NCAIR 2024 Annual Conference

Institutional Innovation

Enrollment Projection with Monte Carlo Simulation

Franklin Zhou
IPAR, East Carolina University

What is a Monte Carlo Simulation?

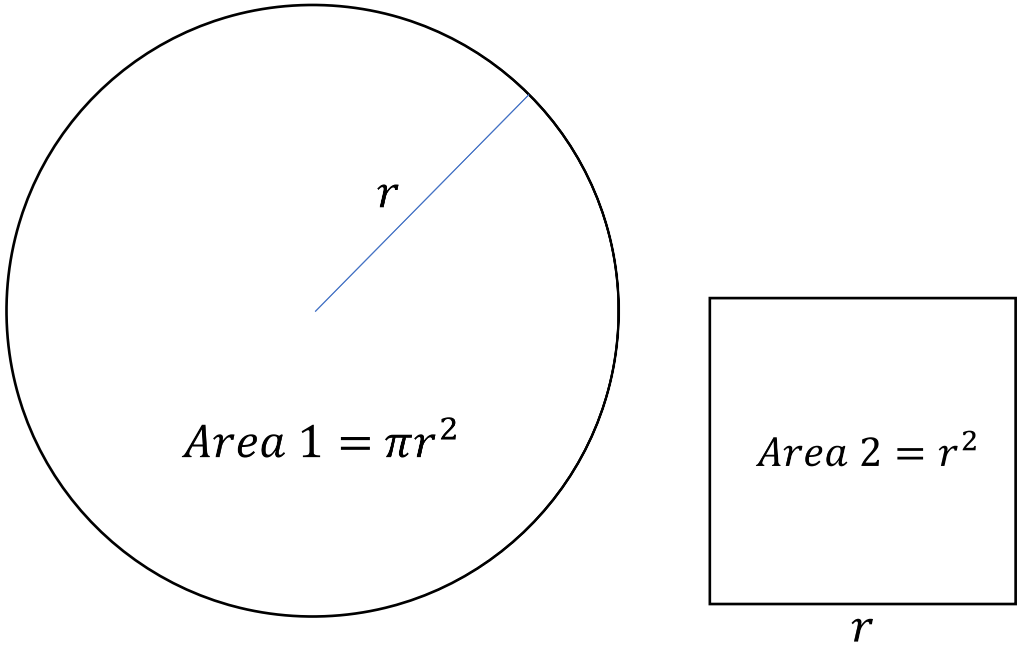
Monte Carlo: a casino in Monaco

"Monte Carlo Simulation, also known as the Monte Carlo Method or a multiple probability simulation, is a mathematical technique, which is used to estimate the possible outcomes of an uncertain event."

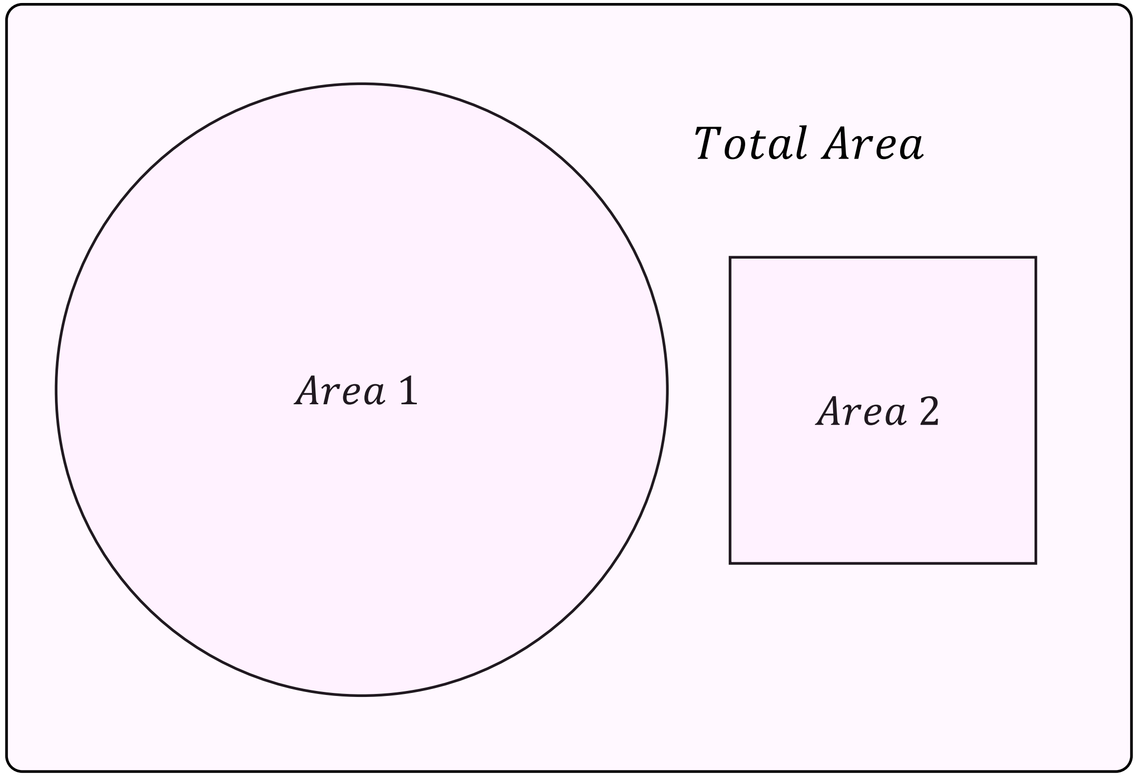
https://www.ibm.com/topics/monte-carlo-simulation

Example of Monte Carlo Simulation

How to find π?

