

First Simulation

Dr. Ekstrom

Purpose: Imagine every hitter in a game either hits a homerun (with probability $p = 0.1$) or strikes out (with probability $1 - p = 0.9$), how many runs do we expect the team to score in a nine inning game?

Sports Analytics: Estimate the impact of a strategy.

Statistics: Estimating statistics using a simulation.

R: Simulating in *R*

SIMULATING A SINGLE INNING

We will begin by simulating an inning. The key to our simulation is the `sample()` command. It samples an entry (or multiple entries) from a vector - with the probability of each entry being selected given by a second vector. For example

```
# creates a vector for the results of a batter
# 0 represents a strikeout
# 1 represents a homerun
batter=c(0,1)
# creates a vector that gives the probability of each result
pr=c(0.9,0.1)
# samples 1 element from the batter vector based on the probabilities in pr
sample(batter,1,prob=pr)

## [1] 1
```

The `sample` command randomly selects an element of the `batter` vector - where 0 has a probability of 0.9 of being selected and 1 has a probability of 0.1 of being selected.

In simulating an inning, we need to keep track of the number of runs scored, and the number of outs - the third out ends the inning. So a `while` loop seems appropriate.

```
# probability of a homerun
p=0.1
# creates a vector for the results of a batter
# 0 represents a strikeout
# 1 represents a homerun
batter=c(0,1)
# creates a vector that gives the probability of each result
pr=c(1-p,p)
# keeps track of the number of runs scored
runs=0
# keeps track of the number of outs in the inning
outs=0
# creates a while loop that continues until 3 outs are recorded
while(outs<3){
  # samples 1 element from the batter vector based on the probabilities in pr
  atbat=sample(batter,1,prob=pr)
  # creates an if statement - that adds a run if a homerun, and an out
  # if a strikeout
  if(atbat==1){
    runs=runs+1
  } else {
    outs=outs+1
  }
}
# prints out the number of runs from the inning
runs

## [1] 0
```

SIMULATING MANY INNINGS

Once we get the simulation of one inning working, we wrap it in a `for` loop to simulate as many innings as we would like. We will keep track of the result of each simulated inning in the vector `output`.

```
# number of innings to be simulated
N=100000
# probability of a homerun
p=0.1
# vector to hold the output of each inning
output=rep(-1,N)
# creates a vector for the results of a batter
# 0 represents a strikeout
# 1 represents a homerun
batter=c(0,1)
# creates a vector that gives the probability of each result
pr=c(1-p,p)
# creates a for loop to simulate N innings
for(i in 1:N){
  # keeps track of the number of runs scored
  runs=0
  # keeps track of the number of outs in the inning
  outs=0
  # creates a while loop that continues until 3 outs are recorded
  while(outs<3){
    # samples 1 element from the batter vector based on the probabilities in pr
    atbat=sample(batter,1,prob=pr)
    # creates an if statement - that adds a run if a homerun, and an out
    # if a strikeout
    if(atbat==1){
      runs=runs+1
    } else {
      outs=outs+1
    }
  }
  # stores the output of the inning to the output vector
  output[i]=runs
}
# calculates the average of the output vector
mean(output)

## [1] 0.33029
```

We see the average per inning after simulating `N` innings. Multiply this number by 9 to determine the average per game.

```
9*mean(output)
```

```
## [1] 2.97261
```

SIMULATING A GAME

Now imagine you had two teams - one hits a homerun with probability $p_1 = 0.1$ and the second with probability $p_2 = 0.12$. If they played a game, we would expect the second team to win. But if they played 100 games against each other, would we expect the second team to win all 100 games? If not, how many would we expect them to win? We will answer this question by simulating 100 games of such teams.

