1. Machine Maintenance
2. Formulate this problem as a finite-horizon MDP. Define the states, actions, rewards, and transition probabilities. Provide and briefly explain the Bellman optimality equations. (5 pts)

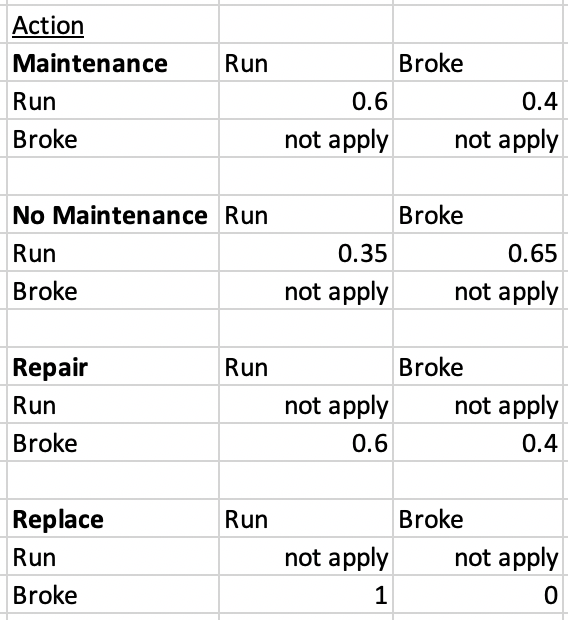
**States**: s ∈{R, B}

R= running

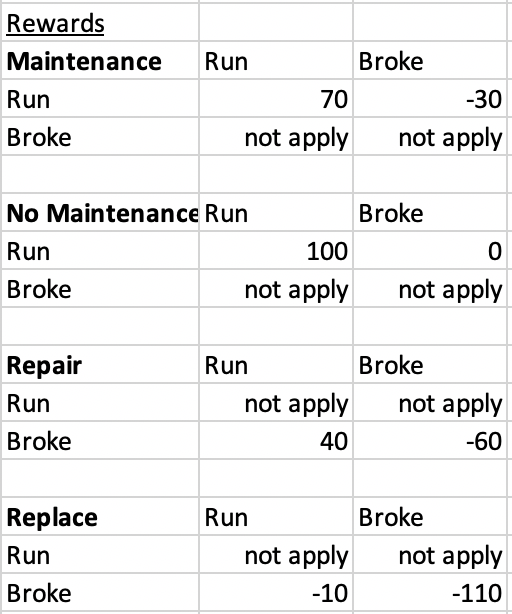
B = broken

**Actions**: A ∈{maintenance, no maintenance, repair, replace}

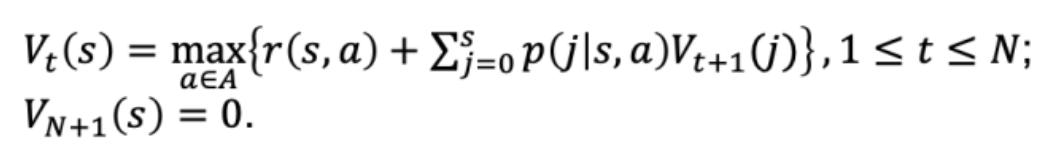
**Transition Probabilities(P):**



**Rewards(R):**



**Bellman Equation:**



Bellman Equation is set up for us to find the optimal value of being in state s (running or broke down) at period t (1 ~ N).

By comparing the sums of different rewards brought by each action (maintenance, no maintenance, repair, replace) and the expected value we got from the future periods, the highest value becomes the optimal value.

In detail, first, we calculate the immediate reward at the current state. That is to compute the reward brought by taking action a in state s.

Second, we calculate the expected future reward. That is to use *the probability* of reaching the next state j by taking action a, *times the optimal value of the next state j*, starting from period t+1. The optimal value of the next state j has been calculated because we use the method of backward induction to find optimal values.

Since we only have N weeks to run through, at period N, we don’t need to make any decision, so we set the value at N+1 to be 0.

(b) Develop R code to find the optimal solution (including the value function and policy). Report and discuss your solution for 𝑁 = 10. Will the optimal policy change when 𝑁 = 20? (10 pts)

Policy when N =10:

From week 1 to week 9:

If the machine is running: the optimal action is to perform preventive maintenance;

If the machine is broken: the optimal action is to replace the machine.

