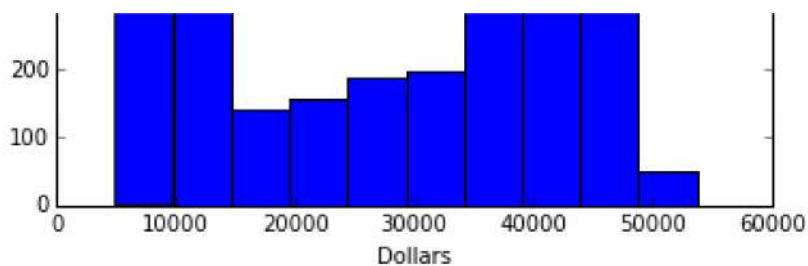






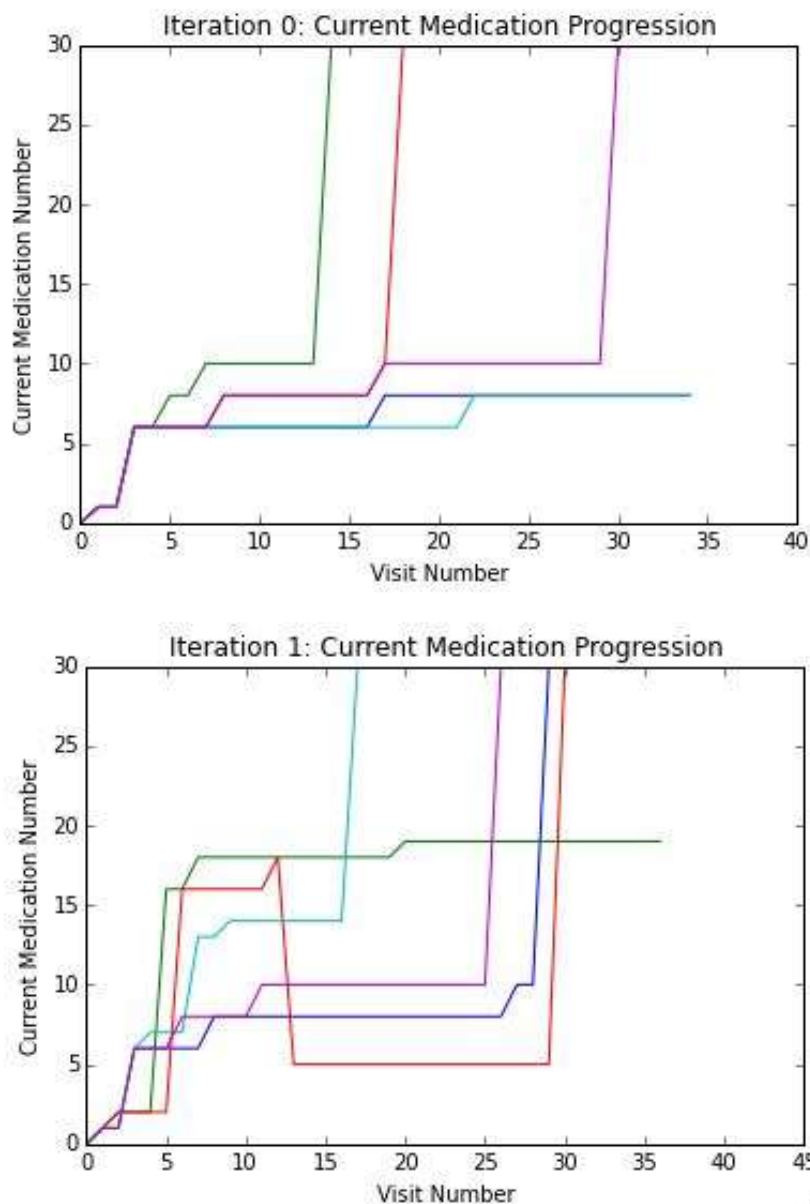






How Patients are moved within the system

In order to track what type of medication/treatment patients are currently under, we can use this chart:



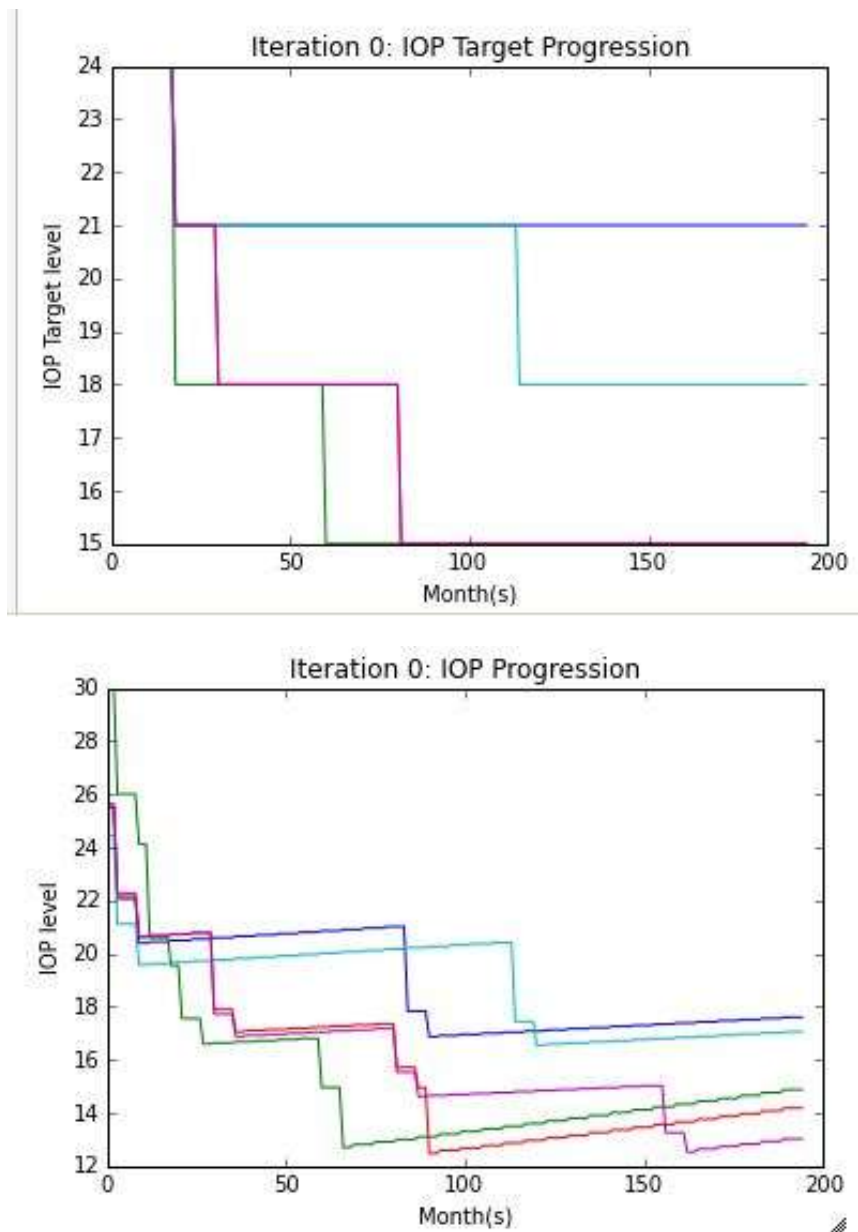
From 1 to 23 are the medications/combinations of Block 1.

Number 30 is Trabeculectomy, number 31 is implant

The key map to identify which medication/combination the patients are currently on is this:



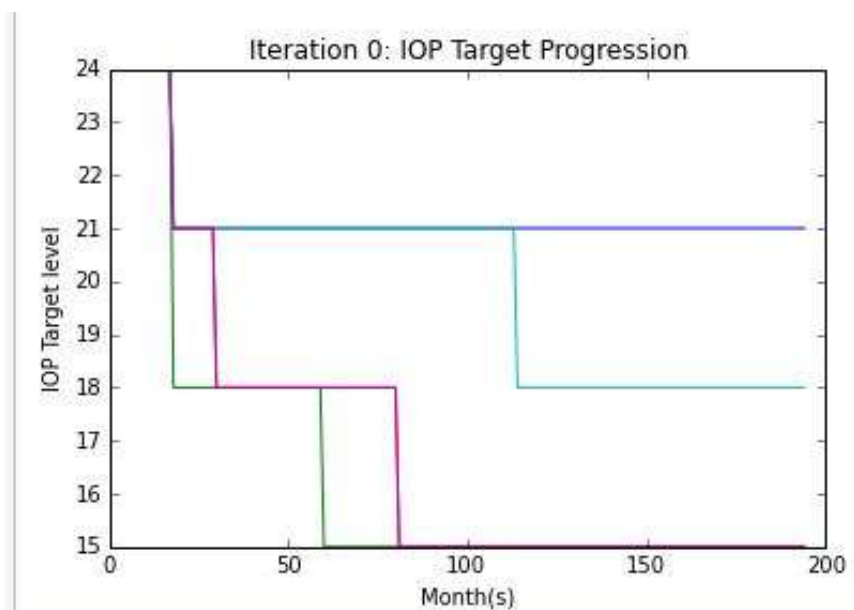
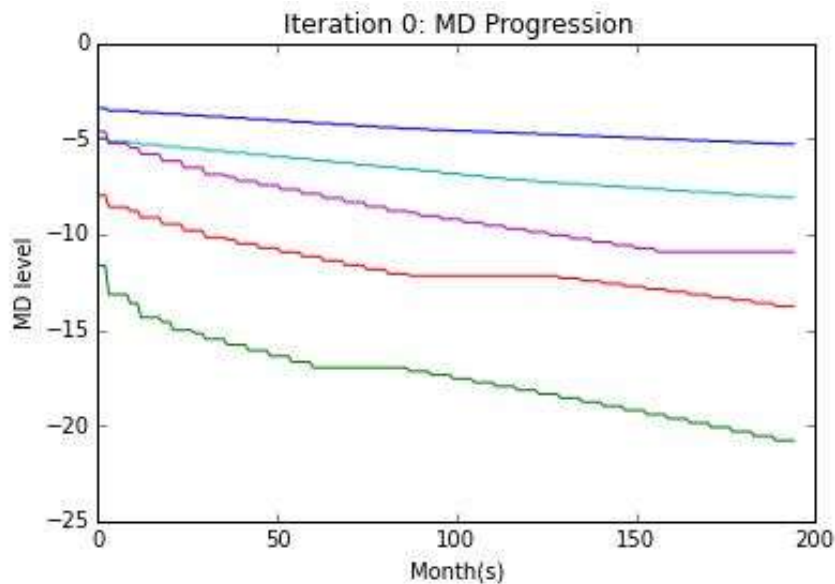
1. IOP Progression and IOP Target



New medication/treatment will be indicated once the IOP > IOP Target

2. When IOP Target will change

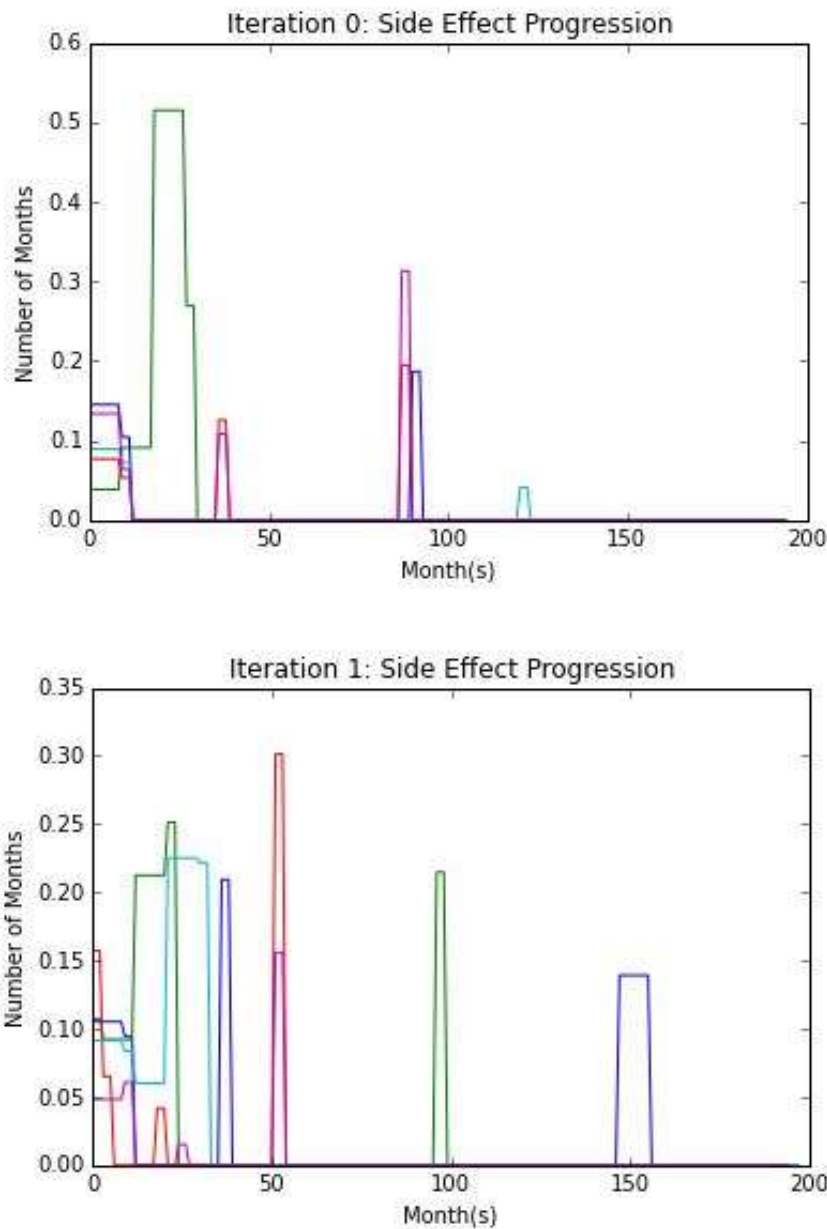
This depends on the Progression (MD decrease)



