YZV231E - Prob. Stat. for Data Science

Nazrin Abdinli, 150220925 12/11/2023

Problem 1

```
import random
import math
import matplotlib.pyplot as plt
random.seed(0)

# Q1 - PART1 SOLUTION
N = 1000000    #number of game trials

#defining players and their choice probabilities
players_probs = {
    'Alice': {'rock': 0.8, 'paper': 0.2, 'scissors': 0.2},
    'Bob': {'rock': 0.2, 'paper': 0.4, 'scissors': 0.4},
    'Carol': {'rock': 1/3, 'paper': 1/3, 'scissors': 1/3}
}

possible_game_choices = ['rock', 'paper', 'scissors']


#loop for each player and their respective choice probabilities
for player, probs in players_probs.items():
    prob_indices = list(range(3))    #for representing the indices for rock, paper, and scissors
    prob_names = list(probs.keys())
    S = [0] * 3        #for initializing the count of rock, paper, and scissors

    #simulating game plays for N trials
    for n in range(N):
        #for selecting choices randomly for players X and Y based on their probabilities
        player_X = random.choices(prob_indices, weights=list(probs.values()))[0]    #for
        ↪ extracting a single element from a list or sequence of randomly sampled elements
        player_Y = random.choices(prob_indices, weights=list(probs.values()))[0]

        #for updating the count for the choice of player_X
        S[player_X] += 1
```

```
    #for calculating the probabilities based on the simulated counts
    total_simulated_probs = sum(S)
    S = [count / total_simulated_probs for count in S]

    #for plotting the probability mass function (PMF) for player's choices
    plt.bar(prob_names, S, label=player)

#for setting labels and title for the plot
plt.xlabel('Choices')
plt.ylabel('Probability')
plt.title('PMF of S for Players')
plt.legend
plt.show()
```