We start with simulating one game. To make this easier, I am going to create a function, called `inning`, that simulates a teams at bat (really it should be called half-inning).

```
# creates the function inning:
# input - probability value p
# output - number of runs
inning<-function(p)
{
  # keeps track of the number of runs scored
  runs=0
  # keeps track of the number of outs in the inning
  outs=0
  # creates a vector for the results of a batter
  # 0 represents a strikeout
  # 1 represents a homerun
  batter=c(0,1)
  # creates a vector that gives the probability of each result
  pr=c(1-p,p)
  # creates a while loop based on the number of outs
  while(outs<3){
    # samples 1 element from the batter vector based on the probabilities in pr
    atbat=sample(batter,1,prob=pr)
    # creates an if statement - that adds a run if a homerun, and an out
    # if a strikeout
    if(atbat==1){
      runs=runs+1
    } else {
      outs=outs+1
    }
  }
  return(runs)
}

# probability of a homerun for the first team
p1=0.1
# probability of a homerun for the second team
p2=0.12
# keeps track of the number of runs scored by the first team
score1=0
# keeps track of the number of runs scored by the second team
```

```

score2=0
# keeps track of the innings of the game
ginning=1
# while loop that simulates the game - at least 9 innings and continues
# if the game is tied
while((ginning<=9)|(score1==score2)){
  # adds to score of team 1 using a function call to inning
  score1=score1+inning(p1)
  # adds to score of team 2 using a function call to inning
  score2=score2+inning(p2)
  # updates the inning of the game
  ginning=ginning+1
}
score1

## [1] 0
score2

## [1] 4

```

SIMULATING MANY GAMES

Once we get the simulation of one game working, we wrap it in a `for` loop to simulate as many games as we would like. We will keep track of the result of each simulated game in the vector `series`.

```
# creates the function inning:  
# input - probability value p  
# output - number of runs  
inning<-function(p)  
{  
  # keeps track of the number of runs scored  
  runs=0  
  # keeps track of the number of outs in the inning  
  outs=0  
  # creates a vector for the results of a batter  
  # 0 represents a strikeout  
  # 1 represents a homerun  
  batter=c(0,1)  
  # creates a vector that gives the probability of each result  
  pr=c(1-p,p)  
  # creates a while loop based on the number of outs  
  while(outs<3){  
    # samples 1 element from the batter vector based on the probabilities in pr  
    atbat=sample(batter,1,prob=pr)  
    # creates an if statement - that adds a run if a homerun, and an out  
    # if a strikeout  
    if(atbat==1){  
      runs=runs+1  
    } else {  
      outs=outs+1  
    }  
  }  
  return(runs)  
}  
  
# number of games to be simulated  
games=100  
# vector to hold the output of each game  
series=rep(-1,games)  
# probability of a homerun for the first team  
p1=0.1  
# probability of a homerun for the second team  
p2=0.12  
for(i in 1:games){  
  # keeps track of the number of runs scored by the first team  
  score1=0
```

```

# keeps track of the number of runs scored by the second team
score2=0
# keeps track of the innings of the game
ginning=1
# while loop that simulates the game - at least 9 innings and continues
# if the game is tied
while((ginning<=9)|(score1==score2)){
  # adds to score of team 1 using a function call to inning
  score1=score1+inning(p1)
  # adds to score of team 2 using a function call to inning
  score2=score2+inning(p2)
  # updates the inning of the game
  ginning=ginning+1
}
#print("game")
#print(score1)
#print(score2)
# adds a 0 to the series when team 1 wins and 1 to the series when team 2 wins
if(score1<score2){
  series[i]=1
} else {
  series[i]=0
}
}
# calculates the average of the output vector
mean(series)

## [1] 0.66

```

GROUP ACTIVITY - FOOTBALL SIMULATION

Imagine a football team that gains 5 yards with probability 0.25, 4 yards with probability 0.35, 0 yards with probability 0.3, and -2 yards with probability 0.1. What is the probability they get a first down within 4 plays? Write and run a simulation to answer this question.

Submit your RMD file to the appropriate Gradescope assignment. It is due at the end of class.

GROUP ACTIVITY - BASKETBALL SIMULATION

Imagine a team that has a shooting percentage of 52% for 2-point field goals, and 36% for 3-point field goals. They shoot 3-point field goals on 33% of their possessions, and 2-point field goals on 67% of their possessions.

Now imagine a second team that has the exact same shooting percentage for both 2-point and 3-point field goals, but they shoot 3-point field goals on 43% of their possessions, and 2-point field goals on 57% of their possessions.

If these two teams play, what is the probability that the second team wins? Assume that each team gets 100 possessions per game. For simplicity, we will allow for ties.

Does this change if we assume that each team gets 110 possessions per game? If so, by how much?

Write and run a simulation that estimates these probabilities.
