SEJONG UNIVERSITY
VISION 2030 WORLD TOP100 UNIVERSITY

P.E.L.
ROUGH CONSENSUS & RUNNING CODE

대기행렬의 기초
- 2장. 확률이론의 기초 -

2025.06.30.

Wooyoung Son
wooyoung@pel.sejong.ac.kr
Protocol Engineering Lab., Sejong University

CONTENTS

1 개요

2 확률변수

3 기대치

4 확률의 근사 해석법

개요

I

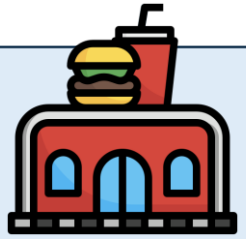
I 1 개요 (1/2)

A. 확률이론 기초의 필요성

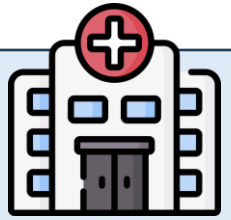
| 대기행렬 이론은 예측이 어려운 현실의 문제를 수학적으로 모델링하기 위해 고안되었으며, 대기행렬 시스템은 불확실한 현상들로 구성되어 있음

- 현실세계에서의 고객 도착 시간, 서비스 시간, 대기 시간 등은 모두 고정된 수치가 아니라 확률적으로 변화함

| 이러한 불확실성을 정량적으로 예측하고, 시스템의 성능을 평가하기 위해서는 확률이론이 필수적임



점심시간 고객 도착량을 예측하여 적절한 인력을 배치



환자 도착과 치료 시간의 변동성을 고려하여 대기 시간을 분석

특정 시간대에 몰리는 통화량을 예측하여 상담 인력 수를 결정

확률이론은 대기행렬 해석의 출발점이며,
복잡하고 불확실한 현실을 수학적으로 바꾸는 언어이다!

A. 대기행렬 해석을 위한 확률 도구

| 대기행렬 시스템을 수학적으로 이해하고 분석하기 위해서는, 불확실한 현상을 수치로 표현하고(확률변수), 그 평균적 행동을 파악하며(기대치), 복잡한 구조도 다룰 수 있는 도구(근사 해석법)가 필요함

1 확률변수

| 현실의 불확실한 현상을 수치로 표현하는 수학적 도구

- 고객 도착 시간, 서비스 시간, 대기 시간은 모두 매 순간 달라질 수 있음에 따라 이를 하나의 수치로 정리하고 모델링하기 위해 확률변수로 표현함

2 기대치

| 확률변수가 평균적으로 어떤 값을 가질지를 나타냄

- '고객은 평균적으로 얼마나 기다려야 할까?', '하루 평균 몇 명이 대기하는가?' 등과 같은 질문은 확률변수를 기반으로 한 기대치 계산을 통해 정량적으로 답할 수 있음

3 근사 해석법

| 복잡한 시스템을 해석 가능한 형태로 단순화하여, 빠르게 성능을 예측할 수 있도록 하는 기법

- 실제 시스템은 복잡하기 때문에, 정확한 해를 도출하는 것은 불가능하거나 계산이 매우 복잡하므로, 확률적 근사 해석이 필요함

확률변수

II

II 2 확률변수 (1/30)

A. 확률변수 개요

| 정의: 어떤 확률실험에서 결과가 확률에 따라 결정될 때, 그 결과를 수치로 표현한 변수

| 예시: 주사위를 던졌을 때 나올 수 있는 확률변수 (1, 2, 3, 4, 5, 6)

| 종류: 이산확률변수, 연속확률변수

이산확률변수 (Discrete Random Variable)

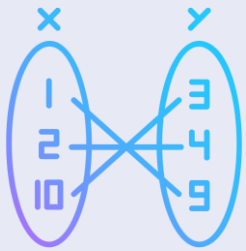
취할 수 있는 값의 집합이 **유한하거나 셀 수 있을 만큼 무한한 확률변수**

대기행렬에서의 예시:

- 대기열에 있는 고객 수
- 일정 시간 내 도착한 요청 횟수

대표 확률변수:

- 이항확률변수
- 기하확률변수
- 포아송확률변수



연속확률변수 (Continuous Random Variable)

어떤 구간 내의 **모든 실수 값을 가질 수 있는 확률변수**

대기행렬에서의 예시:

- 두 고객 도착 간격
- 서비스 완료까지 걸리는 시간

대표 확률변수:

- 지수확률변수
- 정규확률변수

A. 연속확률변수 (Continuous Random Variable)

| 정의: 어떤 구간 내의 모든 실수 값을 가질 수 있는 확률변수

| 특징: 확률은 개별 값이 아닌, 구간 단위로 정의되며, 확률밀도함수(PDF, Probability Density Function) $f(x)$를 통해 확률을 계산함

- 특정 구간 [a, b]에 대한 확률은 다음과 같이 도출됨: $P(a \leq X \leq b) = \int_a^b f(x)\,dx$
- 확률밀도함수 $f(x)$의 조건: $f(x) \geq 0$, $\int_{-\infty}^{\infty} f(x)\,dx = 1$

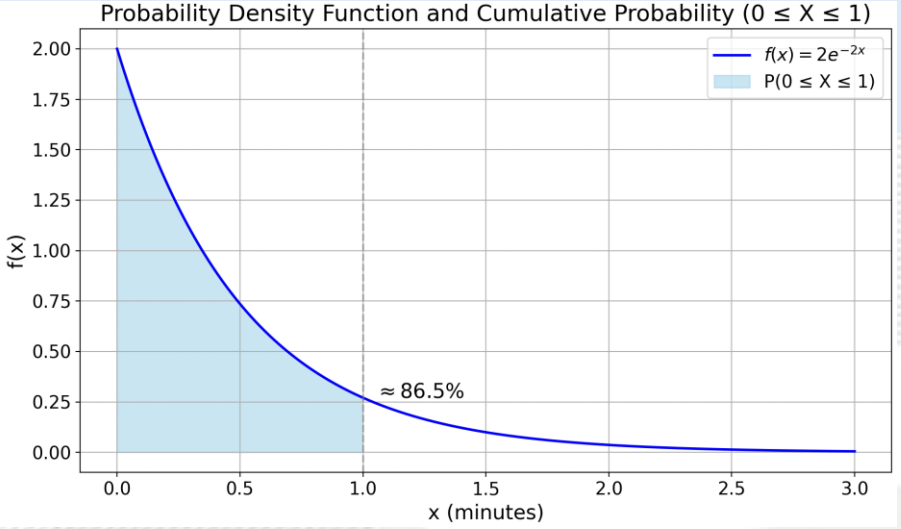
AI센터 1층 투썸플레이스에서 커피를 주문한 후, 음료를 받기까지의 시간이 X이고, 이 시간이 확률밀도함수 $f(x) = 2e^{-2x}$를 따른다고 하자.

이 때, 커피가 <u>1분 이내에 준비될 확률</u>은?

- $P(0 \leq X \leq 1) = \int_0^1 2e^{-2x}\,dx$

$= [-e^{-2x}]_0^1 = -e^{-2} + 1 \approx 0.8647$

- 약 86.5% 확률로 1분 이내에 커피를 받을 수 있다!



(그림 1) 확률밀도함수를 통한 확률 예측 그래프

A. 연속확률변수 (Continuous Random Variable)

| 대표적인 연속확률변수: 지수확률변수, 정규확률변수

대기행렬을 공부하기 위해서는 왜 하필 지수확률변수, 정규확률변수를 알아야할까?!

대기행렬 (Queueing Theory)에서 해결하고자 하는 문제

- "고객이 도착하는 **시간 간격**은 어떻게 되는가?"
- "서비스를 받는 데 **걸리는 시간**은 얼마나 되는가?"
- "평균적으로 **얼마나 오래** 기다려야 하는가?"

지수분포는 **Memoryless의 성질로 인해, '다음 사건까지 기다리는 시간'을 모델링**하는 데 특화된 분포임

정규분포는 **여러 확률변수의 합을 다루며, 평균 성능 분석이나 근사 해석**에 사용됨

대기행렬의 시간 분석에는 지수확률변수,
시스템 평균 성능 분석에는 정규확률변수가 필수적으로 활용됨

A. 지수확률변수 (Exponential Random Variable)

| 정의: 확률밀도함수가 다음과 같은 형태를 가지는 연속확률변수

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

| 지수분포 표기법: $X \sim Exp(\lambda)$

| 특징: 기다리는 시간, 이벤트 간 간격을 모델링할 때 사용됨

- 기억 없음 성질 (Memoryless Property): 과거에 얼마나 기다렸는지와 무관하게 앞으로 기다릴 시간의 확률 분포가 동일함
  - 연속확률변수 $X \geq 0$가
    $P(X > s + t | X > s) = P(X > t)$
    for all $s, t \geq 0$ 를 만족하면, $\rightarrow X \sim Exp(\lambda)$

$* S(x) = P(X > x)$

- $\frac{P(X>s+t)}{P(X>s)} = \frac{S(s+t)}{S(s)} = S(t)$
- $S(s+t) = S(s) \cdot S(t)$
- $S(x) = e^{-\lambda x}$

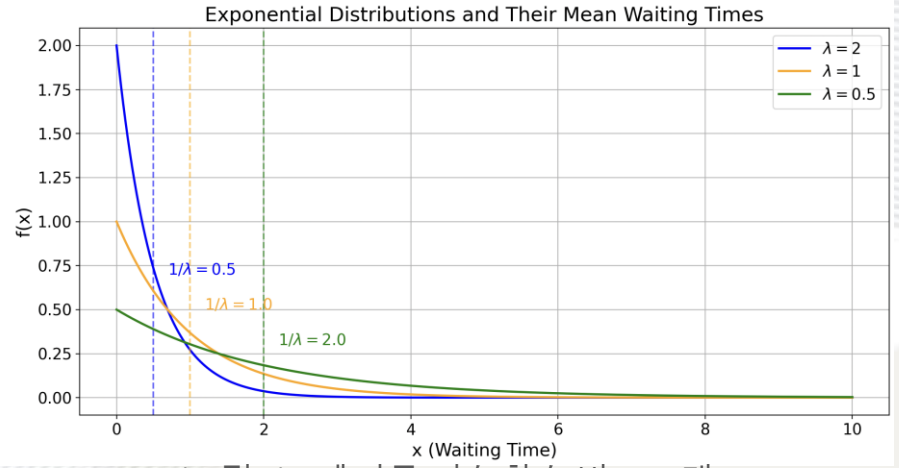
λ: 람다 (lambda)

단위 시간당 평균 사건 발생 횟수

〉 e.g., 분당 2번 도착 $\rightarrow \lambda = 2$

〉 λ가 클수록 사건이 자주 발생함 (평균 대기시간 짧음)

〉 평균 대기시간$= \frac{1}{\lambda}$



(그림 2) λ에 따른 지수 함수 비교 그래프

A. 지수확률변수 (Exponential Random Variable)

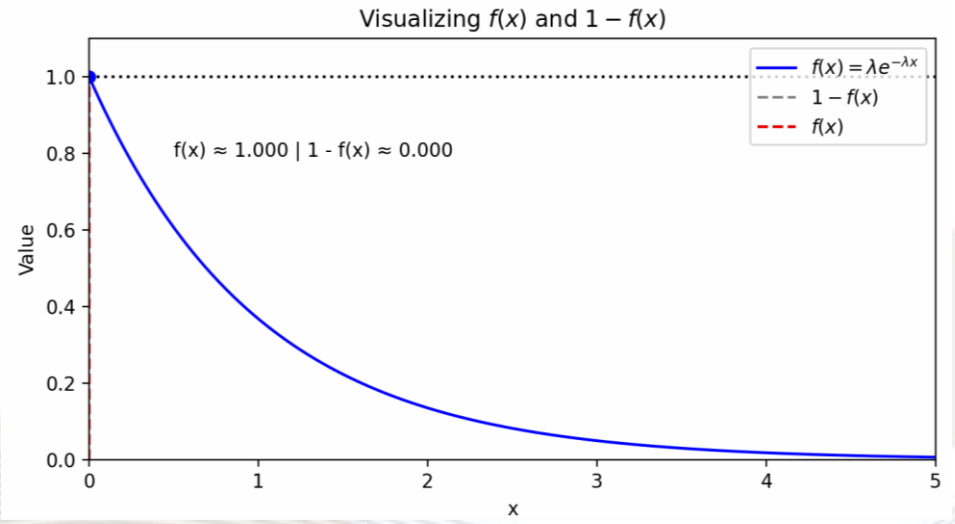
| 누적확률함수 (CDF, Cumulative Distribution Function): PDF를 아래의 값에서 부터 임의의 값 x까지 누적하여 적분한 값으로, $F(x)$로 표기함

- 의미: x초 이내에 사건이 발생할 확률

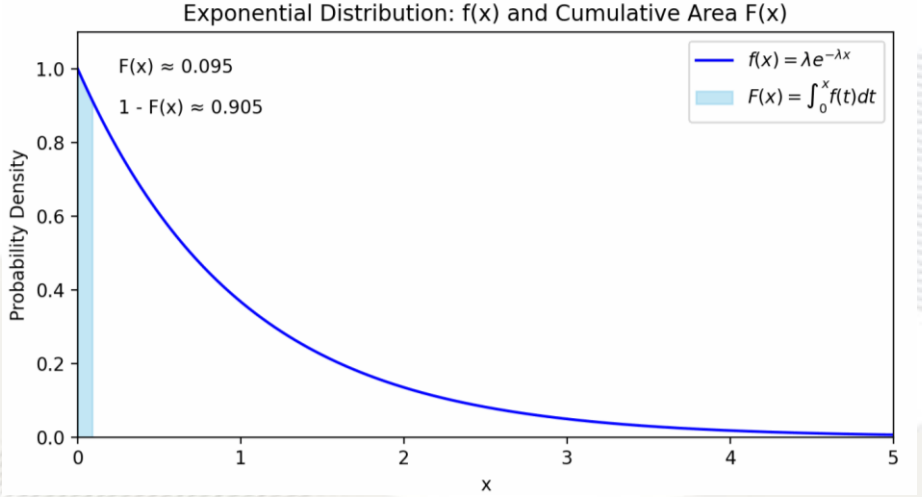
$$F(x) = \int_0^x \lambda e^{-\lambda x}\, dt = 1 - e^{-\lambda x}, x \geq 0$$

| 지수확률변수의 수학적 해석

- 특정 순간 사건이 발생할 확률밀도: $f(x)$
- 사건이 그 전에 발생했을 확률: $F(x)$
- 사건의 발생을 기다리는 동안 아무 일도 안 일어났을 확률: $P(X > x) = 1 - F(x)$



(그림 3) 지수분포의 확률밀도함수 변화 시각화



(그림 4) 지수분포의 누적확률함수 변화 시각화

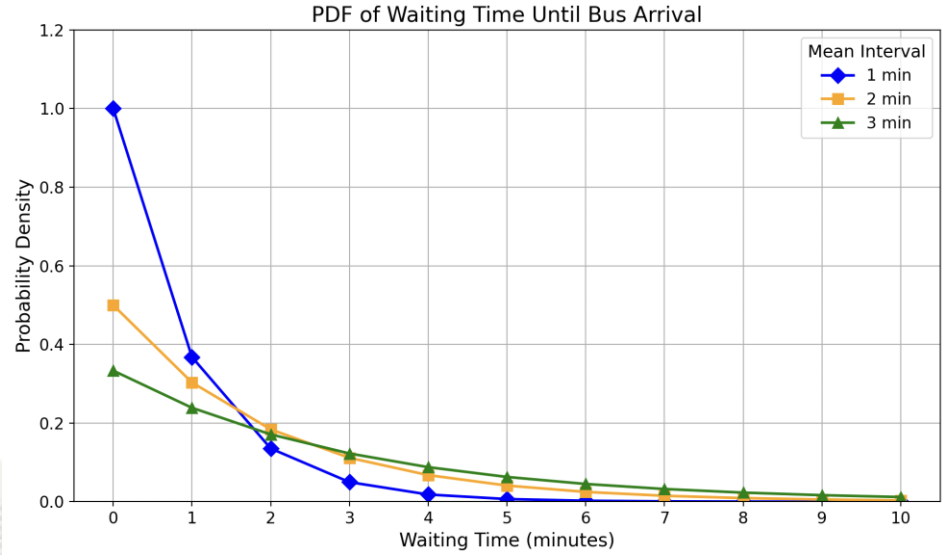
A. 지수확률변수 (Exponential Random Variable)

예제 2-1

버스의 도착간격이 **평균 1분, 2분 혹은 3분 간격으로 지수함수분포를 따라서 도착한다고 가정**할 때, 고객이 버스정류장에 가서 차가 올 때까지 기다려야 할 시간이 1분에서 10분일 경우에 대한 확률밀도함수를 각각 구하여 그래프로 그려라

문제 2-1

예제 2-1의 그래프에서 1분 및 4분일 사건의 확률밀도함수에 대하여 세 가지 파라미터에 대해서 PDF의 크기의 순서가 바뀐 것을 알 수 있다. 그 이유에 대한 물리적 의미는 무엇인가?



(그림 5) 고객이 기다려야 할 시간에 대한 확률밀도함수

- 도착간격의 평균이 작으면(λ가 크면), 사건이 더 빨리 발생할 확률이 높음
  → 앞쪽(PDF 초반)이 큼
- 하지만 감소 속도도 빠름
  → 시간이 조금만 지나도 확률이 빠르게 감소
- 반대로 도착간격의 평균이 크면(λ가 작으면), 초반 확률은 작지만 천천히 감소
  → 뒤쪽 (PDF 후반) 값은 더 큼

A. 정규확률변수 (Normally Distributed Random Variable)

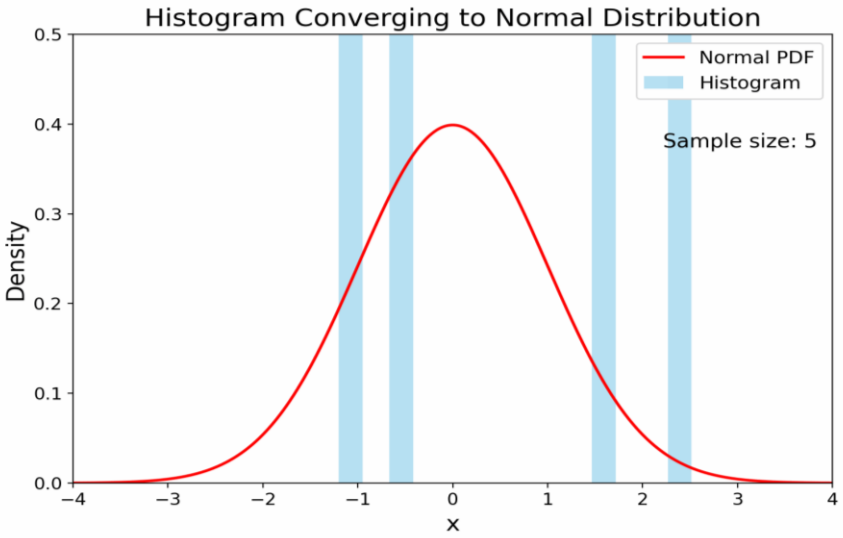
| 정의: 평균 μ, 분산 σ^2를 가지며 다음의 확률밀도함수를 따르는 연속확률변수

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty$$

| 정규분포 표기법: $X \sim N(\mu, \sigma^2)$

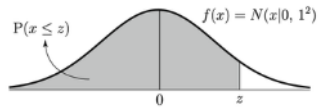
| 특징: 충분히 많은 샘플에서 나타나는 대칭적 분포를 가지며, 표준정규분포로의 변환 가능성이 존재함

- 많은 양의 샘플을 측정하고 관측하면, 평균을 중심으로 좌우 대칭적인 종(bell) 모양의 분포가 자연스럽게 나타남
- 평균과 분산이 다른 정규분포는 평균 0, 분산 1의 표준정규분포로 변환할 수 있음



(그림 6) 표본 수 증가에 따른 정규분포로의 수렴 시각화

Standard Normal Distribution table



z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319

A. 정규확률변수 (Normally Distributed Random Variable)

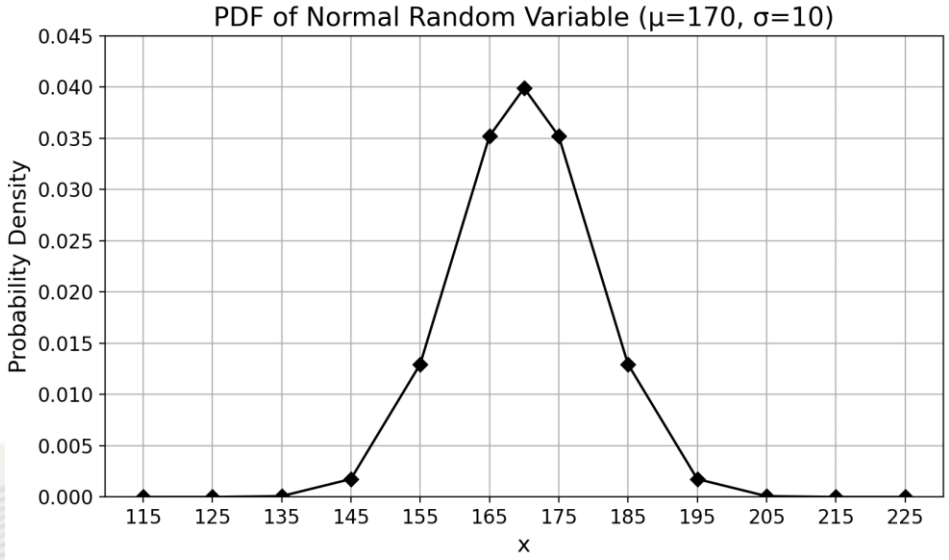
예제 2-2

평균이 170이고 표준편차가 10인 어떤 정규확률변수의 PDF를 $x = [115, 225]$에 대하여 구하라.