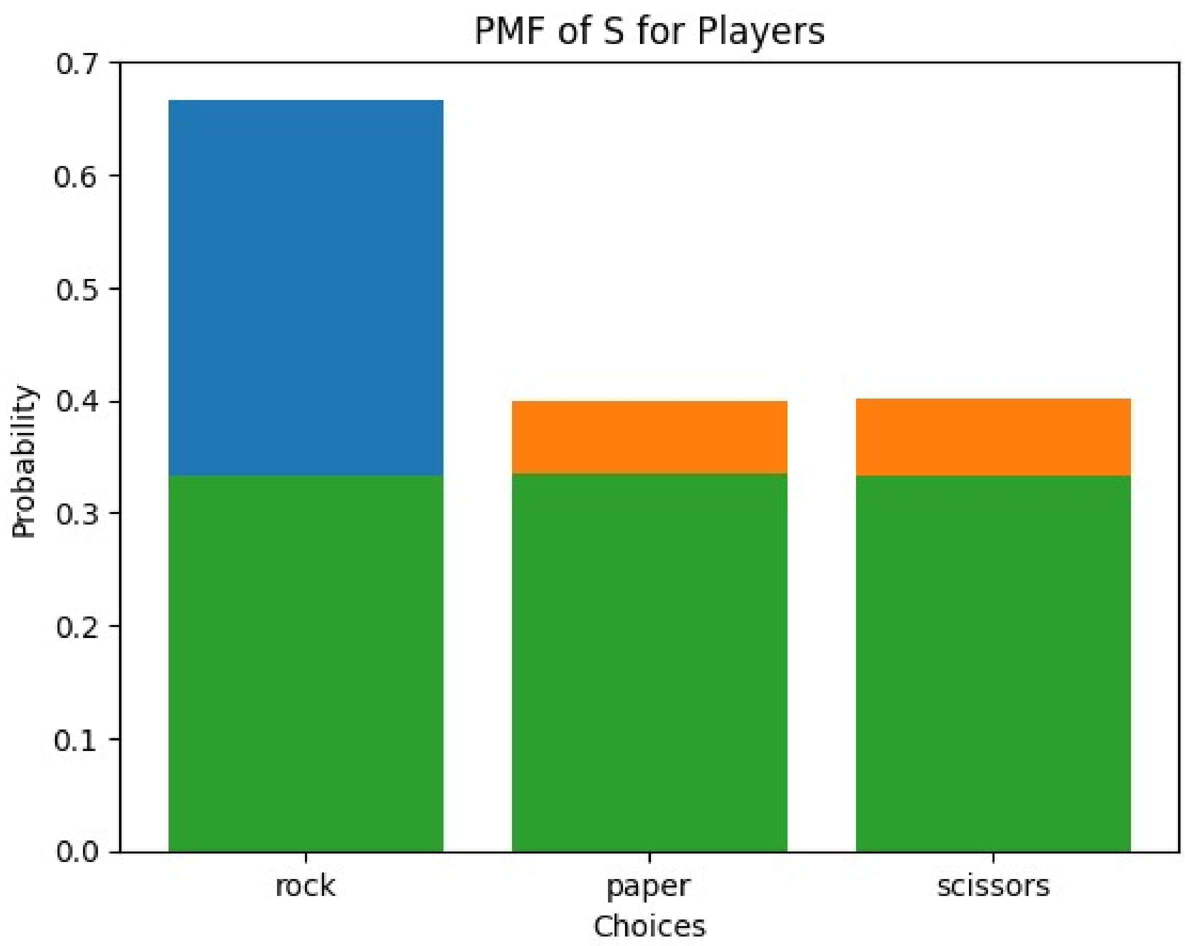


Figure 1: PMF of S for Players

```python
# Q1 - PART2 SOLUTION
#for calculating the combined probability of each choice (rock, paper, scissors) for all players
rock_choice_prob = 1 - (1- players_probs['Alice']['rock']) * (1 - players_probs['Bob']['rock']) * (1 - players_probs['Carol']['rock'])
paper_choice_prob = 1 - (1- players_probs['Alice']['paper']) * (1 - players_probs['Bob']['paper']) * (1 - players_probs['Carol']['paper'])
scissors_choice_prob = 1 - (1- players_probs['Alice']['scissors']) * (1 - players_probs['Bob']['scissors']) * (1 - players_probs['Carol']['scissors'])

print("Probability of selecting rock: ", rock_choice_prob)
print("Probability of selecting paper: ", paper_choice_prob)
print("Probability of selecting scissors: ", scissors_choice_prob)
```

```
Probability of selecting rock:  0.8933333333333333
Probability of selecting paper:  0.6799999999999999
Probability of selecting scissors:  0.6799999999999999
```

Figure 2: Probabilities of actions selecting rock, paper, scissors

```python
# Q1 - PART3 SOLUTION
num_wins_alice = 0
num_wins_bob = 0
num_wins_carol = 0
num_draws = 0

#simulating N games and keeping track of wins and draws
for n in range(N):
    #simulating choices for each player based on their defined probabilities
    player_X_result = random.choices(prob_names,
    ↪ weights=list(players_probs['Alice'].values()))[0]
    player_Y_result = random.choices(prob_names,
    ↪ weights=list(players_probs['Bob'].values()))[0]
    player_Z_result = random.choices(prob_names,
    ↪ weights=list(players_probs['Carol'].values()))[0]

    #determining winners or draws based on the rules
    if (player_X_result == 'rock' and player_Y_result == 'scissors') or (player_X_result ==
    ↪ 'scissors' and player_Y_result == 'paper') or (player_X_result == 'paper' and
    ↪ player_Y_result == 'rock'):
        num_wins_alice += 1
    elif (player_Y_result == 'rock' and player_X_result == 'scissors') or (player_Y_result
    ↪ == 'scissors' and player_X_result == 'paper') or (player_Y_result == 'paper' and
    ↪ player_X_result == 'rock'):
        num_wins_bob += 1
    elif (player_Z_result == 'rock' and player_X_result == 'scissors') or (player_Z_result
    ↪ == 'scissors' and player_X_result == 'paper') or (player_Z_result == 'paper' and
    ↪ player_X_result == 'rock'):
        num_wins_carol += 1
    else:
        num_draws += 1

#calculating the probabilities of wins and draws
prob_wins_alice = num_wins_alice / N
prob_wins_bob = num_wins_bob / N
prob_wins_carol = num_wins_carol / N
prob_draws = num_draws / N


print("Probability of Alice winning:", prob_wins_alice)
print("Probability of Bob winning:", prob_wins_bob)
```

```
print("Probability of Carol winning:", prob_wins_carol)
print("Probability of a draw:", prob_draws)
```

```
Probability of Alice winning: 0.366308
Probability of Bob winning: 0.366333
Probability of Carol winning: 0.089464
Probability of a draw: 0.177895
```