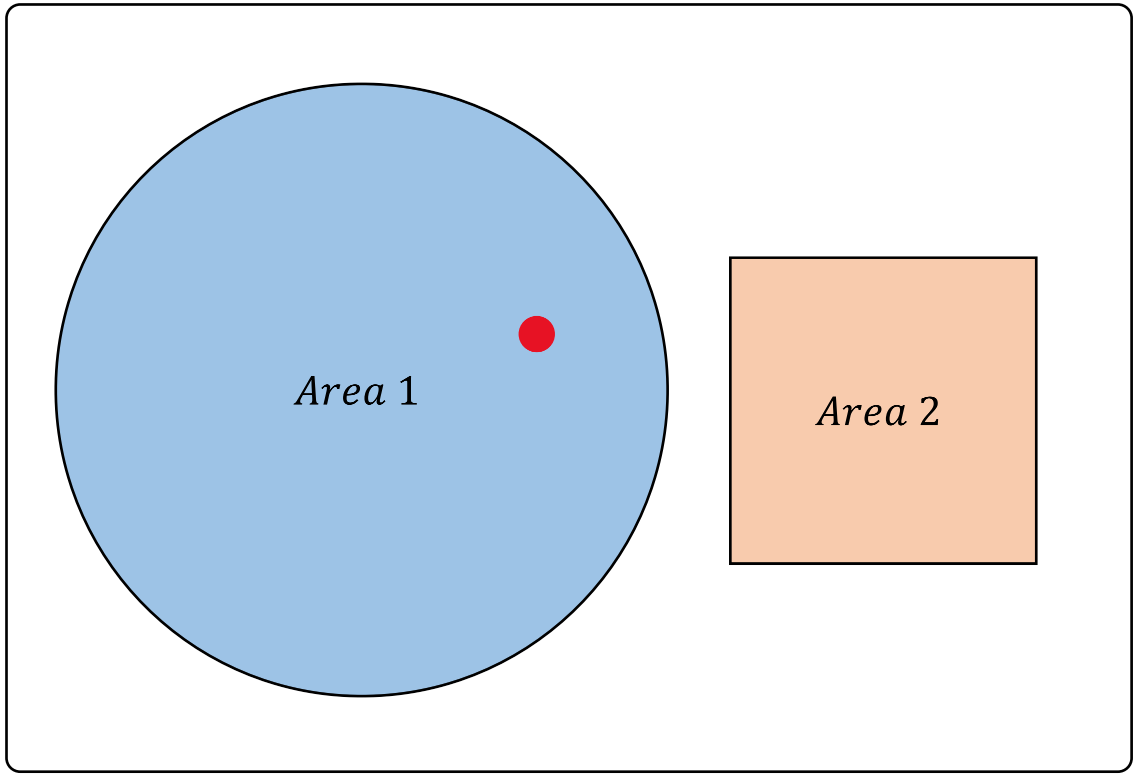


$$\frac{Area\ 1}{Area\ 2} = \pi$$



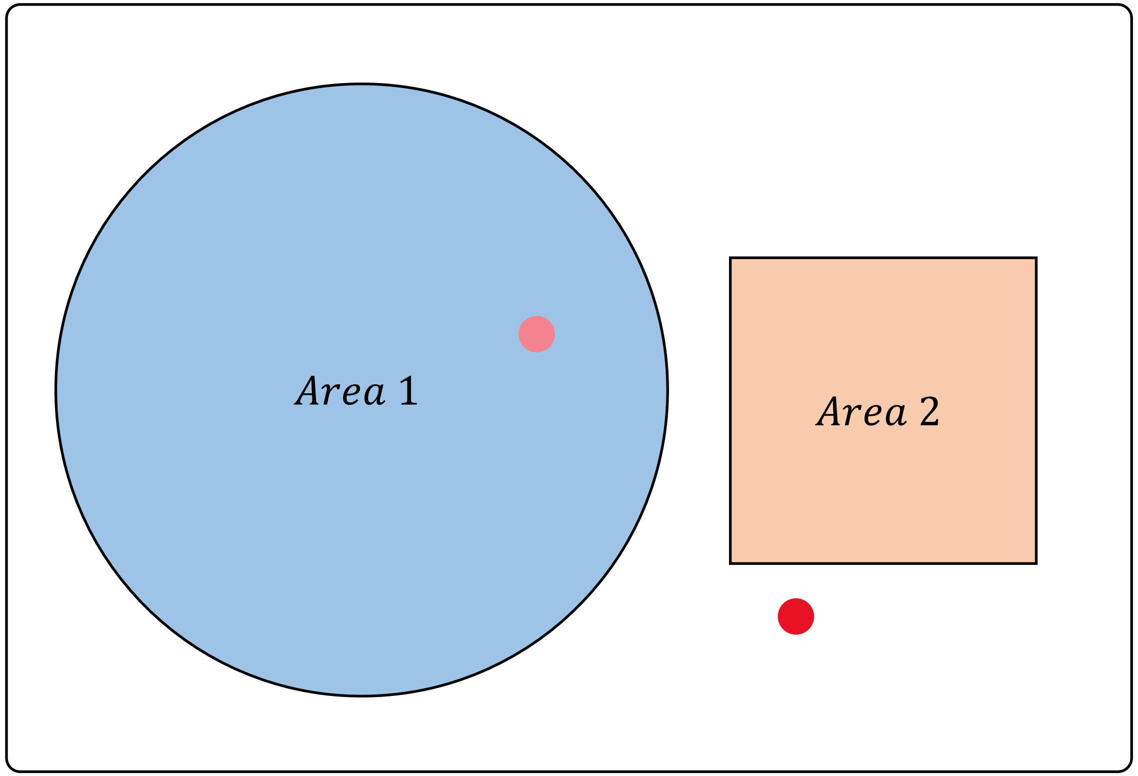
$$\frac{Area\ 1/TotalArea}{Area\ 2/TotalArea} = \pi$$

Example of Monte Carlo Simulation