문제 2-2

예제 2-2의 그래프에서 어떤 확률의 값이 평균을 중심으로 대칭을 이루는 것의 물리적인 의미는 무엇인가?

$$\mu = 170, \sigma = 10 \rightarrow f(x) = \frac{1}{\sqrt{2\pi}10} e^{-\frac{(x-170)^2}{200}}, -115 \leq x \leq 225$$



(그림 7) 정규확률변수의 확률밀도함수

- 정규분포에서 평균은 가장 "중심적인" 값이며, 데이터가 그 평균을 중심으로 같은 확률로 퍼져 있음을 의미함
- 즉, 평균보다 a만큼 작은 값이 나올 확률과 평균보다 a만큼 큰 값이 나올 확률이 동일함을 의미함

A. 정규확률변수 (Normally Distributed Random Variable)

| 선형결합된 확률변수: 평균 μ, 분산 σ^2를 가지는 정규확률변수 X를 변수로 하는 함수 $Y = \alpha X + \beta$는 $Y \sim N(\alpha\mu + \beta, \alpha^2\sigma^2)$로 표기함

증명

누적확률분포의 정의에 의하여, X의 분포함수

$$F_Y(x) = P\{Y \le x\} = P\{\alpha X + \beta \le x\} = P\left\{X \le \frac{x-\beta}{\alpha}\right\}$$

$$F_Y(x) = P\left\{X \le \frac{x-\beta}{\alpha}\right\} = F_X\left(\frac{x-\beta}{\alpha}\right)$$

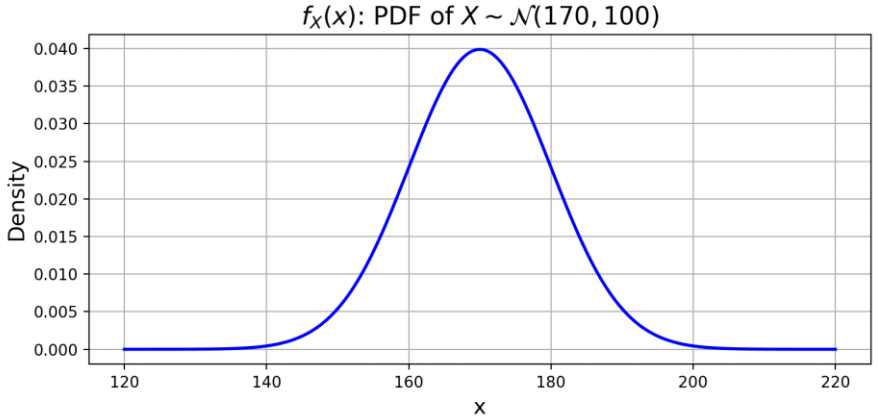
확률밀도함수의 정의에 의하여,

$$f_Y(x) = \frac{d}{dx}F_Y(x) = \frac{1}{\alpha}F'_X\left(\frac{x-\beta}{\alpha}\right) = \frac{1}{\alpha}f_X\left(\frac{x-\beta}{\alpha}\right)$$

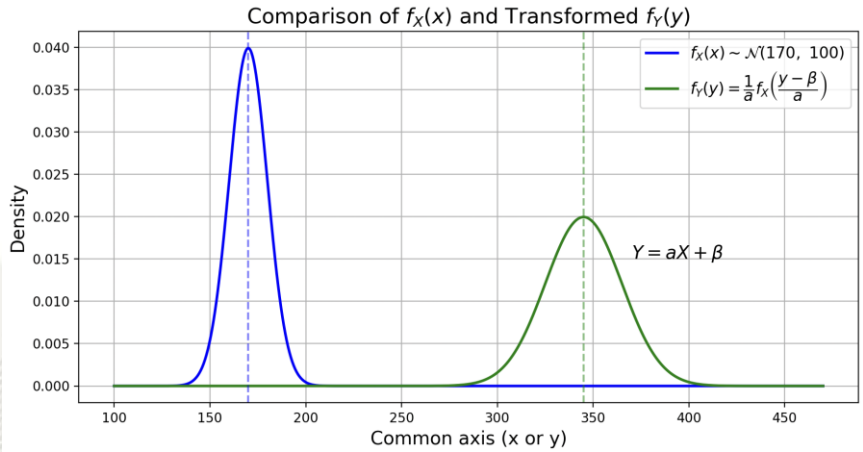
따라서,

$$f_Y(x) = \frac{1}{\sqrt{2\pi}\alpha\sigma}e^{-\frac{(x-(\alpha\mu+\beta))^2}{2\alpha^2\sigma^2}}$$

평균이 $\alpha\mu + \beta$, 분산이 $\alpha^2\sigma^2$인 정규분포를 따르는 확률변수



(그림 8) 정규분포 $f(x)$ 그래프



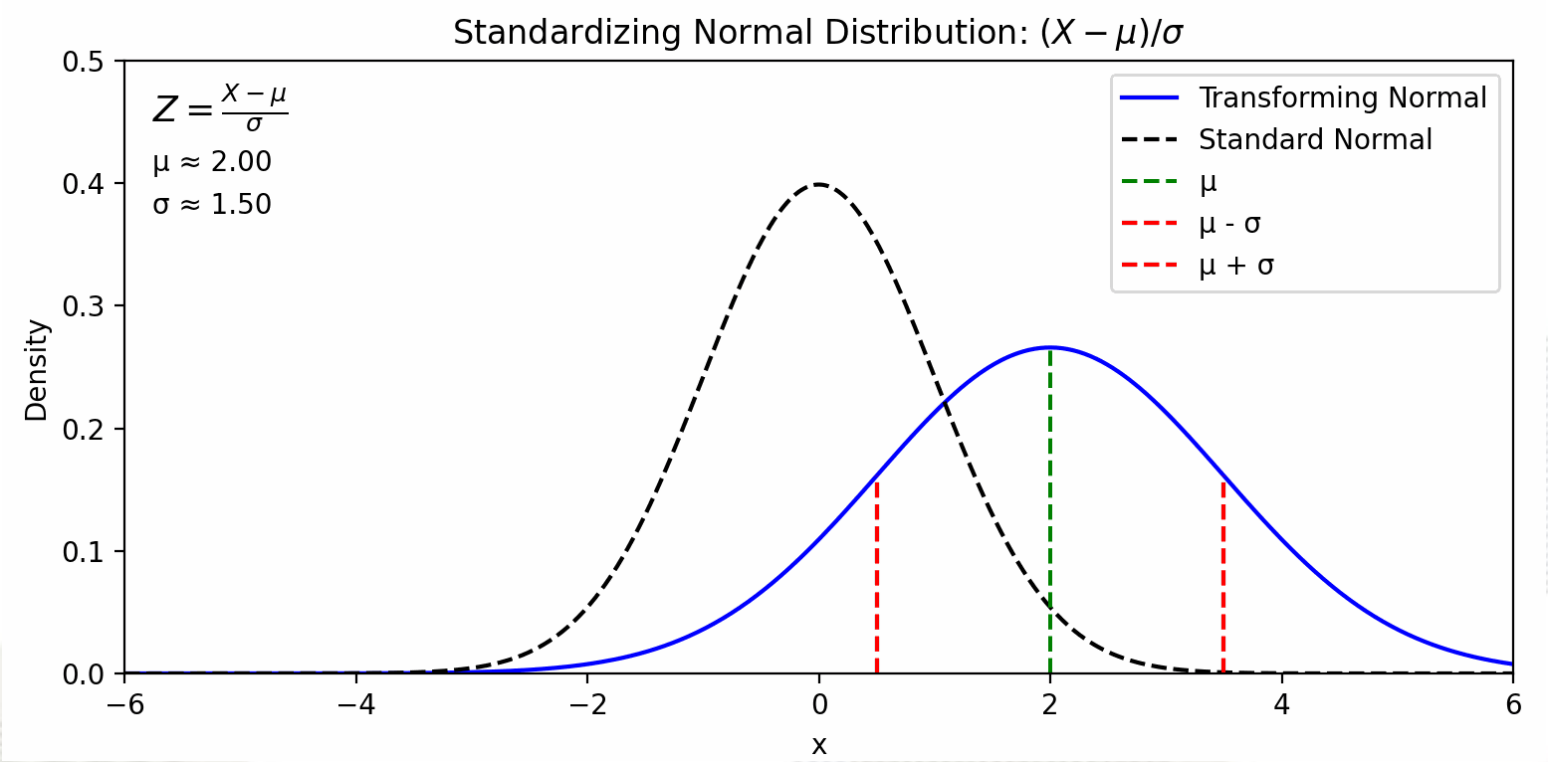
(그림 9) 정규분포 $f(x)$ 및 $f(y)$ 그래프 비교

A. 정규확률변수 (Normally Distributed Random Variable)

| 표준정규분포: 확률변수 Z가 평균 0, 분산 1을 갖고 다음과 같은 확률밀도함수를 따를 때, Z는 표준정규분포를 따른다고 함

$$Z \sim N(0,1),\ f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}, -\infty < z < \infty$$

$$Z = \frac{X-\mu}{\sigma} \leftrightarrow X = \mu + \sigma Z$$



(그림 10) 표준정규분포로의 변화 그래프

A. 정규확률변수 (Normally Distributed Random Variable)

| 누적분포함수 (CDF, Cumulative Distribution Function): 확률변수 X가 어떤 값 이하일 확률을 나타내는 함수

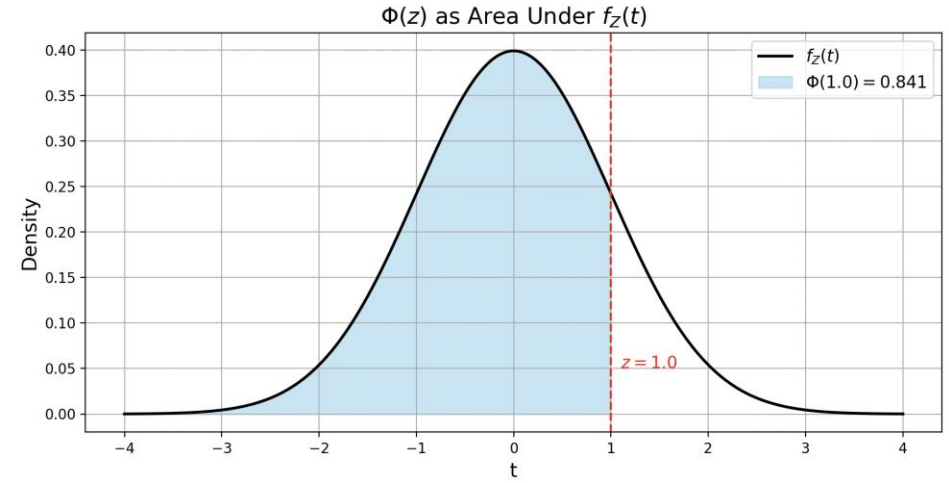
$$\Phi(x) = \Pr(X < x) = \int_{-\infty}^{x} f(x)\,dx$$

- $\Phi(0) = 0.5, \Phi(-x) = 1 - \Phi(x)$의 대칭성을 활용한 계산 수행 가능

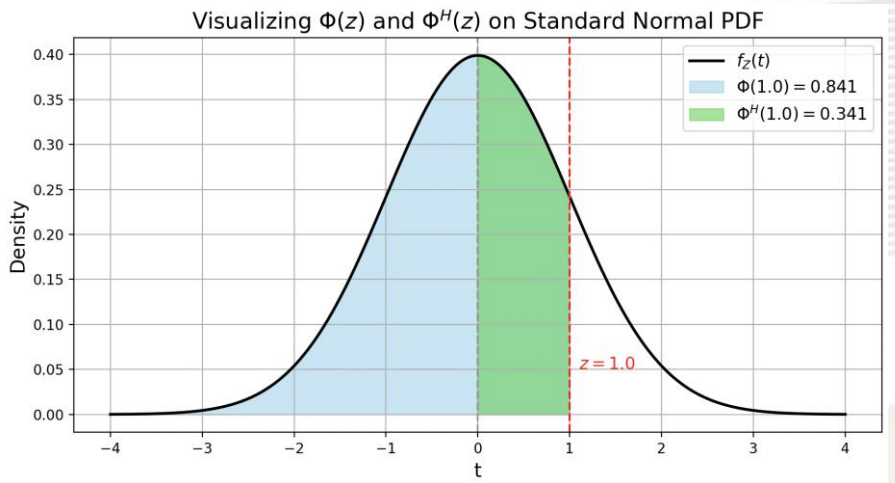
| 편측 누적분포함수 (One-Sided CDF): 확률변수 X가 0보다 크고 x보다 작을 확률을 나타내는 함수

$$\Phi^H(x) = \Pr(0 < X < x) = \int_{0}^{x} f(x)\,dx$$

- 일반 누적분포함수에서 $\Phi^H(x) = \Phi(x) - 0.5$ 형태로 계산 가능



(그림 11) 표준정규분포에서의 $\Phi(1)$ 값



(그림 12) 표준정규분포에서의 $\Phi(1)$ 및 $\Phi^H(1)$ 값

A. 정규확률변수 (Normally Distributed Random Variable)

예제 2-3

어느 마을에서 그 마을에 사는 어른과 아이를 포함한 사람의 평균 키는 160cm이고, 표준 편차는 10cm일 때 어느 한 사람을 세워 놓고 키를 재었을 때 그의 키가 190cm보다 더 클 확률은 얼마인가? 만약 다른 값은 같고 표준편차가 30이면 그 확률은 어떻게 되는가?

문제 2-3

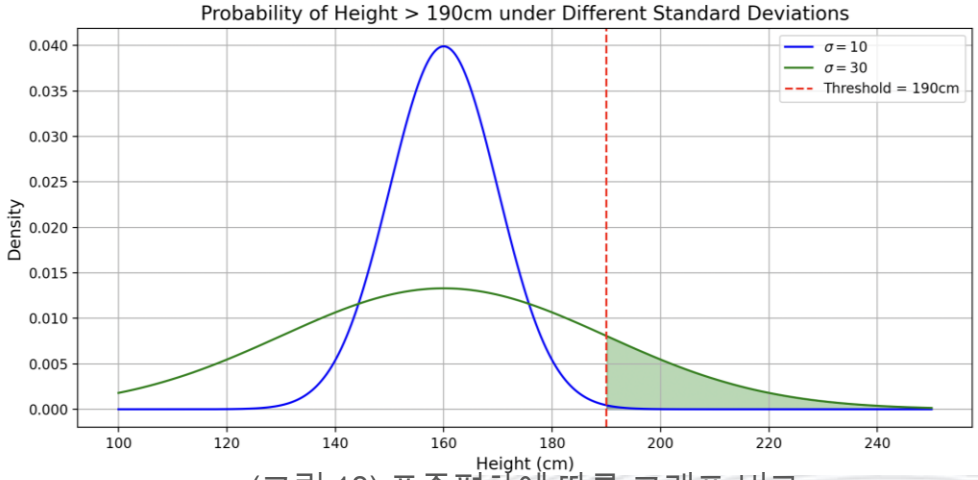
위의 두 결과가 의미하는 것은 무엇인가?

표준편차 10cm인 경우

- $P\{X > 190\} = P\left\{\frac{X-160}{10} > \frac{190-160}{10}\right\}$
  $= P\{Z > 3\} = 1 - \Phi(3)$
  $= 1 - 0.9987 = 0.0013$

표준편차 30cm인 경우

- $P\{X > 190\} = P\left\{\frac{X-160}{30} > \frac{190-160}{30}\right\}$
  $= P\{Z > 1\} = 1 - \Phi(1)$
  $= 1 - 0.8413 = 0.1587$



(그림 13) 표준편차에 따른 그래프 비교

- 표준편차가 작을수록 ($\sigma = 10$), 분포가 평균 160을 중심으로 더 좁고 집중되어 있음
  → 190cm는 매우 극단적인 값 → 확률 매우 작음(0.135%)
- 표준편차가 클수록 ($\sigma = 30$), 분포가 넓게 퍼짐
  → 190cm는 덜 극단적인 값 → 상대적으로 확률 큼 (15.87%)

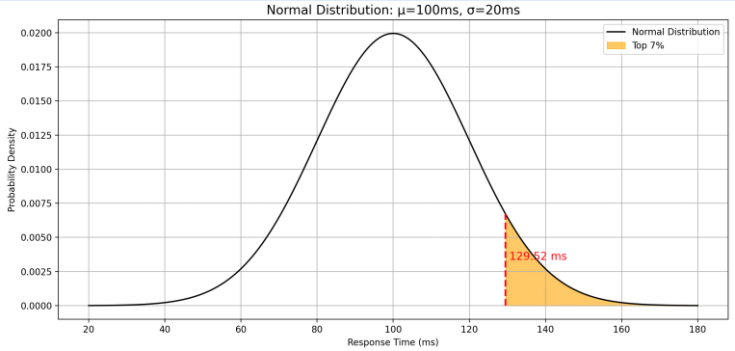
A. 정규확률변수 (Normally Distributed Random Variable)

운영체제의 프로세스 처리시간

CPU가 처리하는 프로세스의 평균 처리시간은 100ms, 표준편차는 20ms이며, 정규분포를 따른다.

활용1 - 처리시간 기반 작업 분류 기준 설정

- 시스템에서는 처리시간이 상위 **7%에 해당하는 작업을 '과도하게 오래 걸리는 작업'으로 정의**하고, 스케줄링 선점 대상이 될 수 있음
- 표준정규분포를 통해 상위 7%에 해당하는 $z \approx 1.475 \rightarrow T = 100 + 1.475 \times 20 = 129.5\text{ms}$
- **정규분포를 기반으로 이상 작업 자동 분류 및 스케줄링 정책 설정 가능**



(그림 14) 성공 횟수에 따른 확률질량함수 변화

활용2 - 확률 기반 시스템 안정성 진단

- 한 프로세스가 130ms 이상 걸릴 확률을 도출함으로써 일반적으로 발생할 확률을 구함
- 정규화: $z = \frac{130-100}{20} = 1.5$
- $P(X > 130) = P(Z > 1.5) \approx 0.0668$
- 운영 시스템에서 일정 시간 동안 long task 비율이 15% 이상으로 증가했다면?

→ CPU 병목 또는 시스템 내 리소스 과점 현상 발생 가능성 시사

→ 시스템은 해당 확률의 변화 추이를 모니터링함으로써 부하 상황을 사전에 감지/자원 재분배, 우선순위 조정 등 대응 가능

A. 이산확률변수 (Discrete Random Variable)

| 정의: 취할 수 있는 값의 집합이 유한하거나 셀 수 있을 만큼 무한한 확률변수

| 특징: 확률은 개별 값에 대해 정의되며, 확률질량함수(PMF, Probability Mass Function) $p(x)$를 통해 각 값에 대한 확률을 직접 정의함

- 특정 값 x에 대해: $P(X = x) = p(x)$
- 확률질량함수 $p(x)$의 조건: $p(x) \geq 0, \sum_x p(x) = 1$

AI센터 1층 투썸플레이스에서 커피 추첨 이벤트가 열리고 있다.
고객은 1인 1회 추첨 기회를 가지며, 추첨 성공 확률은 20%라고 하자.
이 때, 고객이 **2명 이하로만 성공**할 확률은?

