SAYANTAN RAHA

## Roll # : BAI09056

### IIMB - BAI09 - Assignment 4

Out[1]: Toggle on/off Code

# WRITE ALL EQUATIONS AND Provide Explanations

# Q-1-1

- Calculating the Aggregate TPM

The **MLE** estimate of the probability $P_{ij}$ (probability of moving from state i to satge j) in one Step is given by:

$$\hat{P}_{ij} = \frac{N_{ij}}{\sum_{k=1}^{m} N_{ik}}$$

where $N_{ij}$ is the number of cases in which $X_n$ (state in time n is i) and $X_{n+1} = j$ (state in time n+1 is j). For aggreagte Matrix we will consider all $N_{ij}$ from all the monthly data

Out[13]:

| | Lost | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|---|
| **Lost** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **Stage A** | 0.208707 | 0.510885 | 0.280408 | 0.000000 | 0.000000 |
| **Stage B** | 0.144502 | 0.000000 | 0.380024 | 0.475475 | 0.000000 |
| **Stage C** | 0.431441 | 0.000000 | 0.000000 | 0.286625 | 0.281934 |
| **Won** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

- Month 1 TPM

Out[14]:

| | Lost | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|---|
| **Lost** | 1.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 |
| **Stage A** | 0.199601 | 0.5 | 0.300399 | 0.000000 | 0.000000 |
| **Stage B** | 0.147739 | 0.0 | 0.398492 | 0.453769 | 0.000000 |
| **Stage C** | 0.457415 | 0.0 | 0.000000 | 0.299098 | 0.243487 |
| **Won** | 0.000000 | 0.0 | 0.000000 | 0.000000 | 1.000000 |

- Month 2 TPM

Out[15]:

| | Lost | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|---|
| **Lost** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **Stage A** | 0.211665 | 0.520226 | 0.268109 | 0.000000 | 0.000000 |
| **Stage B** | 0.135371 | 0.000000 | 0.392043 | 0.472586 | 0.000000 |
| **Stage C** | 0.420976 | 0.000000 | 0.000000 | 0.290478 | 0.288545 |
| **Won** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

- Month 3 TPM

Out[16]:

| | Lost | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|---|
| **Lost** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.00000 |
| **Stage A** | 0.213612 | 0.511685 | 0.274703 | 0.000000 | 0.00000 |
| **Stage B** | 0.152256 | 0.000000 | 0.341479 | 0.506266 | 0.00000 |
| **Stage C** | 0.415618 | 0.000000 | 0.000000 | 0.269392 | 0.31499 |
| **Won** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.00000 |

# Q-1-2

In order to show the Aggregate Data Follows Markov chain we will do the following tests

- The time homogeneity of transition matrix using **Likelihood Ratio test**
- Perform the **Anderson Goodman Test** on monthly TPMs and Aggregated TPM for time Dependence test.

### Anderson Goodman Test

$H_O$: The sequence of transitions are independant

$H_A$: The sequence of transitions are dependant

The corresponding Test Statistic is:

$$\chi^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where

- $O_{ij}$ = Observed number of transitions from state i to state j
- $E_{ij}$ = Expected number of transitions from state i to state j

Alpha / Significance = 0.05

```
Aggregate TPM- Test
Null Hypothesis :: ho = Distributions are independent
Alternate Hypothesis :: ha = Distributions are dependent


Chisq 12987.24859805583 is more than critical Chisq value 26.29622760486423 for significance alpha 0.05 and df 16
Corresponding p-value 0.0 is below alpha / significance 0.05 for df 16
Hence we reject the NULL Hypothesis, the distributions are dependent

Month 1 - Test
Null Hypothesis :: ho = Distributions are independent
Alternate Hypothesis :: ha = Distributions are dependent


Chisq 13648.629894861813 is more than critical Chisq value 26.29622760486423 for significance alpha 0.05 and df 16
Corresponding p-value 0.0 is below alpha / significance 0.05 for df 16
Hence we reject the NULL Hypothesis, the distributions are dependent

Month 2 - Test
Null Hypothesis :: ho = Distributions are independent
Alternate Hypothesis :: ha = Distributions are dependent


Chisq 12764.088812408034 is more than critical Chisq value 26.29622760486423 for significance alpha 0.05 and df 16
Corresponding p-value 0.0 is below alpha / significance 0.05 for df 16
Hence we reject the NULL Hypothesis, the distributions are dependent

Month 3 - Test
Null Hypothesis :: ho = Distributions are independent
Alternate Hypothesis :: ha = Distributions are dependent


Chisq 12576.629383030904 is more than critical Chisq value 26.29622760486423 for significance alpha 0.05 and df 16
Corresponding p-value 0.0 is below alpha / significance 0.05 for df 16
Hence we reject the NULL Hypothesis, the distributions are dependent
```

### Likelihood Ratio test

$H_O$: $P_{ij}(t) = P_{ij}$ for t = 1,2,3

$H_A$: $P_{ij}(t) \neq P_{ij}$ for t = 1,2,3

where $P_{ij}(t)$ is the estimated transition probability

The corresponding Test Statistic is:

$$\lambda = \prod_{t} \prod_{ij} \frac{\hat{P}_{ij}}{\hat{P}_{ij}(t)}$$

which is equivalent to

$$\chi^2 = \sum_{t} \sum_{i} \sum_{j} \frac{n_i(t)[\hat{P}_{ij}(t) - \hat{P}_{ij}]^2}{\hat{P}_{ij}}$$

where $n_i(t)$ is the number of customers in state i at time t.

Alpha / Significance = 0.05

```
Time Homogeneity Test / Likelihood Ratio Test - for Aggreagted TPM
Null Hypothesis :: ho = PTMx elements == PTMAggregate elements
Alternate Hypothesis :: ha = PTMx elements != PTMAggregate elements


Chisq 48.381890967159826 is below critical Chisq value 55.75847927888702 for significance alpha 0.05 and df 40
Corresponding p-value 0.17053157682892028 is more than alpha / significance 0.05
Hence we retain the NULL Hypothesis
```

From the test results above we can conclude that the process is a **First Order Markov Chain**

## Q-1-3

- we will be using the following Aggregate TPM for answering the question. It is derived from the aggregate TPM of counts / frequencies provided in the excel.

Out[78]:

| | Lost | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|---|
| **Lost** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **Stage A** | 0.208707 | 0.510885 | 0.280408 | 0.000000 | 0.000000 |
| **Stage B** | 0.144502 | 0.000000 | 0.380024 | 0.475475 | 0.000000 |
| **Stage C** | 0.431441 | 0.000000 | 0.000000 | 0.286625 | 0.281934 |
| **Won** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

- Since the TPM has two Absorbing states we cannot represent this matrix into its Canonical Form and find out F & R and find time to absorption, as those absorption number are absorption to either of the two Absorbing States.
- We will remove the unwanted Absorbing state from the Matrix and recompute the TPM (normlize the values wrt to omittied State and use that TMP to fnd out the time to absorption from State B to State Won
- The corresponding new TPM is:

Out[87]:

| | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|
| **Stage A** | 0.645633 | 0.354367 | 0.000000 | 0.000000 |
| **Stage B** | 0.000000 | 0.444214 | 0.555787 | 0.000000 |
| **Stage C** | 0.000000 | 0.000000 | 0.504125 | 0.495875 |
| **Won** | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

- To solve the problem we will represent the Matrix in its canonical form: P= $\begin{bmatrix} I & O \\ R & Q \end{bmatrix}$

  where:
  - I = Identity Matrix
  - O = Zero matrix
  - R = Probability of absoption from a transient state to absorbing state
  - Q = Probability of transition between transient states

To calculate the eventual probability of absorption we will compute the Limiting probability / long running probability. When we multiply **P** we get the following form of matrix:

$$P^n = \begin{bmatrix} I & O \\ \sum_{k=0}^{n-1}(Q^k)R & Q^n \end{bmatrix}$$

as $n \to \infty$ $\sum_{k=0}^{n-1}(Q^k)$ = F = $(I - Q)^{-1}$

- F = Fundamental Matrix
- Expected time to absorption = Fc, where c = unit vector