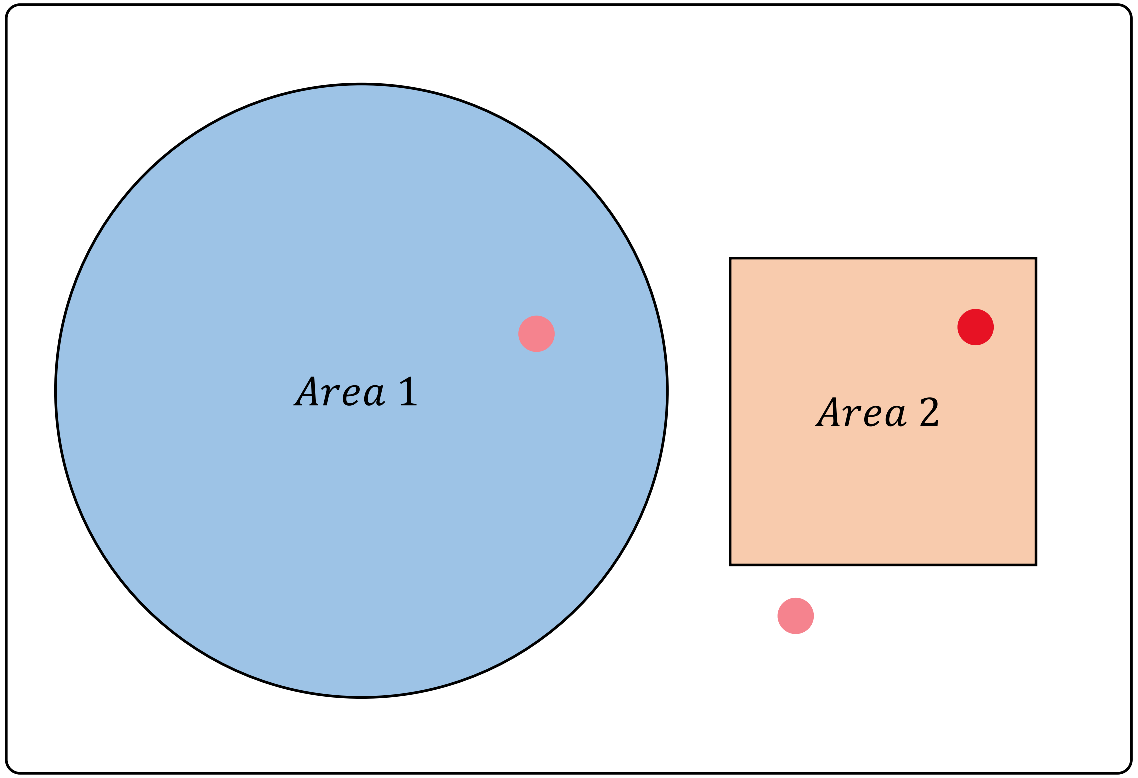
For each single dot, the probability of falling within the circle is $\frac{Area\ 1}{Total\_Area}$, the probability of falling within the square is $\frac{Area\ 2}{Total\_Area}$