- $P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2)$

$$= \binom{5}{0}(0.2)^0(0.8)^5 + \binom{5}{1}(0.2)^1(0.8)^4 + + \binom{5}{2}(0.2)^2(0.8)^3$$

$$\approx 0.32768 + 0.4096 + 0.2048 = 0.94208$$

- 약 94.2% 확률로 5명 중 2명 이하만 당첨된다!

A. 이산확률변수 (Discrete Random Variable)

| 대표적인 이산확률변수: 이항확률변수, 기하확률변수, 포아송확률변수

대기행렬을 공부하기 위해서는 왜 하필 이항, 기하, 포아송 확률변수를 알아야 할까?!

대기행렬 (Queueing Theory)에서 해결하고자 하는 문제

- "**100명의 고객**에게 설문을 보낼 때, 응답하는 사람이 **20명 이하**일 확률은?"
- "**첫 번째 고객이 도착하기까지 몇 번의 실패**가 있을까?"
- "**일정 시간 동안 고객이 몇 명 도착**할까?"

이항분포는 **고정된 시행 횟수 안에서의 성공 횟수에 대한 모델링을 수행함**

기하분포는 **처음 성공이 나타날 때까지 걸리는 시행 횟수에 대한 모델링을 수행함**

포아송분포는 **단위 시간당 도착 고객 수**를 확률적으로 표현할 때 사용됨

반복적 성공/실패 분석에는 이항분포 / 첫 성공까지의 시행 수 분석에는 기하분포 / 단위 시간 내 사건 발생 수 분석에는 포아송분포가 핵심적으로 활용됨

A. 이항확률변수 (Binomial Random Variable)

| 정의: 동일한 조건에서 독립적으로 반복되는 베르누이 시행을 n 번 수행했을 때, 성공 횟수(k)를 확률변수로 표현하며 다음의 확률질량함수를 따르는 이산확률변수

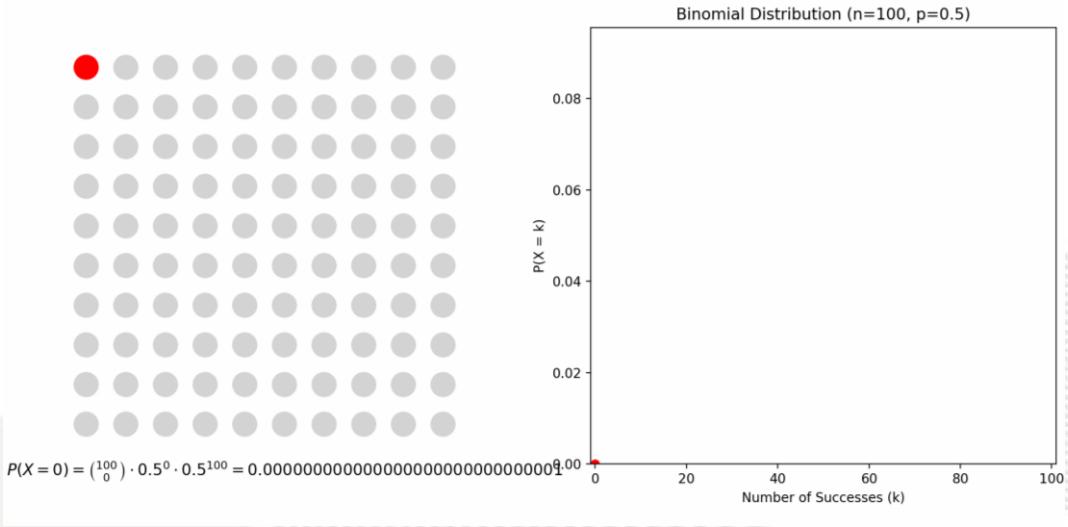
$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

| 이항분포 표기법: $X \sim Binomial(n, p)$

베르누이 시행 (Bernoulli Trial)

- 1회의 실험 결과가 '성공' 또는 '실패' 둘 중 하나로만 나뉘는 시행
- 성공할 확률: p, 실패할 확률: $1-p$
- 결과를 확률변수 X로 나타내면

$$X = \begin{cases} 1\ (\text{성공}) \\ 0\ (\text{실패}) \end{cases}$$



(그림 15) 성공 횟수에 따른 확률질량함수 변화

A. 이항확률변수 (Binomial Random Variable)

예제 2-4

앞면과 뒷면이 나올 확률이 같은 **500원짜리 동전을 4개 이용**하여 윷놀이를 한다고 하자. 이때 도, 개, 걸, 윷 그리고 모 중에서 나올 확률이 높은 순서대로 나열하라.

- 동전에서 앞면과 뒷면이 나올 확률은 같으므로, $p = 1 - p = 0.5$
- 도 (앞:3, 뒤: 1) $\rightarrow P(\text{도}) = \binom{4}{3}\frac{1}{2}^4 = \frac{1}{4}$
- 개 (앞:2, 뒤: 2) $\rightarrow P(\text{개}) = \binom{4}{2}\frac{1}{2}^4 = \frac{3}{8}$
- 걸 (앞:1, 뒤: 3) $\rightarrow P(\text{걸}) = \binom{4}{1}\frac{1}{2}^4 = \frac{1}{4}$
- 윷 (앞:0, 뒤: 4) $\rightarrow P(\text{윷}) = \binom{4}{0}\frac{1}{2}^4 = \frac{1}{16}$
- 모 (앞:4, 뒤: 0) $\rightarrow P(\text{모}) = \binom{4}{4}\frac{1}{2}^4 = \frac{1}{16}$

문제 2-4

아마추어 야구선수 나승엽 선수는 동네의 야구연습장에서 공을 칠 때 **평균타율이 3할**이라고 한다. 그가 어느 날 거기에 가서 **10개의 공을 쳤을 때 3번 안타를 칠 확률**은 얼마인가?

- 평균타율이 3할이라고 하였으므로, $p = 0.3, 1 - p = 0.7$
- $n = 10, k = 3$ 이므로, 구하고자 하는 값은 $\rightarrow P(X = k) = \binom{10}{3}0.3^3 0.7^7$

A. 기하확률변수 (Geometric Random Variable)

| 정의: 일정한 성공확률(p)을 가지는 베르누이 시행을 반복할 때, 첫번째 성공이 나타나는 시행횟수(n)를 확률변수 X로 표현하며, 다음의 확률질량함수를 따르는 이산확률변수

$$P(X = n) = (1-p)^{n-1}p, n = 1,2,3, \dots$$

| 기하분포 표기법: $X \sim Geo(p)$

기하확률변수의 PMF 도출 과정

- F_i: i번째 시행에서 실패한 사건
- S_n: n번째 시행에서 처음으로 성공할 사건

- 사건 $X = n$: 앞선 $n-1$은 모두 실패하고, n번째에 처음 성공하는 사건

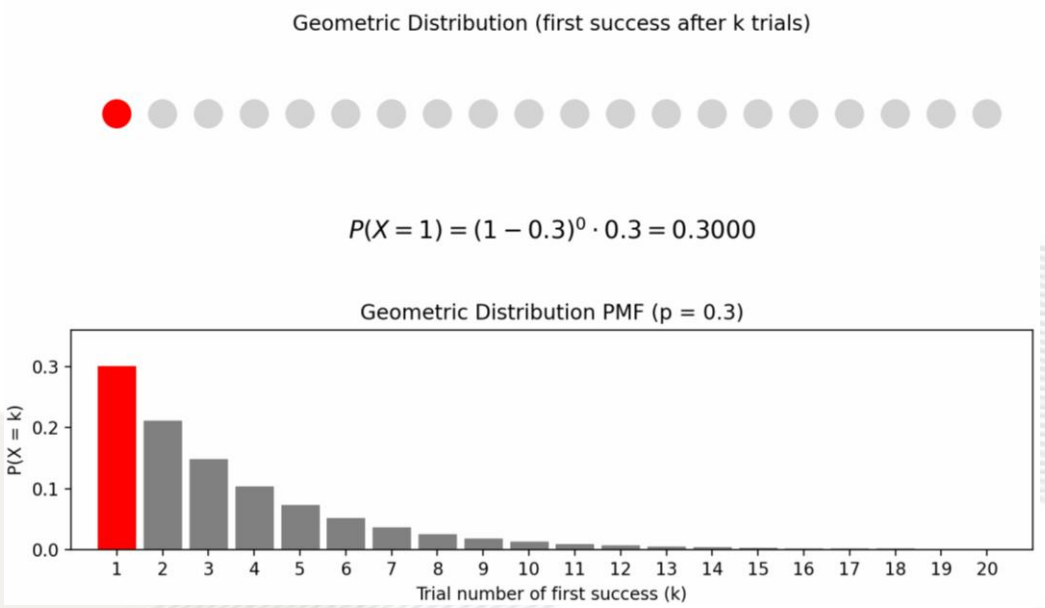
$$P(X = n) = P\left(\bigcap_{i=1}^{n-1} F_i \cap S_n\right)$$

- 각 시행은 서로 독립이므로,

$$P(X = n) = \prod_{i=1}^{n-1} P(F_i) \cdot P(S_i)$$

- 확률을 대입해보면..

$$P(X = n) = (1-p)^{n-1}p$$



(그림 16) 첫번째 성공이 나타나는 시행횟수(n)에 따른 확률질량함수 변화

A. 기하확률변수 (Geometric Random Variable)

예제 2-5

어떤 남자가 선을 보아서 결혼에 성공할 확률이 30%라고 한다. 그렇다면 이 남자가 **세 번 만에 결혼에 성공**할 확률은 얼마인가?

- $p = 0.3,\ n = 3$이므로, 구하고자 하는 값은 $\rightarrow P(X = 3) = 0.7^2 \times 0.3 = 0.147$

예제 2-6

어떤 남자가 선을 보아서 결혼에 성공할 확률이 1%라고 한다. 그렇다면 이 남자가 **언젠가는 결혼에 성공**할 확률은 얼마일까?

- $P(X < \infty) = \sum_{n=1}^{\infty}(1-p)^{n-1}p = \frac{p}{1-(1-p)} = 1$

문제 2-5

농구선수는 센터라인에서 바스켓을 향하여 공을 던졌을 때 골이 바스켓 안으로 들어갈 확률이 10%라고 한다. 이때 이 사람이 공을 던져서 골을 넣기 위하여 **최소한 5번 이상 던져야 확률은?**

- 최초의 성공이 5번째 이상에 발생할 확률은 1~4번째까지 모두 실패할 확률과 동일함
- $p = 0.1, 1-p = 0.9$이므로, 구하고자 하는 값은,

$$\rightarrow P(X \geq 5) = (1-p)^4 = 0.9^4 = 0.6561$$

A. 기하확률변수 (Geometric Random Variable)

웹 요청 재시도 성공률 기반 정책 수립

웹 API 요청은 통신 오류나 네트워크 불안정으로 인해 실패할 수 있으며, 서비스 운영자는 사용자 경험 보장을 위해 최소 95% 이상의 요청 성공률을 보장하려고 한다. 한 번의 요청이 성공할 확률이 $p = 0.7$일 때, 최대 3번까지 요청을 반복하는 정책이 이 기준을 만족하는가?

활용

- 첫 성공이 발생하는 시도 횟수를 확률변수 X로 모델링을 수행하면,

$$X \sim Geo(p = 0.7)$$

- 누적 성공 확률을 계산하면 (3번 이내 성공 확인),

$$\begin{aligned} P(X \leq 3) &= P(1) + P(2) + P(3) \\ &= 0.7 + (0.3)(0.7) + (0.3)^2(0.7) \\ &= 0.7 + 0.21 + 0.063 = 0.973 \end{aligned}$$

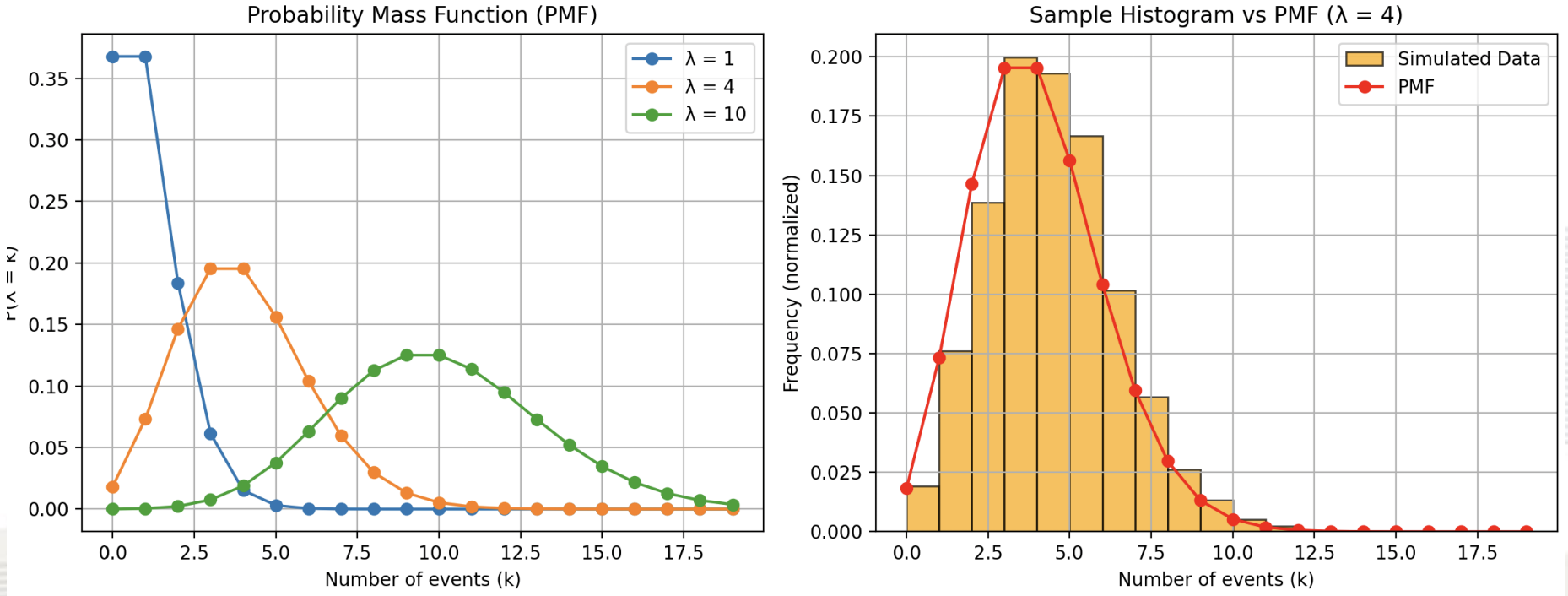
- 따라서, 3번 이내에 성공할 확률이 약 97.3%이므로, “최대 3회까지 재시도” 정책은 95% 이상 성공률 보장 요구 조건을 만족함 (성공률 보장과 동시에 과도한 시도 방지 가능)

A. 포아송확률변수 (Poisson Random Variable)

| 정의: 단위시간 당 평균 사건 발생 횟수가 λ일 때, 주어진 시간 구간 내 사건이 k번 발생할 확률을 나타내며, 다음의 확률질량함수를 따르는 이산확률변수

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, 2, \ldots$$

| 기하분포 표기법: $X \sim Poisson(\lambda)$



(그림 17) λ에 따른 포아송분포 그래프

A. 포아송확률변수 (Poisson Random Variable)

포아송 분포의 유도

가정 1. 임의의 시간구간 내 사건 발생은 서로 독립임

가정 2. 충분히 작은 시간구간 $\Delta t \to 0$ 내에서는 사건이 오직 한 번 발생하며, 확률은 $p = \lambda \Delta t$임
- 사건이 발생하지 않을 확률은 $q = 1 - p = 1 - \lambda \Delta t$이며, 이 사건은 베르누이 과정을 따름

$P(k, t)$: 임의의 시간구간 t 동안 k개의 사건이 일어날 확률

- Step 1. 시간 [0, $t + \Delta t$] 동안 사건이 전혀 발생하지 않을 확률
  - $P(0, t + \Delta t) = P\{(0, t)$간에 사건이 하나도 일어나지 않고 시간 $(t, t + \Delta t)$에 역시 사건이 하나도 일어나지 않음$\}$
  - = $P((0, t)$간에 사건이 하나도 일어나지 않음$) \times P((t, t + \Delta t)$ 간에 사건이 하나도 일어나지 않음$)$
  - $P(0, t + \Delta t) = P(0, t)q = P(0, t)(1 - \lambda \Delta t)$
  - $\frac{P(0, t+\Delta t) - P(0,t)}{\Delta t} = -\lambda P(0, t), t \geq 0$
  - $\frac{d}{dt} P(0, t) = -\lambda P(0, t)$
  - $Log P(0, t) = -\lambda t + C$
  - $P(0, t) = e^{-\lambda t + C} = A e^{-\lambda t + C}$ $(A = e^C)$
  - $P(0, t) = e^{-\lambda t}$

A. 포아송확률변수 (Poisson Random Variable)

포아송 분포의 유도

가정 1. 임의의 시간구간 내 사건 발생은 서로 독립임

가정 2. 충분히 작은 시간구간 $\Delta t \to 0$ 내에서는 사건이 오직 한 번 발생하며, 확률은 $p = \lambda \Delta t$임
- 사건이 발생하지 않을 확률은 $q = 1 - p = 1 - \lambda \Delta t$이며, 이 사건은 베르누이 과정을 따름

$P(k,t)$: 임의의 시간구간 t 동안 k개의 사건이 일어날 확률

- Step 2. 시간 $[0, t + \Delta t]$ 동안 k개의 사건이 발생할 확률
  - $P(k, t+\Delta t) = P(k,t)q + P(k-1,t)p = P(k,t)(1-\lambda\Delta t) + P(k-1,t)\lambda\Delta t$
  - $\frac{P(k,\, t+\Delta t) - P(k,t)}{\Delta t} = -\lambda P(k,t) + \lambda P(k-1,t)$
  - $\frac{d}{dt}P(k,t) + \lambda P(k,t) = \lambda P(k-1,t)$
  - $k = 1$인 경우
    - $\frac{d}{dt}P(1,t) + \lambda P(1,t) = \lambda P(0,t)$
    - 동차해: $\frac{d}{dt}P_h(1,t) + \lambda P_h(1,t) = 0, P_h(1,t) = Ce^{-\lambda t}$
      - $P(1,0) = 0$이므로 $C = 0$
    - 특별해: $\frac{d}{dt}P_p(1,t) + \lambda P_p(1,t) = \lambda P(0,t) \to \frac{d}{dt}P_p(1,t) + \lambda P_p(1,t) = \lambda e^{-\lambda t}$
    - 적분 적용 시, $P_p(1,t) = (\lambda t + C)e^{-\lambda t} \to P(1,t) = \lambda t e^{-\lambda t}$

k	$P(k,t)$
0	$e^{-\lambda t}$
1	$\lambda t e^{-\lambda t}$
2	$\frac{(\lambda t)^2}{2} e^{-\lambda t}$
3	$\frac{(\lambda t)^3}{6} e^{-\lambda t}$

$$P(k,t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

A. 포아송확률변수 (Poisson Random Variable)

예제 2-7

AI 센터 앞을 지나가는 공항버스의 수는 1분에 평균 5대인 포아송 분포를 따른다고 하자. 어느 날 건물 앞에 서서 1시간 동안 지나가는 버스의 수를 관찰하였을 때 지나간 버스의 수가 1,000대 이상일 확률은 얼마인가?

- 1분을 기준으로 $\lambda = 5$이므로, 1시간을 기준으로 $\lambda = 300$
- 구하고자 하는 것은 1시간 동안 지나간 버스의 수가 1,000대 이상일 확률이기 때문에, 전체 확률 1에서 버스가 지나갈 확률이 999대 이하일 확률을 빼면 됨

$$P(\text{1시간 동안 지나간 버스가 1,000대 이상}) = 1 - \sum_{k=0}^{999} \frac{300^k}{k!} e^{-300}$$

A. 포아송확률변수 (Poisson Random Variable)

문제 2-6

주변에서 포아송분포를 따르는 확률변수에는 어떠한 것이 있는지 찾아보고 현실에 맞게 모형화해 보자.