3. When IOP will increase

```
def onNoMedicationOrTrabeculectomy(self):
    self.params['SideEffect'] = 0
    #IOP is supposed to increase 0.5% annually, without medication
    if self.medicalRecords['OnTrabeculectomy'] == True or self.medicalRecords['OnImplant'] == True:
        self.Attribute['IOP'] = self.Attribute['IOP'] * (1 + (1.5/100)*(self.params['time_next_visit']/12))
        self.medicalRecords['MedicationIntake'] += 1
    else:
        self.Attribute['IOP'] = self.Attribute['IOP'] * (1 + (0.5/100)*(self.params['time_next_visit']/12))
```

4. Side Effect Progression



Side Effect is nullified if:

I. Patients are on Trabeculectomy or Implant

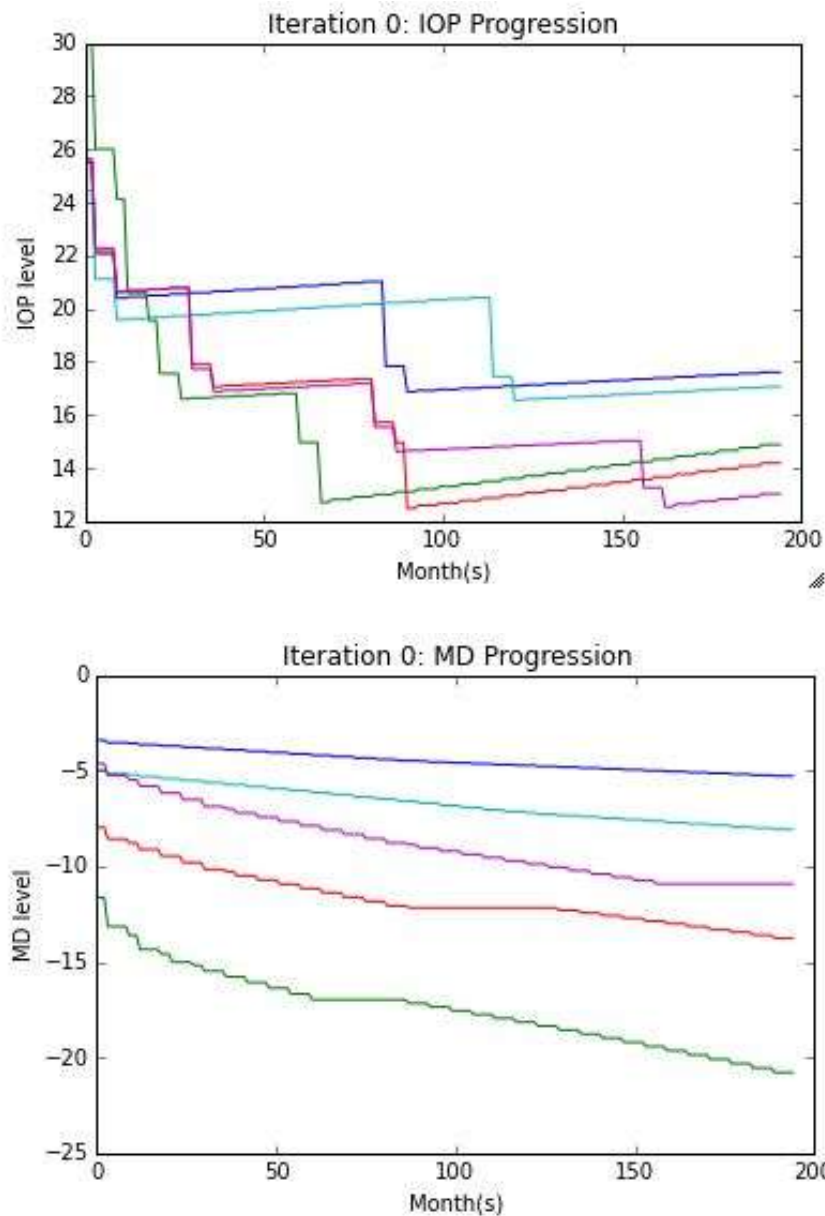
II. Patients' current medication amount is > 10

```
def IOPandSideEffectEvaluation(self):
    if self.medicalRecords['MedicationIntake'] > 10 :
        self.params['SideEffect'] = 0
    if self.PatientAttribute['IOP'] > self.PatientAttribute['IOPTarget']:
        self.medicalRecords['TreatmentOverallStatus'] = 2
        self.medicalRecords['ContinueTreatment'] = True
    else:
        self.medicalRecords['ContinueTreatment'] = False
```