Example of Monte Carlo Simulation



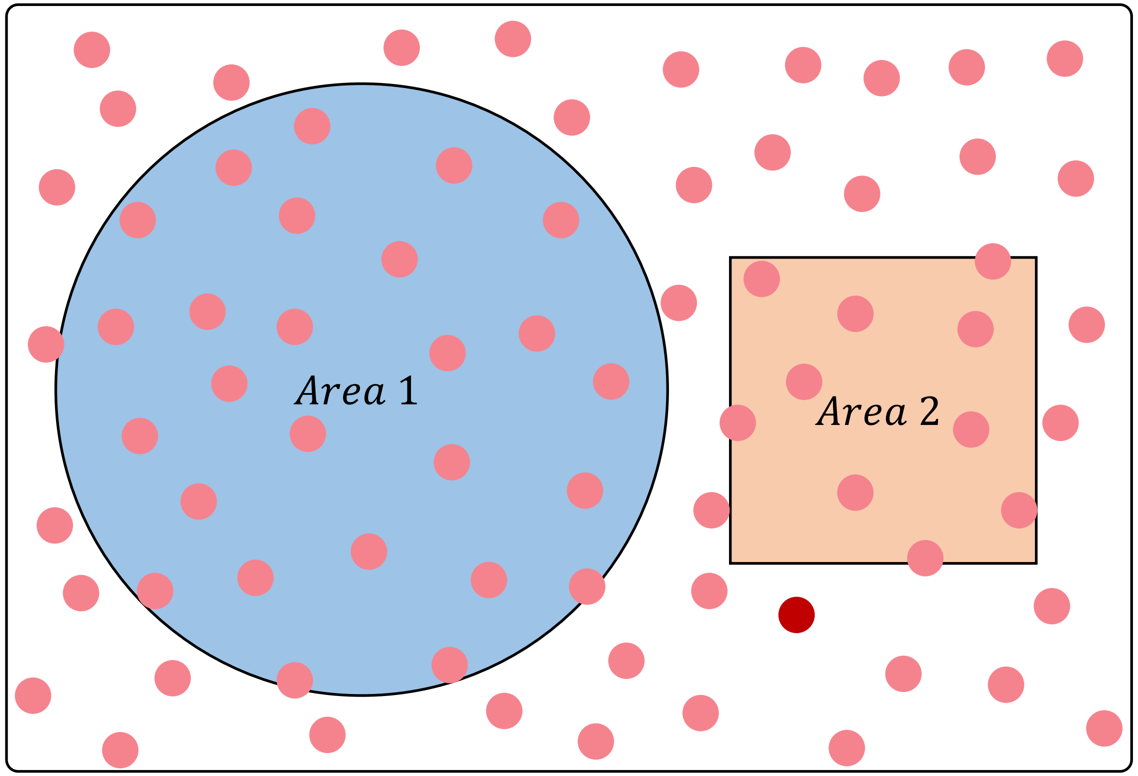
For each single dot, the probability of falling within the circle is $\frac{Area\ 1}{Total\_Area}$, the probability of falling within the square is $\frac{Area\ 2}{Total\_Area}$

Example of Monte Carlo Simulation



For each single dot, the probability of falling within the circle is $\frac{Area\ 1}{Total\_Area}$, the probability of falling within the square is $\frac{Area\ 2}{Total\_Area}$

Example of Monte Carlo Simulation



For each single dot, the probability of falling within the circle is $\frac{Area\ 1}{Total\_Area}$, the probability of falling within the square is $\frac{Area\ 2}{Total\_Area}$

Since $\frac{Area\ 1/Total\_Area}{Area\ 2/Total\_Area} = \pi$

In the long run,

$$\frac{\#\ of\ dots\ within\ the\ circle}{\#\ of\ dots\ within\ the\ square} = \pi$$

Example of Monte Carlo Simulation

Dot number	Result
1	Circle
2	Circle
3	Square
4	Out
5	Circle
…	…
n	Square

When n is getting larger, according to Law of Large Numbers:

$$\frac{\#\ of\ dots\ within\ the\ circle}{\#\ of\ dots\ within\ the\ square} \approx \pi$$

$$P_1(Circle) = P_2(Circle) = \cdots = P_n(Circle)$$

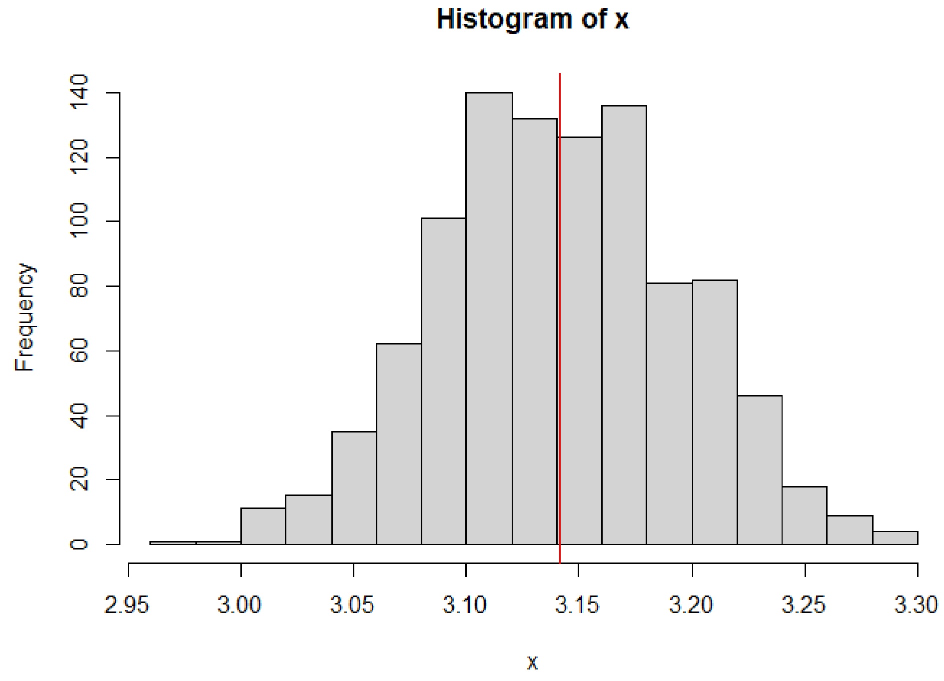
$$P_1(Square) = P_2(Square) = \cdots = P_n(Square)$$