예시	설명	모형화
콜센터	1시간 동안 몇 명의 고객이 전화하는가?	X: 시간 t 동안 전화 수
응급실	1일 동안 응급환자가 몇 명 오는가?	X: 하루 응급환자 수
도서관	10분 동안 입장하는 사람 수	X: 단위 시간 입장 인원
인터넷 서버	1초 동안 서버에 도달한 요청 수	X: 초당 패킷 수

어떤 확률변수가 포아송 분포를 따르기 위한 조건
- 사건 발생은 독립적
- 단위 시간 당 평균 발생 횟수 (λ)가 일정함
- 짧은 시간 간격에는 사건이 한 번만 발생함
- 동시에 여러 사건이 발생할 확률은 0에 수렴함

저녁 8시~9시 사이, AI센터 편의점에는 평균적으로 15명이 계산을 하러 온다. 이때, 오늘 저녁 8~9시에 손님이 정확히 20명 올 확률은?

- $\lambda = 15$이며 확률변수 X는 '1시간 동안 계산대에 온 손님 수'

$$X \sim Poisson(15) \rightarrow P(X = k) = \frac{15^k}{k!} e^{-15}$$

- 따라서, 구하고자 하는 값은

$$P(X = 20) = \frac{15^{20}}{20!} e^{-15}$$

A. 포아송확률변수 (Poisson Random Variable)

DDoS 탐지를 위한 임계치 설정

기업들은 자신들의 서비스 특성과 보안 민감도에 따라 '얼마나 많은 요청이면 비정상인가'를 정량적으로 결정하고 싶어한다. 평균적으로 초당 100건의 요청이 발생하는 어느 기업은 1초 내 요청 수가 k건 이상일 확률이 1% 이하라면 DDoS 공격이 발생하였다고 판단하려고 한다.

활용

- 평균 초당 요청 수는 100건임에 따라, $\lambda = 100$
- '1초 내 요청 수가 k건 이상일 확률이 1% 이하'이면 DDoS 공격으로 판단하고자 함에 따라, 다음의 식을 만족하는 최소 정수 k를 찾고자 함

$$P(X \geq k) \leq 0.01$$

- $P(X \geq k) = 1 - P(X < k) = 1 - P(X \leq k-1)$이므로,
- $1 - P(X \leq k-1) \leq 0.01 \Rightarrow P(X \leq k-1) \geq 0.99$
- 따라서,

$$\sum_{i=0}^{k-1} \frac{100^i}{i!} e^{-100} \geq 0.99$$

- $P(X \leq 122) \approx 0.99003 \Rightarrow\ 1 - P(X \geq 123) \approx 1 - 0.99003 = 0.00997 \leq 0.01$
- 즉, $k = 123$가 최소 조건을 만족하는 이상 트래픽 탐지 임계값임을 확인할 수 있음

A. 포아송확률변수 (Poisson Random Variable)

클라우드 환경에서의 로그 차단 이벤트 이상 탐지

AWS 방화벽에서는 평소 평균 2건/분의 차단 이벤트가 발생한다. 이벤트 수집기 이상 여부를 판단하기 위하여 차단 이벤트가 0건일 확률이 3% 이하가 되는 최소 시간을 구하여라

활용

- $P(k,t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$ 의 수식을 갖는 포아송 분포에서, $\lambda = 2, k = 0$임을 알 수 있음

$$P(0,x) = \frac{(2t)^0}{0!} e^{-2x} \leq 0.03$$

- 양번에 자연로그를 취함으로써 풀어보면,

$$e^{-2x} \leq 0.03$$
$$-2x \leq \ln(0.03) \approx -3.5066$$
$$x \geq \frac{3.5066}{2} \approx 1.7533$$

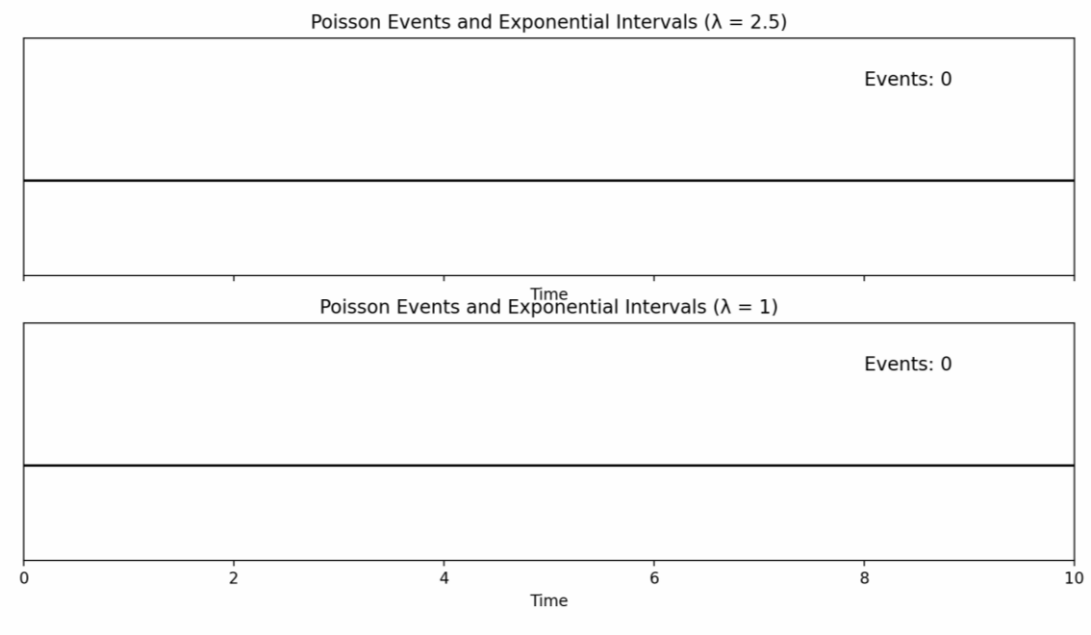
- 따라서, '2분 동안 차단 이벤트가 0건 발생'일 경우 이상 상태로 판단하는 정책 설정 가능

A. 포아송분포와 지수분포

어떤 사건이 무작위로 일어날 때...
- 지수분포: 사건이 일어나는 시간 간격을 모델링
- 포아송분포: 일정 시간 내에 사건이 k번 발생할 확률을 모델링

→ **시간 간격과 횟수는 서로 관련되어 있지 않을까?!**

→ e.g., 시간 간격이 짧을수록 사건 횟수는 많아짐



(그림 18) 사건 건 시간 간격에 따른 단위시간 당 사건 횟수 비교

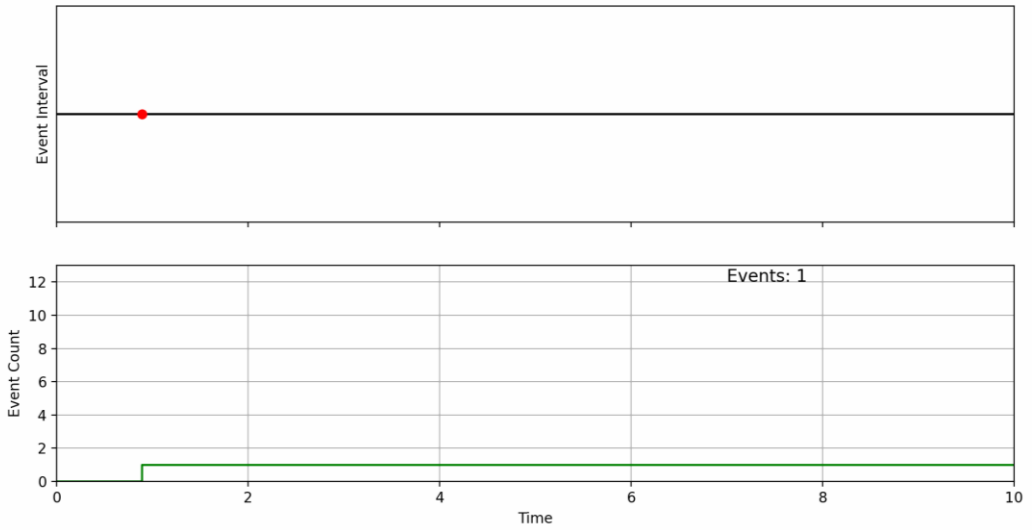
사건 간 시간 간격이 지수분포를 따르며, 이 간격들이 쌓여가는 속도에 따라 주어진 시간 내 사건 횟수는 포아송 분포를 따름

A. 포아송분포와 지수분포

| 지수분포로 생성된 사건 간 시간 간격을 누적하면 단위시간 내 발생 횟수가 포아송 분포를 따름

$$X_1, X_2, \cdots \sim Exp(\lambda) \Rightarrow N(t) = max\left\{k: \sum_{i=1}^{k} X_i \leq t\right\} \sim Poisson(\lambda t)$$

Exponential Intervals → Poisson Event Count



(그림 19) 사건 건 시간 간격 누적에 따른 단위시간 당 사건 횟수

대기행렬 (Queueing Theory)에서의 지수분포/포아송분포

〉 M/M/1 큐 시스템에서는 다음의 조건을 가짐
- 첫 번째 M: 도착 간격이 지수분포

→ 도착 횟수는 포아송 분포
- 두 번째 M: 서비스 시간이 지수분포
- 1: 서버 1대

구성 요소	분포	설명
고객 도착 간격 T	지수분포$\sim Exp(\lambda)$	고객이 도착하는 간격
단위시간 내 도착 수 $N(t)$	포아송분포$\sim Poisson(\lambda t)$	도착 횟수 (도착 간격 누적값으로부터 파생)
서비스 시간	지수분포$\sim Exp(\mu)$	각 고객 처리시간

A. 확률생성함수 (PGF, Probability Generating Function)

| 정의: 확률변수 X에 대한 확률값들을 하나의 함수로 요약한 것으로, 이산확률변수 X의 확률질량함수가 $p(k)$일 때, 다음과 같이 표현됨

$$G(z) = E[z^X] = \sum_{k=0}^{\infty} p(k) z^k$$

| 특징: $G(z)$은 z 에 대한 함수이며, 확률변수 X의 정보를 담고 있음에 따라, X의 값들을 전부 다루지 않고, z라는 도구를 통해 쉽게 원하는 값을 유도함

| 활용: $p(k) = \frac{G^{(k)}(0)}{k!}$ 을 통해 확률질량함수 도출 가능

| 예시:
- 상수확률변수 $X = c$: $G(z) = z^c$
- 포아송 확률변수: $G(z) = e^{\lambda(z-1)}$

예제 2-8

어떤 확률변수 X의 확률생성함수 $G(z)$가 아래와 같이 주어진다고 할 때 이 확률변수의 확률질량함수를 구하고 이 확률변수가 무슨 분포를 따르는지 논하라: e^{z-1}

- $G(z) = e^{z-1} = e^{-1} \sum_{n=0}^{\infty} \frac{z^n}{n!} = \sum_{n=0}^{\infty} \frac{e^{-1}}{n!} z^n$ $\left(e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!} \text{ 이므로}\right)$
- $p(n) = \frac{e^{-1}}{n!}$이므로, 확률변수 X는 $\lambda = 1$인 포아송 분포를 따름

기대치

Ⅲ

A. 확률변수의 기대값 (Expected value)

| 정의: 확률변수가 취할 수 있는 각각의 값에 그 값이 발생할 확률을 곱한 후 모두 더하여 계산된 확률변수의 평균값

| 표기법: $\mu_X, \mu, E[X]$

| 공식:

- X 가 이산형인 경우: $\mu = E[X] = \sum_x xp(x)$
- X가 연속형인 경우: $\mu = E[X] = \int_{-\infty}^{\infty} xf(x)\,dx$

| 도출 방법: 1) 확률함수를 활용한 방법, 2) 모멘트생성함수를 활용한 방법

6면체 주사위의 값(1, 2, 3, 4, 5, 6)의 기대치는 얼마일까?

- $\frac{1+2+3+4+5+6}{6} = \frac{21}{6} = 3.5$
- $\frac{1+2+3+4+5+6}{6} = (1)\left(\frac{1}{6}\right) + (2)\left(\frac{1}{6}\right) + (3)\left(\frac{1}{6}\right) + (4)\left(\frac{1}{6}\right) + (5)\left(\frac{1}{6}\right) + (6)\left(\frac{1}{6}\right)$

A. 확률함수를 활용한 기대치 도출 – 연속확률변수의 기대치

| 앞서, 연속확률변수의 기대치는 다음과 같이 도출됨을 확인함

$$E[X] = \int_{-\infty}^{\infty} xf(x)\,dx$$

예제 2-9

도착간격이 지수분포를 따르고 평균도착률이 한 시간당 6대인 버스가 있다. 이때 이 버스의 평균도착간격은 얼마일까?

- 지수분포함수를 가지는 확률변수 X에 대한 기대치를 구하면,

$$E[X] = \int_{-\infty}^{\infty} xf(x)\,dx = \int_{0}^{\infty} x\lambda e^{-\lambda x}\,dx = -xe^{-\lambda x}\Big|_0^{\infty} + \int_{0}^{\infty} e^{-\lambda x}\,dx = -\frac{1}{\lambda}e^{-\lambda x}\Big|_0^{\infty} = \frac{1}{\lambda}$$

- $\lambda = 6$이며, 지수확률변수의 기대치는 $E[X] = \frac{1}{\lambda} = \frac{1}{6}$

A. 확률함수를 활용한 기대치 도출 – 연속확률변수의 기대치

예제 2-10

확률변수의 PDF가 파라미터 (μ, σ^2)로 정의되는 정규확률변수 X의 기대치와 분산을 구하여 이 파라미터의 물리적인 의미를 논하여라.

- 파라미터 (μ, σ^2)로 정의되는 정규확률변수 X의 PDF는,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty$$

- $E[X] = \int_{-\infty}^{\infty} x f(x)\, dx = \int_{-\infty}^{\infty} x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}(x-\mu)e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx +$
$\mu\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} y e^{-\frac{y^2}{2\sigma^2}}\, dy + \mu\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx = \mu$

- $Var[X] = E[(x-\mu)^2] = \int_{-\infty}^{\infty}(x-\mu)^2 f(x)\, dx = \sigma^2\int_{-\infty}^{\infty} y^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, dy = \sigma^2$

A. 확률함수를 활용한 기대치 도출 – 이산확률변수의 기대치

| 앞서, 이산확률변수의 기대치는 다음과 같이 도출됨을 확인함

$$E[X] = \sum_{x} xp(x)$$

예제 2-11

어느 팀이 한 번 만에 풍선을 터트릴 확률은 20%라고 하자. 그럼 어떤 **한 팀이 통과할 때까지** 터트려야 하는 횟수의 기대치는 얼마인가?

- 기하분포를 가지는 기하확률변수 X에 대한 기대치를 구하면,

$$E[X] = \sum_{n=1}^{\infty} np(n) = \sum_{n=1}^{\infty} n(1-p)^{n-1}p = p\sum_{n=1}^{\infty} n(1-p)^{n-1}$$

- $E[X] = p\sum_{n=1}^{\infty} nq^{n-1} = p\frac{d}{dq}\left(\sum_{n=1}^{\infty} q^n\right) = p\frac{d}{dq}\frac{q}{1-q} = \frac{p}{(1-q)^2} = \frac{1}{p}$
- $p = \frac{1}{5}$임에 따라, $E[X] = \frac{1}{\frac{1}{5}} = 5 \rightarrow$ 풍선을 터트리고 통과할 때까지 평균 5번 시도해야 함

A. 확률함수를 활용한 기대치 도출 – 이산확률변수의 기대치

예제 2-12

확률질량함수가 다음과 같이 정의되는 푸아송확률변수 X에 대하여 기대치가 λ임을 증명하여라.

- 푸아송분포를 가지는 푸아송확률변수 X에 대한 기대치를 구하면,

$$E[X] = \sum_{n=0}^{\infty} np(n) = \sum_{n=0}^{\infty} n\frac{\lambda^n e^{-\lambda}}{n!} = \sum_{n=1}^{\infty} n\frac{e^{-\lambda}\lambda^n}{(n-1)!}$$

$$= \lambda e^{-\lambda}\sum_{n=1}^{\infty} n\frac{\lambda^{n-1}}{(n-1)!} = \lambda e^{-\lambda}\sum_{k=0}^{\infty}\frac{\lambda^k}{k!} = \lambda e^{-\lambda}e^{\lambda} = \lambda$$

A. 모멘트생성함수를 활용한 기대치 도출

모멘트생성함수.. 그리고 그것을 활용한 기대치를 공부하기 전에,
'모멘트'가 무엇인지부터 알아야겠지!

모멘트 = 적률(積率)

積: 쌓을 적 → "곱하는 것, 누적하는 것"

率: 비율 률 → "평균값"

곱한 뒤 평균을 낸 것
(중심에서 얼마나 떨어져 있는지를 수치화)

수학에서의 n차 적률: $E[X^n]$

- 1차 적률 ($E[X]$): 기대치
- 2차 적률 ($E[X^2]$): 분산 계산에 활용
  - $\sigma^2 = E[X^2] - (E[X])^2$

즉, 모먼트(Moment)란 확률분포의 특성을 정량화하는데 활용된다!

증명

- $\sigma^2 = \sum_x (x-\mu)^2 f(x) = \sum_x (x^2 - 2\mu x + \mu^2) f(x)$

$$= \sum_x x^2 f(x) - 2\mu \sum_x x f(x) + \mu^2 \sum_x f(x)$$

- 정의에 의해서 $\mu = \sum_x x f(x)$이고, $\sum_x f(x) = 1$이므로

$$\sigma^2 = \sum_x x^2 f(x) - \mu^2 = E(X^2) - \mu^2$$

A. 모멘트생성함수를 활용한 기대치 도출

왜 모먼트생성함수(MGF, Moment Generating Function)를 정의할까?!

| 목표: 모먼트를 한 번에 계산할 수 있는 도구 함수를 만들고자 함

| 핵심 아이디어: 모든 n차 모멘트를 구하는 것이 가능하며 함수의 도함수에 정보가 포함된 함수

$$\Phi(t) = E[e^{tX}]$$

왜 e^{tX}를 활용할까?!

- 지수함수의 테일러 전개를 살펴보면,

$$e^{tX} = 1 + tX + \frac{t^2X^2}{2!} + \frac{t^3X^3}{3!} + \cdots$$

- 양변에 기대값을 취하면,

$$E[e^{tX}] = 1 + tE[X] + \frac{t^2E[X^2]}{2!} + \frac{t^3E[X^3]}{3!} + \cdots$$

- 미분과 $t = 0$ 대입을 통해 원하는 값을 도출할 수 있음을 확인할 수 있음

A. 모멘트생성함수를 활용한 기대치 도출

| 따라서, 연속확률변수 X에 대하여 MGF를 다음과 같이 정의함

$$\Phi(t) = E[e^{tX}] = \int_{-\infty}^{\infty} e^{tX} f(x)dx$$

MGF의 활용 방법에 알아보면..