Solving the problem we get:

```
F Matrix::
[[2.0445098  0.92470628 0.61632968]
 [0.         1.61296487 1.07506365]
 [0.         0.         1.40178691]]

Time to Churn: Fc =
```

Out[103]:

```
array([[3.58554576],
       [2.68802852],
       [1.40178691]])
```

- We are getting approximately on an average 3 Months for an Opportunity is **Stage B** to get converted to into **Contract Signing**

# Q-1-4

- Revenue can only be realised once a Opportunity reaches the Won state
- Multiplying TPM with the Initial distribution will be used to find the revenue

Out[147]:

| | Lost | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|---|
| **Lost** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **Stage A** | 0.208707 | 0.510885 | 0.280408 | 0.000000 | 0.000000 |
| **Stage B** | 0.144502 | 0.000000 | 0.380024 | 0.475475 | 0.000000 |
| **Stage C** | 0.431441 | 0.000000 | 0.000000 | 0.286625 | 0.281934 |
| **Won** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

From TPM above we can see that only .281934 or 28% of Stage 3 Revenue has probability of being converted after one month. Hence: Revenue = 0.281934 * 1.6

```
Expected Revenue after a Month 0.45109440000000006 Billion
```

Out[152]:

| | Lost | Stage A | Stage B | Stage C | Won |
|---|---|---|---|---|---|
| **Lost** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **Stage A** | 0.355852 | 0.261004 | 0.249818 | 0.133327 | 0.000000 |
| **Stage B** | 0.404555 | 0.000000 | 0.144418 | 0.316974 | 0.134052 |
| **Stage C** | 0.555103 | 0.000000 | 0.000000 | 0.082154 | 0.362743 |
| **Won** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

For 2 Months the new TPM is given by $P^2$.

From TPM above we can see that 0.134052 or 13% of Stage B and 0.362743 or 36% of Stage 3 Revenue has probability of being converted after two months. Hence: Revenue =0.1340523 * 1.8 + 0.3627429*1.6

```
Expected Revenue after two Months (end of March) 0.8216827800000001 Billion
```

# Q-2-1

- Possible States = (NN, YN, NY, YY)
- NN = No Accidents is past 2 years
- YN = No Accident in last year, but had accident in prior year
- NY = Accident in last year, but had no accident in prior year
- YY = Accident in last year, and had accident in prior year

$P_I = Initial State = [1, 0, 0, 0]$

$Cost = [-200, 300, 600, 1000]$

The corresponding TPM is given by:

Out[154]:

| | NN | YN | NY | YY |
|---|---|---|---|---|
| **NN** | 0.9 | 0.0 | 0.1 | 0.0 |
| **YN** | 0.9 | 0.0 | 0.1 | 0.0 |
| **NY** | 0.0 | 0.9 | 0.0 | 0.1 |
| **YY** | 0.0 | 0.9 | 0.0 | 0.1 |

# Q-2-2

State after n periods is given by $P_I x P^n$ where:

- $P_I = Initial Distribution$
- P = Transition Probability Matrix
- n = Number of periods

Out[12]:
```
(array([[0.81, 0.09, 0.09, 0.01],
        [0.81, 0.09, 0.09, 0.01],
        [0.81, 0.09, 0.09, 0.01],
        [0.81, 0.09, 0.09, 0.01]]), -70.99999999999999)
```

His premium will possibly reduce by INR 70.99

# Q-3

- To solve the problem we will represent the TPM Matrix in its canonical form: P= $\begin{bmatrix} I & O \\ R & Q \end{bmatrix}$

  where:
  - I = Identity Matrix
  - O = Zero matrix
  - R = Probability of absoption from a transient state to an absorbing state
  - Q = Probability of transition between transient states

To calculate the eventual probability of absorption we will compute the Limiting probability / long running probability. When we take dot product of **P** n times with itself we get the following form of matrix:

$P^n$ = $\begin{bmatrix} I & O \\ \sum_{k=0}^{n-1}(Q^k)R & Q^n \end{bmatrix}$

as $n \to \infty$ $\sum_{k=0}^{n-1}(Q^k)$ = F = $(I - Q)^{-1}$

- F = Fundamental Matrix
- Expected time to absorption = Fc, where c = unit vector

Hence Fc is the matrix we will derive from the provided data points.

Solving the problem we get:

Out[162]:

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **1** | 1.00 | 0.00 | 0.0 | 0.00 | 0.00 | 0.00 |
| **2** | 0.00 | 1.00 | 0.0 | 0.00 | 0.00 | 0.00 |
| **3** | 0.05 | 0.05 | 0.9 | 0.00 | 0.00 | 0.00 |
| **4** | 0.10 | 0.05 | 0.0 | 0.80 | 0.05 | 0.00 |
| **5** | 0.20 | 0.10 | 0.0 | 0.05 | 0.60 | 0.05 |
| **6** | 0.10 | 0.20 | 0.0 | 0.00 | 0.00 | 0.70 |

```
Fundamental Matrix ::
```

Out[168]:

```
array([[10.        ,  0.        ,  0.        , -0.        ],
       [ 0.        ,  5.16129032,  0.64516129,  0.10752688],
       [ 0.        ,  0.64516129,  2.58064516,  0.43010753],
       [ 0.        ,  0.        ,  0.        ,  3.33333333]])
```

```
R Matrix ::
```

Out[169]:

```
array([[0.05, 0.05],
       [0.1 , 0.05],
       [0.2 , 0.1 ],
       [0.1 , 0.2 ]])
```

```
Q Matrix ::
```

Out[170]:

```
array([[0.9 , 0.  , 0.  , 0.  ],
       [0.  , 0.8 , 0.05, 0.  ],
       [0.  , 0.05, 0.6 , 0.05],
       [0.  , 0.  , 0.  , 0.7 ]])
```

```
Time to Churn = Fc ::
```

Out[174]:

| | Time2Absorption |
|---|---|
| **3** | 10.000000 |
| **4** | 5.913978 |
| **5** | 3.655914 |
| **6** | 3.333333 |

# Q-3.1

- From the calculations above we can see that time to churn is highest from State 3

```
Probability of Absorption to Absorbing State = FR ::
```

Out[177]:

| | 1 | 2 |
|---|---|---|
| **3** | 0.500000 | 0.500000 |
| **4** | 0.655914 | 0.344086 |
| **5** | 0.623656 | 0.376344 |
| **6** | 0.333333 | 0.666667 |

# Q-3.2

- From the FR Matrix above we can see that from State 6 the eventual absorption to State 2 is .666 or 67% approx

# Q-3.3

The CLV for N periods is given by (Pfeifer and Carraway):

CLV = $\sum_{t=0}^{N} \frac{P_I x P^t R}{(i+i)t}$

where

- i = Interest rate
- d = 1/(1+i) = Discount factor = .99

Initial Distribution : $P_I$ = [0, 0, 0, 0, 0, 1]

Margin = [0, 200, 300, 400, 600, 800]

```
If Expected Duration is assumed to be 3 (Actual 3.33) then CLV :: 2196.0942379999997
If Expected Duration is assumed to be 4 (Actual 3.33) then CLV :: 2477.9331073339995
```

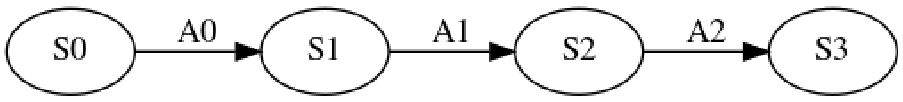
# Q-4.1

We will use Markov Decision Process to solve the following problem.