$$P_1(Out) = P_2(Out) = \cdots = P_n(Out)$$

Example of Monte Carlo Simulation

Dot number	Simulation 1 result	Simulation 2 result	…	Simulation n result
001	Circle	Circle	…	Square
002	Circle	Square	…	Out
003	Square	Out	…	Circle
004	Out	Circle	…	Circle
005	Circle	Circle	…	Square
…	…	…	…	…
500	Circle	Out	…	Circle

π_1 π_2 … π_n



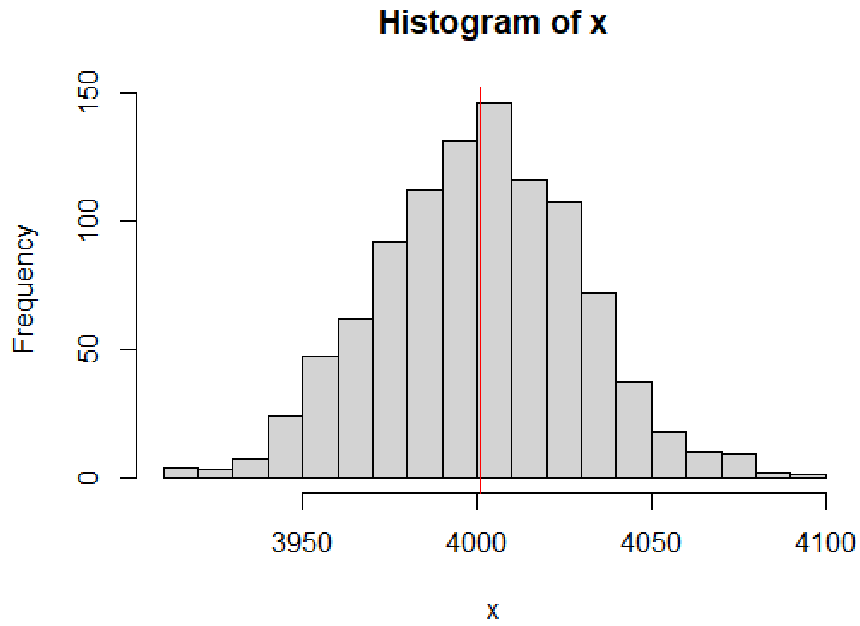
Simulation of 1000 times

```
Min.  1st Qu. Median Mean  3rd Qu. Max.
2.976  3.104  3.144 3.144 3.180   3.284
```

Similarly…

Student ID	Simulation 1 result	Simulation 2 result	…	Simulation n result
001	Retained	Retained	…	Not Retained
002	Not Retained	Retained	…	Retained
003	Retained	Retained	…	Retained
004	Retained	Not Retained	…	Retained
005	Retained	Retained	…	Not Retained
…	…	…	…	…
500	Not Retained	Retained	…	Retained

