

AML5103 Odd Semester 2024 Lab Exam-1

Problem-1: suppose you are the head of a team that comprises 5 newly recruited software engineers. All 5 of them are equally skilled and are working on a major project under your guidance for which each one of them works on their respective code modules and merges it with the master project code. All modules have identical workloads. This merger happens on a weekly basis after which you compile the master project code. During a particular week's check, you find 5 compilation errors when compiling the master project code, with 4 of those errors tracing back to the module submitted by software engineer-1. The other software engineers thereafter call software engineer-1 clumsy. However, software engineer-1 claims that it was just bad luck and that any of them could have had their modules resulting in 4 out of the 5 compilation errors that you saw. As the team lead, the challenge for you here is to see if software engineer-1 has some valid statistical support for their position. To that end, you, as the team lead, first hypothesize that software engineer-1 is not clumsy. This means, you assume that any of the 5 software engineers' modules is equally likely to result in a compilation error. Under this hypothesis, using simulation calculate the probability that accused software engineer-1 submits a module that leads to at least 4 of the 5 compilation errors when compiling the master project code. Using the resulting probability, settle the claim as to whether software engineer-1 is clumsy as their colleagues are accusing or actually not.

Sampling space and probability vector

```
s = c(1:5)
p = replicate(length(s), 1/ 5)
```

Function to simulate the one trial of the random experiment

```
simTrial = function(){
return(sample(s, 5, replace = TRUE, prob = p))
}
```

Function to check if software engineer-1's module has resulted in at least 4 out of the 5 compilation errors

```
checkEvent = function(data){
return(sum(data == 1) >= 4)
}
```

Simulate the trials of the random experiment a large number of time and calculate the probability that software engineer-1's module caused at least 4 out of the 5 compilation errors

```
n simulations = 10000
simulatedData = replicate(10000, simTrial())
mean(apply(simulatedData, 2, checkEvent))

## [1] 0.0068
```

Problem-2: Consider the following model for a bus-ridership analysis:

- at each one of the stops stop, each passenger is likely to get off the bus independent of others with the following chances: 80% for stops 1, 2, 3, 8, 9, 10 and 20% for the remaining stops;
- at every stop, there is a 50% / 40% / 10% chance of 0 / 1 / 2 passengers getting on board;
- the bus never gets full; new passengers at any stop can always be accommodated;
- bus is empty when it arrives at the first stop.

Calculate using simulation the probability that the bus is empty after visiting the tenth stop.

Sampling spaces

```
# Sampling space for leaving the bus
s1 = c(1, 0) # 1 corresponds to leaving the bus
p1 = array(c(replicate(3, c(0.8, 0.2)), replicate(4, c(0.2, 0.8)),
replicate(3, c(0.8, 0.2))), dim = c(2, 10))
# each column corresponds to the

#Leaving and staying on probability for that stop

# Sampling space for number of passengers boarding the bus
s2 = c(0, 1, 2)
p2 = c(0.5, 0.4, 0.1)
```

Simulate one trial of the random experiment

```
simTrial = function(nstops){
# Initial number of passengers in the bus
npassengers = 0
# Run through every stop
for (j in 1:nstops){
  # If we have at least one passenger in the bus
  # simulate the leaving process
  if (npassengers > 0){
    left_the_bus = sum(sample(s1, size = npassengers, replace = TRUE, prob =
p1[,j]))
    npassengers = npassengers - left_the_bus
  }
  # Else if no passengers on the bus, then nobody Leaves the bus
  else if (npassengers == 0){
    left_the_bus = 0
  }
}
```

```

# Simulating the boarding process
boarded_the_bus = sample(s2, size = 1, replace = TRUE, prob = p2)
npassengers = npassengers + boarded_the_bus
# Print the status at every stop (uncomment to see what happens in one
trial)
#cat(sprintf('No. of passengers after stop-%d is %d (left_the_bus = %d,
boarded_the_bus = %d)\n', j, npassengers, left_the_bus, boarded_the_bus))
}
return(npassengers)
}

```

Function to check if the bus is empty at the tenth stop

```

checkEvent = function(data){
return(data == 0)
}

```

Simulate the trials of the random experiment a large number of times and calculate the probability that bus will be empty at the tenth stop

```

simulatedData=replicate(10000,simTrial(10))
mean(sapply(simulatedData,checkEvent))

## [1] 0.4166

```