- $\Phi'(t) = \frac{d}{dt}E[e^{tX}] = E\left[\frac{d}{dt}e^{tX}\right] = E[Xe^{tX}]$
- $\Phi''(t) = \frac{d}{dt}\Phi'(t) = E[X^2e^{tX}]$
- 따라서,

$$\Phi'(0) = E[X], \Phi''(0) = E[X^2]$$

- $\sigma^2 = E[X^2] - (E[X])^2 = \Phi''(0) - (\Phi'(0))^2$

A. 모멘트생성함수를 활용한 기대치 도출

예제 2-13

어떤 확률변수의 MGF가 다음과 같이 주어질 때 그 확률변수의 평균과 분산을 구하라.

$$\Phi(t) = e^{\lambda(e^t-1)}$$

- MGF를 통해 확률변수의 평균과 분산을 도출할 수 있으므로, 이에 대한 미분을 수행하면,
- $\Phi'(t) = \lambda e^t e^{\lambda(e^t-1)}$
- $\Phi''(t) = \lambda^2 e^{2t} e^{\lambda(e^t-1)} + \lambda e^t e^{\lambda(e^t-1)} = e^{\lambda(e^t-1)}(\lambda e^t + \lambda^2 e^{2t})$
- $\Phi'(0) = \lambda, \Phi''(0) = \lambda + \lambda^2$
- 따라서, $E[X] = \Phi'(0),\ Var[X] = \Phi''(0) - (\Phi'(0))^2$이므로,
- $E[X] = \lambda, Var[X] = \Phi''(0) - \left(\Phi'(0)\right)^2 = \lambda$

A. 모멘트생성함수를 활용한 기대치 도출

| 목표: 두 정규분포 $X \sim N(\mu_1, \sigma_1^2), Y \sim N(\mu_2, \sigma_2^2)$의 MGF를 통해, 이들의 합 $X+Y$도 정규분포가 됨을 보이고자 함

| 도출하고 하는 결과: $X+Y \sim N(\mu_1+\mu_2, \sigma_1^2+\sigma_2^2)$

- 정규분포의 PDF는 다음과 같으므로,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty$$

- $\Phi(t) = E[e^{tX}] = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} e^{tX} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = exp\left(\mu t + \frac{\sigma^2 t^2}{2}\right)$

- 두 정규분포 X, Y는 독립이므로, $\Phi_{X+Y}(t) = \Phi_X(t) \cdot \Phi_X(t)$이고,
- MGF가 지수함수 꼴로 표현됨에 따라,

$$\Phi_{X+Y}(t) = exp\left(\mu_1 t + \frac{1}{2}\sigma_1^2 t^2\right) \times exp\left(\mu_2 t + \frac{1}{2}\sigma_2^2 t^2\right) = exp\left((\mu_1+\mu_2)t + \frac{1}{2}(\sigma_1^2+\sigma_2^2)t^2\right)$$

MGF 없이 증명하려고 했다면..

- $Z = X+Y$의 분포를 구하기 위하여 합성된 확률분포함수를 다음과 같이 도출해야 함

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(z-x-\mu_1)^2}{2\sigma_2^2}} dx$$

A. 모멘트생성함수를 활용한 기대치 도출

| 이산확률변수는 다음과 같은 적률생성함수를 가지며, 이는 이산확률변수의 확률생성함수 (PGF, Probability Generating Function)이라고 함

$$G(z) = E[z^X] = \sum_{k=0}^{\infty} p(k) z^k$$

구분	연속형 (MGF)	이산형 (PGF)
함수	$\Phi(t) = E[e^{tX}]$	$G(z) = E[z^X]$
사용 이유	• 실수 전 구간 표현 가능 • 지수함수를 전개하여 모든 적률 생성 가능	• 거듭제곱 계수로 확률 및 모멘트 추출 용이
해석 방식	• t에 대한 연속 함수	• z에 대한 이산 항들의 합
장점	• 모멘트 계산, 중심극한정리 등 분석에 유리	• 간결하며 확률/팩토리얼 모멘트 추출 용이 • 계산 쉬움

e^{tX}는 연속적인 실수 영역을 커버하기 위해 적합하고,
z^X는 이산확률변수 X에 충분한 구조임

A. 모멘트생성함수를 활용한 기대치 도출

| 따라서, 이산확률변수 X에 대하여 PGF를 다음과 같이 정의함

$$G(z) = E[z^X] = \sum_{k=0}^{\infty} p(k) z^k$$

PGF의 활용 방법에 알아보면..

- $G^{(1)}(z) = \frac{d}{dt} E[z^X] = E\left[\frac{d}{dt} z^X\right] = E[X z^{X-1}]$
- 따라서,

$$G^{(1)}(1) = E[X]$$

확률생성함수 G(z)의 k차 도함수 표현

- $G^{(k)}(z) = \sum_{x=k}^{\infty} p(k) \cdot x(x-1)\cdots(x-k+1) \cdot z^{x-k}$

- k차 팩토리얼 모멘트를 구해보면,

$$E[X(X-1)\cdots(X-k+1)] = \frac{X!}{(X-k)!}$$

$$E\left[\frac{X!}{(X-k)!}\right] = G^{(k)}(1)$$

A. 모멘트생성함수를 활용한 기대치 도출

PGF의 활용 방법에 알아보면..

- $Var[X] = E[X^2] - (E[X])^2 \rightarrow X^2 = X(X-1) + X$를 이용하면,
- $E[X^2] = E[X(X-1)] + E[X] = G^{(2)}(1) + G^{(1)}(1)$
- 따라서, $Var[X] = G^{(2)}(1) + G^{(1)}(1) - (G^{(1)}(1))^2$

예제 2-14

어떤 확률변수 X의 확률생성함수 $G(z)$가 아래와 같이 주어진다고 할 때 이 확률변수의 기대치와 분산을 구하라.

$$G(z) = e^{z-1}$$

- $E[X] = G^{(1)}(1)$이므로, $E[X] = e^{z-1}|_{z=1} = 1$
- $Var[X] = G^{(2)}(1) + G^{(1)}(1) - \left(G^{(1)}(1)\right)^2$ 이며, $G^{(2)}(1) = e^{z-1}|_{z=1} = 1$ 이므로,

$$Var[X] = G^{(2)}(1) + G^{(1)}(1) - \left(G^{(1)}(1)\right)^2 = 1 + 1 - 1 = 1$$

확률의 근사 해석법

IV

IV 4 확률의 근사 해석법 (1/19)

A. 근사 해석법의 필요성

| 앞서 복잡한 시스템을 해석 가능한 형태로 단순화하여, 빠르게 성능을 예측할 수 있도록 근사해석법을 활용함을 언급한 바 있음

1 마르코프 부등식 (Markov's Inequality)

| (활용) 확률분포를 정확히 몰라도 기대치만 알면 상대적으로 큰 값이 나올 확률의 상한을 구할 수 있음

- (예시) 고객 대기시간의 평균이 5분일 때, 15분 이상 기다릴 확률은 최대 $\frac{5}{15} = \frac{1}{3}$ 임

2 체비셰프 부등식 (Chebyshev's Inequality)

| (활용) 분포의 형태와 관계없이 평균으로부터 멀어질 확률의 상한을 제공함

- (예시) 평균 대기시간 10분, 표준편차 2분이면, 6분 이상 벗어날 확률은 최대 $\frac{1}{3^2} = \frac{1}{9}$ 임

3 대수의 법칙 (Law of Large Numbers)

| (활용) 샘플 평균은 전체 평균(기대값)에 수렴하므로, 많은 시행을 통해 확률적 성질을 안정적으로 추정 가능

- (예시) 고객이 도착하는 시간 간격을 여러 번 측정하면, 그 평균은 실제 평균 도착 간격과 가까워짐

4 중심극한정리 (Central Limit Theorem)

| (활용) 실제 분포가 어떤 형태이든, 많은 수의 합이나 평균은 정규분포로 근사하여 계산을 용이하게 수행할 수 있음

- (예시) 한 시간 동안 도착한 고객 수가 일정하지 않아도, 여러 시간 동안 평균을 보면 정규분포로 근사 가능함에 따라 대기행렬에서 서비스 시간 분포 추정에 활용됨

A. 마르코프 부등식 (Markov's Inequality)

| 정의: 0 이상의 값을 가지는 확률변수의 경우, 평균값만으로 그 값이 특정 임계값 이상일 확률의 상한을 추정할 수 있는 부등식

$$P(X \geq a) \leq \frac{E[X]}{a} \; (X \geq 0, a > 0)$$

| 특징: 분포가 어떤 모양이든 상관없이 적용 가능하며, 평균값만 있어도 사용 가능함

증명

- 기대값을 다음처럼 두 구간으로 나누면,

$$E[X] = \int_0^\infty x f(x)\,dx = \int_0^a x f(x)\,dx + \int_a^\infty x f(x)\,dx$$

- 두 번째 항에 대하여 $x \geq a$ 이므로,

$$\int_a^\infty x f(x)\,dx \geq a \int_a^\infty f(x)\,dx = aP(X \geq a)$$

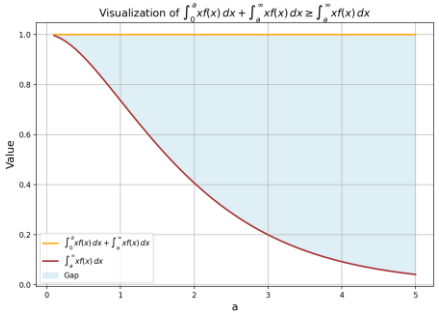
- 따라서,

$$P(X \geq a) \leq \frac{E[X]}{a}$$

A. 마르코프 부등식 (Markov's Inequality)

| 한계점: 큰 오차가 발생할 수 있으며, 분포 형태를 전혀 반영하지 않음

- 평균값만을 보고 확률을 추정함에 따라 훨씬 큰 확률 상한이 도출될 수 있음



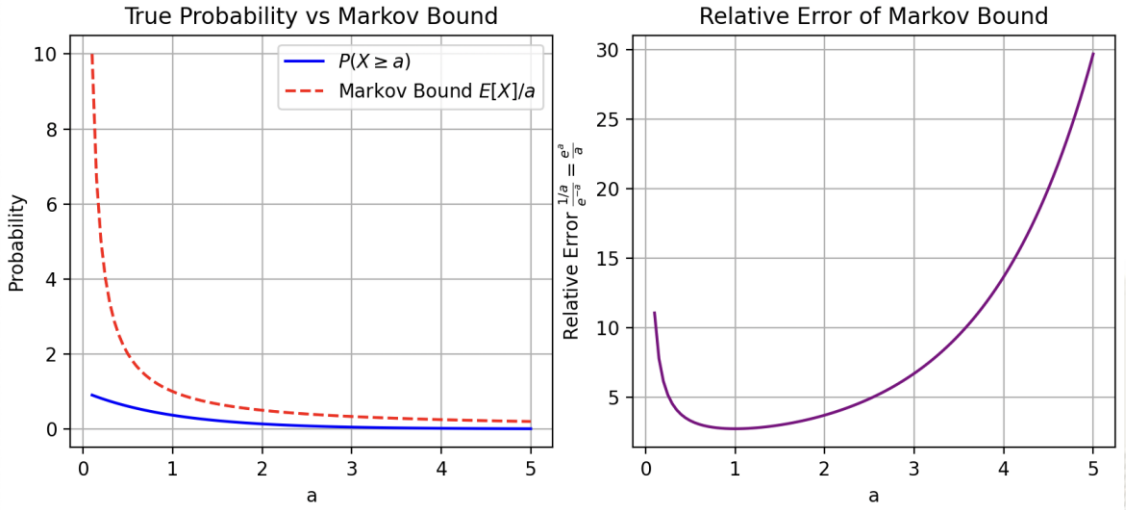
(그림 21) $\lambda = 1$에서의 마르코프 부등식 오차 그래프

마르코프 부등식은 평균만을 가지고 계산함에 따라 비대칭 분포, 치우침(skewness), 집중도(peakedness) 등에 대한 반영을 수행하지 못함

- $\int_a^\infty xf(x)\,dx$의 상한치를 나타내려고 등장하였으나, a 값이 크다면

$$\int_0^a xf(x)\,dx + \int_a^\infty xf(x)\,dx \geq \int_a^\infty xf(x)\,dx$$

의 식은 큰 오차를 가짐



(그림 20) a값 증가에 따른 마르코프 부등식 오차 증가

마르코프 부등식의 활용성

- 최소한의 정보(기대값)만을 필요로 함
- 마르코프 부등식은 간단하고 보수적인 추정을 위한 도구임
- 정확성에는 제한이 존재하므로, 기초 추정용 또는 경계 설정용으로 사용됨
- 분산 정보가 있을 경우 체비셰프 부등식 등으로 더 정밀한 추정이 가능함

A. 마르코프 부등식 (Markov's Inequality)

예제 2-15

세종대학교 정보보호학과 학생의 확률론 과목의 점수가 90점 이상일 확률이 10%라고 한다. 그렇다면 이 학생들의 평균점수는 적어도 얼마보다는 작지 않을지 마르코프부등식을 이용하여 예측하라.

- 마르코프부등식 $P(X \geq a) \leq \frac{E[X]}{a}$에서,
- $a = 90, P(X \geq 90)$이므로,

$$E[X] \geq aP(X \geq a) = 90 \times 0.1 = 9$$

예제 2-16

세종대학교 정보보호학과 학생의 메모리 익스플로잇 과목의 평균점수가 50점이라고 한다. 이때 어떤 한 학생을 선택해서 그의 점수를 물었을 때 그 학생의 점수가 90점 이상일 확률은 가장 클 때 대략 얼마일까?

- 마르코프부등식 $P(X \geq a) \leq \frac{E[X]}{a}$에서,
- $E[X] = 50$이고, 구하고자 하는 것은 $P(X \geq 90)$이므로,

$$P(X \geq 90) \leq \frac{50}{90} = 0.556$$

A. 마르코프 부등식 (Markov's Inequality)

문제 2-7

우리나라 성인 여자의 평균키가 160cm라고 하자. 이때 어떤 성인 여자 한 명을 선택하였을 때 그 사람의 키가 170cm보다 작을 확률은 최소한 얼마인지 대강 예측하라.

- $E[X] = 160$이고, 구하고자 하는 것은 $P(X < 170)$이므로,

$$P(X \geq 170) \leq \frac{E[X]}{170} = \frac{160}{170} = \frac{16}{17}$$

- 따라서,

$$P(X < 170) = 1 - P(X \geq 170) \geq 1 - \frac{16}{17} = \frac{1}{17} \approx 0.0588$$

A. 체비셰프 부등식 (Chebyshev's Inequality)

| 정의: 평균과 분산이 존재하는 확률변수의 경우, 해당 확률변수가 평균에서 k 이상 떨어질 확률의 상한을 다음과 같이 추정할 수 있는 부등식

$$P(|X-\mu| \geq k) \leq \frac{\sigma^2}{k^2} \ (k>0)$$

| 특징: 데이터의 분포를 정확히 알기 어려운 경우에도 데이터의 흩어짐 정도를 파악할 수 있음

증명

- $P(|X-\mu| \geq k)$을 $P((X-\mu)^2 \geq k^2)$로 바꾸어 표현
→ $P[X \geq a]$을 가지는 마르코프 부등식의 꼴과 유사함

- 마르코프부등식 $P[X \geq a] \leq \frac{E[X]}{a}$에서,

$$P\left((X-\mu)^2 \geq k^2\right) \leq \frac{E[(X-\mu)^2]}{k^2} = \frac{\sigma^2}{k^2}$$

- 따라서,

$$P(|X-\mu| \geq k) \leq \frac{\sigma^2}{k^2}$$

체비셰프 부등식의 활용성

〉 평균과 분산만으로 상한 추정 가능

〉 어떤 분포든 적용 가능하며, 실험 결과의 신뢰구간 설정, 이상치 탐지 등에 사용됨

〉 특정 구간 내에 데이터가 위치할 최소 확률 보장 (e.g., 95% 이상이 평균 ±4.47 σ 내에 있음)

〉 정규분포 가정이 어려운 경우나, 표본 크기가 작을 때도 적용 가능함

A. 체비셰프 부등식 (Chebyshev's Inequality)

예제 2-17

초고속정보통신망에서 아주 많은 수의 가입자가 발생시키는 트래픽의 양[단위: bps]이 평균이 초당 1메가(메가는 백만을 의미함)비트이고 분산이 0.5라고 하자. 이때 이들 가입자가 발생시키는 트래픽의 양이 **평균값보다 1메가 bps 이상 벌어질 확률**은 얼마일까?

- 체비셰프 부등식 $P(|X-\mu| \geq k) \leq \frac{\sigma^2}{k^2}$ 에서,
- $\mu = 1, \sigma^2 = 0.5$이며, $|X-\mu| \geq 1$인 확률을 구하고자 함

$$P(|X-1| \geq 1) \leq \frac{0.5}{1^2} = 0.5$$

문제 2-7

어떤 확률 실험에 대하여 얻어지는 확률변수가 평균이 0.7이고 분산이 0.3이라고 한다. 이때 임의의 한 실험결과가 **평균으로부터 1 이상 떨어져 있을 확률**은 최대 얼마일까?

- $\mu = 0.7, \sigma^2 = 0.3$이며, $|X-\mu| \geq 1$인 확률을 구하고자 함

$$P(|X-0.7| \geq 1) \leq \frac{0.3}{1^2} = 0.3$$

A. 마르코프 부등식 VS. 체비셰프 부등식

로그 처리 시간 초과 탐지를 위한 상한 추정

한 보안 로그 수집 시스템에서는 1건의 로그를 처리하는 데 걸리는 시간의 기대값은 100ms, 분산은 400이다. 서비스 품질 기준에 따라, 로그 처리 시간이 200ms 이상일 확률이 중요한 성능 지표가 된다. 하지만 정확한 분포 형태는 알 수 없는 상황에서, 상한 추정을 통해 해당 확률을 추정하고자 한다.

활용

1. 마르코프 부등식 적용

$$P(X \geq 200) \leq \frac{E[X]}{200} = \frac{100}{200} = 0.5$$

- 분포 형태는 모르지만, 기대값만으로 상한 50%를 추정

2. 체비셰프 부등식 적용

- $P(|X-\mu| \geq k) \leq \frac{\sigma^2}{k^2}$ 에서, $\sigma = 20, k = 100$이므로,

$$P(|X-\mu| \geq 100) \leq \frac{400}{10000} = 0.04$$

- 분산까지 고려한 결과, 훨씬 더 낮은 상한값(4%)으로 추정 가능
- 즉, 로그 수집 시간이 200ms 넘을 확률 추정 시, 분산을 알고 있으면 훨씬 정확하게 위험도를 평가할 수 있음

A. 대수의 법칙 (Law of Large Numbers)

| 정의: 동일한 분포를 가지는 확률변수를 반복적으로 측정했을 때, 표본평균이 모평균에 수렴하는 현상

$$\frac{1}{n}\sum_{k=1}^{n} X_k \to \mu \ (\text{as } n \to \infty)$$

| 특징: 표본 수가 클수록 정확도 향상되며, 확률변수들이 독립이고 동일한 분포를 따르는 경우, 분포의 구체적인 모양과 상관없이 적용 가능함

대표 수식

- 확률변수 $X_1, X_2, \dots, X_n$이 평균 μ를 갖는 독립이고 동일한 분포(i.i.d.)일 때,

$$\frac{S_n}{n} \xrightarrow{P} \mu \text{ (대수의 약법칙)}$$

$$\frac{S_n}{n} \xrightarrow{a.s.} \mu \text{ (대수의 강법칙)}$$

*$S_n = \sum_{k=1}^{n} X_k$
(X_k: 독립 동분포 확률변수)

대수의 법칙의 활용성

〉 반복된 실험이나 관측을 통해 신뢰성 있는 평균값을 추정 가능함

〉 분포의 형태에 관계없이 평균 추정이 가능하며, 샘플의 크기가 커질수록 모평균에 수렴하므로 신뢰도가 높아짐

〉 실험 결과의 재현성과 일관성을 판단하는 근거가 됨

〉 통계적 추론과 데이터 분석 모델링의 이론적 근거로 활용됨

A. 대수의 약법칙 (Weak Law of Large Numbers)

| 정의: 충분히 많은 표본을 모으면 그 표본평균이 확률적으로 모평균에 가까워진다는 법칙

$$\lim_{n\to\infty} P\left(\left|\frac{S_n}{n}-\mu\right|>\epsilon\right)=0$$

*$S_n=\sum_{k=1}^{n} X_k$
(X_k: 독립 동분포 확률변수)

| 의미 및 해석:

- $\lim_{n\to\infty} P\left(\left|\frac{S_n}{n}-\mu\right|<\epsilon\right)=1$ 로 표현 가능: 표본 수 n 이 충분히 크면, 평균이 모평균에 가까울 확률이 거의 1이라는 의미
- $S_n=\sum_{k=1}^{n} X_k$이므로, $\lim_{n\to\infty} P\left(\left|\frac{1}{n}\sum_{k=1}^{n} X_k-\mu\right|<\epsilon\right)=1$로 표현 가능: n개의 확률변수의 표본평균은 모평균에 수렴함

증명

- 체비셰프 부등식 $P(|X-\mu|\geq k)\leq\frac{\sigma^2}{k^2}$에서,

$$\lim_{n\to\infty} P\left(\left|\frac{S_n}{n}-\mu\right|>\epsilon\right)\leq\frac{Var\left[\frac{S_n}{n}\right]}{\epsilon^2}=\frac{\sigma^2}{n\epsilon^2}\to 0$$

확실히 수렴하지는 않더라도,
"확률적으로는 수렴"한다는 의미에서
"대수의 약법칙"이라 부름

A. 대수의 약법칙 (Weak Law of Large Numbers)

예제 2-18

모평균 μ 와 모분산 σ^2 이 모두 유한한 확률변수에 대하여 충분히 많은 개수의 표본을 추출하여 평균을 구하였을 때 그 표본평균이 모평균에 근접할 확률은 얼마인가?