\# Retained \# Retained \# Retained



Simulation of 1000 times

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
3913 3982    4002   4001 4021    4094
```

How to produce the simulation in Excel

Data set

- Dependent Variable
  - Retained in Fall (Y/N)
- Independent Variables
  - Age
  - Gender
  - Race/Ethnicity
  - Full Time/Part Time
  - Cumulative GPA
  - …

Step 1: Use predictive model to get the predicted retention probability for each student.

Student ID	Probability of Retention
0001	$P_{001}(Retention)$
0002	$P_{002}(Retention)$
0003	$P_{003}(Retention)$
0004	$P_{004}(Retention)$
0005	$P_{005}(Retention)$
…	…
5000	$P_{500}(Retention)$

How to produce the simulation in Excel

	Col A	Col B	Col C
Row 1	**Student ID**	**Probability of Retention**	**Simulation 1**
Row 2	0001	0.389937	0
Row 3	0002	0.757576	1
Row 4	0003	0	0
Row 5	0004	0.111111	1
	0005	0.15	0
	…	…	…
Row 5001	5000	0.142857	0

Step 2: Run simulation for all students.

=IF($B2>=RAND(),1,0)

=IF($B3>=RAND(),1,0)

…

=IF($B501>=RAND(),1,0)

How to produce the simulation in Excel

Step 3: Run as many simulations as you want

	Col A	Col B	Col C	Col D	…	Col AAA
Row 1	**Student ID**	**Probability of Retention**	**Simulation 1**	**Simulation 2**	**…**	**Simulation n**
Row 2	0001	0.389937	0	1	…	0
Row 3	0002	0.757576	1	1	…	1
Row 4	0003	0	0	0	…	0
Row 5	0004	0.111111	1	0	…	0
	0005	0.15	0	0	…	1
	…	…	…	…	…	
Row 5001	5000	0.142857	0	1	…	0

ECU

How to produce the simulation in Excel

Step 4: Calculate the sum for each simulation as expected total retention number

	Col A	Col B	Col C	Col D	…	Col AAA
Row 1	**Student ID**	**Probability of Retention**	**Simulation 1**	**Simulation 2**	**…**	**Simulation n**
Row 2	0001	0.389937	0	1	…	1
Row 3	0002	0.757576	1	1	…	1
Row 4	0003	0	0	0	…	0
Row 5	0004	0.111111	1	0	…	0
	0005	0.15	0	0	…	1
	…	…	…	…	…	
Row 5001	5000	0.142857	0	1	…	0
			=SUM(C2:C5001)	=SUM(D2:D5001)		=SUM(AAA2:AAA5001)

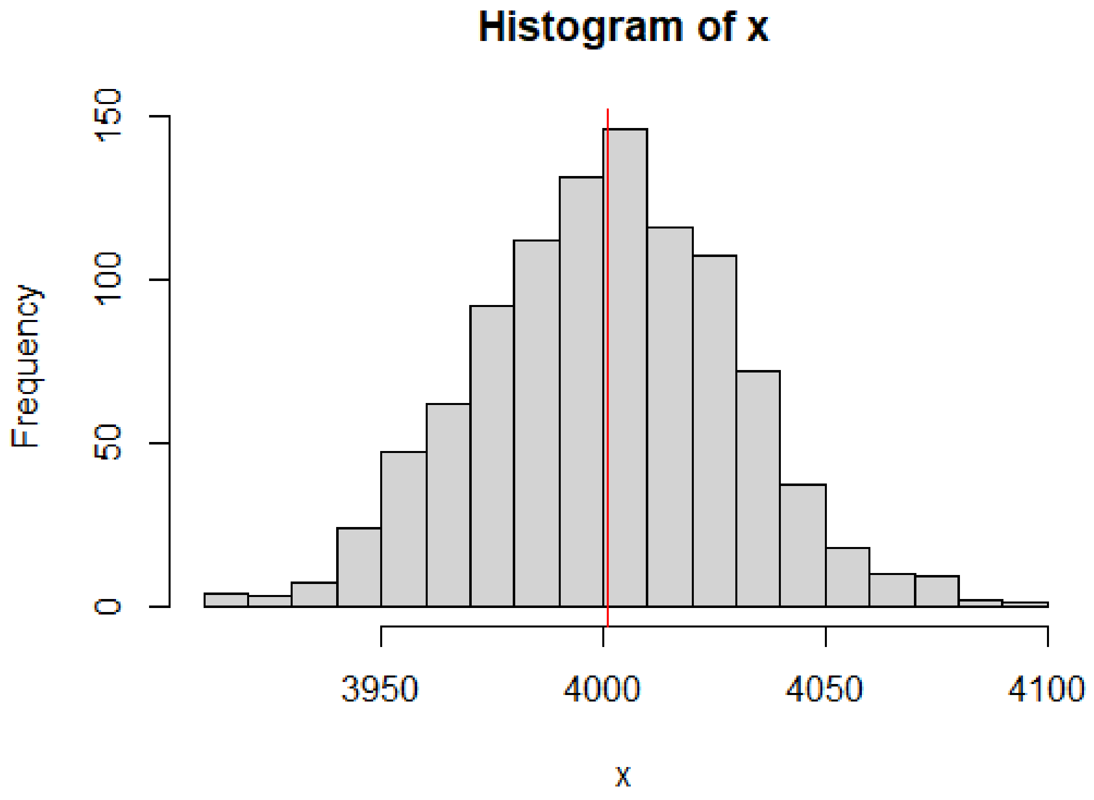
ECU

How to produce the simulation in Excel

Step 5: Descriptive analysis of all expected retention numbers

Simulation of 1000 times

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3913	3982	4002	4001	4021	4094



How to produce the simulation in R

- Step 1: Use predictive model to get the predicted retention probability for each student

Example: binary logistic regression

```
# build the logistic regression
lm_ug <- glm(RETAINED ~., data = UG_train, family = "binomial")