In week 10:

if the machine is running: the optimal action is not to perform preventive maintenance

if the machine is broken: the optimal action is to repair the machine.

Discussion:

From week 1 to week 9:

For week 1 to week 9, our policy is to maximize the chance that the machine is running, regardless of the machine’s state at the current period. That translates to maintenance (0.6 percent chance for machine to run throughout the week and have a gross profit of $100 that week) when the machine is currently running, and replace(100 percent chance for machine to run throughout the week and have a gross profit of $100 that week) when the machine is currently broken.

In week 10:

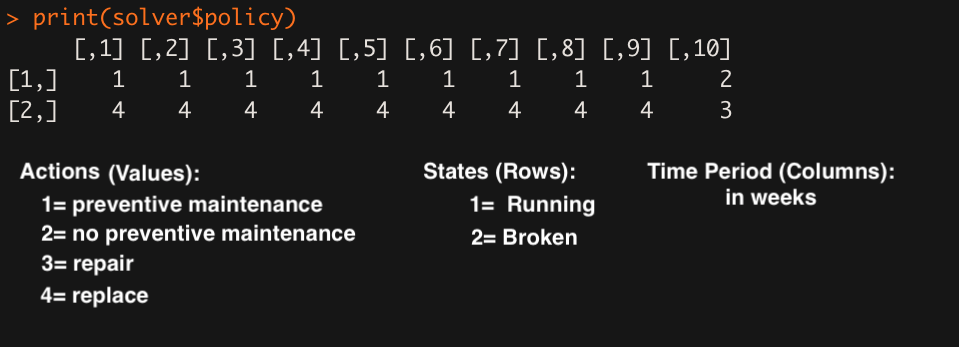
At week 10, our policy prefers the actions with the lower cost.

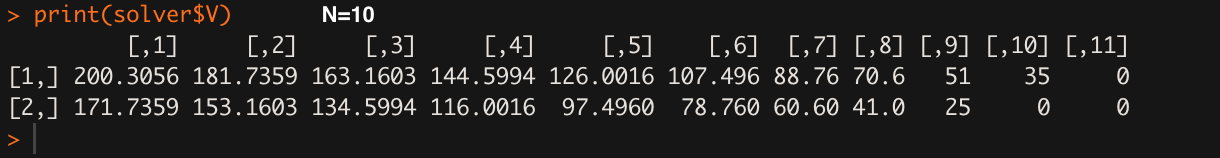
if the machine is running: the optimal action is not to perform preventive maintenance because we do not need to run the machine in the next week. Therefore, we don’t need to spend extra $30 on it.

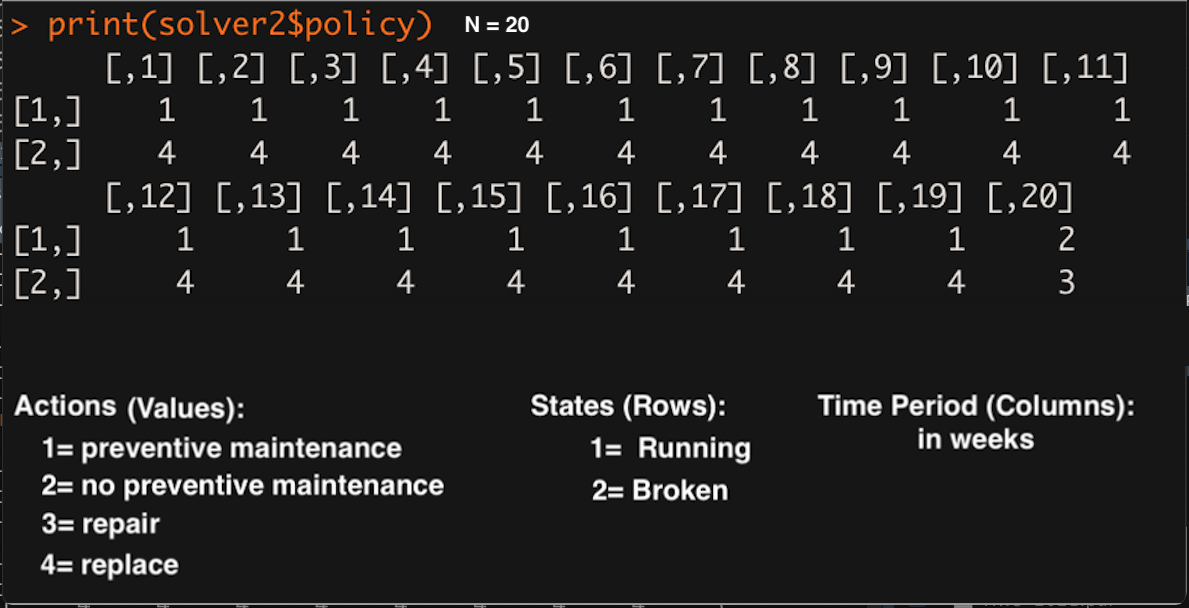
if the machine is broken: the optimal action is to repair the machine. As we do not need the machine to run in the next week (no expected reward), the best policy is to repair (lower cost at $60 than replace at $110)