- Discount factor is 0.95
- State Space = [1,2,3,4]
- Action set = [1,2,3]
- TPMs for State-Action are provided in the Question (For saving space I will not be displaying the same in Answers)

IN MDP we have an Initial State (S0) in the diagram below and we take Action (A0), then S1 is the State in the next time period and this continues. Such a sequence is shown for 4 Stages in the diagram below.

Out[200]:




The Objective is to **Maximise** the expected return obtained over a period of return. Based on State and Reward the Reward generated can be represented as:

$R(S_0, a_0) + \beta R(S_1, a_1) + \beta^2 R(S_2, a_2) + \beta^3 R(S_3, a_3) + ...$

where :

- $R(S_0, a_0)$ = Reward generated from Initial Stage where initial State = $S_0$ and action taken = $a_0$. The rewards obtained from future states is discounted by factor $\beta$

Objective :: $Maximise_{a_i \epsilon A}[R(S_0, a_0) + \beta R(S_1, a_1) + \beta^2 R(S_2, a_2) + \beta^3 R(S_3, a_3) + ...]$

We will use **POLICY ITEARTION ALGORITHM** to obtain the long term revenue generated from policy (2,2,1,3)

The **Value Function** for a Policy $\Pi$ starting at State $S_i$ is given by:

$V^{\Pi}(i) = R^{\Pi}(i) + \beta \sum_{j \epsilon S} P_{ij}^{\Pi} x V^{\Pi}(j)$

where:

- $R^{\Pi}(i)$ :: Immediate Reward
- $\beta \sum_{j \epsilon S} P_{ij}^{\Pi} x V^{\Pi}(j)$ :: Discounted Future Reward

We will develop a System of Linear Equation by using the above equation for each State and Action provided in the Policy.

**Equations**:

- 0.62 v1 - 0.285 v2 - 0.1425 v3 - 0.1425 v4 = 160
- 0. v1 + 0.2875 v2 - 0.1 v3 - 0.1425 v4 = 200
- 0. v1 - 0.095 v2 - 0.24 v3 - 0.095 v4 = 270
- 0. v1 + 0. v2 - 0.285 v3 - 0.335 v4 = 500

The Matrix of the coefficients of the system of Equations are :

Out[391]:
```
array([[ 0.62  , -0.285 , -0.1425, -0.1425],
       [ 0.    ,  0.2875, -0.1   , -0.1425],
       [ 0.    , -0.095 ,  0.24  , -0.095 ],
       [ 0.    ,  0.    , -0.285 ,  0.335 ]])
```

The Immediate Reward Matrix for the State Actions provided:

Out[215]:
```
array([[160],
       [200],
       [270],
       [500]])
```

Policy to evaluate = (2, 2, 1, 3)

Solving the Above set of **Linear Equations** we get the following Values for the Policy:

Out[216]:

| | Policy Value |
|---|---|
| **V12** | 6222.414034 |
| **V22** | 6335.801092 |
| **V31** | 6368.248847 |
| **V43** | 6910.301258 |

Where $V_{ij}$ = Policy Value when Initial State was i and action taken was j The Overall Policy Value is :::

Out[217]:
```
25836.765229462104
```

## Q-4.2

To Check if Policy (2, 2, 1, 2) is better than the previous Policy (2, 2, 1, 3) we will use the **Policy Evaluation Step**. The objectibe is to check if the value function obtained in Q-4-1 is less thena the new Policy.

The Policy improvement evaluation step includes the following:

$T^{\Pi^{new}}(i) = Max_{a_{i,new}}[R(i, a_{i,new}) + \beta \sum_{j \epsilon S} P(j|i, a_{i,new}) x V^{\Pi}(j)]$

where :

- $i \epsilon S$
- $a_{i,new}$ = A new action chosen for state i
- $T^{\Pi^{new}}(i)$ = Value function when the current policy for state i is changed to $a_{i,new}$
- R(i, a_{i,new}) = Reward when action for state i is replaced with new action $a_{i,new}$

Solving the new equation $T^{\Pi^{new}}(4)$ =

Out[218]: 6930.921658077932

$T^{\Pi^{new}}(4) > V^{\Pi}(4)$ value. Hence this is a better Policy.

We will resolve the problem for the new Policy. The Matrix of the coefficients of the system of New Equations are :

Out[219]:
```
array([[ 0.62  , -0.285 , -0.1425, -0.1425],
       [ 0.    ,  0.2875, -0.1   , -0.1425],
       [ 0.    , -0.095 ,  0.24  , -0.095 ],
       [ 0.    ,  0.    , -0.095 ,  0.145 ]])
```

The Immediate Reward Matrix for the New State Actions provided:

Out[220]:
```
array([[160],
       [200],
       [270],
       [400]])
```

Solving the Above set of **Linear Equations** we get the following Values for the New Policy:

Out[222]:

| | Policy Value |
|---|---|
| **V12** | 6249.595436 |
| **V22** | 6363.327540 |
| **V31** | 6393.980092 |
| **V42** | 6947.780060 |

Where $V_{ij}$ = Policy Value when Initial State was i and action taken was j The Overall Policy Value is :::

Out[223]: array([25954.68312784])

## Q-4.3

The Optimal Policy can be obtained by solving a sustem of Linear Equations:

**Decision Variables**: x_si, i = 1,2,3,4 are the Best Policy Value function for Policy

**Minimize Objective** : x_s1 + x_s2 + x_s3 + x_s4

**Subject To**:

**Constraint_for_State_S1_Policy_1**: 0.525 x_s1 - 0.2375 x_s2 - 0.1425 x_s3 - 0.095 x_s4 >= 180

**Constraint_for_State_S1_Policy_2**: 0.62 x_s1 - 0.285 x_s2 - 0.1425 x_s3 - 0.1425 x_s4 >= 160

**Constraint_for_State_S1_Policy_3**: 0.43 x_s1 - 0.19 x_s2 - 0.095 x_s3 - 0.095 x_s4 >= 200

**Constraint_for_State_S2_Policy_1**: 0.2875 x_s2 - 0.1425 x_s3 - 0.095 x_s4 >= 225

**Constraint_for_State_S2_Policy_2**: 0.2875 x_s2 - 0.095 x_s3 - 0.1425 x_s4 >= 200

**Constraint_for_State_S2_Policy_3**: - 0.095 x_s1 + 0.335 x_s2 - 0.095 x_s3 - 0.095 x_s4 >= 250

**Constraint_for_State_S3_Policy_1**: - 0.095 x_s2 + 0.24 x_s3 - 0.095 x_s4 >= 270

**Constraint_for_State_S3_Policy_2**: - 0.0475 x_s2 + 0.24 x_s3 - 0.1425 x_s4 >= 240

**Constraint_for_State_S3_Policy_3**: - 0.19 x_s2 + 0.335 x_s3 - 0.095 x_s4 >= 300

**Constraint_for_State_S4_Policy_1**: - 0.19 x_s3 + 0.24 x_s4 >= 450

**Constraint_for_State_S4_Policy_2**: - 0.095 x_s3 + 0.145 x_s4 >= 400

**Constraint_for_State_S4_Policy_3**: - 0.285 x_s3 + 0.335 x_s4 >= 500

**Non Zero Constraint** :All Decision Variable (x_si >0)

```
 s1 :: 6285.43 ::
 s2 :: 6358.91 ::
 s3 :: 6491.42 ::
 s4 :: 7015.09 ::
Objective 26150.850000000002
```

```
# %load MDPPolicy.sen
GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   1

Problem:
Objective:  Objective = 26150.85187 (MINimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 Constraint_for_State_S1_Policy_1
                    BS     198.14905     -18.14905     180.00000           +Inf      -6.28153   24906.17364
Constraint_for_State_S1_Policy_2
                                                 .          +Inf      198.14905          +Inf          +Inf
```

```
     2 Constraint_for_State_S1_Policy_2
                    NL     160.00000          .        160.00000       143.88421      -5.34746    26064.67331 Constraint_for_State_S1_Policy_3
                                           5.34746          +Inf            +Inf          +Inf           +Inf

     3 Constraint_for_State_S1_Policy_3
                    BS     211.42385     -11.42385     200.00000            +Inf      -7.54374    24555.92470 Constraint_for_State_S1_Policy_2
                                              .             +Inf       211.42385          +Inf           +Inf

     4 Constraint_for_State_S2_Policy_1
                    BS     236.72520     -11.72520     225.00000            +Inf     -26.21858    19944.25242 Constraint_for_State_S2_Policy_3
                                              .             +Inf       195.32355     338.22724   106217.76397 Constraint_for_State_S3_Policy_3

     5 Constraint_for_State_S2_Policy_2
                    BS     211.85095     -11.85095     200.00000            +Inf     -25.11468    20830.28398 Constraint_for_State_S2_Policy_3
                                              .             +Inf       197.31463     218.95940    72537.61038 Constraint_for_State_S4_Policy_3

     6 Constraint_for_State_S2_Policy_3
                    NL     250.00000          .        250.00000       246.92242     -24.37290    26075.84219 Constraint_for_State_S4_Policy_1
                                          24.37290          +Inf            +Inf          +Inf           +Inf

     7 Constraint_for_State_S3_Policy_1
                    BS     287.41130     -17.41130     270.00000       392.08067     -37.88866    15261.22234 Constraint_for_State_S3_Policy_3
                                              .             +Inf       287.41130          +Inf           +Inf

     8 Constraint_for_State_S3_Policy_2
                    BS     256.24270     -16.24270     240.00000       354.82678     -40.22741    15842.87140 Constraint_for_State_S3_Policy_3
                                              .             +Inf       243.00060     195.95670    76363.32403 Constraint_for_State_S4_Policy_3

     9 Constraint_for_State_S3_Policy_3
                    NL     300.00000          .        300.00000       297.99286     -29.68893    26091.26207 Constraint_for_State_S4_Policy_1
                                          29.68893          +Inf       433.57783          +Inf    30116.63439 Constraint_for_State_S2_Policy_1

    10 Constraint_for_State_S4_Policy_1
                    BS     450.25150       -.25150     450.00000       548.60557     -26.38302    14271.85677 Constraint_for_State_S4_Policy_3
                                              .             +Inf       450.25150          +Inf           +Inf

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   2

Problem:
Objective:  Objective = 26150.85187 (MINimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
    11 Constraint_for_State_S4_Policy_2
                    BS     400.50301       -.50301     400.00000       471.18937     -36.70973    11448.49379 Constraint_for_State_S4_Policy_3
                                              .             +Inf       400.50301          +Inf           +Inf

    12 Constraint_for_State_S4_Policy_3
                    NL     500.00000          .        500.00000       499.67775     -20.59071    26144.21646 Constraint_for_State_S4_Policy_1
                                          20.59071          +Inf       626.02178          +Inf    28745.72946 Constraint_for_State_S2_Policy_2

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   3

Problem:
Objective:  Objective = 26150.85187 (MINimum)

   No. Column name  St      Activity      Obj coef   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 x_s1         BS    6285.43074       1.00000          .             +Inf      -1.11226    12874.36722 Constraint_for_State_S1_Policy_2
                                              .             +Inf      6285.43074          +Inf           +Inf

     2 x_s2         BS    6358.90913       1.00000          .             +Inf      -2.16109     6049.78880 Constraint_for_State_S2_Policy_3
                                              .             +Inf      6358.90913          +Inf           +Inf

     3 x_s3         BS    6491.42180       1.00000          .        7671.88260      -2.35952     4342.78573 Constraint_for_State_S3_Policy_3
                                              .             +Inf      6491.42180          +Inf           +Inf

     4 x_s4         BS    7015.09019       1.00000          .        7875.46343      -2.01599     4993.40589 Constraint_for_State_S4_Policy_3
                                              .             +Inf      7015.09019          +Inf           +Inf

End of report
```

Optimal Policy(2,3,3,3)

Optimal Policy is obtained by checking the Sensitivity Report. Binding Constraints provide the best Policy Values

## Q-4.4

The number of time steps in a MDP defines the number of Stages in the process. Since here the number of Time periods is 4, hence we have 4 Stages in this Problem.

The Dynamic Programming recursive equation for the Value Iteration Algorithm is given by:

$$V^*_t(i) = MAX_{a\epsilon A}\left[R(i,a_i) + \beta\Sigma_{k=S_i}^{S_n} P_{ij}(a_i)V^*_{t+1}(j)\right]$$

where $V^*_t(i)$ is the Optimal value for a policy when current period is t and the current State is $S_i$

We also assume if the duration of the planning horizon is n, $V^*_{n+1}(i) = 0$ for all states $S_i$

Variable notations that will be used to solve the problem:

$$V_{-t-i-j}$$
$$\text{Where t} = 1..4 \text{ defines the time periods or Stages of MDP}$$
$$\text{and i} = 1..4 \text{ defines the 4 States of MDP}$$
$$\text{and j} = 1..3 \text{ defines the 3 Actions of MDP}$$

- e.g. Variable V_1_1_1 implies Value of Policy for Stage 1, State 1 and Action Taken is 1
- e.g. Variable Vo_1_1_1 implies Optimal Value of Policy for Stage 1, State 1 and Action Taken is 1

