Probability and Monte Carlo Simulations

POL 212: Quantitative Analysis I

Winter 2024

Probability distributions

One helpful way to think of probability distributions is that they are really just a set of **instructions**. For instance, if we want to draw random numbers, the choice of a distribution and its parameterization dictate which numbers can be drawn and which are more or less likely to be drawn.

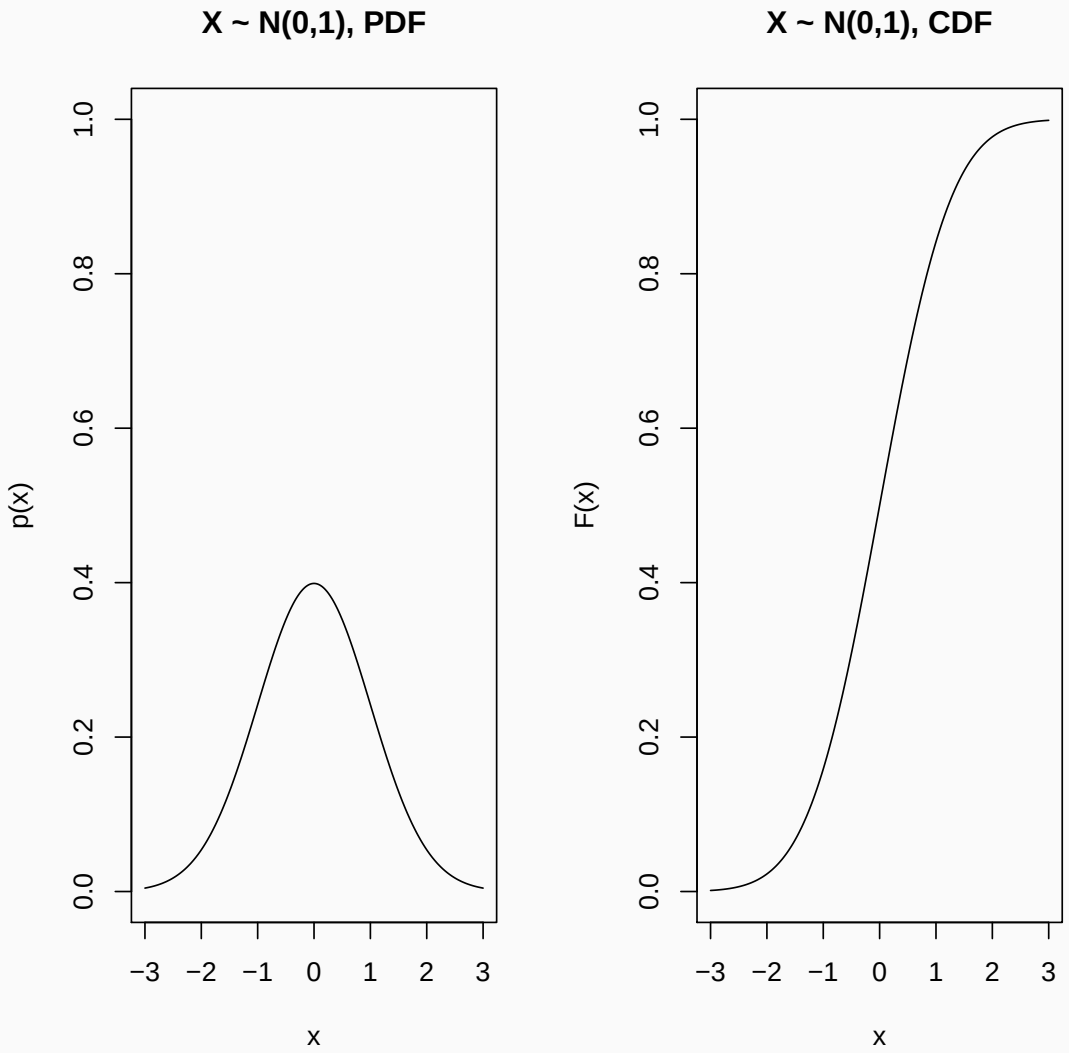
Probability distributions

Examples:

- The normal distribution: an appropriate choice when we want to draw continuous numbers on between $-\infty$ and ∞. It has two parameters: μ (the mean, shifts the distribution left or right) and σ^2 (the variance, stretches or compresses the distribution).
- The uniform distribution: useful when we want equally probable continuous numbers between *a* and *b* (its two parameters).
- The binomial distribution: appropriate when we only want to draw discrete numbers (usually a binary 0/1, failure/success, etc.). Its parameters are *p* (the probability of a 1/success/etc. in each trial) and *n* (the number of trials).