- 대수의 약법칙에서, $\lim_{n\to\infty} P\left(\left|\frac{S_n}{n}-\mu\right|>\epsilon\right)=0$ 이며, 이를 변형하면,

$$\lim_{n\to\infty} P\left(\left|\frac{S_n}{n}-\mu\right|<\epsilon\right)=1$$

- 따라서, 표본평균이 모평균에 근접할 확률은 거의 1임

A. 대수의 강법칙 (Strong Law of Large Numbers)

| 정의: 충분히 많은 표본을 모으면, 표본평균이 거의 확실하게 모평균에 수렴하는 법칙 ($\frac{S_n}{n} \xrightarrow[a.s.]{} \mu$)

$$P\left(\lim_{n\to\infty} \frac{\sum_{k=1}^{n} X_k}{n} = \mu\right) = 1$$

*$S_n = \sum_{k=1}^{n} X_k$
(X_k: 독립 동분포 확률변수)
*a.s.: almost surely

| 의미 및 해석: '상대도수의 극한 = 확률'이라는 이론적 기반을 제공함

- 상대도수 (Relative Frequency): 사건 A의 발생 횟수 ÷ 전체 실험 횟수
- 확률 (Probability): 사건 A가 발생할 '이론적인 가능성'

〉 $\lim_{n\to\infty} \frac{\text{사건 } A \text{ 발생 횟수}}{n} = P(A)$: 실험을 반복할수록 상대도수는 확률에 점점 가까워짐

즉, 관측 횟수가 무한히 많아질수록 불확실성이 사라지고 예측 가능해짐을 의미함

| 예시: 동전을 1만 번 던질 때의 앞면 비율 ≈ 0.5
→ '앞면이 나올 확률'이 상대도수의 극한으로 정의됨을 보여줌

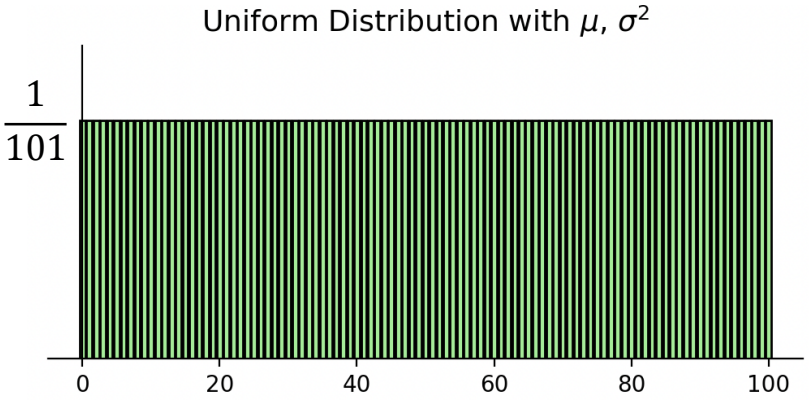
예제 2-19

유한한 모평균 μ를 가지는 확률변수에 대하여 충분히 많은 개수의 표본을 추출하여 평균을 구하였을 때 그 표본평균이 모평균에 근접할 확률은 얼마인가?

- 대수의 약법칙에서, $P\left(\lim_{n\to\infty} \frac{\sum_{k=1}^{n} X_k}{n} = \mu\right) = 1$이므로, 표본평균이 모평균에 근접할 확률은 1임

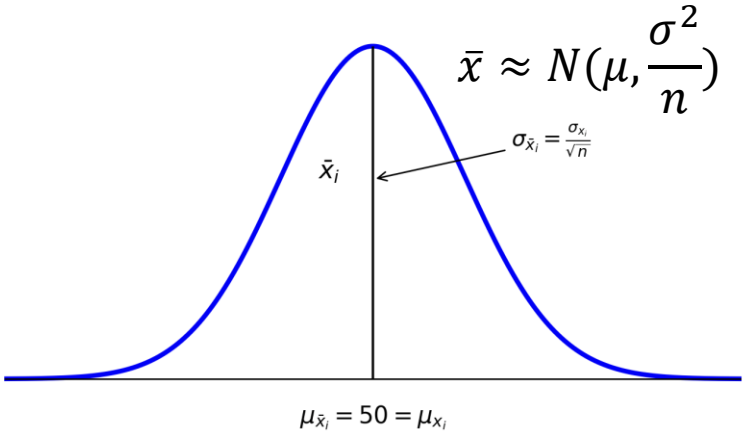
A. 중심극한정리 (CLT, Central Limit Theorem)

| 정의: 서로 독립이고 동일한 분포 (i.i.d.)를 따르는 확률변수들의 합 또는 평균은 그 개수가 충분히 클 경우 정규분포로 수렴하는 현상

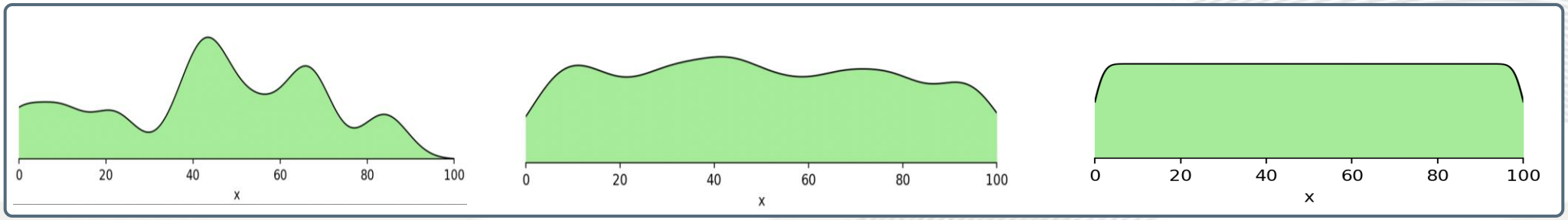


(그림 22) 0부터 100까지 모든 값의 확률이 1/101인 균등분포 (평균 μ, 분산 σ^2)

$\bar{x}_i$



(그림 24) 정규분포에 수렴하는 샘플 평균의 분포



(그림 23) 샘플 수에 따른 균등분포 근사 (n=10, 80, 101)

A. 중심극한정리 (CLT, Central Limit Theorem)

| 필요성: 데이터의 합이나 평균에 대해 예측 가능한 분포(정규분포)로 근사시킴으로써, 복잡한 현상을 수학적으로 단순화하고 분석 가능하게 만들어줌

| 예시: 공장 품질 관리 시, 하루에 생산된 부품의 무게를 일일이 알 수는 없지만, 여러 개의 무게 평균은 거의 정규분포를 따르므로 → 불량 여부를 평균 기준으로 판단 가능

중심극한정리 유도

Step 1. 확률변수 설정
- 확률변수 $X_1, X_2, \ldots, X_n$은 i.i.d.를 만족함
- 각 변수는 평균 μ, 분산 σ^2을 가짐
- $S_n = \sum_{i=1}^{n} X_i$

Step 2. 정규분포로의 수렴을 위한 준비
- S_n 자체는 n이 커질수록 무한히 커지므로 비교 불가 → 따라서 정규화 수행

$$\frac{S_n - n\mu}{\sigma\sqrt{n}}$$

Step 3. 정규분포 식의 변형
- 식을 항목별로 분해하고, 다음과 같이 치환:

$$Y_i = \frac{X_i - \mu}{\sigma}$$

- 정규화된 합은 다음과 같이 표현 가능:

$$Z_n = \sum_{i=1}^{n} \frac{Y_i}{\sqrt{n}}$$

→ 이것이 중심극한정리에서 정규분포로 수렴함을 보이고 싶은 대상!

A. 중심극한정리 (CLT, Central Limit Theorem)

중심극한정리 유도

Step 4. 정규분포로의 수렴 확인

- Z_n의 정규분포로 수렴함을 보이기 위해 MGF 사용

$$M_{\frac{Y_i}{\sqrt{n}}}(t) = E\left[e^{t\left(\frac{Y_i}{\sqrt{n}}\right)}\right] = M(\frac{t}{\sqrt{n}})$$

- Z_n은 i.i.d. 항의 합이므로,

$$M_{Z_n}(t) = \left(M(\frac{t}{\sqrt{n}})\right)^n$$

- 테일러 전개를 활용하면,

$$\left(M(\frac{t}{\sqrt{n}})\right)^n \xrightarrow[n\to\infty]{} e^{\frac{t^2}{2}}$$

- $e^{\frac{t^2}{2}}$은 표준정규분포 $N(0,1)$의 MGF
- 이는 해당 확률변수(Z_n)의 분포가 표준정규분포에 수렴함을 의미함

아직 확률이 얼마인지,
어떤 구간에 들 확률이 얼마나 되는지는
직접적으로 알 수 없음

예를 들어..
"100개의 데이터를 샘플링했을 때, 평균이 55를 넘을 확률은?"
→ 단순히 분포가 정규분포와 닮았다고 말하는 것만으로는 해결되지 않음
→ "55를 넘을 확률" = 정규분포의 CDF 가 필요!

$$P\left\{\frac{S_n - n\mu}{\sigma\sqrt{n}} \le a\right\} = P\left\{\frac{E[S_n] - \mu}{\sigma/\sqrt{n}} \le a\right\}$$

- $E[S_n] = \frac{S_n}{n} = \frac{\sum_{i=1}^{n} X_i}{n} = \bar{X}$

$$P\left\{\frac{E[S_n] - \mu}{\sigma/\sqrt{n}} \le a\right\} = P\left\{\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \le a\right\} \to \Phi(a)$$

A. 중심극한정리를 활용한 근사

Step 1. 이항분포의 확률근사

- 성공 확률이 p, 실패확률이 $1-q$인 베르누이 시행을 n번 반복할 때,
- 성공 횟수 $X \sim Bin(n,p)$는 다음의 확률을 가짐:

$$P(X=k) = \binom{n}{k} p^k q^{n-k}$$

- 기대값: $E[X] = np$
- 분산: $Var[X] = npq$

$$P(n\text{번의 시행에서 } k\text{번 성공}) = \binom{n}{k} p^k q^{n-k} \cong \frac{1}{\sqrt{2\pi npq}} e^{-\frac{(k-np)^2}{2npq}}$$

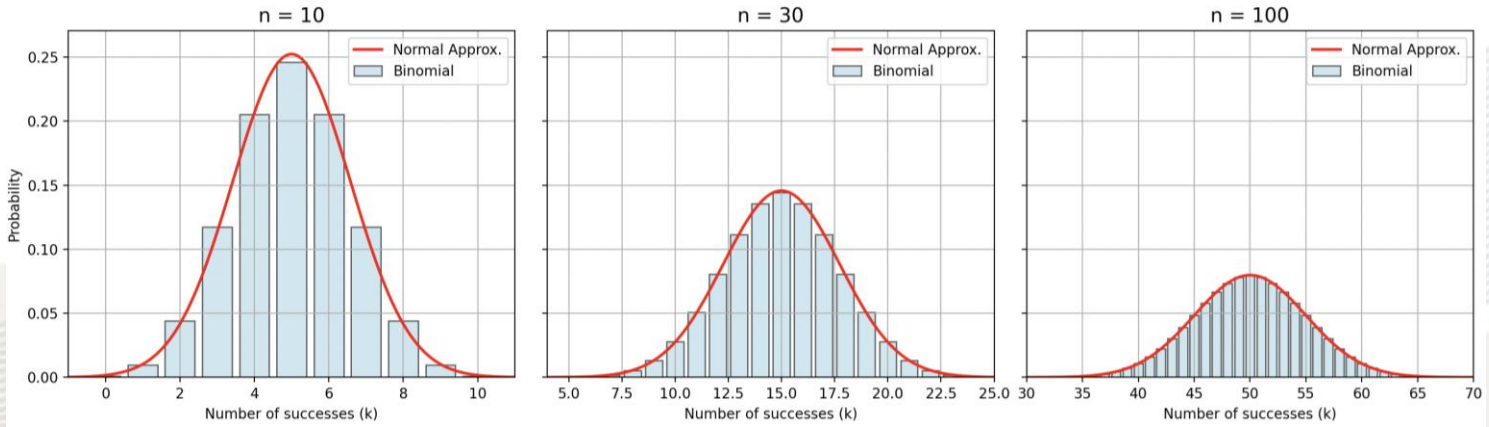
Step 2. 중심극한정리를 이용한 근사

- $X = \sum_{i=1}^{n} X_i \sim Bin(n,p)$는 베르누이 합으로 볼 수 있으며,
- 중심극한정리에 따라,

$$\frac{X-n\mu}{\sqrt{npq}} \to N(0,1)$$

- 따라서, $X \sim N(np, npq)$로 근사 가능하며,
- 정규분포의 밀도함수를 이용해 이항분포를 근사하면,

$$P(X=k) = \frac{1}{\sqrt{2\pi npq}} e^{-\frac{(k-np)^2}{2npq}}$$



(그림 25) n 증가에 따른 이항분포의 정규분포로의 근사

A. 중심극한정리 (CLT, Central Limit Theorem)

예제 2-20

N개의 i.i.d. 확률변수 $X_i, i = 1, 2, \dots, N$가 있고 각 확률변수는 아래와 같은 확률밀도함수를 가진다고 한다.

$$f_X(x) = \begin{cases} e^{-x}, x \geq 0 \\ 0, \quad x < 0 \end{cases}$$

여기서 이들 N개의 i.i.d. 확률변수 $X_i, i = 1, 2, \dots, N$ 를 더한 값 Y를 다음과 같이 나타낼 때

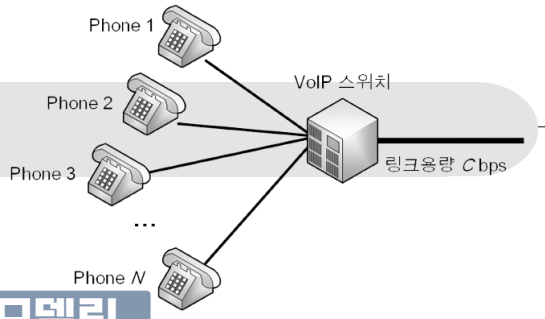
$$Y = \lim_{N \to \infty} \sum_{i=1}^{N} X_i$$

Y의 확률밀도함수를 구하라.

- Y의 확률밀도함수를 구하기 위해서는 Y의 평균과 분산을 구해야함
- $E[X_i] = \int_0^\infty x f_X(x) dx = \int_0^\infty x e^{-x} dx = 1$
- $E[X_i^2] = \int_0^\infty x^2 f_X(x) dx = \int_0^\infty x^2 e^{-x} dx = 2$
- $\sigma_{X_i}^2 = E[X_i^2] - (E[X_i])^2 = 1$
- 따라서, $E[Y] = \sum_{i=1}^{N} E[X_i] = N,\ \sigma_Y^2 = \sigma_{X_1}^2 + \sigma_{X_2}^2 + \cdots + \sigma_{X_N}^2 = N$이므로,

$$f_Y(y) = \frac{1}{\sqrt{2\pi N}} e^{-\frac{(y-N)^2}{2N}}$$

IV 4 확률의 근사 해석법 (18/19)



A. 중심극한정리의 활용

1. 문제 상황

- N명의 가입자들이 Cbps의 대역용량을 가지는 링크를 통해 VoIP 서비스를 받음

- 모든 가입자가 동시에 통신 시도 시, 링크 용량 초과 가능성이 존재함
  → "링크 용량이 어느 정도 이상이면 품질을 보장할 수 있을까?!"

2. 확률 변수 정의 및 모델링

- X_i: i번째 가입자의 순간 비트 도착률
- 각 X_i의 평균: m, 분산: σ^2
- 전체 트래픽 총합

$$Y = \sum_{i=1}^{N} X_i$$

- Y의 평균: $E[Y] = Nm$, 분산: $Var(Y) = N\sigma^2$

3. 성능 실패 확률 한계 설정

- 네트워크 용량: C
- 실패 확률 정의: $P(Y > C) < \delta$
- 정규화 수행:

$$P\left(\frac{Y - Nm}{\sqrt{N}\sigma} > \frac{C - Nm}{\sqrt{N}\sigma}\right) < \delta$$

$$1 - P\left(\frac{Y - Nm}{\sqrt{N}\sigma} \leq \frac{C - Nm}{\sqrt{N}\sigma}\right) < \delta$$

$$1 - \Phi\left(\frac{C - Nm}{\sqrt{N}\sigma}\right) < \delta$$

4. 역함수 적용과 C 계산

- 표준정규분포 역함수 Q^{-1} 사용

$$\frac{C - Nm}{\sqrt{N}\sigma} > Q^{-1}(\delta)$$

$$C > Nm + \sqrt{N}\sigma Q^{-1}(\delta)$$

5. 해석 및 활용

- 네트워크 용량은 평균 트래픽 Nm보다 더 크게 설정하고, 트래픽의 변동성을 고려한 여유량 $\sqrt{N}\sigma Q^{-1}(\delta)$을 포함해야 함
- 네트워크 설계 시 QoS 기준 설정 가능

A. 중심극한정리의 활용

예제 2-20

어떤 네트워크에서 접속되어 있는 **가입자의 수 $N = 36$명, 각 가입자당 평균대역폭 $m = 1$ Mbps, 가입자당 대역폭의 분산 $\sigma^2 = 0.09$**, 그리고 서비스품질 요구조건으로 정의된 순간 대역 요구치가 네트워크가 제공 가능한 **대역치인 C를 초과할 확률의 목표치인 $\delta = 0.01$**일 때 이 네트워크에서 가입자가 요구하는 서비스품질을 만족시켜주기 위하여 필요한 소요 대역폭 C는 최소한 얼마 이상이어야 하는가?