Figure 3: Ali, Bob, Carol’s winning and draw probabilities

Problem 2

```
import random
import math
import matplotlib.pyplot as plt
random.seed(0)

# Q2 - PART0 SOLUTION
num_weeks = 14
all_probs = []

#loop for each week to calculate the probability of students being present
for week in range(1, num_weeks+1):
    weekly_prob = 0.6 * (1 - 4 * ((week - 1) / 13) * (1 - (week - 1) / 13)) + 0.2
    ↪  #Calculating the probability for the current week based on a complex function
    all_probs.append(weekly_prob)      #appending the calculated probability to the list for
    ↪  each week

plt.bar(range(1, num_weeks + 1), all_probs)
plt.xlabel('Week')
plt.ylabel('Probability (weekly_prob)')
plt.title('Probability of Students Present in Each Week')
plt.show()
```

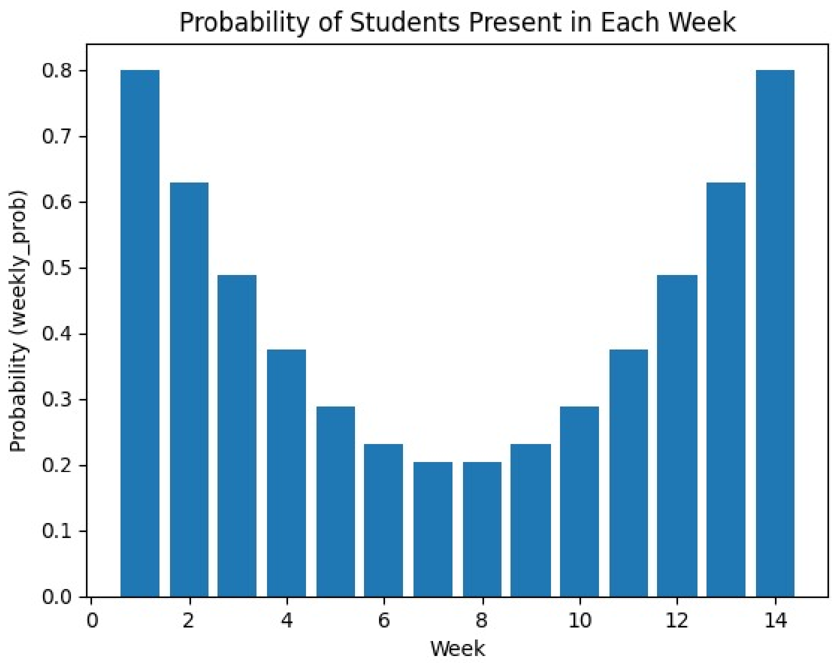
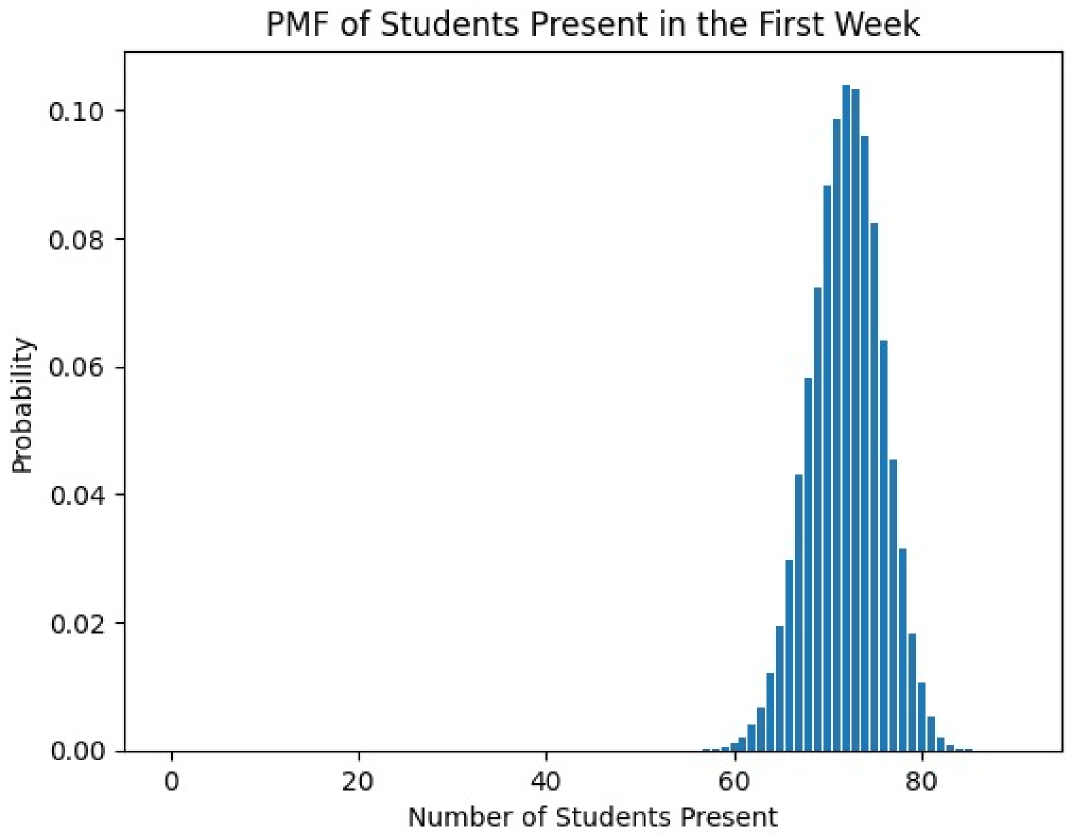
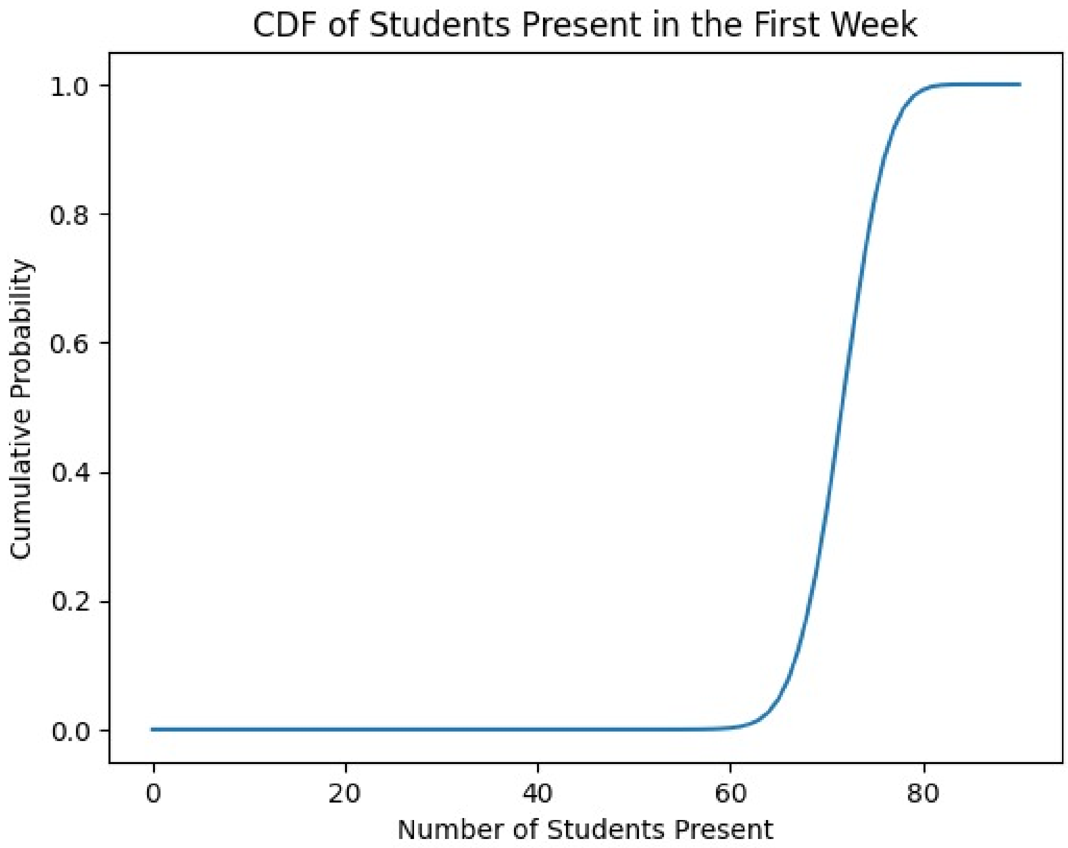