# ….
# validation part omitted

# predict the retention by applying the model on the new data
projection_24 <- predict(lm_ug , newdata = UG_23_DATA, type = "response")
```

This gives you the probability of Response = 1

How to produce the simulation in R

- Step 2: Run Monte Carlo Simulation

Store the predicted total numbers in the vector

runif(): *generate a random number in a uniform distribution*

lapply(): *apply a function over a List or Vector*

```
# set total number of simulations
N=3000

# build up a vector
total_ug <- NULL

# create the function
pred_fun <- function(x){
    ifelse(x >= runif(1),1,0)
  }

# run the simulation N times and store the sum into the vector
for (i in 1:N){
        df_pred <- lapply(projection_24,pred_fun)
        total_ug[i] = sum(unlist(df_pred))
  }
```

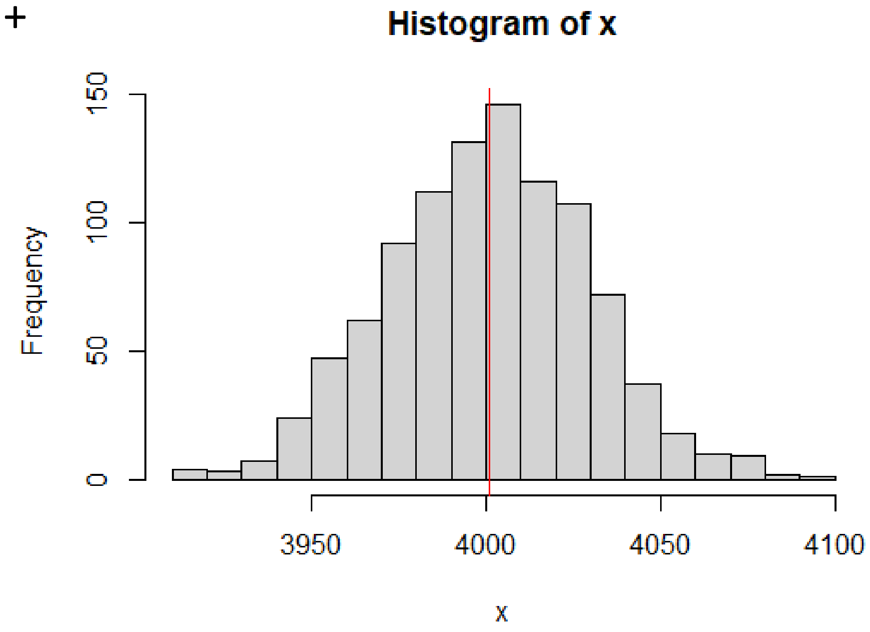
ECU

How to produce the simulation in R

- Step 3: Check the result and make plot

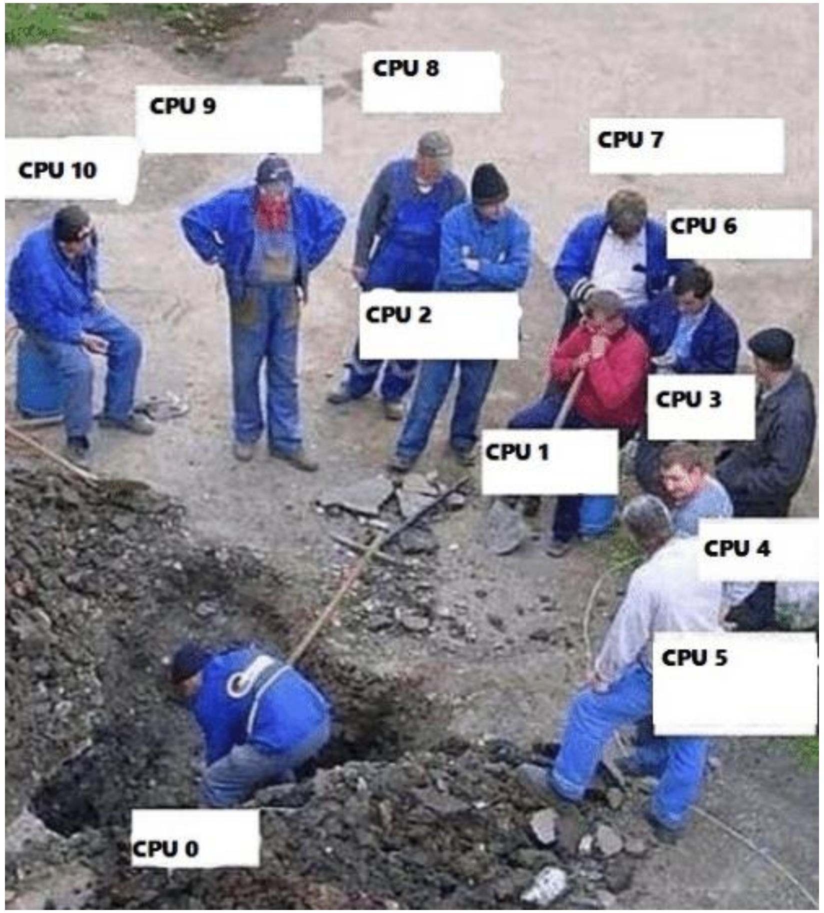
```
ggplot(data = as.data.frame(total_ug), mapping = aes(x = total_ug)) +
    geom_histogram(bins = 50) +
    labs(title = "Retention projection of 2024 undergraduates") +
    xlab("Estimated Retention") +
    theme_light()