- 앞서 구한 수식을 살펴보면,

$$C > Nm + \sqrt{N}\sigma Q^{-1}(\delta)$$

- $N = 36, m = 1, \sigma^2 = 0.09 \rightarrow \sigma = 0.3, \delta = 0.01$이므로,

$$C > 36 \cdot 1 + \sqrt{36} \cdot 0.3 \cdot Q^{-1}(\delta) \approx 36 + 4.1868$$

$$C > 40.1868 \text{ Mbps}$$

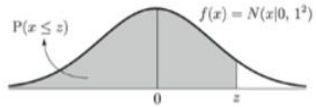
- $\delta = 0.01$일 때,
- $Q^{-1}(0.01) = z$를 알기 위해서는

$\rightarrow \Phi(z) = 1 - \delta = 0.99$인 z 값 도출

$\rightarrow \Phi(2.32) = 0.9898, \Phi(2.33) = 0.9901$이므로,

$$Q^{-1}(\delta) \approx 2.326$$

Standard Normal Distribution table



z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952

감 사 합 니 다

손 우 영 (wooyoung@pel.sejong.ac.kr)

V 부록 #1 - Python 코드

- (그림 1)

```python
import numpy as np
import matplotlib.pyplot as plt

# Probability density function
def f(x):
    return 2 * np.exp(-2 * x)

# x range
x = np.linspace(0, 3, 1000)
y = f(x)

# Plot
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(x, y, label=r'$f(x) = 2e^{-2x}$', color='blue', linewidth=2)

# Fill area under curve from 0 to 1
x_fill = np.linspace(0, 1, 500)
y_fill = f(x_fill)
ax.fill_between(x_fill, y_fill, color='skyblue', alpha=0.5, label='P(0 ≤ X ≤ 1)')

# Annotation
ax.text(1.05, f(1), r"$\approx 86.5\%$", fontsize=16)
ax.axvline(x=1, linestyle='--', color='gray', alpha=0.6)

# Labels and styling
ax.set_title('Probability Density Function and Cumulative Probability (0 ≤ X ≤ 1)', fontsize=18)
ax.set_xlabel('x (minutes)', fontsize=16)
ax.set_ylabel('f(x)', fontsize=16)
ax.tick_params(labelsize=14)
ax.legend(fontsize=14)
ax.grid(True)

plt.tight_layout()
plt.show()
```

V 부록 #2 - Python 코드

- (그림 2)

```python
import numpy as np
import matplotlib.pyplot as plt

# Define exponential PDFs with different λ values
lambdas = [2, 1, 0.5]
x = np.linspace(0, 10, 1000)
colors = ['blue', 'orange', 'green']
labels = [r'$\lambda=2$', r'$\lambda=1$', r'$\lambda=0.5$']
y_positions = [0.7, 0.5, 0.3]

# Plot settings
plt.figure(figsize=(10, 6))

for i, lam in enumerate(lambdas):
    y = lam * np.exp(-lam * x)
    plt.plot(x, y, label=labels[i], linewidth=2, color=colors[i])

    avg_wait = 1 / lam
    plt.axvline(avg_wait, color=colors[i], linestyle='--', alpha=0.6)
    plt.text(avg_wait + 0.2, y_positions[i], f'$1/\\lambda = {avg_wait:.1f}$',
             fontsize=14, color=colors[i], fontweight='bold')

# Titles and labels
plt.title('Exponential Distributions and Their Mean Waiting Times', fontsize=18)
plt.xlabel('x (Waiting Time)', fontsize=16)
plt.ylabel('f(x)', fontsize=16)
plt.legend(fontsize=14)
plt.grid(True)
plt.tick_params(labelsize=14)

plt.tight_layout()
plt.show()
```

V 부록 #3 - Python 코드

- (그림 3)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

lam = 1.0
x_vals = np.linspace(0, 5, 500)
f_vals = lam * np.exp(-lam * x_vals)

fig, ax = plt.subplots(figsize=(8, 4))
line_f, = ax.plot(x_vals, f_vals, label=r'$f(x) = \lambda e^{-\lambda x}$', color='blue')

ax.hlines(1, 0, 5, colors='black', linestyles='dotted')

vline_upper = ax.vlines(0, 0, 1, colors='gray', linestyles='dashed', label=r'$1 - f(x)$')
vline_lower = ax.vlines(0, 0, 0, colors='red', linestyles='dashed', label=r'$f(x)$')

point_f, = ax.plot([], [], 'bo')

text_val = ax.text(0.1, 0.72, '', transform=ax.transAxes)

ax.set_xlim(0, 5)
ax.set_ylim(0, 1.1)
ax.set_title('Visualizing $f(x)$ and $1 - f(x)$')
ax.set_xlabel('x')
ax.set_ylabel('Value')

ax.legend()

def init():
    point_f.set_data([], [])
    text_val.set_text('')
    return line_f, vline_upper, vline_lower, point_f, text_val

def update(frame):
    x = x_vals[frame]
    fx = f_vals[frame]
    point_f.set_data([x], [fx])
    vline_upper.set_segments([[(x, fx), (x, 1)]])  # 1 - f(x)
    vline_lower.set_segments([[(x, 0), (x, fx)]])  # f(x)
    text_val.set_text(f'f(x) ≈ {fx:.3f} | 1 - f(x) ≈ {1 - fx:.3f}')
    return line_f, vline_upper, vline_lower, point_f, text_val

ani = animation.FuncAnimation(
    fig, update,
    frames=range(0, len(x_vals), 10),
    init_func=init,
    interval=200,
    blit=True
)

ani.save("f3.gif", writer='pillow', fps=10)

```

V 부록 #4 - Python 코드

- (그림 4)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

lam = 1.0
x_vals = np.linspace(0, 5, 500)
f_vals = lam * np.exp(-lam * x_vals)

fig, ax = plt.subplots(figsize=(8, 4))
line_f, = ax.plot(x_vals, f_vals, label=r'$f(x) = \lambda e^{-\lambda x}$', color='blue')
fill_F = ax.fill_between([], [], [], color='skyblue', alpha=0.5, label=r'$F(x) = \int_0^x f(t)dt$')
text_F = ax.text(0.05, 0.9, '', transform=ax.transAxes)
text_1mF = ax.text(0.05, 0.8, '', transform=ax.transAxes)

ax.set_xlim(0, 5)
ax.set_ylim(0, 1.1)
ax.set_title('Exponential Distribution: f(x) and Cumulative Area F(x)')
ax.set_xlabel('x')
ax.set_ylabel('Probability Density')
ax.legend()

def init():
    global fill_F
    fill_F.remove()
    fill_F = ax.fill_between([], [], [], color='skyblue', alpha=0.5)
    text_F.set_text('')
    text_1mF.set_text('')
    return line_f, fill_F, text_F, text_1mF

def update(frame):
    global fill_F
    x = x_vals[frame]
    y_fill = lam * np.exp(-lam * x_vals[:frame])
    fill_F.remove()
    fill_F = ax.fill_between(x_vals[:frame], y_fill, color='skyblue', alpha=0.5)
    F_x = 1 - np.exp(-lam * x)
    text_F.set_text(f'F(x) ≈ {F_x:.3f}')
    text_1mF.set_text(f'1 - F(x) ≈ {1 - F_x:.3f}')
    return line_f, fill_F, text_F, text_1mF

ani = animation.FuncAnimation(
    fig, update,
    frames=range(10, len(x_vals), 5),
    init_func=init,
    blit=False,
    interval=200
)

ani.save("exponential_animation.gif", writer='pillow', fps=10)

plt.show()
```

V 부록 #5 - Python 코드

- (그림 5)

```python
import numpy as np
import matplotlib.pyplot as plt

# Time range: 0 to 10 minutes
x = np.arange(0, 11)
lambdas = [1, 0.5, 1/3]  # Mean intervals: 1, 2, 3 minutes
labels = ['1 min', '2 min', '3 min']
markers = ['D', 's', '^']
colors = ['blue', 'orange', 'green']

# Plot
plt.figure(figsize=(8, 6))

for lam, label, marker, color in zip(lambdas, labels, markers, colors):
    fx = lam * np.exp(-lam * x)
    plt.plot(x, fx, label=label, marker=marker, markersize=8,
             color=color, linewidth=2)

# Axis settings
plt.xlabel('Waiting Time (minutes)', fontsize=14)
plt.ylabel('Probability Density', fontsize=14)
plt.title('PDF of Waiting Time Until Bus Arrival', fontsize=16)
plt.xticks(np.arange(0, 11, 1), fontsize=12)
plt.yticks(np.linspace(0, 1.2, 7), fontsize=12)
plt.ylim(0, 1.2)
plt.grid(True)
plt.legend(title='Mean Interval', fontsize=12, title_fontsize=13)
plt.tight_layout()
plt.show()
```

V 부록 #6 - Python 코드

- (그림 6)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from scipy.stats import norm

# 정규분포 파라미터
mu = 0
sigma = 1

# 샘플 수 단계
sample_sizes = [5, 10, 50, 100, 500, 1000, 2000, 5000, 10000, 20000, 100000, 10000000]
# x축 범위 및 PDF
x = np.linspace(-4, 4, 1000)
pdf = norm.pdf(x, mu, sigma)

# figure 및 기본 설정
fig, ax = plt.subplots(figsize=(9, 6))

# 히스토그램 초기화 (빈 막대)
hist_plot = ax.bar(x, np.zeros_like(x), width=0.25, alpha=0.6, label='Histogram', color='skyblue')

# 정규분포 PDF 라인
pdf_line, = ax.plot(x, pdf, 'r-', lw=2, label='Normal PDF')

# 제목, 축
ax.set_title('Histogram Converging to Normal Distribution', fontsize=18)
ax.set_xlabel('x', fontsize=16)
ax.set_ylabel('Density', fontsize=16)

# 샘플 수 텍스트 (legend 아래 고정)
text_size = ax.text(0.98, 0.78, '', transform=ax.transAxes,
                    fontsize=14, ha='right', va='top')

# 범위 및 눈금
ax.set_xlim(-4, 4)
ax.set_ylim(0, 0.5)
ax.tick_params(labelsize=12)
ax.legend(loc='upper right', fontsize=13)

# 초기화 함수
def init():
    for rect in hist_plot:
        rect.set_height(0)
    text_size.set_text('')
    return hist_plot + (pdf_line,) + (text_size,)

# 업데이트 함수
def update(frame):
    size = sample_sizes[frame]
    samples = np.random.normal(mu, sigma, size)
    counts, bins = np.histogram(samples, bins=30, range=(-4, 4), density=True)
    bin_centers = 0.5 * (bins[:-1] + bins[1:])

    # 막대 그래프 갱신
    for i, rect in enumerate(hist_plot):
        if i < len(bin_centers):
            rect.set_x(bin_centers[i])
            rect.set_height(counts[i])
        else:
            rect.set_height(0)
    text_size.set_text(f'Sample size: {size}')
    return hist_plot + (pdf_line,) + (text_size,)

# 애니메이션 실행
ani = animation.FuncAnimation(
    fig, update,
    frames=len(sample_sizes),
    init_func=init,
    interval=1000,
    repeat=False
)

# GIF 저장
ani.save("normal_convergence.gif", writer='pillow', fps=1)

plt.show()
```

V 부록 #7 - Python 코드

- (그림 7)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# 평균과 표준편차
mu = 170
sigma = 10

# x 값: 115 ~ 225 (간격 10) + 170 추가
x_base = np.arange(115, 226, 10)
x = np.append(x_base, 170)
x = np.sort(x)  # 정렬하여 선이 매끄럽게 이어지도록
y = norm.pdf(x, mu, sigma)

# 그래프 그리기
plt.figure(figsize=(9, 5))
plt.plot(x, y, marker='D', linestyle='-', color='black', linewidth=1.5, markersize=6)

# 축 설정
plt.xlabel('x', fontsize=14)
plt.ylabel('Probability Density', fontsize=14)
plt.title('PDF of Normal Random Variable (μ=170, σ=10)', fontsize=16)
plt.xticks(np.arange(115, 226, 10), fontsize=12)
plt.yticks(np.linspace(0, 0.045, 10), fontsize=12)
plt.ylim(0, 0.045)
plt.grid(True)

plt.tight_layout()
plt.show()
```

V 부록 #8 - Python 코드

- (그림 8)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# X ~ N(170, 10^2)
mu = 170
sigma = 10
x = np.linspace(120, 220, 1000)
fx = norm.pdf(x, mu, sigma)

plt.figure(figsize=(6, 4))
plt.plot(x, fx, color='blue', linewidth=2)
plt.title(r'$f_X(x)$: PDF of $X \sim \mathcal{N}(170, 100)$', fontsize=16)
plt.xlabel('x', fontsize=14)
plt.ylabel('Density', fontsize=14)
plt.grid(True)
plt.tight_layout()
plt.show()
```

V 부록 #9 - Python 코드

- (그림 9)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# X ~ N(170, 100)
mu_X = 170
sigma_X = 10

# 선형변환 Y = aX + β
a = 2
beta = 5
mu_Y = a * mu_X + beta       # 345
sigma_Y = abs(a) * sigma_X   # 20

# x축 통합 범위 (넓게 설정)
x_all = np.linspace(100, 470, 1000)

# f_X(x)
fx = norm.pdf(x_all, mu_X, sigma_X)

# f_Y(y) = (1/a) * f_X((y - beta)/a)
inv_x = (x_all - beta) / a
fy = norm.pdf(inv_x, mu_X, sigma_X) / abs(a)

# 그래프
plt.figure(figsize=(10, 5))
plt.plot(x_all, fx, label=r'$f_X(x) \sim \mathcal{N}(170,\ 100)$', color='blue', linewidth=2)
plt.plot(x_all, fy, label=r'$f_Y(y) = \dfrac{1}{a} f_X\left( \dfrac{y - \beta}{a} \right)$', color='green',
linewidth=2)

# 평균선 시각화
plt.axvline(mu_X, color='blue', linestyle='--', alpha=0.5)
plt.axvline(mu_Y, color='green', linestyle='--', alpha=0.5)

# 식 텍스트 추가 (그래프 내 빈 공간에 삽입)
plt.text(370, 0.015, r'$Y = aX + \beta$', fontsize=14)

# 그래프 기본 설정
plt.title('Comparison of $f_X(x)$ and Transformed $f_Y(y)$', fontsize=16)
plt.xlabel('Common axis (x or y)', fontsize=14)
plt.ylabel('Density', fontsize=14)
plt.grid(True)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()
```

V 부록 #10 - Python 코드

- (그림 10)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from scipy.stats import norm

# 시작 파라미터
mu_start = 2.0
sigma_start = 1.5

# 목표: 표준정규분포
mu_end = 0.0
sigma_end = 1.0

# x축 범위
x_vals = np.linspace(-6, 6, 500)
std_pdf = norm.pdf(x_vals, mu_end, sigma_end)

# figure 생성
fig, ax = plt.subplots(figsize=(9, 4))
line, = ax.plot([], [], color='blue', label='Transforming Normal')
std_line, = ax.plot(x_vals, std_pdf, 'k--', label='Standard Normal')

# 수식 텍스트 고정
formula_text = ax.text(0.02, 0.9, r'$Z = \frac{X - \mu}{\sigma}$', transform=ax.transAxes, fontsize=13)
text_mu = ax.text(0.02, 0.82, '', transform=ax.transAxes)
text_sigma = ax.text(0.02, 0.75, '', transform=ax.transAxes)

# μ와 ±σ 표시용 수직선
mu_line, = ax.plot([], [], 'g--', label='μ')
sigma_left, = ax.plot([], [], 'r--', label='μ - σ')
sigma_right, = ax.plot([], [], 'r--', label='μ + σ')

# 축 설정
ax.set_xlim(-6, 6)
ax.set_ylim(0, 0.5)
ax.set_xlabel('x')
ax.set_ylabel('Density')
ax.set_title('Standardizing Normal Distribution: $(X - \mu)/\sigma$')
ax.legend(loc='upper right')
```

```python

# 초기화 함수
def init():
    line.set_data([], [])
    mu_line.set_data([], [])
    sigma_left.set_data([], [])
    sigma_right.set_data([], [])
    text_mu.set_text('')
    text_sigma.set_text('')
    return line, std_line, mu_line, sigma_left, sigma_right, text_mu, text_sigma

# 업데이트 함수
def update(frame):
    alpha = frame / 100  # 0~1
    mu = (1 - alpha) * mu_start + alpha * mu_end
    sigma = (1 - alpha) * sigma_start + alpha * sigma_end

    y_vals = norm.pdf(x_vals, mu, sigma)
    line.set_data(x_vals, y_vals)

    # 수직선 위치 갱신
    mu_line.set_data([mu, mu], [0, norm.pdf(mu, mu, sigma)])
    sigma_left.set_data([mu - sigma, mu - sigma], [0, norm.pdf(mu - sigma, mu, sigma)])
    sigma_right.set_data([mu + sigma, mu + sigma], [0, norm.pdf(mu + sigma, mu, sigma)])

    text_mu.set_text(f'μ ≈ {mu:.2f}')
    text_sigma.set_text(f'σ ≈ {sigma:.2f}')
    return line, std_line, mu_line, sigma_left, sigma_right, text_mu, text_sigma

# 애니메이션 실행 (blit=False로 변경!)
ani = animation.FuncAnimation(
    fig, update,
    frames=range(0, 101, 2),
    init_func=init,
    interval=100,
    blit=False
)

ani.save("f10.gif", writer='pillow', fps=10)

```

V 부록 #11 - Python 코드

- (그림 11)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# 설정: 표준정규분포 Z ~ N(0, 1)
mu = 0
sigma = 1
z_val = 1.0  # 예시로 z = 1

# x 범위 및 PDF
x = np.linspace(-4, 4, 1000)
pdf = norm.pdf(x, mu, sigma)

# Φ(z) 계산
phi_z = norm.cdf(z_val)

# 그래프
plt.figure(figsize=(8, 5))
plt.plot(x, pdf, label=r'$f_Z(t)$', color='black', linewidth=2)

# 누적 면적 shading: (-∞, z]
x_fill = np.linspace(-4, z_val, 500)
plt.fill_between(x_fill, norm.pdf(x_fill), color='skyblue', alpha=0.5, label=rf'$\Phi({z_val}) = {phi_z:.3f}$')

# 수직선 z
plt.axvline(z_val, color='red', linestyle='--')
plt.text(z_val + 0.1, 0.05, rf'$z = {z_val}$', fontsize=12, color='red')

# 설정
plt.title(r'$\Phi(z)$ as Area Under $f_Z(t)$', fontsize=16)
plt.xlabel('t', fontsize=14)
plt.ylabel('Density', fontsize=14)
plt.grid(True)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()
```

V 부록 #12 - Python 코드

- (그림 12)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# 설정
mu = 0
sigma = 1
z_val = 1.0

# x 범위 및 pdf
x = np.linspace(-4, 4, 1000)
pdf = norm.pdf(x, mu, sigma)

# 값 계산
phi_z = norm.cdf(z_val)
phi_H_z = phi_z - 0.5  # 편측 누적분포

# 그래프 시작
plt.figure(figsize=(9, 5))
plt.plot(x, pdf, color='black', linewidth=2, label=r'$f_Z(t)$')

# 전체 누적분포 Φ(z): (-∞, z]
x_fill_phi = np.linspace(-4, z_val, 500)
plt.fill_between(x_fill_phi, norm.pdf(x_fill_phi), color='skyblue', alpha=0.5, label=rf'$\Phi({z_val}) = {phi_z:.3f}$')

# 편측 누적분포 Φ^H(z): (0, z]
x_fill_phi_H = np.linspace(0, z_val, 300)
plt.fill_between(x_fill_phi_H, norm.pdf(x_fill_phi_H), color='limegreen', alpha=0.5, label=rf'$\Phi^H({z_val}) = {phi_H_z:.3f}$')

# 수직선 표시
plt.axvline(0, color='gray', linestyle='--', alpha=0.6)
plt.axvline(z_val, color='red', linestyle='--')
plt.text(z_val + 0.1, 0.05, rf'$z = {z_val}$', fontsize=12, color='red')

# 설정
plt.title(r'Visualizing $\Phi(z)$ and $\Phi^H(z)$ on Standard Normal PDF', fontsize=16)
plt.xlabel('t', fontsize=14)
plt.ylabel('Density', fontsize=14)
plt.legend(fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```

V 부록 #13 - Python 코드

- (그림 13)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# 평균과 두 가지 표준편차
mu = 160
sigma1 = 10
sigma2 = 30

# 기준값
threshold = 190

# x 축 범위
x = np.linspace(100, 250, 1000)

# 두 정규분포의 PDF
pdf1 = norm.pdf(x, mu, sigma1)
pdf2 = norm.pdf(x, mu, sigma2)

# 그래프 그리기
plt.figure(figsize=(10, 5))
plt.plot(x, pdf1, label=r'$\sigma=10$', color='blue')
plt.plot(x, pdf2, label=r'$\sigma=30$', color='green')

# 기준값보다 클 확률 부분 색칠
x_fill1 = x[x > threshold]
x_fill2 = x[x > threshold]
plt.fill_between(x_fill1, norm.pdf(x_fill1, mu, sigma1), color='blue', alpha=0.3)
plt.fill_between(x_fill2, norm.pdf(x_fill2, mu, sigma2), color='green', alpha=0.3)

# 기준선
plt.axvline(x=threshold, color='red', linestyle='--', linewidth=1.5, label='Threshold = 190cm')
plt.title('Probability of Height > 190cm under Different Standard Deviations', fontsize=14)
plt.xlabel('Height (cm)', fontsize=12)
plt.ylabel('Density', fontsize=12)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