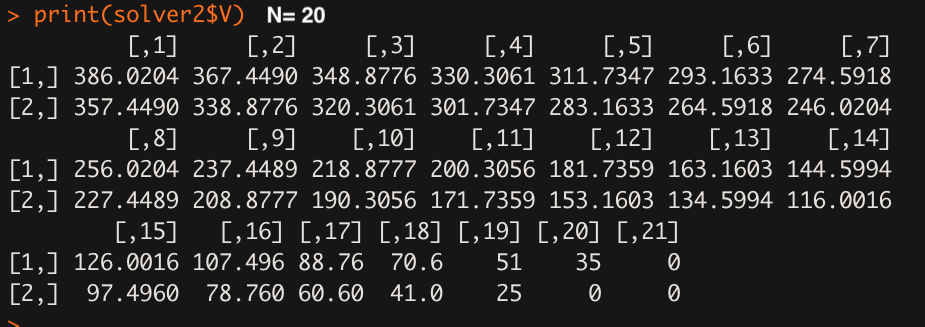
The optimal policy structure doesn’t change when N = 20.

As the machine’s conditions are independent from week to week, it’s based on the transition probabilities, the best action we can take at each period is depending on the current state, whether it’s running or broken. Therefore, no matter how many periods we take here, the first N-1 weeks will follow the same policy. And the last week will take different actions as there will be no more rewards in the week after the last week.









1. Gambler’s Problem:
2. Formulate this problem as an infinite-horizon MDP. Define the states, actions, rewards, and transition probabilities. Provide and explain the Bellman optimality equations. (10 pts)

Problem Formulation:

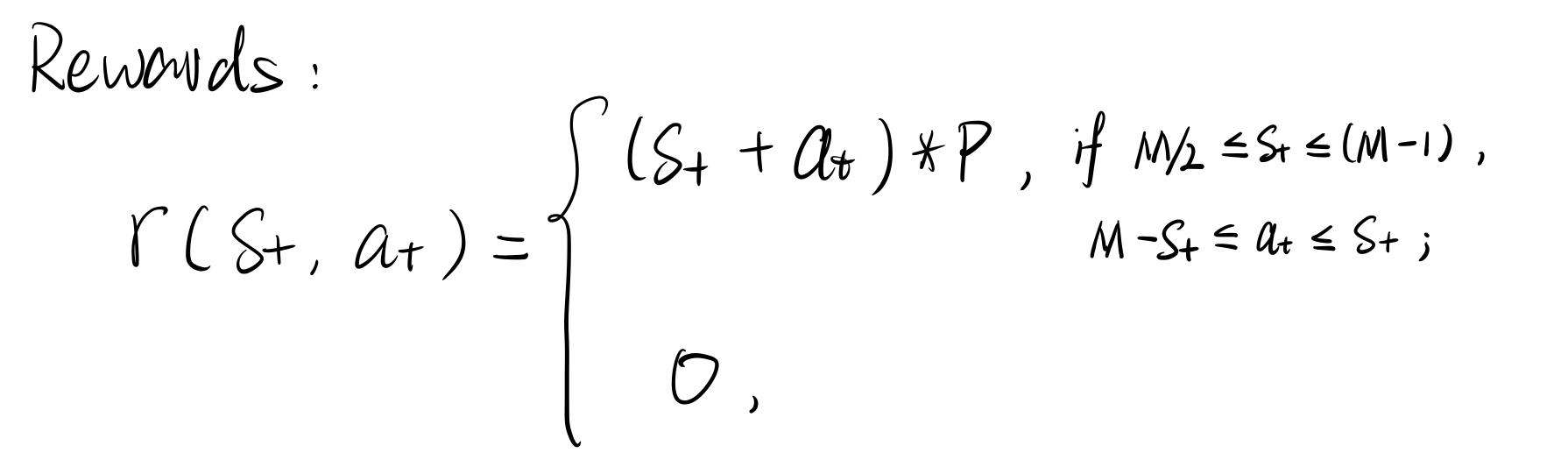
M=100

P=0.6

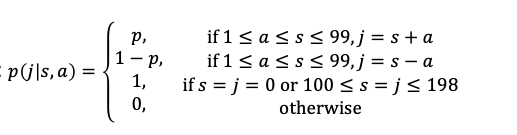
**States(S)**: stakes the gambler owns s ∈{0,1,2,…,2M-1,2M-2}

**Actions(A):** bet stake amount of a ∈{1,2,….,M-1}

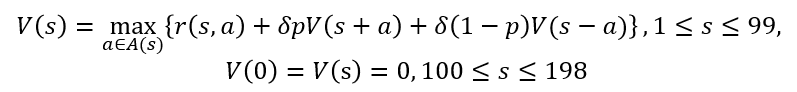
**Rewards(R):**



**Transition Probabilities(A):**



**Bellman Equation:**

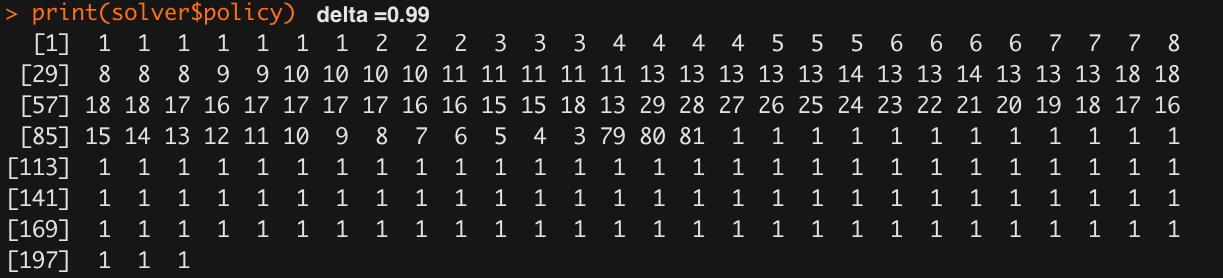


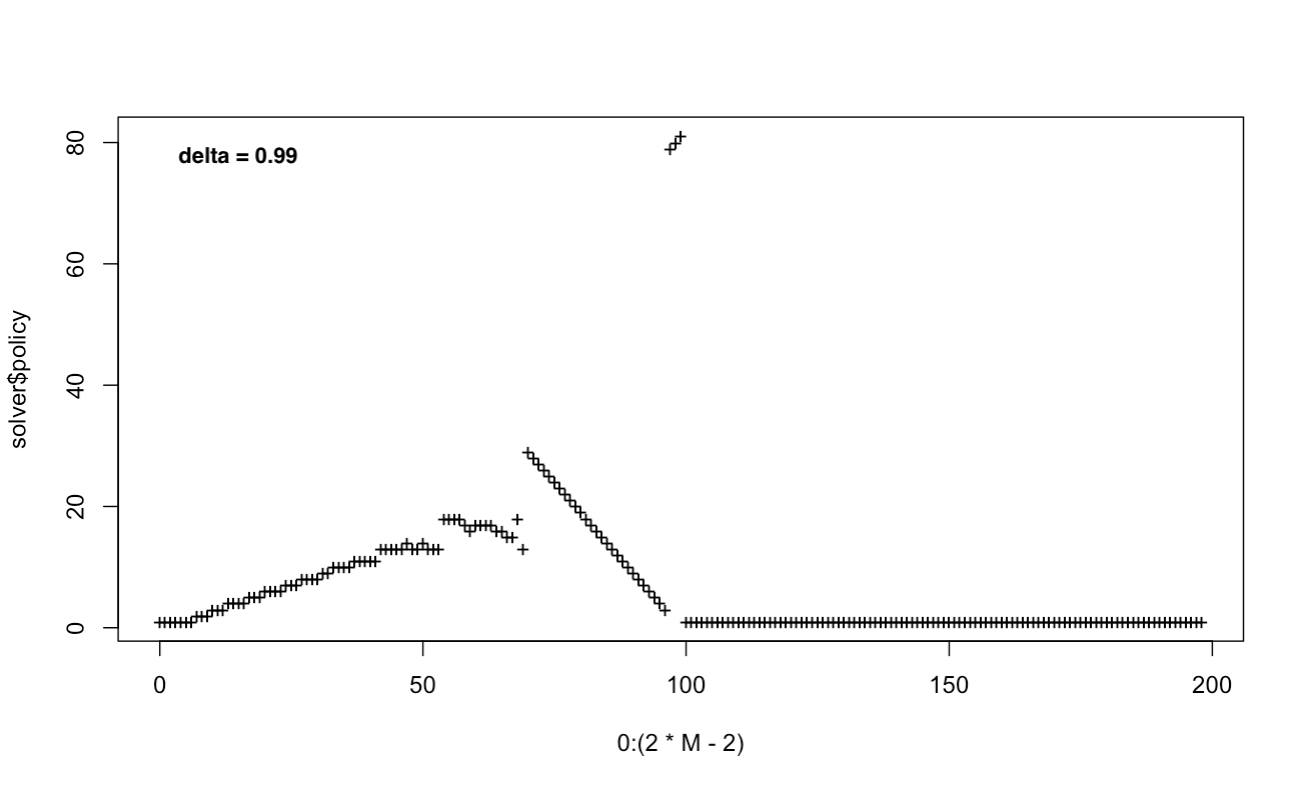
The optimal value here is max of the sum of immediate reward and future expected value. But in this case, it’s an infinite-horizon MDP problem. We won’t be able to calculate the expected value in the same way we used in the previous question. To approximate the expected value, a discount factor is involved in the equation. As a discount factor (delta) is smaller than 1, the term for the future expected value will converge to a finite value.