III. $IOP < IOP \text{ Target}$

```
def onNoMedicationOrTrabeculectomy(self):
    self.params['SideEffect'] = 0
```

5. Effect of IOP on MD



You would notice that there are some flat areas in the MD curves.

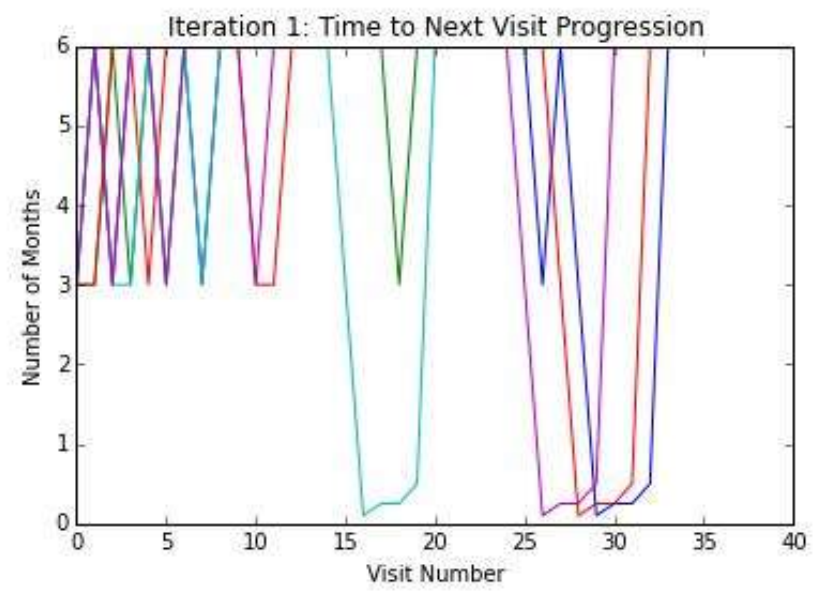
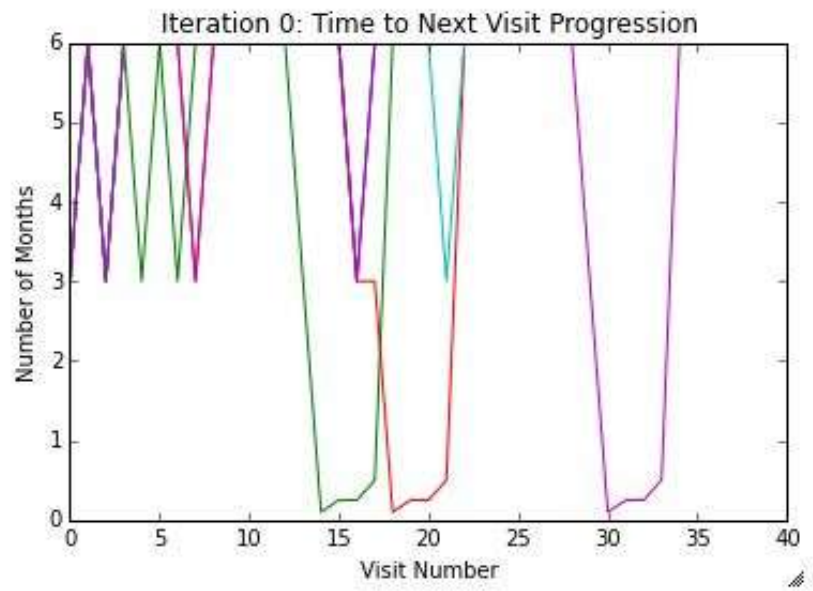
This is because of this behaviour in the system

```
if self.Attribute['IOP'] > 13:
    difference = self.Attribute['MDR'] * (1.13 ** (self.Attribute['IOP'] - 15.5)) * (self.params['time_next_visit'])
else:
    difference = 0
```