Following are the sets of Dynamic Programming Equations and their solutions:

```
V511 = 0
V512 = 0
V513 = 0
V521 = 0
V522 = 0
V523 = 0
V531 = 0
V532 = 0
V533 = 0
V541 = 0
V542 = 0
V543 = 0
```

```
V411 = 200 * .9 + 0
V412 = 200 * .8 + 0
V413 = 200 * 1 + 0
V421 = 250 * .9 + 0
V422 = 250 * .8 + 0
V423 = 250 * 1 + 0
V431 = 300 * .9 + 0
V432 = 300 * .8 + 0
V433 = 300 * 1 + 0
V441 = 500 * .9 + 0
V442 = 500 * .8 + 0
V443 = 500 * 1 + 0
```

```
Vmax for t=4, State = 1 :: 200
Policy :: 3

Vmax for t=4, State = 2 :: 250
Policy :: 3

Vmax for t=4, State = 3 :: 300
Policy :: 3

Vmax for t=4, State = 4 :: 500
Policy :: 3
```

```
V311 = 200 * .9 + 0.95 * (.5*V41M + .25*V42M + .15 *V43M + .1 * V44M)
V312 = 200 * .8 + 0.95 * (.4*V41M + .3*V42M + .15 *V43M + .15 * V44M)
V313 = 200 * 1 + 0.95 * (.6*V41M + .2*V42M + .1 *V43M + .1 * V44M)
```

```
V311 :: 424.625 V312 :: 421.25 V313 :: 437.5
Vmax for t=3, State = 1 :: 437.5
Policy :: 3
```

```
V321 = 250 * .9 + 0.95 * (.0*V41M + .75*V42M + .15 *V43M + .1 * V44M)
V322 = 250 * .8 + 0.95 * (.0*V41M + .75*V42M + .1 *V43M + .15 * V44M)
V323 = 250 * 1 + 0.95 * (.1*V41M + .7*V42M + .1 *V43M + .1 * V44M)
```

```
V321 :: 493.375 V322 :: 477.875 V323 :: 511.25
Vmax for t=3, State = 2 :: 511.25
Policy :: 3
```

```
V331 = 300 * .9 + 0.95 * (.0*V41M + .1*V42M + .8 *V43M + .1 * V44M)
V332 = 300 * .8 + 0.95 * (.0*V41M + .05*V42M + .8 *V43M + .15 * V44M)
V333 = 300 * 1 + 0.95 * (.0*V41M + .2*V42M + .7 *V43M + .1 * V44M)
```

```
V331 :: 569.25 V332 :: 551.125 V333 :: 594.5
Vmax for t=3, State = 3 :: 594.5
Policy :: 3
```

```
V341 = 500 * .9 + 0.95 * (.0*V41M + .0*V42M + .2 *V43M + .8 * V44M)
V342 = 500 * .8 + 0.95 * (.0*V41M + .0*V42M + .1 *V43M + .9 * V44M)
V343 = 500 * 1 + 0.95 * (.0*V41M + .0*V42M + .3 *V43M + .7 * V44M)
```

```
V341 :: 887.0 V342 :: 856.0 V343 :: 918.0
Vmax for t=3, State = 4 :: 918.0
Policy :: 3
```

```
V211 = 200 * .9 + 0.95 * (.5*V31M + .25*V32M + .15 *V33M + .1 * V34M)
V212 = 200 * .8 + 0.95 * (.4*V31M + .3*V32M + .15 *V33M + .15 * V34M)
V213 = 200 * 1 + 0.95 * (.6*V31M + .2*V32M + .1 *V33M + .1 * V34M)
```

```
V211 :: 681.160625 V212 :: 687.4875 V213 :: 690.2
Vmax for t=2, State = 1 :: 690.2
Policy :: 3
```

```
V221 = 250 * .9 + 0.95 * (.0*V31M + .75*V32M + .15 *V33M + .1 * V34M)
V222 = 250 * .8 + 0.95 * (.0*V31M + .75*V32M + .1 *V33M + .15 * V34M)
V223 = 250 * 1 + 0.95 * (.1*V31M + .7*V32M + .1 *V33M + .1 * V34M)
```

```
V221 :: 761.191875 V222 :: 751.5581249999999 V223 :: 775.2312499999999
Vmax for t=2, State = 2 :: 775.2312499999999
Policy :: 3
```

```
V231 = 300 * .9 + 0.95 * (.0*V31M + .1*V32M + .8 *V33M + .1 * V34M)
V232 = 300 * .8 + 0.95 * (.0*V31M + .05*V32M + .8 *V33M + .15 * V34M)
V233 = 300 * 1 + 0.95 * (.0*V31M + .2*V32M + .7 *V33M + .1 * V34M)
```

```
V231 :: 857.5987500000001 V232 :: 846.919375 V233 :: 879.69
Vmax for t=2, State = 3 :: 879.69
Policy :: 3
```

```
V241 = 500 * .9 + 0.95 * (.0*V31M + .0*V32M + .2 *V33M + .8 * V34M)
V242 = 500 * .8 + 0.95 * (.0*V31M + .0*V32M + .1 *V33M + .9 * V34M)
V243 = 500 * 1 + 0.95 * (.0*V31M + .0*V32M + .3 *V33M + .7 * V34M)
```

```
V241 :: 1260.635 V242 :: 1241.3675 V243 :: 1279.9025
Vmax for t=2, State = 4 :: 1279.9025
Policy :: 3
```

```
V111 = 200 * .9 + 0.95 * (.5*V21M + .25*V22M + .15 *V23M + .1 * V24M)
V112 = 200 * .8 + 0.95 * (.4*V21M + .3*V22M + .15 *V23M + .15 * V24M)
V113 = 200 * 1 + 0.95 * (.6*V21M + .2*V22M + .1 *V23M + .1 * V24M)
```

```
V111 :: 938.9089843749999 V112 :: 950.9588375 V113 :: 945.869225
Vmax for t=1, State = 1 :: 950.9588375
Policy :: 2
```

```
V121 = 250 * .9 + 0.95 * (.0*V21M + .75*V22M + .15 *V23M + .1 * V24M)
V122 = 250 * .8 + 0.95 * (.0*V21M + .75*V22M + .1 *V23M + .15 * V24M)
V123 = 250 * 1 + 0.95 * (.1*V21M + .7*V22M + .1 *V23M + .1 * V24M)
```

```
V221 :: 1024.298828125 V222 :: 1018.308921875 V223 :: 1036.2590687499999
Vmax for t=1, State = 2 :: 1036.2590687499999
Policy :: 3
```

```
V131 = 300 * .9 + 0.95 * (.0*V21M + .1*V22M + .8 *V23M + .1 * V24M)
V132 = 300 * .8 + 0.95 * (.0*V21M + .05*V22M + .8 *V23M + .15 * V24M)
V133 = 300 * 1 + 0.95 * (.0*V21M + .2*V22M + .7 *V23M + .1 * V24M)
```

```
V131 :: 1133.8021062500002 V132 :: 1127.773990625 V133 :: 1153.878525
Vmax for t=1, State = 3 :: 1153.878525
Policy :: 3
```

```
V141 = 500 * .9 + 0.95 * (.0*V21M + .0*V22M + .2 *V23M + .8 * V24M)
V142 = 500 * .8 + 0.95 * (.0*V21M + .0*V22M + .1 *V23M + .9 * V24M)
V143 = 500 * 1 + 0.95 * (.0*V21M + .0*V22M + .3 *V23M + .7 * V24M)
```

```
V141 :: 1589.867 V142 :: 1577.8871874999998 V143 :: 1601.8468125
Vmax for t=1, State = 4 :: 1601.8468125
Policy :: 3
```

**Optimal Values are**:

Out[59]:

| | t=1 Optimal Action | t=2 Optimal Action | t=3 Optimal Action | t=4 Optimal Action |
|---|---|---|---|---|
| **State 1** | 2 | 3 | 3 | 3 |
| **State 2** | 3 | 3 | 3 | 3 |
| **State 3** | 3 | 3 | 3 | 3 |
| **State 4** | 3 | 3 | 3 | 3 |

# Q-5-1

Flipkart's Customer Analytics Churn problem deals with a dynamic problem namely the shift customer behavior / preference in the next period from the current period. The shift of preferences by an individual customer can be described as a sequence of certain states. Hence this is a stochastic process recorded over discrete time periods.

A Markovian stochastic process has the memory-less property, which means that the future state can be predicted from the knowledge of the present state.The **First Order Markov Chain** is given by: $P[X_{n+1}|X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = P[X_{n+1}|X_n = i_n]$

The customer behavior state space (defined by frequency of purchase) is discrete and the process is observed over a period of time. Moreover the current state is dependent only on the prior state. Hence the process satisfies the Markovian properties and can be the problem can be modelled as a **Discrete Time Markov Chain**.

Assumptions:

1. The current state is dependent only on the prior state
2. The Transition Probability Matrix time homogeneos

# Q-5-2

In this problem Churn is defined as a period of inactivity (not buying from Flipkart). The Period is defines as 13 Months. If Customer does not buy in 12 prior months he/she is considered as Churned. This is depicted by state 13.

# Q-5-3

- To solve the problem we will represent the TPM Matrix in its canonical form: P= $\begin{bmatrix} I & O \\ R & Q \end{bmatrix}$

  where:
  - I = Identity Matrix
  - O = Zero matrix
  - R = Probability of absoption from a transient state to an absorbing state
  - Q = Probability of transition between transient states

To calculate the eventual probability of absorption we will compute the Limiting probability / long running probability. When we take dot product of **P** n times with itself we get the following form of matrix:

$P^n = \begin{bmatrix} I & O \\ \sum_{k=0}^{n-1}(Q^k)R & Q^n \end{bmatrix}$

as $n \to \infty$ $\sum_{k=0}^{n-1}(Q^k)$ = F = $(I - Q)^{-1}$

- F = Fundamental Matrix
- Expected time to absorption = Fc, where c = unit vector

Hence Fc is the matrix we will derive from the provided data points.

