MONTE CARLO SIMULATIONS

Noratiqah Mohd Ariff

Simulation

Simulation is a method used to examine the “what if” without having real data. Simulation and model fitting are related but opposite processes.

In **simulation**, the data generating process is known. We will know the form of the model as well as the value of each of the parameters. We will often control the distribution and parameters which define the randomness, or noise in the data.

In **model fitting**, the data is known. We will then assume a certain form of model and find the best possible values of the parameters given the observed data. We will attempt to fit many models, and we will learn metrics to assess which model fits best.

Data Generating Process (DGP)

- A DGP describes how values of a variable of interest are produced in the population.
- Most DGP's of interest include a systematic component and a stochastic component.
- We use statistical analysis to infer characteristics of the DGP by analyzing observable data sampled from the population.
- In applied statistical work, we never know the DGP – if we did, we wouldn't need statistical estimates of it.

Synthetic Data

- At a conceptual level, synthetic data is not real data. Synthetic data is a proxy for real data. There are three types of synthetic data. The first type is generated from actual/real datasets, the second type does not use real data and the third type is a hybrid of these two.
- Why use simulation/synthetic data?

❖ Analysis where observable data is not available/not enough

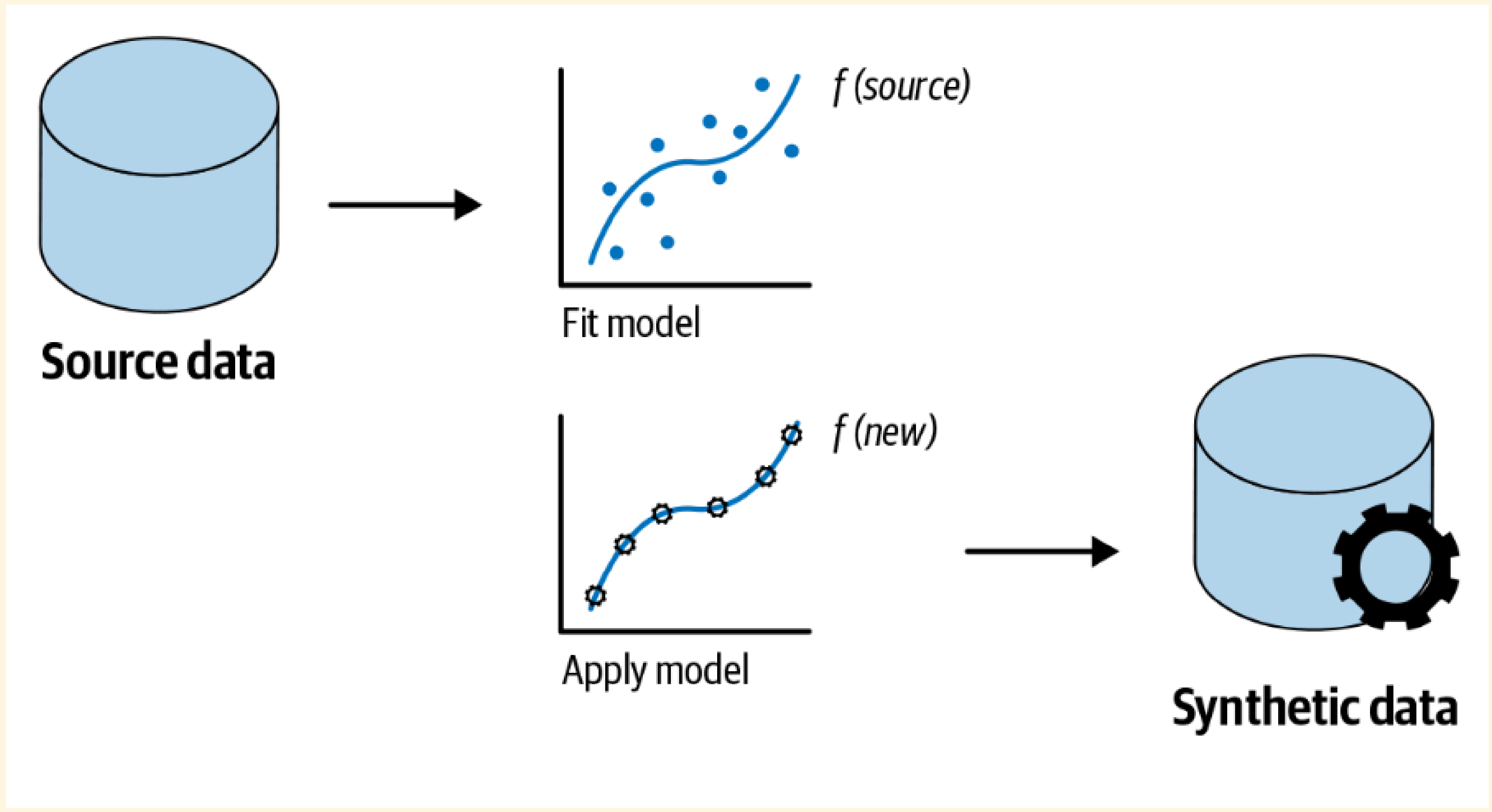
❖ Mimic the repeated sampling framework of classical frequentist statistics/ experimental lab results

❖ Provide solutions where analytic solutions are not available

❖ Test hypothetical processes, check robustness

Synthetic Data from Real Data

This means that the analyst has some real datasets and then builds a model to capture the distributions and structure of that real data. Here *structure* means the multivariate relationships and interactions in the data. Once the model is built, the synthetic data is sampled or generated from that model. If the model is a good representation of the real data, then the synthetic data will have statistical properties similar to those of the real data.