V 부록 #14 - Python 코드

- (그림 14)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

mu = 100
sigma = 20
x = np.linspace(mu - 4*sigma, mu + 4*sigma, 1000)
y = norm.pdf(x, mu, sigma)

x_thresh = norm.ppf(0.93, loc=mu, scale=sigma)
y_thresh = norm.pdf(x_thresh, mu, sigma)
x_fill = np.linspace(x_thresh, mu + 4*sigma, 1000)
y_fill = norm.pdf(x_fill, mu, sigma)

plt.figure(figsize=(10, 6))
plt.plot(x, y, label='Normal Distribution', color='black')
plt.fill_between(x_fill, y_fill, color='orange', alpha=0.6, label='Top 7%')

plt.vlines(x=x_thresh, ymin=0, ymax=y_thresh, color='red', linestyle='--', linewidth=2)
plt.text(x_thresh + 1, y_thresh / 2, f'{x_thresh:.2f} ms', color='red', fontsize=12)

plt.title('Normal Distribution: μ=100ms, σ=20ms', fontsize=14)
plt.xlabel('Response Time (ms)')
plt.ylabel('Probability Density')
plt.legend()
plt.grid(True)
plt.show()
```

V 부록 #15 - Python 코드

- (그림 15)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from matplotlib.patches import Circle
from math import comb

# 설정
n = 100
p = 0.5

# 이항분포 계산
k_vals = np.arange(n + 1)
binom_pmf = [comb(n, k) * p**k * (1 - p)**(n - k) for k in k_vals]

fig, (ax_left, ax_right) = plt.subplots(1, 2, figsize=(14, 6))

# 왼쪽: 10x10 그리드 점 생성
circle_objs = []
for i in range(10):
    for j in range(10):
        idx = i * 10 + j
        c = Circle((j, 9 - i), 0.3, color='lightgray')
        ax_left.add_patch(c)
        circle_objs.append(c)

ax_left.set_xlim(-1, 10)
ax_left.set_ylim(-1, 10)
ax_left.set_aspect('equal')
ax_left.axis('off')
title_left = ax_left.set_title('')

# 수식 텍스트 위치 (점 밑)
formula_text = ax_left.text(5, -1.3, '', ha='center', fontsize=12)
value_text = ax_left.text(5, -1.8, '', ha='center', fontsize=12)

# 오른쪽: 점 그래프
dot_objs = ax_right.plot([], [], 'ro')[0]
x_data, y_data = [], []

ax_right.set_xlim(-1, n + 1)
ax_right.set_ylim(0, max(binom_pmf) * 1.2)
ax_right.set_xlabel('Number of Successes (k)')
ax_right.set_ylabel('P(X = k)')
ax_right.set_title(f'Binomial Distribution (n={n}, p={p})')
```

```python
# 업데이트 함수
def update(frame):
    k = frame
    if k > n or k >= 100:
        return

    # 색상 변경
    circle_objs[k].set_color('red')

    # 수식 계산 및 표시
    binom_coeff = comb(n, k)
    prob = binom_coeff * (p ** k) * ((1 - p) ** (n - k))
    formula_text.set_text(
    rf'$P(X={k}) = \binom{{{n}}}{{{k}}} \cdot {p}^{{{k}}} \cdot {(1 - p):.1f}^{{{n - k}}} = {prob:.30f}$')


    # 점 추가
    x_data.append(k)
    y_data.append(prob)
    dot_objs.set_data(x_data, y_data)

    return circle_objs + [dot_objs]

ani = animation.FuncAnimation(
    fig, update,
    frames=range(101),  # 0부터 100까지 포함되도록 수정
    interval=100,
    blit=False,
    repeat=False
)

ani.save("f14.gif", writer='pillow', fps=10)
```

V 부록 #16 - Python 코드

- (그림 16)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from matplotlib.patches import Circle
from math import pow

# 파라미터 설정
p = 0.3
max_k = 20
geom_pmf = [pow(1 - p, k - 1) * p for k in range(1, max_k + 1)]

# Figure 및 서브플롯 생성
fig = plt.figure(figsize=(10, 6))
gs = fig.add_gridspec(3, 1, height_ratios=[1, 0.3, 1])

# === [1] 윗줄: 시도 점 표시 ===
ax_circles = fig.add_subplot(gs[0])
circle_objs = []
for i in range(max_k):
    circ = Circle((i + 1, -0.6), 0.3, color='lightgray')
    ax_circles.add_patch(circ)
    circle_objs.append(circ)

ax_circles.set_xlim(0, max_k + 1)
ax_circles.set_ylim(-1, 1)
ax_circles.set_aspect('equal')
ax_circles.axis('off')
ax_circles.set_title('Geometric Distribution (first success after k trials)')

# === [2] 중간: 수식 ===
ax_text = fig.add_subplot(gs[1])
ax_text.axis('off')
formula_text = ax_text.text(0.5, 1, '', ha='center', va='center', fontsize=14)

# === [3] 하단: 확률 막대그래프 ===
ax_bar = fig.add_subplot(gs[2])
bars = ax_bar.bar(range(1, max_k + 1), [0] * max_k, color='gray')
ax_bar.set_ylim(0, max(geom_pmf) * 1.2)
ax_bar.set_xlim(0, max_k + 1)
ax_bar.set_xticks(np.arange(1, max_k + 1, 1))
ax_bar.set_xlabel('Trial number of first success (k)')
ax_bar.set_ylabel('P(X = k)')
ax_bar.set_title(f'Geometric Distribution PMF (p = {p})')

# === 애니메이션 업데이트 함수 ===
def update(k):
    if k >= max_k:
        return

    for c in circle_objs:
        c.set_color('lightgray')
    for i in range(k):
        circle_objs[i].set_color('blue')  # 실패
    circle_objs[k].set_color('red')      # 성공

    prob = pow(1 - p, k) * p
    formula_text.set_text(
        rf'$P(X={k+1}) = (1 - {p})^{{{k}}} \cdot {p} = {prob:.4f}$'
    )

    for i, bar in enumerate(bars):
        bar.set_height(geom_pmf[i])
        bar.set_color('red' if i == k else 'gray')

    return circle_objs + list(bars)

# === 애니메이션 실행 ===
ani = animation.FuncAnimation(
    fig, update,
    frames=range(max_k),
    interval=500,
    repeat=False
)

ani.save("f15.gif", writer='pillow', fps=5)
```

V 부록 #17 - Python 코드

- (그림 17)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import poisson

def static_poisson_visualizations():
    lambda_vals = [1, 4, 10]
    k = np.arange(0, 20)

    fig, axs = plt.subplots(1, 2, figsize=(12, 5))
    fig.suptitle("Understanding the Poisson Distribution")

    # PMF for different λ
    for lam in lambda_vals:
        pmf = poisson.pmf(k, mu=lam)
        axs[0].plot(k, pmf, marker='o', label=f'λ = {lam}')
    axs[0].set_title("Probability Mass Function (PMF)")
    axs[0].set_xlabel("Number of events (k)")
    axs[0].set_ylabel("P(X = k)")
    axs[0].legend()
    axs[0].grid(True)

    # Simulation: histogram vs PMF
    np.random.seed(42)
    sim_lambda = 4
    samples = np.random.poisson(sim_lambda, 10000)
    axs[1].hist(samples, bins=range(0, 15), density=True, alpha=0.7,
                color='orange', edgecolor='black', label='Simulated Data')
    axs[1].plot(k, poisson.pmf(k, mu=sim_lambda), 'o-', color='red', label='PMF')
    axs[1].set_title(f"Sample Histogram vs PMF (λ = {sim_lambda})")
    axs[1].set_xlabel("Number of events (k)")
    axs[1].set_ylabel("Frequency (normalized)")
    axs[1].legend()
    axs[1].grid(True)

    plt.tight_layout()
    plt.show()

static_poisson_visualizations()
```

V 부록 #18 - Python 코드

- (그림 18)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

# Parameters
lambda_vals = [2.5, 1]
t_end = 10
seeds = [0, 42]

# Generate event times
event_times = []
for lam, seed in zip(lambda_vals, seeds):
    rng = np.random.default_rng(seed)
    t = 0
    times = []
    while t < t_end:
        t += rng.exponential(1 / lam)
        if t < t_end:
            times.append(t)
    event_times.append(np.array(times))

# Animation settings
frames_per_second = 5
total_frames = t_end * frames_per_second

# Set up figure
fig, axs = plt.subplots(2, 1, figsize=(12, 6), sharex=True)
titles = [f"λ = {lam}" for lam in lambda_vals]

for ax, title in zip(axs, titles):
    ax.set_xlim(0, t_end)
    ax.set_ylim(-2, 3)
    ax.set_yticks([])
    ax.hlines(0, 0, t_end, colors='black')
    ax.set_title(f"Poisson Events and Exponential Intervals ({title})")
    ax.set_xlabel("Time")

# Plot elements
dots_list = []
curves_list = [[] for _ in axs]
count_texts = []

for ax in axs:
    dot, = ax.plot([], [], 'ro')
    dots_list.append(dot)
    count_text = ax.text(t_end * 0.8, 2, '', fontsize=12)
    count_texts.append(count_text)

# Init
def init():
    for dot in dots_list:
        dot.set_data([], [])
    for txt in count_texts:
        txt.set_text('')
    return dots_list + count_texts

# Update
def update(frame):
    current_time = frame / frames_per_second

    for i, times in enumerate(event_times):
        visible_times = times[times <= current_time]
        dots_list[i].set_data(visible_times, np.zeros_like(visible_times))
        count_texts[i].set_text(f"Events: {len(visible_times)}")

        # Draw curves only once per interval
        while len(curves_list[i]) < len(visible_times) - 1:
            j = len(curves_list[i])
            x0, x1 = visible_times[j], visible_times[j + 1]
            t_curve = np.linspace(x0, x1, 100)
            curve_y = -0.5 * np.sin(np.pi * (t_curve - x0) / (x1 - x0))
            curve, = axs[i].plot(t_curve, curve_y, 'b-')
            curves_list[i].append(curve)

    return []

# Run animation
ani = animation.FuncAnimation(
    fig, update, frames=total_frames, init_func=init,
    blit=False, repeat=False, interval=200
)

ani.save("f17.gif", writer='pillow', fps=5)
```

V 부록 #19 - Python 코드

- (그림 19)

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

# Parameters
lambda_val = 1.2
t_end = 10
rng = np.random.default_rng(seed=1)

# Generate exponential inter-arrival times and arrival moments
inter_arrivals = rng.exponential(1 / lambda_val, size=100)
arrival_times = np.cumsum(inter_arrivals)
arrival_times = arrival_times[arrival_times <= t_end]
n_events = len(arrival_times)

# Set up figure and axes
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 6), sharex=True)
fig.suptitle("Exponential Intervals → Poisson Event Count")

# Top axis: event timeline
ax1.set_xlim(0, t_end)
ax1.set_ylim(-1, 1)
ax1.set_yticks([])
ax1.set_ylabel("Event Interval")
ax1.hlines(0, 0, t_end, color='black')
dots_line, = ax1.plot([], [], 'ro')
curves = []

# Bottom axis: cumulative event count
ax2.set_xlim(0, t_end)
ax2.set_ylim(0, n_events + 2)
ax2.set_ylabel("Event Count")
ax2.set_xlabel("Time")
ax2.grid(True)
count_line, = ax2.step([], [], where='post', color='green')
count_text = ax2.text(0.7 * t_end, n_events + 1, '', fontsize=12)

# Init function
def init():
    dots_line.set_data([], [])
    count_line.set_data([], [])
    count_text.set_text('')
    for c in curves:
        c.remove()
    curves.clear()
    return [dots_line, count_line, count_text]

# Update function
def update(frame):
    t = arrival_times[:frame + 1]
    y = np.zeros_like(t)
    dots_line.set_data(t, y)

    # Draw blue interval curve
    if frame > 0:
        x0, x1 = arrival_times[frame - 1], arrival_times[frame]
        t_curve = np.linspace(x0, x1, 100)
        curve_y = 0.3 * np.sin(np.pi * (t_curve - x0) / (x1 - x0))
        curve, = ax1.plot(t_curve, curve_y, 'b-')
        curves.append(curve)

    # Update step plot for Poisson count
    steps_x = np.concatenate([[0], arrival_times[:frame + 1], [t_end]])
    steps_y = np.concatenate([[0], np.arange(1, frame + 2), [frame + 1]])
    count_line.set_data(steps_x, steps_y)
    count_text.set_text(f"Events: {frame + 1}")

    return [dots_line, count_line, count_text] + curves

# Create animation
ani = animation.FuncAnimation(
    fig, update, frames=n_events, init_func=init,
    blit=True, repeat=False, interval=3000
)

ani.save("f18.gif", writer='pillow', fps=3)
```

V 부록 #20 - Python 코드

- (그림 20)

```python
import numpy as np
import matplotlib.pyplot as plt

# 설정: X ~ 지수분포(lambda=1), 평균 E[X] = 1
a_vals = np.linspace(0.1, 5, 100)
true_prob = np.exp(-a_vals)
markov_bound = 1 / a_vals

# 상대 오차 비율
relative_error = markov_bound / true_prob  # = e^a / a

# 첫 번째 그래프: 실제 확률 vs 마르코프 상한
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
plt.plot(a_vals, true_prob, label=r'$P(X \geq a)$', color='blue')
plt.plot(a_vals, markov_bound, label=r'Markov Bound $E[X]/a$', linestyle='--', color='red')
plt.xlabel('a')
plt.ylabel('Probability')
plt.title('True Probability vs Markov Bound')
plt.legend()
plt.grid(True)

# 두 번째 그래프: 상대 오차 비율
plt.subplot(1, 2, 2)
plt.plot(a_vals, relative_error, color='purple')
plt.xlabel('a')
plt.ylabel(r'Relative Error $\frac{1/a}{e^{-a}} = \frac{e^a}{a}$')
plt.title('Relative Error of Markov Bound')
plt.grid(True)

plt.tight_layout()
plt.show()
```

V 부록 #21 - Python 코드

- (그림 21)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad

# f(x) = e^{-x}, 지수분포 (λ=1)
f = lambda x: np.exp(-x)
xf = lambda x: x * np.exp(-x)

a_vals = np.linspace(0.1, 5, 100)
lhs_vals = []
rhs_vals = []

for a in a_vals:
    left_part = quad(xf, 0, a)[0]
    right_part = quad(xf, a, np.inf)[0]
    lhs_vals.append(left_part + right_part)
    rhs_vals.append(right_part)

plt.figure(figsize=(10, 6))
plt.plot(a_vals, lhs_vals, label=r'$\int_0^a x f(x)\,dx + \int_a^\infty x f(x)\,dx$', lw=2, color='orange')
plt.plot(a_vals, rhs_vals, label=r'$\int_a^\infty x f(x)\,dx$', lw=2, color='brown')
plt.fill_between(a_vals, lhs_vals, rhs_vals, color='lightblue', alpha=0.4, label='Gap')
plt.xlabel('a', fontsize=14)
plt.ylabel('Value', fontsize=14)
plt.title(r'Visualization of $\int_0^a xf(x)\,dx + \int_a^\infty xf(x)\,dx \geq \int_a^\infty xf(x)\,dx$', fontsize=14)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

V 부록 #22 - Python 코드

- (그림 22)

```python
import numpy as np
import matplotlib.pyplot as plt

x_discrete = np.arange(0, 101)
prob = 1 / len(x_discrete)

fig, ax = plt.subplots(figsize=(6, 2.5))
ax.bar(x_discrete, [prob] * len(x_discrete), color='lightgreen', edgecolor='black')
ax.set_xlim(-5, 105)
ax.set_ylim(0, 0.013)
ax.set_title(r"Uniform Distribution with $\mu$, $\sigma^2$", fontsize=13)
ax.set_xlabel("x")
ax.set_yticks([])  # y축 눈금 제거
ax.spines['left'].set_position('zero')  # y축 0 위치로 이동
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.grid(False)
plt.show()
```

V 부록 #23 - Python 코드

- (그림 23) (1/2)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gaussian_kde

x_discrete = np.arange(0, 101)
np.random.seed(42)
sample_10 = np.random.choice(x_discrete, size=10, replace=False)
x_vals = np.linspace(0, 100, 500)

kde = gaussian_kde(sample_10, bw_method=0.2)
y_vals = kde(x_vals)

fig, ax = plt.subplots(figsize=(6, 2.5))
ax.fill_between(x_vals, y_vals, color='lightgreen')
ax.plot(x_vals, y_vals, color='black', linewidth=1)
ax.set_xlim(0, 100)
ax.set_ylim(0, 0.03)
ax.margins(y=0.03)

ax.set_xlabel("x")
ax.set_yticks([])
ax.spines['left'].set_color('none')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.grid(False)

plt.tight_layout()
plt.show()
```

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gaussian_kde

x_discrete = np.arange(0, 101)
np.random.seed(42)
sample_10 = np.random.choice(x_discrete, size=80, replace=False)
x_vals = np.linspace(0, 100, 500)

kde = gaussian_kde(sample_10, bw_method=0.2)
y_vals = kde(x_vals)

fig, ax = plt.subplots(figsize=(6, 2.5))
ax.fill_between(x_vals, y_vals, color='lightgreen')
ax.plot(x_vals, y_vals, color='black', linewidth=1)
ax.set_xlim(0, 100)
ax.set_ylim(0, 0.02)
ax.margins(y=0.03)

ax.set_xlabel("x")
ax.set_yticks([])
ax.spines['left'].set_color('none')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.grid(False)

plt.tight_layout()
plt.show()
```

V 부록 #24 - Python 코드

- (그림 23) (2/2)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gaussian_kde

x_discrete = np.arange(0, 101)
np.random.seed(42)
sample_10 = np.random.choice(x_discrete, size=101, replace=False)
x_vals = np.linspace(0, 100, 500)

kde = gaussian_kde(sample_10, bw_method=0.07)
y_vals = kde(x_vals)

fig, ax = plt.subplots(figsize=(6, 2.5))
ax.fill_between(x_vals, y_vals, color='lightgreen')
ax.plot(x_vals, y_vals, color='black', linewidth=1)
ax.set_xlim(0, 100)
ax.set_ylim(0, 0.02)
ax.margins(y=0.03)

ax.set_xlabel("x")
ax.set_yticks([])
ax.spines['left'].set_color('none')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.grid(False)

plt.tight_layout()
plt.show()
```

V 부록 #25 - Python 코드

- (그림 24)

```python
import numpy as np
import matplotlib.pyplot as plt

mu = 50
sigma = 10
x = np.linspace(mu - 4*sigma, mu + 4*sigma, 500)
y = 1 / (sigma * np.sqrt(2 * np.pi)) * np.exp(- (x - mu)**2 / (2 * sigma**2))

fig, ax = plt.subplots(figsize=(6, 4))
ax.plot(x, y, color='blue', linewidth=3)

ax.vlines(mu, 0, max(y), color='black', linewidth=1.5)
ax.hlines(0, mu - 4*sigma, mu + 4*sigma, color='black', linewidth=1)
ax.text(mu - 6, max(y)*0.6, r'$\bar{x}_i$', fontsize=16)
ax.annotate(r'$\sigma_{\bar{x}_i} = \frac{\sigma_{x_i}}{\sqrt{n}}$',
            xy=(mu, max(y)*0.6),
            xytext=(mu + 15, max(y)*0.7),
            arrowprops=dict(arrowstyle='->', color='black'),
            fontsize=14)

ax.text(mu, -0.003, r'$\mu_{\bar{x}_i} = 50 = \mu_{x_i}$',
        fontsize=15, ha='center', va='top')

# 축 설정
ax.set_xlim(mu - 4*sigma, mu + 4*sigma)
ax.set_ylim(-0.005, max(y)*1.1)
ax.axis('off')

plt.tight_layout()
plt.show()
```

V 부록 #26 - Python 코드

- (그림 25)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binom, norm

# Parameters
p = 0.5
n_list = [10, 30, 100]

# 공통 ymax 계산
ymax = max([binom.pmf(int(n * p), n, p) for n in n_list]) * 1.1

fig, axs = plt.subplots(1, 3, figsize=(18, 5), sharey=True)

for i, n in enumerate(n_list):
    mu = n * p
    sigma = np.sqrt(n * p * (1 - p))

    # x축 범위 통일: [μ - 4σ, μ + 4σ]
    x_min = int(mu - 4 * sigma)
    x_max = int(mu + 4 * sigma)
    x = np.arange(x_min, x_max + 1)

    # Binomial PMF (범위 자르는 것 주의)
    pmf = binom.pmf(x, n, p)
    axs[i].bar(x, pmf, alpha=0.6, label='Binomial', color='lightblue', edgecolor='black')

    # Normal approximation PDF
    x_norm = np.linspace(x_min, x_max, 500)
    pdf = norm.pdf(x_norm, mu, sigma)
    axs[i].plot(x_norm, pdf, color='red', lw=2, label='Normal Approx.')

    axs[i].set_title(f'n = {n}', fontsize=14)
    axs[i].set_xlabel('Number of successes (k)')
    axs[i].set_xlim(x_min, x_max)
    axs[i].set_ylim(0, ymax)
    axs[i].legend()
    axs[i].grid(True)

axs[0].set_ylabel('Probability')

plt.suptitle('Comparison of Binomial Distribution and Normal Approximation', fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```