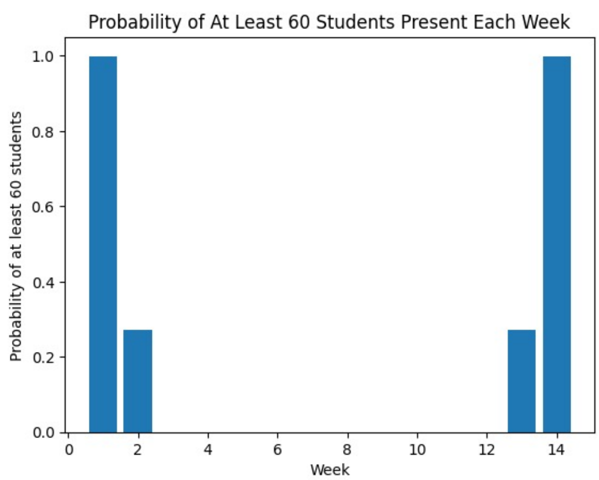
Figure 4: Probability of Students Present in Each Week

```
# Q2 - PART1 SOLUTION
N = 100000
present_students_week1 = []

#loop for the range of N
for n in range(N):
    present_students = sum(random.random() < all_probs[0] for x in range(90))
    present_students_week1.append(present_students)

#PMF
#creating a PMF by counting the occurrences of each count of present students and dividing
↪    by the total number of simulations
pmf = [present_students_week1.count(y) / N for y in range(91)]
plt.bar(range(91), pmf)
plt.xlabel('Number of Students Present')
plt.ylabel('Probability')
plt.title('PMF of Students Present in the First Week')
plt.show()

#CDF
#creating CDF by calculating the cumulative sum of probabilities up to each count of present
↪    students
cdf = [sum(pmf[0:i+1]) for i in range(len(pmf))]
plt.plot(range(91), cdf)
plt.xlabel('Number of Students Present')
plt.ylabel('Cumulative Probability')
plt.title('CDF of Students Present in the First Week')
plt.show()
```