TPM:

Out[227]:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.511 | 0.489 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **2** | 0.365 | 0.000 | 0.635 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **3** | 0.300 | 0.000 | 0.000 | 0.7 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **4** | 0.244 | 0.000 | 0.000 | 0.0 | 0.756 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **5** | 0.205 | 0.000 | 0.000 | 0.0 | 0.000 | 0.795 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **6** | 0.180 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.82 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **7** | 0.153 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.847 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **8** | 0.137 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.863 | 0.000 | 0.000 | 0.000 | 0.000 |
| **9** | 0.105 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.895 | 0.000 | 0.000 | 0.000 |
| **10** | 0.103 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.897 | 0.000 | 0.000 |
| **11** | 0.091 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.909 | 0.000 |
| **12** | 0.079 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.921 |
| **13** | 0.000 | 0.000 | 0.000 | 0.0 | 0.000 | 0.000 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 |

```
Time to Churn = Fc ::
```

Out[231]:

| | Time2Absorption |
|---|---|
| 1 | 52.787206 |
| 2 | 50.742216 |
| 3 | 47.991946 |
| 4 | 44.508264 |
| 5 | 40.513473 |
| 6 | 36.090686 |
| 7 | 31.206084 |
| 8 | 26.127086 |
| 9 | 20.736082 |
| 10 | 15.858576 |
| 11 | 10.503338 |
| 12 | 5.170189 |

# Q-5-4

State after n periods is given by $P_I \, x \, P^n$ where:

- $P_I = Initial Distribution$
- P = Transition Probability Matrix
- n = 4 = Number of periods

Out[240]:

| | Number Customers |
|---|---|
| States | |
| 1 | 1074.388975 |
| 2 | 539.320687 |
| 3 | 354.210981 |
| 4 | 255.615948 |
| 5 | 164.324538 |
| 6 | 267.153390 |
| 7 | 344.985480 |
| 8 | 0.000000 |
| 9 | 0.000000 |
| 10 | 0.000000 |
| 11 | 0.000000 |
| 12 | 0.000000 |
| 13 | 0.000000 |

# Q-5-5

The long-run CLV is given by:

**CLV** = $Lim_{t \to \infty} CLV_t$ = $(I - dP)^{-1} R$

Where:

- I = Identity Matrix
- P = PTM
- R = Reward Matrix = [1000, -200, -200,-200,-200,-200,-200,-200,-200,-200,-200,-200,0]
- d = discount = (1-.2) = 0.8

Out[250]:

| | CLV |
|---|---|
| 1 | 2149.629668 |
| 2 | 692.385122 |
| 3 | 521.049722 |
| 4 | 366.318932 |
| 5 | 242.578076 |
| 6 | 141.570457 |
| 7 | 48.816745 |
| 8 | -21.100835 |
| 9 | -82.126661 |
| 10 | -87.563622 |
| 11 | -90.152044 |
| 12 | -64.143405 |
| 13 | 0.000000 |

## Q-5-6

- Both are same as both are in State 1 at end of September. Time to churn in 53 months (Details in Q2)

## Q-5-7

TPM for Oct 2013 is:

Out[255]:

| TPM 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 64 | 65 | 66 | 67 | 68 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.148977 | 0.098666 | 0.130860 | 0.089315 | 0.075125 | 0.082522 | 0.374536 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **2** | 0.071390 | 0.096742 | 0.161302 | 0.116978 | 0.084461 | 0.084992 | 0.000000 | 0.384136 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **3** | 0.037220 | 0.058843 | 0.155777 | 0.135224 | 0.106183 | 0.101329 | 0.000000 | 0.000000 | 0.405423 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **4** | 0.022045 | 0.035943 | 0.103449 | 0.132588 | 0.118100 | 0.125870 | 0.000000 | 0.000000 | 0.000000 | 0.462004 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **5** | 0.016725 | 0.024355 | 0.075653 | 0.105075 | 0.121460 | 0.144371 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **6** | 0.012655 | 0.018166 | 0.052100 | 0.078913 | 0.108510 | 0.158960 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **7** | 0.038378 | 0.039901 | 0.091342 | 0.070000 | 0.081312 | 0.089105 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **8** | 0.031331 | 0.047041 | 0.088130 | 0.080580 | 0.064245 | 0.082553 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **9** | 0.022114 | 0.032076 | 0.088292 | 0.095641 | 0.085781 | 0.095442 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **10** | 0.014880 | 0.019548 | 0.064006 | 0.089855 | 0.093334 | 0.102373 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **11** | 0.012086 | 0.015493 | 0.046713 | 0.079734 | 0.094546 | 0.120323 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **12** | 0.008103 | 0.011395 | 0.035222 | 0.052263 | 0.085796 | 0.131776 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **13** | 0.036615 | 0.025722 | 0.057987 | 0.061645 | 0.053156 | 0.091498 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **14** | 0.024974 | 0.037029 | 0.056548 | 0.058001 | 0.057887 | 0.075090 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **15** | 0.020251 | 0.020355 | 0.056187 | 0.069626 | 0.071345 | 0.088516 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **16** | 0.011048 | 0.015392 | 0.054849 | 0.068390 | 0.075628 | 0.095819 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **17** | 0.009678 | 0.011185 | 0.036651 | 0.058248 | 0.080963 | 0.113236 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **18** | 0.007839 | 0.009602 | 0.028281 | 0.046069 | 0.063663 | 0.116859 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **19** | 0.032727 | 0.026537 | 0.038821 | 0.048967 | 0.049382 | 0.064102 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **20** | 0.022617 | 0.018165 | 0.045809 | 0.041448 | 0.044705 | 0.071578 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **21** | 0.016271 | 0.021240 | 0.050948 | 0.045912 | 0.057802 | 0.070804 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **22** | 0.011354 | 0.018140 | 0.038926 | 0.059803 | 0.055409 | 0.091033 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **23** | 0.009844 | 0.008807 | 0.029932 | 0.053384 | 0.063165 | 0.088705 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **24** | 0.005657 | 0.009608 | 0.023047 | 0.033312 | 0.051370 | 0.096860 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **25** | 0.008844 | 0.006035 | 0.040530 | 0.028641 | 0.032940 | 0.062849 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **26** | 0.031763 | 0.029316 | 0.042026 | 0.056701 | 0.055596 | 0.057518 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **27** | 0.009682 | 0.016984 | 0.041078 | 0.052383 | 0.040507 | 0.073169 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **28** | 0.009180 | 0.011238 | 0.033369 | 0.049322 | 0.051598 | 0.078945 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **29** | 0.007011 | 0.005838 | 0.024930 | 0.034825 | 0.050412 | 0.076224 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **30** | 0.006357 | 0.006947 | 0.017955 | 0.026996 | 0.042998 | 0.085079 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **44** | 0.010408 | 0.022934 | 0.026150 | 0.027944 | 0.028578 | 0.044131 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **45** | 0.006371 | 0.016682 | 0.015604 | 0.028233 | 0.029970 | 0.036419 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **46** | 0.005157 | 0.010176 | 0.019815 | 0.038687 | 0.033024 | 0.053367 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **47** | 0.003620 | 0.006867 | 0.020282 | 0.029435 | 0.036276 | 0.054734 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **48** | 0.003415 | 0.003915 | 0.009023 | 0.017479 | 0.030725 | 0.057304 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **49** | 0.011111 | 0.026474 | 0.036674 | 0.026897 | 0.020803 | 0.027214 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **50** | 0.003953 | 0.000000 | 0.027466 | 0.007905 | 0.004132 | 0.038480 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **51** | 0.007038 | 0.002970 | 0.022499 | 0.017766 | 0.022214 | 0.028625 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **52** | 0.004429 | 0.001874 | 0.012862 | 0.019643 | 0.018954 | 0.044645 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **53** | 0.002873 | 0.002363 | 0.010974 | 0.021125 | 0.026538 | 0.048210 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **54** | 0.005390 | 0.001352 | 0.008703 | 0.012110 | 0.023731 | 0.050262 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **55** | 0.012879 | 0.014234 | 0.003953 | 0.043667 | 0.021641 | 0.033360 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **56** | 0.005051 | 0.010781 | 0.008838 | 0.024851 | 0.008838 | 0.011905 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **57** | 0.004648 | 0.011274 | 0.024897 | 0.028392 | 0.033134 | 0.019872 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **58** | 0.004758 | 0.003197 | 0.013485 | 0.020187 | 0.027115 | 0.040592 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.890667 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **59** | 0.000751 | 0.002830 | 0.016329 | 0.017115 | 0.017284 | 0.044030 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.90166 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **60** | 0.003686 | 0.001865 | 0.008667 | 0.012957 | 0.021385 | 0.049761 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.901678 | 0.000000 | 0.000000 | 0.000 |
| **61** | 0.009091 | 0.020022 | 0.020238 | 0.014069 | 0.000000 | 0.028211 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.908369 | 0.000000 | 0.000 |
| **62** | 0.005051 | 0.010732 | 0.028706 | 0.039470 | 0.005682 | 0.015251 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.895108 | 0.000 |
| **63** | 0.000000 | 0.013242 | 0.009644 | 0.014461 | 0.019641 | 0.027330 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.915 |
| **64** | 0.002797 | 0.005084 | 0.010247 | 0.019226 | 0.018129 | 0.032927 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **65** | 0.004553 | 0.004568 | 0.014347 | 0.020216 | 0.017341 | 0.031431 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **66** | 0.002390 | 0.001359 | 0.011291 | 0.012139 | 0.019185 | 0.043169 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **67** | 0.000000 | 0.010101 | 0.015449 | 0.005348 | 0.017677 | 0.023529 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **68** | 0.011841 | 0.005682 | 0.014758 | 0.016711 | 0.028016 | 0.022334 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **69** | 0.001855 | 0.007005 | 0.021047 | 0.007559 | 0.016117 | 0.031069 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **70** | 0.004611 | 0.008425 | 0.014012 | 0.015338 | 0.015760 | 0.026578 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **71** | 0.001990 | 0.004218 | 0.005469 | 0.010458 | 0.023384 | 0.028805 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **72** | 0.001754 | 0.003724 | 0.007960 | 0.010953 | 0.019647 | 0.033428 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |
| **73** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000 |