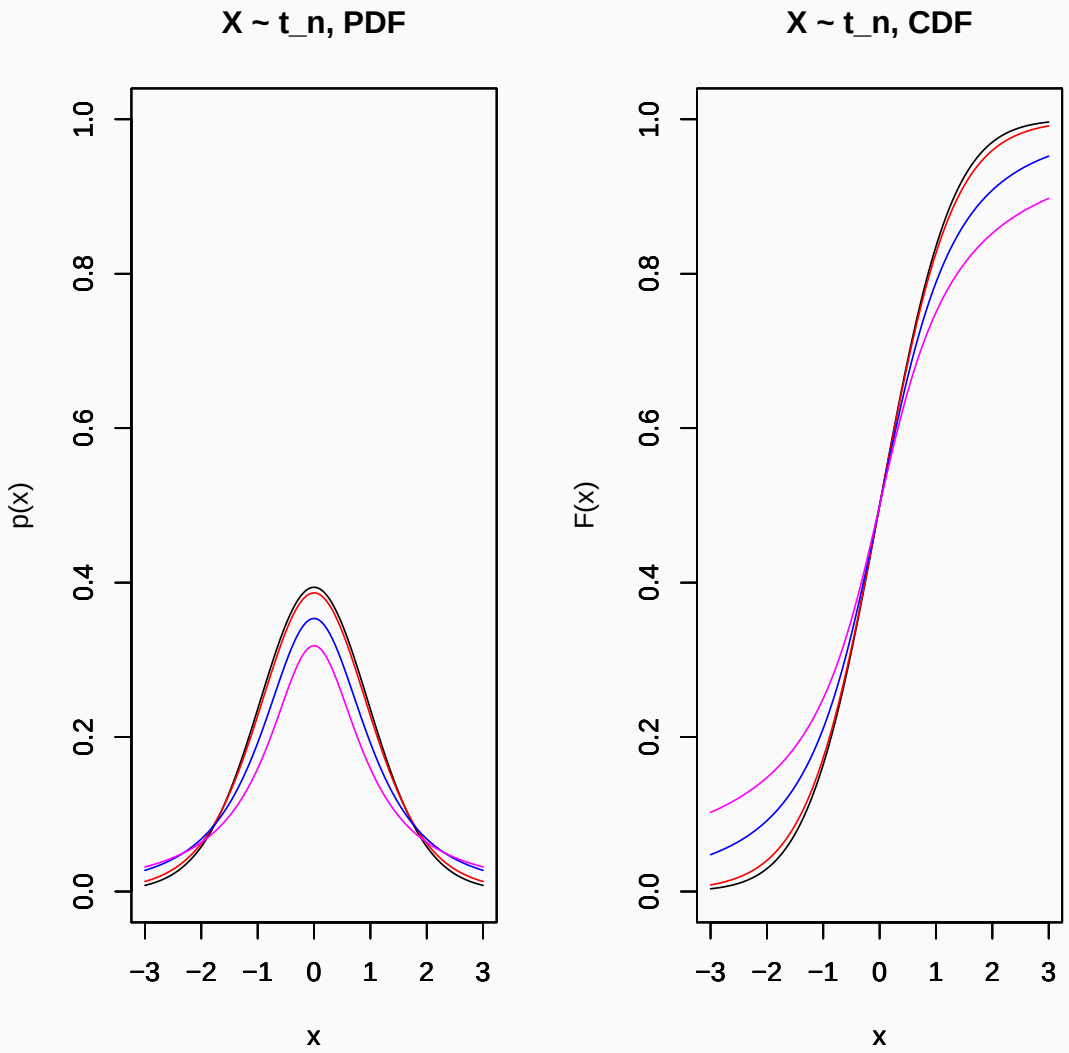
The normal distribution

Note the distinction between the PDF and CDF.



Student's t distribution

Note the distinction between the PDF and CDF.



Probability distributions

- It's never a bad idea (in face, it's often a really good idea) to play around with distributions and their parameterizations in **R** to get more familiar with them.
- Specifically, we can **simulate** their realizations by taking lots and lots of random draws from them and summarizing those draws in various ways (e.g., plotting them, taking their mean, etc.).
    - With the commands `rnorm()`, `runif()`, `rbinom()`, etc.
- This is a subtle but important point: in doing so, we are of course not seeing the distributions themselves **but** we are learning about them via simulation!

Monte Carlo simulations

- We can of course either **simulate** distributions by specifying their parameters and sampling from them, or (more frequently) **estimate** the parameters of distributions based on real-world data.
    - For example, what is the probability a civil unrest in a country given certain conditions?
- But both ways that we use distributions are **complementary**; indeed, here we'll focus on using **simulations** as a way to develop intuition about popular estimation methods (e.g., the linear regression model) and also to experiment with such methods.

Monte Carlo simulations

- This ability to experiment via simulations is oh so useful in answering a variety of “What if?” monsters you’ll encounter as you move forward with research projects.
    - Are your results driven by outliers? How sensitive are your results to various kinds of misspecification? Are your estimates of uncertainty (e.g., standard errors) too small? And on and on it goes!
- But first, let’s talk a bit more about data generating processes (DGPs).