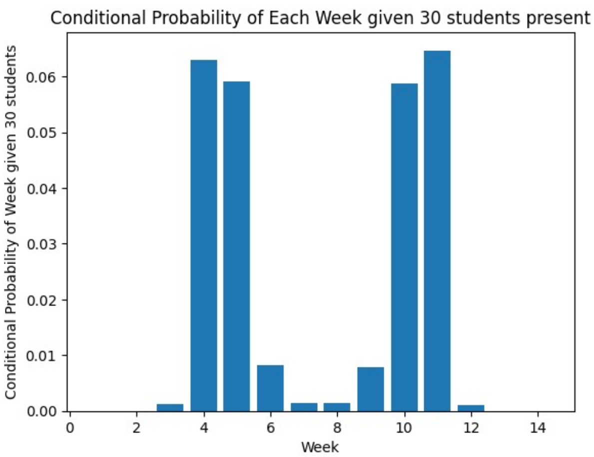

PMF of Students Present in the First Week
Probability
0.10
0.08
0.06
0.04
0.02
0.00
0
20
40
60
80
Number of Students Present


CDF of Students Present in the First Week
Cumulative Probability
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
Number of Students Present

```
# Q2 - PART2 SOLUTION
at_least_60_all_probs = []

#loop for 14 weeks
for week in range(14):
    weekly_present_students = []

    #simulating N times for a chosen week
    for n in range(N):
        #simulating the number of students present using a probability distribution for this
        ↪ specific week
        present_students = sum(random.random() < all_probs[week] for x in range(90))
        weekly_present_students.append(present_students)        #appending the count of
        ↪ present students to the list

    at_least_60_count = sum(1 for at_least_60_students in weekly_present_students if
    ↪ at_least_60_students >= 60)        #counting the occurrences where at least 60
    ↪ students were present in the simulations for this week
    at_least_60_prob = at_least_60_count / N        #calculating the probability of having at
    ↪ least 60 students present for this week

    at_least_60_all_probs.append(at_least_60_prob)        #appending the calculated
    ↪ probability to the list for all weeks

plt.bar(range(1, 15), at_least_60_all_probs)
plt.xlabel('Week')
plt.ylabel('Probability of at least 60 students')
plt.title('Probability of At Least 60 Students Present Each Week')
plt.show()
```



```
# Q2 - PART3 SOLUTION
students_present_30 = 30
conditional_pmf = []

#loop for 14 weeks
for week in range(14):
    count_30_students = sum(1 for present_count in range(N) if sum(random.random() <
    ↪   all_probs[week] for _ in range(90)) == students_present_30)      #initializing a
    ↪   count for the occurrence of exactly 30 students present for a particular week
    prob_30_students = count_30_students / N      #calculating the probability of exactly
    ↪   30 students being present for a particular week

    #probability of being week 7 given 30 students
    if week == 6:  #week 7 corresponds to index 6
        week_7_30_students_prob = prob_30_students

    conditional_pmf.append(prob_30_students)      #appending the calculated probability to
    ↪   the list for all weeks


plt.bar(range(1, 15), conditional_pmf)
plt.xlabel('Week')
plt.ylabel('Conditional Probability of Week given 30 students')
plt.title('Conditional Probability of Each Week given 30 students present')
plt.show()

print(f"Probability of Week 7 given 30 students present: {week_7_30_students_prob}")
```



1 Problem 1 Handwriting

1.1 Subsection A

if Ali wins:

$$
\begin{aligned}
BBB &\to \frac{3!}{3!} = 1 \\
BBBW &\to \frac{4!}{3! \cdot 1!} = 4 \\
BBBWW &\to \frac{5!}{3! \cdot 2!} = \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}{2 \cdot 3 \cdot 2} = 10
\end{aligned}
$$

if Ahmet wins:

$$
\begin{aligned}
WWW &\to \frac{3!}{3!} = 1 \\
WWWB &\to \frac{4!}{3! \cdot 1!} = 4 \\
WWWBB &\to \frac{5!}{3! \cdot 2!} = \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}{2 \cdot 3 \cdot 2} = 10 \\
\Omega =& \{BBB, BBBW, BBWB, BWBB, WBBB, \\
& BBBWW, BBWWB, BWWBB, WWBBB, \\
& WBWBB, WBBWB, WBBBW, BWBWB, \\
& BWBBW, BBWBW \\
& WWW, WWWB, WWBW, WBWW, BWWW, \\
& WWWBB, WWBBW, WBBWW, BBWWW, \\
& BWBWW, BWWBW, BWWWB, WBWBW, \\
& WBWWB, WWBWB\} \\
|\Omega| =& 30
\end{aligned}
$$

1.2 Subsection B

P(Ali) $\to$ probability that Ali wins the game

$E_{Ali}:$ Ali wins $= \{BBB, BBBW, BBWB, BWBB, WBBB$

$BBBWW, BBWWB, BWWBB, WWBBB,$

$WBWBB, WBBWB, WBBBW, BWBWB,$

$BWBBW, BBWBW\}$

$$P(Ali) = \frac{|E_{Ali}|}{|\Omega|} = \frac{15}{30} = \frac{1}{2}$$

1.3 Subsection C

$P($ AliBlack $)=$ probability that Ali wins the game, given that the first car is black

$$
\begin{aligned}
& E_{\text{Aliblack}}: \text{ Ali wins} \\
& E_{\text{Aliblack}}=\{BBB, BBBW, BBWB, BWBB, \\
& BBWWW, BBWWB, BWWBB, BWBWB, \\
& BWBBW, BBWBW\} \\
& P\left(E_{\text{Aliblack}}\right)=\frac{\left|E_{\text{Aliblack}}\right|}{|\Omega|}=\frac{10}{30}=\frac{1}{3}
\end{aligned}
$$

2 Problem 2 Handwriting

2.1 Subsection A

$P($ first rode) : probability of choosing first road
P (no traffic | first road): probability of not getting stuck in tragic in first road

$$P(\text{ first road })P(\text{ no traffic } \mid \text{ first road }) = 0.5 \cdot 0.1 = 0.05$$

2.2 Subsection B

$P($ second rode) : probability of choosing second road
P (no traffic | second road): probability of not getting stuck in traffic in second road

$$P(\text{ second road })P(\text{ no traffic } \mid \text{ second road }) = 0.3 \cdot 0.08 = 0.024$$

2.3 Subsection C

$P($ third rode) : probability of choosing third road
P (no traffic | third road): probability of not getting stuck in traffic in third road

$$P(\text{ third road })P(\text{ no traffic } \mid \text{ third road }) = 0.2 \cdot 0.12 = 0.024$$

3 Problem 3 Handwriting

3.1 Subsection A

$P(k \text{ errors }) = \begin{pmatrix} n \\ k \end{pmatrix} \times p^k \times (1-p)^{n-k}$

$n = 64$ (number of bits)

$p = 0.008$ (probability of error for a single bit)

$k :$ the number of errors

the probabilities of having 0,1 , or 2 errors are:

$$P(0 \text{ error }) = \begin{pmatrix} 64 \\ 0 \end{pmatrix} \times 0.008^0 \times (1-0.008)^{64} \approx 0.2131$$

$$P(1 \text{ error }) = \begin{pmatrix} 64 \\ 1 \end{pmatrix} \times 0.008^1 \times (1-0.008)^{63} \approx 0.3457$$

$$P(2 \text{ errors }) = \begin{pmatrix} 64 \\ 2 \end{pmatrix} \times 0.008^2 \times (1-0.008)^{62} \approx 0.2571$$

$$P(\text{ accepted }) = P(0 \text{ error }) + P(1 \text{ error }) + P(\text{ 2 errors })$$

$$\downarrow$$

$$P(\text{ accepted }) \approx 0.2131 + 0.3457 + 0.2571 \approx \mathbf{0.8159}$$

$$P(\text{accepted}) = P(\text{no retransmission})$$

3.2 Subsection B

$(P(\text{ no retransmission }))^5 \rightarrow$ the probability that 5 blocks are transmitted without any retransmission