73 rows × 73 columns

Inital Customer distribution:

Out[258]:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Counts-10/13** | 315 | 425 | 1013 | 1265 | 1381 | 1624 | 107 | 162 | 396 | 592 | ... | 68 | 132 | 253 | 9 | 16 | 42 | 56 | 122 | 258 | 11299 |

1 rows × 73 columns

Customer distribution in Nov 2013 is: $P_I * P * R$

where:

- P = TPM
- $P_I$ = Initial State
- R = Revenue = [22032, 6977, 3114, 1423, 720, 304], which we will repeat 12 times for each Frequency. Revenue =0 for State 73

Out[390]:

| | Counts-11/13 | Revenue |
|---|---|---|
| **1** | 305.49 | 6730654.27 |
| **2** | 388.81 | 2712755.11 |
| **3** | 995.10 | 3098738.97 |
| **4** | 1216.59 | 1731206.59 |
| **5** | 1349.80 | 971856.92 |
| **6** | 1807.95 | 549617.69 |
| **7** | 117.98 | 2599306.55 |
| **8** | 163.26 | 1139048.85 |
| **9** | 410.69 | 1278900.10 |
| **10** | 584.43 | 831650.75 |
| **11** | 707.57 | 509451.52 |
| **12** | 926.81 | 281750.44 |
| **13** | 63.13 | 1390791.85 |
| **14** | 98.19 | 685080.18 |
| **15** | 229.94 | 716030.32 |
| **16** | 364.67 | 518931.90 |
| **17** | 510.56 | 367606.11 |
| **18** | 673.42 | 204718.84 |
| **19** | 47.81 | 1053345.04 |
| **20** | 56.62 | 395028.45 |
| **21** | 188.64 | 587430.58 |
| **22** | 291.92 | 415396.02 |
| **23** | 382.28 | 275242.78 |
| **24** | 530.48 | 161266.78 |
| **25** | 40.67 | 896053.01 |
| **26** | 65.74 | 458695.83 |
| **27** | 131.93 | 410821.33 |
| **28** | 192.94 | 274552.15 |
| **29** | 285.03 | 205224.39 |
| **30** | 491.49 | 149413.74 |
| **31** | 27.89 | 614373.04 |
| **32** | 39.26 | 273933.42 |
| **33** | 113.40 | 353118.51 |
| **34** | 155.57 | 221374.38 |
| **35** | 198.59 | 142983.56 |
| **36** | 403.58 | 122688.11 |
| **37** | 21.58 | 475442.15 |
| **38** | 32.71 | 228227.99 |
| **39** | 72.22 | 224900.69 |
| **40** | 116.36 | 165581.50 |
| **41** | 161.76 | 116470.76 |
| **42** | 314.63 | 95647.86 |
| **43** | 18.91 | 416734.10 |
| **44** | 28.05 | 195684.86 |
| **45** | 53.47 | 166505.53 |
| **46** | 86.32 | 122832.06 |
| **47** | 130.23 | 93765.83 |
| **48** | 278.85 | 84769.51 |
| **49** | 11.84 | 260751.62 |
| **50** | 18.48 | 128912.99 |
| **51** | 49.40 | 153841.23 |
| **52** | 68.02 | 96794.83 |
| **53** | 112.04 | 80668.59 |
| **54** | 230.95 | 70208.98 |
| **55** | 11.06 | 243690.45 |
| **56** | 16.53 | 115296.09 |
| **57** | 49.44 | 153952.72 |
| **58** | 65.52 | 93241.02 |
| **59** | 82.58 | 59455.04 |
| **60** | 191.37 | 58176.53 |
| **61** | 14.79 | 325953.32 |
| **62** | 19.52 | 136222.16 |
| **63** | 48.28 | 150337.84 |

| | Counts-11/13 | Revenue |
|---|---|---|
| **64** | 84.61 | 120404.82 |
| **65** | 126.23 | 90887.29 |
| **66** | 272.31 | 82781.28 |
| **67** | 18.17 | 400263.90 |
| **68** | 17.01 | 118658.23 |
| **69** | 32.96 | 102651.58 |
| **70** | 61.99 | 88209.08 |
| **71** | 119.80 | 86253.02 |
| **72** | 230.35 | 70025.85 |
| **73** | 11762.41 | 0.00 |

## Q-5-8

From TPM (**EXHIBIT 14**) we see that :

1. In state 5 29% of Customers in this state move to Inactivate State
2. In state 6 30% of Customers in this state move to Inactivate State
3. In state 7 30% of Customers in this state move to Inactivate State
4. In state 8 38% of Customers in this state move to Inactivate State

Also Customers in 5,6,7 are high revenue generators for Flipkart.

Once in Inactive state a high proportion of Customers remain in Inactive state, and may finally churn out. Hence if Flipkart intervenes in the states 5,6,7,8 and the intervention is successful they may be able to generate more revenue

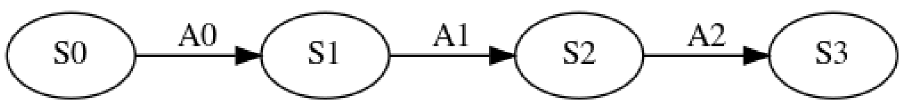
## Q-6-1

We will use Markov Decision Process to solve the following problem.

- Discount factor is 0.9
- State Space = [1,2]
- Action set = [1,2]
- TPMs for State-Action are provided in the Question (For saving space I will not be displaying the same in Answers)

IN MDP we have an Initial State (S0) in the diagram below and we take Action (A0), then S1 is the State in the next time period and this continues. Such a sequence is shown for 4 Stages in the diagram below.