Synthetic Data from DGP

This means that the analyst used existing models, or the analyst has background knowledge. If either is accurate, then the synthetic data will behave in a manner that is consistent with real world data. When a process is new or not well understood by the analyst, and there is no real historical data to use, then an analyst can make some simple assumptions about the distributions and correlations among the variables involved in the process. This type of data will likely not have the same properties as real data but can still be useful for some purposes.

Getting Started

When we **do** a simulation, we must make many assumptions. One major assumption is the choice of the distribution to use for a particular variable. Each particular distribution has *parameters* that are integral to generating data from the distribution. We need to *set* a value for these parameters to simulate a value from a distribution. Given a particular distribution and known parameters, we can generate *values* from that distribution.

Probability Distributions in R

- For most probability distributions, R has built-in functions to

1. Calculate the pdf/pmf of a distribution.
2. Calculate the cdf of the distribution.
3. Calculate the quantile function of the distribution.
4. Generate random numbers based on the distribution.

- All the functions related to above have the same “form” for all distributions - “d,p,q,r” function.

Probability Distributions in R

- "d" is for "density“. It is used to find values of the probability density function.
- "p" is for "probability“. It is used to find the probability that the random variable lies to the left of a given number.
- "q" is for "quantile“. It is used to find the quantiles of a given distribution.
- "r" is for "random“. It is used to generate random numbers of a given distribution.

Probability Distributions in R

Distributions	Functions			
Beta	`pbeta`	`qbeta`	`dbeta`	`rbeta`
Binomial	`pbinom`	`qbinom`	`dbinom`	`rbinom`
Chi-Square	`pchisq`	`qchisq`	`dchisq`	`rchisq`
Exponential	`pexp`	`qexp`	`dexp`	`rexp`
F	`pf`	`qf`	`df`	`rf`
Gamma	`pgamma`	`qgamma`	`dgamma`	`rgamma`
Log Normal	`plnorm`	`qlnorm`	`dlnorm`	`rlnorm`
Normal	`pnorm`	`qnorm`	`dnorm`	`rnorm`
Poisson	`ppois`	`qpois`	`dpois`	`rpois`
Student t	`pt`	`qt`	`dt`	`rt`
Uniform	`punif`	`qunif`	`dunif`	`runif`

Pseudorandom

- Note that the numbers generated in computer are not exactly
- random. They are “pseudorandom”.
- But they are good enough for most applications.
- You can set the “seed” of your randomly generated numbers
- using `set.seed()` function.

Examples:

- **Normal Distribution** – simulate 1000 values from a normal distribution with a mean of 10 and a standard deviation of 4.

```
rnorm(100, mean=10,sd=4)
```

- **Binomial Distribution** – simulate 6-coin flips using a fair coin with 1=head and 0=tail

```
rbinom(6,size=1,prob=0.5)
```

- **Uniform Distribution** – simulate 50 test scores of students where any score between 0 and 100 is possible

```
runif(50,min=0,max=100)
```

Monte Carlo Simulation

- The Monte Carlo method is a computerized mathematical technique that allows people to quantitatively account for forecasting and decision-making.
- Monte Carlo method is a way to use random samples of parameters to explore the behavior of a complex system.
- A Monte Carlo simulation is used to handle an extensive range of problems in a variety of different fields.
- Computer simulation that generates large number of simulated samples of data based on an assumed DGP that characterizes the population.

History of Monte Carlo Simulation

- The technique was initially developed by Stanislaw Ulam, a mathematician who worked on the Manhattan Project, the secret effort to create the first atomic weapon. He shared his idea with John Von Neumann, a colleague at the Manhattan Project, and the two collaborated to refine the Monte Carlo simulation.
- The Monte Carlo simulation was named after the gambling destination in Monaco because chance and random outcomes are central to this modeling technique, as they are to games like roulette, dice, and slot machines.

Basic Steps of Monte Carlo Simulation

1. Set up the predictive model, identifying both the dependent variable to be predicted and the independent variables (also known as the input, risk or predictor variables) that will drive the prediction.
2. Specify probability distributions of the independent variables. Use historical data and/or the analyst's subjective judgment.
3. Run simulations repeatedly, generating random values of the independent variables. Do this until enough results are gathered to make up a representative sample of the near infinite number of possible combinations.

Example of Monte Carlo Simulation

Assume the true underlying distribution of an exam scores for a given year is Normally distributed with mean 896 and standard deviation of 174. Generate 1000 Monte Carlo samples of size 100 and estimate the mean, median, standard deviation and range for each sample. Lastly, plot the histograms for all the Monte Carlo estimates.

Examples:

```
m <- 1000
n <- 100
x <- matrix(rnorm((m*n),896,174), nrow=m)
means <- apply(x,MARGIN=1,FUN=mean)
sdevs <- apply(x,MARGIN=1,FUN=sd)
medians <- apply(x,MARGIN=1,FUN=median)
ranges <- apply(apply(x,MARGIN=1,FUN=range),MARGIN=2,FUN=diff)
par(mfrow=c(2,2)) # Creates a 2x2 graphics window.
hist(means,xlab="Means",ylab="Frequency", main="Monte Carlo
Means",cex.axis=1.5,cex.lab=1.6,cex.main=1.6)
hist(sdevs,xlab="Std. Deviations",ylab= "Frequency",main="Monte Carlo
Std.Deviations",cex.axis=1.5,cex.lab=1.6,cex.main=1.6)
hist(medians,xlab="Medians",ylab="Frequency", main="Monte Carlo
Medians",cex.axis=1.5,cex.lab=1.6,cex.main=1.6)
hist(ranges,xlab="Ranges",ylab="Frequency", main="Monte Carlo
Ranges",cex.axis=1.5,cex.lab=1.6,cex.main=1.6)
```

Thank You