$P(5 \text{ blocks without retransmission }) = (P(\text{ no retransmission }))^5 \approx (0.8159)^5 \approx$ **0.3615**

4 Problem 4 Handwriting

Choosing a prize from n prizes for each couple:

$$\begin{array}{ccccc}
\underline{1^{\text{st}}\text{ couple:}} & \underline{2^{\text{nd}}\text{ couple:}} & \underline{3^{\text{rd}}\text{ couple:}} & \cdots & \underline{n^{\text{th}}\text{ couple:}} \\
\binom{n}{1} \cdot & \binom{n-1}{1} \cdot & \binom{n-2}{1} & \cdots & \binom{1}{1} \\
\downarrow & \downarrow & \downarrow & & \downarrow \\
\frac{(n!)}{(n-1)!1!} \cdot & \frac{(n-1)!}{(n-2)!1!} \cdot & \frac{(n-2)!}{(n-3)! \cdot 1!} & \cdots & \frac{1}{\underbrace{0! \cdot 1!}_{1}}
\end{array}$$

$n!$ $\rightarrow$ choosing a prize from n prizes for each couple: then, we should take into account that we have 2 person choices for each couple $\rightarrow$ we have n couples so totally we have 2^n person choices.

$$|E| = n! \cdot 2^n$$

$\Omega :$ our sampletpace is choosing n people prom $2n$ people

$$|\Omega| = \binom{2n}{n}$$

so, our probability is:

$$P(E) = \frac{|E|}{|\Omega|} = \frac{\mathbf{n! \cdot 2^n}}{\binom{2n}{n}}$$

5 Problem 5 Handwriting

5.1 Subsection A

$P(T = t) = \begin{pmatrix} t-1 \\ r-1 \end{pmatrix} \cdot p^r \cdot (1-p)^{t-r}$

t : the number of trials

r : the number of successes we want (5 toys for ages 3-8)

p is the probability of success (0.3)

$1 - p$ is the probability of failure (0.7)

Given that $P(T = t)$ represents the probability of finding 5 toys suitable for ages 3-8 on the t^{th} trial, we can calculate this PMF for $t > 14$

$$p_T(t) = \begin{pmatrix} t-1 \\ 5-1 \end{pmatrix} \times (0.3)^5 \times (1-0.3)^{t-5}$$

5.2 Subsection B

The conditional probability $P(T = t \mid T > 14)$ is the probability of needing t trials given that we haven't found the 5 toys even after 14 trials

- $P(T = t \cap T > 14)$: the probability of requiring t trials and it being greater than 14
- $P(T > 14)$: the probability that the number of trials needed is more than 14

$$P(T > 14) = \sum_{t=15}^{\infty} P(T = t)$$
$$P(T > 14) = 1 - \sum_{t=0}^{14} P(T = t)$$
$$P(T > 14) = 1 - \sum_{t=0}^{14} \begin{pmatrix} t-1 \\ 5-1 \end{pmatrix} \times 0.3^5 \times (1-0.3)^{t-5}$$

6 Problem 6 Handwriting

6.1 Subsection A

$P(i \geqslant k)=\sum_{i=k}^{N}\left(\begin{array}{c}N \\ i\end{array}\right) \times p_1^i \times\left(1-p_1\right)^{N-i}$

Above is the formula for calculating the probability of at least k students present in the 1^{st} week

6.2 Subsection B

$P(n \text{ students in week } w)=\left(\begin{array}{c}N \\ n\end{array}\right) \times\left(p_w\right)^n \times\left(1-p_w\right)^{N-n}$

$P(\text{ current week } w)=\frac{1}{W_{\text{max}}}$

P (current ween $w \ln$ students $)=P(n$ students in week $w) \times \frac{1}{W_{\max }}$