Out[200]:




The Objective is to **Maximise** the expected return obtained over a period of return. Based on State and Reward the Reward generated can be represented as:

$R(S_0, a_0) + \beta R(S_1, a_1) + \beta^2 R(S_2, a_2) + \beta^3 R(S_3, a_3) + ...$

where :

- $R(S_0, a_0)$ = Reward generated from Initial Stage where initial State = $S_0$ and action taken = $a_0$. The rewards obtained from future states is discounted by factor $\beta$

Objective :: $Maximise_{a_i \epsilon A}[R(S_0, a_0) + \beta R(S_1, a_1) + \beta^2 R(S_2, a_2) + \beta^3 R(S_3, a_3) + ...]$

We will use **POLICY ITEARTION ALGORITHM** to obtain the long term revenue generated from policy (2,2,1,3)

The **Value Function** for a Policy $\Pi$ starting at State $S_i$ is given by:

$V^{\Pi}(i) = R^{\Pi}(i) + \beta \sum_{j \epsilon S} P_{ij}^{\Pi} x V^{\Pi}(j)$

where:

- $R^{\Pi}(i)$ :: Immediate Reward
- $\beta \sum_{j \epsilon S} P_{ij}^{\Pi} x V^{\Pi}(j)$ :: Discounted Future Reward

We will develop a System of Linear Equation by using the above equation for each State and Action provided in the Policy. The Matrix of the coefficients of the system of Equations are :

Out[338]:
```
array([[ 0.46, -0.36],
       [-0.27,  0.37]])
```

The Immediate Reward Matrix for the State Actions provided:

Out[339]:
```
array([[20],
       [-2]])
```

Policy to evaluate = (1, 2)

Solving the Above set of **Linear Equations** we get the following Values for the Policy:

Out[340]:

| | Policy Value |
|---|---|
| V11 | 91.51 |
| V22 | 61.37 |

Where $V_{ij}$ = Policy Value when Initial State was i and action taken was j The Overall Policy Value is :::

Out[30]: 152.87671232876724

To Check if Policy (1, 1) is better than the previous Policy (1, 2) we will use the **Policy Evaluation Step**. The objective is to check if the value function obtained in Q-4-1 is less than the new Policy.

The Policy improvement evaluation step includes the following:

$$T^{\Pi^{new}}(i) = Max_{a_{i,new}}[R(i, a_{i,new}) + \beta \sum_{j \epsilon S} P(j|i, a_{i,new}) x V^{\Pi}(j)]$$

where :

- $i \epsilon S$
- $a_{i,new}$ = A new action chosen for state i
- $T^{\Pi^{new}}(i)$ = Value function when the current policy for state i is changed to $a_{i,new}$
- R(i, a_{i,new}) = Reward when action for state i is replaced with new action $a_{i,new}$

Solving the new equation $T^{\Pi^{new}}(i) =

Out[342]: 65.71609756097563

$T^{\Pi^{new}}(2) > V^{\Pi}(2)$ value. Hence this is a better Policy.

**(1, 1) is better than (1, 2)**

We will resolve the problem for the new Policy. The Matrix of the coefficients of the system of New Equations are :

Out[343]:
```
array([[ 0.46, -0.36],
       [-0.72,  0.82]])
```

The Immediate Reward Matrix for the New State Actions provided:

Out[344]:
```
array([[ 20],
       [-12]])
```

Policy to evaluate = (1, 1)

Solving the Above set of **Linear Equations** we get the following Values for the Policy:

Out[346]:

| | Policy Value |
|---|---|
| V11 | 102.37 |
| V22 | 75.25 |

Where $V_{ij}$ = Policy Value when Initial State was i and action taken was j The Overall Policy Value is :::

Out[347]: 177.627118644068

## Q-6-2

The Optimal Policy can be obtained by solving a sustem of Linear Equations:

**Decision Variables**: x_si, i = 1,2 are the Best Policy Value function for Policy

**Minimize Objective**: x_s1 + x_s2

**Subject To**:

**Constraint_for_State_S1_Policy_1**: 0.46 x_s1 - 0.36 x_s2 >= 20

**Constraint_for_State_S1_Policy_2**: 0.28 x_s1 - 0.18 x_s2 >= 30

**Constraint_for_State_S2_Policy_1**: - 0.27 x_s1 + 0.37 x_s2 >= -2

**Constraint_for_State_S2_Policy_2**: - 0.72 x_s1 + 0.82 x_s2 >= -12

**Non Zero Constraint**: All Decision Variable (x_si >0)

```
 s1 :: 224.4 ::
 s2 :: 182.4 ::
Objective 406.8
```

```
# %load MDPPolicy2.sen
GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   1

Problem:
Objective:  Objective = 332.3636364 (MINimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
```

```
                                               Marginal    Upper bound          range          range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 Constraint_for_State_S1_Policy_1
                    BS      40.47273     -20.47273      20.00000           +Inf       -8.76712     -22.46575
Constraint_for_State_S1_Policy_2
                                                 .          +Inf           -Inf       25.55556    1366.66667

Constraint_for_State_S2_Policy_1

     2 Constraint_for_State_S1_Policy_2
                    NL      30.00000             .      30.00000       14.57534      -11.63636     152.87671
Constraint_for_State_S1_Policy_1
                                          11.63636          +Inf           +Inf           +Inf          +Inf

     3 Constraint_for_State_S2_Policy_1
                    NL      -2.00000             .      -2.00000      -20.21739       -8.36364     180.00000
Constraint_for_State_S2_Policy_2
                                           8.36364          +Inf       60.55556           +Inf     855.55556

Constraint_for_State_S1_Policy_1

     4 Constraint_for_State_S2_Policy_2
                    BS       3.23636     -15.23636     -12.00000       55.55556      -10.00000     300.00000
Constraint_for_State_S1_Policy_1
                                                 .          +Inf        3.23636           +Inf          +Inf

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                                     Page   2

Problem:
Objective:  Objective = 332.3636364 (MINimum)

   No. Column name  St      Activity      Obj coef   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 x_s1         BS     195.27273       1.00000             .           +Inf       -.72973      -5.40541
Constraint_for_State_S1_Policy_2
                                                 .          +Inf      195.27273          +Inf          +Inf

     2 x_s2         BS     137.09091       1.00000             .      455.55556       -.64286     107.14286
Constraint_for_State_S2_Policy_1
                                                 .          +Inf      137.09091          +Inf          +Inf

End of report
```

Optimal Policy (2,1)

Optimal Policy is obtained by checking the Sensitivity Report. Binding Constraints provide the best Policy Values

## Q-7-1

For a Poisson Distribution the number of events by time t, *N(t)* is given by:

$$P[N(t) = n] = \frac{e^{-\hat{\lambda}t}x(\hat{\lambda}t)^n}{n!}$$

Expected Value: $E[N(t)] = \lambda t$

Variance: $Var[N(t)] = \lambda t$

Data Points:

Out[351]:
```
array([ 9, 11,  8, 12,  6,  4, 14, 11,  6, 10, 16,  8,  6,  7,  4,  6,  7,
       13,  8, 16])
```

```
lambda :: 0.10989010989010989

Expected Demand for 12 Months :: 1.3186813186813187
```

## Q-7-2

To ensure that the Demand is met 90% of time:

$$\sum_{i=0}^{k} \frac{e^{-\hat{\lambda}t}x(\hat{\lambda}t)^i}{i!} \geq 0.90$$

Here t = 24 (2 years with frequency of months)

```
The table below shows density and cumulative distribution.
```

```
Expected Number of Demand for 2 years:: 2.6373626373626373
```

Out[388]:

| | K | Poisson Density | Cumulative Density |
|---|---|---|---|
| **0** | 0 | 0.07 | 0.07 |
| **1** | 1 | 0.19 | 0.26 |
| **2** | 2 | 0.25 | 0.51 |
| **3** | 3 | 0.22 | 0.73 |
| **4** | 4 | 0.14 | 0.87 |
| **5** | 5 | 0.08 | 0.95 |
| **6** | 6 | 0.03 | 0.98 |
| **7** | 7 | 0.01 | 0.99 |
| **8** | 8 | 0.00 | 1.00 |
| **9** | 9 | 0.00 | 1.00 |

For k = 5 the CDF >= 0.9. From the table we see that the **smallest k for which the cumulative probability is greater than 0.90 is 5**. Hence he needs to stock 5 parts to meet deman for 24 months