summary(total_ug)
quantile(total_ug, c(0.025,0.975))
```




```
Min.  1st Qu.  Median  Mean  3rd Qu.  Max.
3913  3982     4002    4001  4021     4094
```

Performance Enhancement



By default, R runs only on a single thread on the CPU.

How to enhance the performance?

- Upgrade equipment

- Improve R code: vectorized functions
- Use parallel processing programming

Performance Enhancement – vectorized function

- Speed up Step 2: Monte Carlo Simulation (original code)

system.time()

user system elapsed
161.68 0.11 161.92

The problem: we evaluate this function too many times.

30,000 students
X 3,000 simulations
= 90,000,000 times!

```
# set total number of simulations
N=3000

# build up a vector
total_ug <- NULL

# create the function
pred_fun <- function(x){
        ifelse(x >= runif(1),1,0)
        }

# run the simulation N times and store the sum into the list
for (i in 1:N){
        df_pred <- lapply(projection_24,pred_fun)
        total_ug[i]= sum(unlist(df_pred))
        }
```

Performance Enhancement – vectorized function

- Speed up Step 2: Monte Carlo Simulation (improved code)

system.time()

user system elapsed
 0.61   0.03   0.64

The ">=" function and "runif()" function only be called ONCE in each iteration of the loop. It saves you lots of time!

```
# set total number of simulations
N=3000

# build up a vector
total_ug <- NULL

# run the simulation N times and store the sum into the list
for (i in 1:N){
total_ug[i] <- sum(projection_24 >= runif(length(projection_24)))
}
```

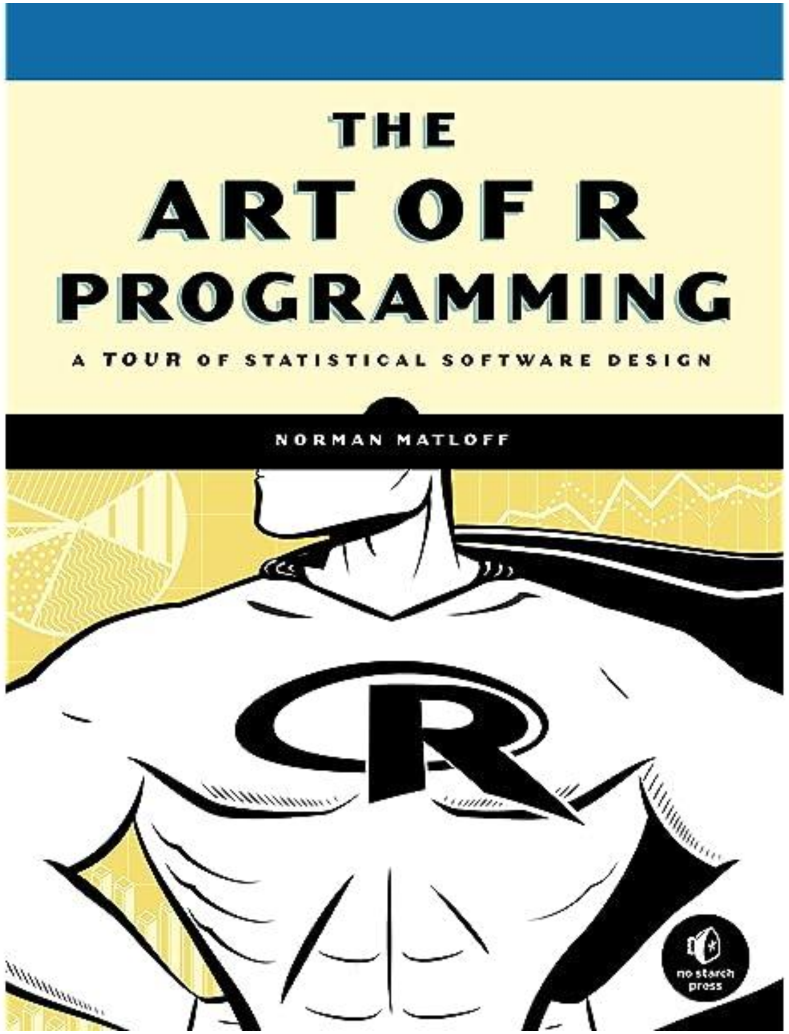
R Programming Resources

Parallel and high performance computing with R

https://youtu.be/NWgOkKorFH4

Parallel Programming with R

https://youtu.be/O8PiX9ofXDI



Thank you

Thank you for attending the 2024 NCAIR Annual Conference!

There's a QR code in your program for a conference evaluation form. You'll also get an e-mail following the conference with a link to the form, which will be available until 4/30.

At your earliest convenience, please take this opportunity to let the planning committee know your thoughts about this year's conference and where you would like to meet next year.