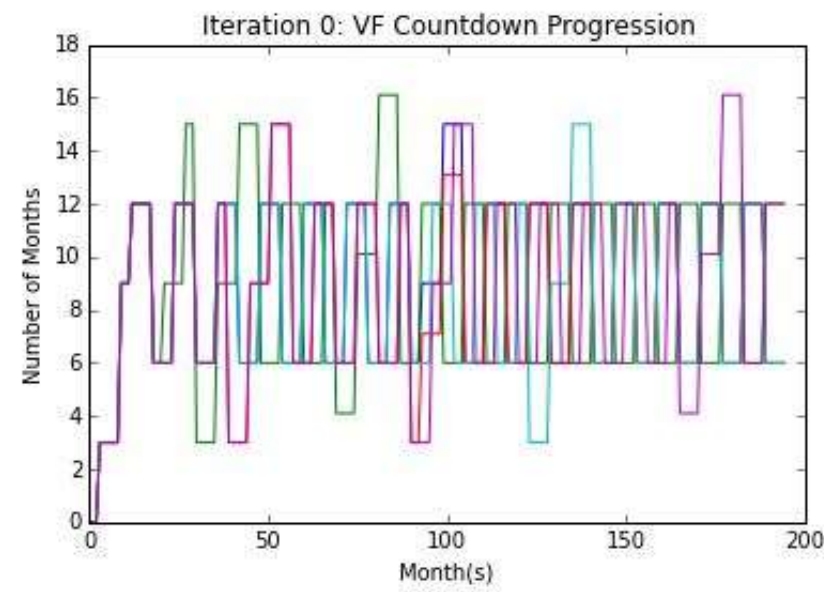
This is actually a very peculiar feature of the simulation system because if IOP = [13.15) then MD free falls!

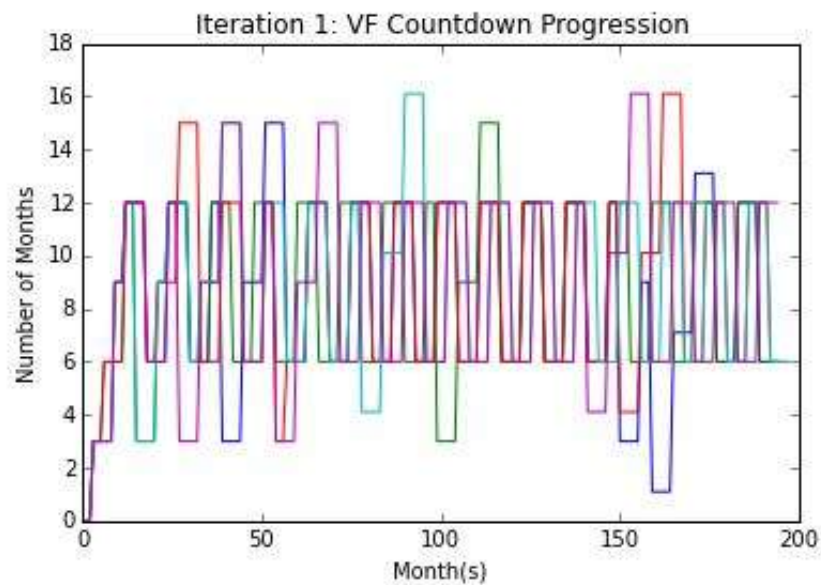
And the system has no way to correct this. The lowest IOP target that warrants change in treatment is 15

6. Time to Next Visit



7. VF Countdown





8. How the VFQ is calculated

8. Other Misc things

Patients > 85 will not be indicated Surgeries (Trabeculectomy and Implant)

Need a baseline for Cataract formation... Not really important for now but we might need it later