This optimal value is calculated based on how much money the gambler owns (s). As the game will finish when the gambler owns more than $100, the expected value (V) when s >= 100 is set to 0. Similarly, when the gambler lost all the money, which is s=0, the expected value (V) is set to 0 as well.

b) Develop R code to find the optimal solution. Report and discuss your solution for the following parameters: 𝑝 = 0.6, 𝑀 = $100, and (discount factor) 𝛿 = 0.99. (5 pts)

The gambler bets $1 for the first 7 states. Then he gradually increases his bet by a small amount for every few steps, considering the winning probability of 0.6. It makes sure he can accumulate some capital. Once reached around $65, he increases his bet significantly. Because he already has some capital, he can take the risk of betting a high amount. However, after several peak bets, he decreases his bet by $1. As the gambler aimed to max the money he earned, the strategy here suggests that approaching a higher value before reaching $100 is better. After getting near to the last few states (around $98), he can bet a very high amount (around $80), in order to maximize his money when the game ends.





c) At a high level, what do you think the form of the optimal policy looks like? That is, is there a simple way to describe the optimal policy to the gambler, without getting into details specific to the parameters in part (b)?

At a high level, we would expect the gambler to bet a small amount at the beginning because the probability of winning is 0.6, which gives him a higher chance to win. Because of the relative high probability of winning, the gambler should generate a small fortune very soon.

Then, we would expect him to gradually increase the betting amount. When he reaches a capital that is more than $60, he may decrease the amount he bets because the risk of betting a high amount is big at this time.

When he reaches a capital that is above 90, we could expect him to bet a very high amount (eg. an amount that is close to how much he owns at that time), because he wants to maximize his money when the game ends and the game is ending soon when he reaches a capital of around $90.

d) Resolve the model in part (b) with a new discount factor 𝛿 = 0.9999. What are the main changes in the optimal policy compared to part (b)? Can you explain the main message to the gambler at the high level? (3 pts)

Comparing to the policy with 𝛿 = 0.99, the policy with 𝛿 = 0.9999 makes many more smaller bets throughout. However, the optimal policy structure does not vary significantly: accumulate wealth in the beginning and at a certain threshold, make a large bet to maximize the optimal return.

As in question a, the discount factor is introduced to control how fast the game will end to approximate an infinite expected value. To the gambler, we can explain the discount factor as a measure of how much time he would like to spend on this game. The higher 𝛿 value means the longer time we can spent on this game. The policy with 𝛿=0.9999, has less “lost time-value of money” impact than the policy with 𝛿=0.99. The gambler can therefore be more patient with his bets, and make small bets to take advantage of the 0.6 probability of winning, as he gradually accumulates wealth. However, the overall policy structure (relatively smaller bets as wealth accumulates, followed by huge bets to maximize return.

