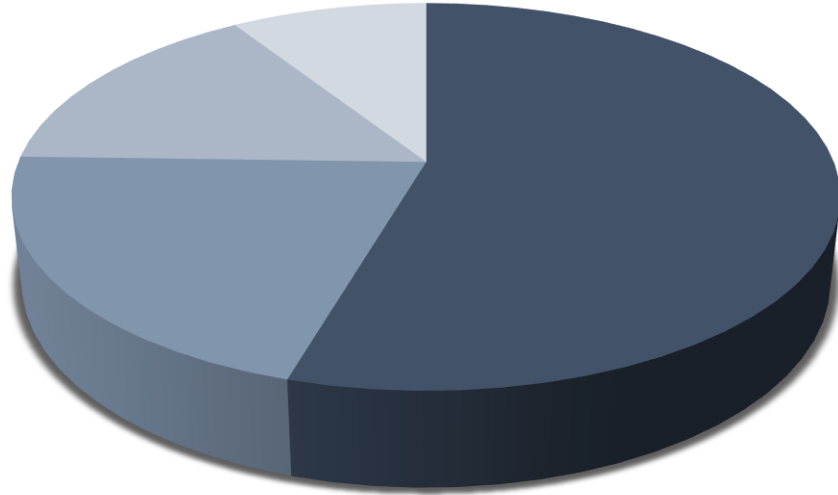


The background of the slide features a variety of 3D statistical charts. On the left, there is a large 3D pie chart with several slices in different colors (yellow, green, blue, purple, orange). To the right of the pie chart, there are several 3D bar charts of varying heights and colors (yellow, orange, pink, purple, blue, green). In the foreground, there is a 3D line chart with a yellow line and a 3D bar chart with blue bars. The overall theme is data visualization.

통계학 실습

15. 시뮬레이션 코딩



I. Simulation Coding

- 시뮬레이션 코딩이란 어떠한 사건과 행동에 대한 결과를 예측하고 프로그래밍을 통하여 확인해보면서 실제 사건이 일어나는 것과 근접하게 만들어 결과를 확인해 보는 것이다.
- 실제로 실험해보기 힘든 것들을 프로그래밍을 통하여 확인해볼 수 있다.
 - 자동차 사고 시뮬레이션
 - 네트워크 패킷 생성률

- 공장에서 제품을 생산한다. 생산라인 하나당 하루에 100개의 제품을 생산할 수 있다. 제품 하나가 만들어지는 와중에 결함이 생길 확률이 3% 라고 한다. 생산 라인 1000개를 가동했을 때, 생산된 제품 전체에서 결함품이 차지하는 비율이 얼마나 되는지 알아보자.

제품 생산 함수 코딩

제품을 생산하여 3% 확률로 결함을 가지게 된다. 결함 : 0, 성공 : 1

```
product <- function() {  
  fault <- sample(1:100, 1)  
  if (fault < 4) {  
    return(0)  
  }  
  else {  
    return(1)  
  }  
}
```

생산 라인 함수 코딩

라인당 100개의 제품을 생산하고 모든 제품의 합을 통해 전체에서의 비율을 확인하는 함수

```
lines <- function(n) {  
  faults <- 0  
  for (i in 1:n) {  
    for (j in 1:100) {  
      if (product() == 0) {  
        faults <- faults + 1  
      }  
    }  
  }  
  return(faults/(n*100)*100)  
}  
# 실행  
lines(1000)
```

- 문제) Tim과 Alice가 주사위를 던져서 내기를 하고자 한다. Tim은 홀수가 나오면 승리하고, Alice는 짝수가 나오면 승리한다. 각 시도에 패자는 승자에게 100원을 줘야 한다. Tim이 5,500원을, Alice가 4,500원을 들고 시작한다. 한 쪽이 모두 돈을 잃을 때 내기가 끝난다고 할 때, Alice가 이길 확률은 얼마나 될지 시뮬레이션 코딩으로 확인해보자.

- 1. 초기설정
 - 출력할 내용에 대한 초기값, 대상이 가지는 초기값
- 2. 각 시도마다의 행동 및 결과
 - 어떤 사건 및 시도에 대한 무작위 결과 설정
- 3. 시도의 반복 실행
 - 2. 에서 정의한 시도의 종료 지점 설정
- 4. 전체 과정을 아우르는 메인 함수 및 출력
- 5. 실행

실습 (1)

- Skeleton 코드를 완성하여 해당 문제를 해결해보자.

```
# 초기화함수
init <- function() {
  tim <- # tim이 처음 들고 있는 돈
  alice <- # Alice가 처음 들고 있는 돈
}
# 시도
trial <- function() {
  result <- # 주사위는 6 숫자 중 숫자 하나가 나와야 한다. sample 함수를 이용하자.

  if ( # 주사위가 홀수면? ) {
    # tim이 이긴다.
  }
  else {
    # alice가 이긴다.
  }
}
```

실습 (2)

- Skeleton 코드를 완성하여 해당 문제를 해결해보자.

```
# 게임내부
ingame <- function() {
  while( # 한 쪽에게 돈이 모두 몰라지면 게임이 끝난다. ) { # 종료되는 시점까지 반복
    trial() # 시행
  }
  if ( # tim이 승자일 경우 ) {
    winner <- c(winner, "tim")
  }
  else {
    winner <- c(winner, "alice")
  }
  init()
}

# n번 만큼 게임 반복
startGame <- function(n) {
  init()
  winner <- c() # 승자가 된 사람의 이름을 저장하는 벡터
  for (i in 1:n) {
    ingame() # n만큼 게임 반복
  }
  # 출력
  prob_win_alice = sum(winner=="alice") / n
  cat("Probability that Alice will win a bet : ", prob_win_alice, "\n");
}

startGame(5000)|
```

실습 (3)

- 완성한 코드의 시뮬레이션을 5000번 돌려서 결과를 확인해보자.

```
> startGame(5000)  
Probability that Alice will win a bet : 0.4514  
> |
```