Data generating processes (DGPs)

- DGPs are fundamentally what we care about as social scientists.
- These are the mechanisms that, well, generate the data we have collected and wish to analyze.
- Population $\rightsquigarrow$ Sample.
- When we refer to “sampling distributions”, this is what we’re sampling from! Hence, our uncertainty and consequent need for tools like confidence intervals and (null) hypothesis testing.

Data generating processes (DGPs)

- We think of DGPs as having both a **systematic** and a **stochastic** component:

$$f(X_i) = E(y|X_i) \tag{1}$$

$$y_i = f(X_i) + \varepsilon_i \tag{2}$$

- *Bonus challenge*: which terms in Equation 2 do we need to place distributions on?

Data generating processes (DGPs)

- Sadly, we of course never get to see an actual DGP. We see their artifacts, but that's it.
- So, we attempt to reconstruct estimates of the DGP from these artifacts (the observed data).

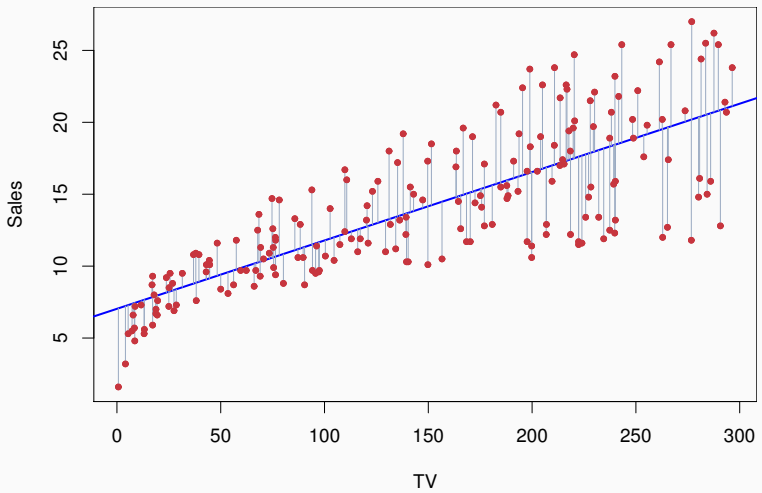
Monte Carlo simulations

- This is where the Monte Carlo simulation method comes into play.
- As defined in *MCS* (p. 4), a Monte Carlo simulation is "any computational algorithm that randomly generates multiple samples of data from a defined population based on an assumed DGP."
- So the usual workflow will be:
  1. Specify the DGP.
  2. Simulate the DGP based on that specification.
  3. Analyze the simulations by using the results to construct estimates of the DGP.
  4. Compare those estimates of the DGP with the true parameters of the DGP.

Monte Carlo simulations

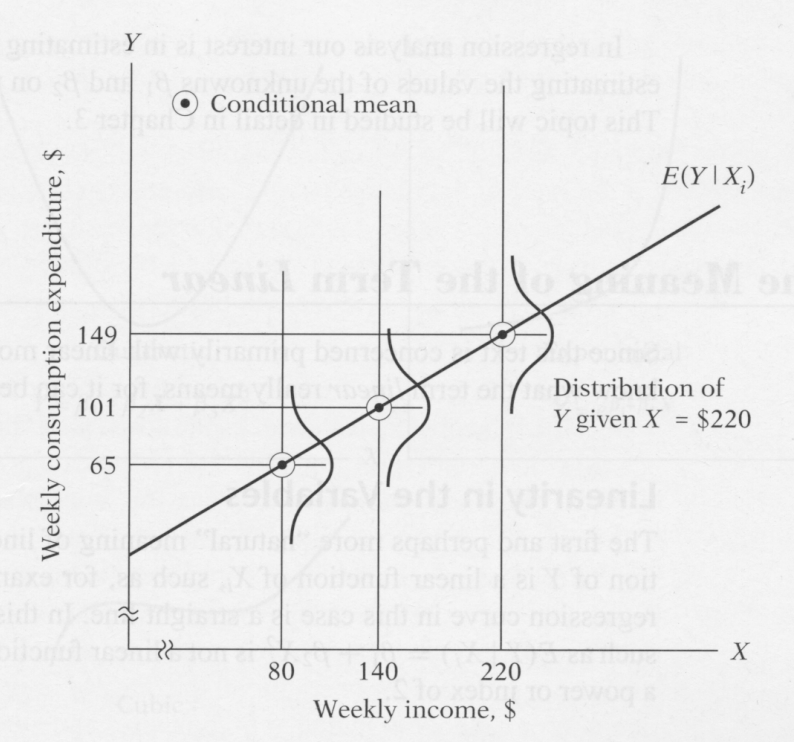
- Example: Let's contrive a relationship between height (x) and weight (y) under various levels of error according to Equations 1 and 2 in `R`.

Regression models



- Far and away, the most popular way to parameterize f in Equations 1–2 is with a set of slope coefficients corresponding to each of the X input variables.
- This gives us the workhorse of social science: the (linear) regression model.
- Think back to the equation for a line: $y = mx + b$.

Regression models

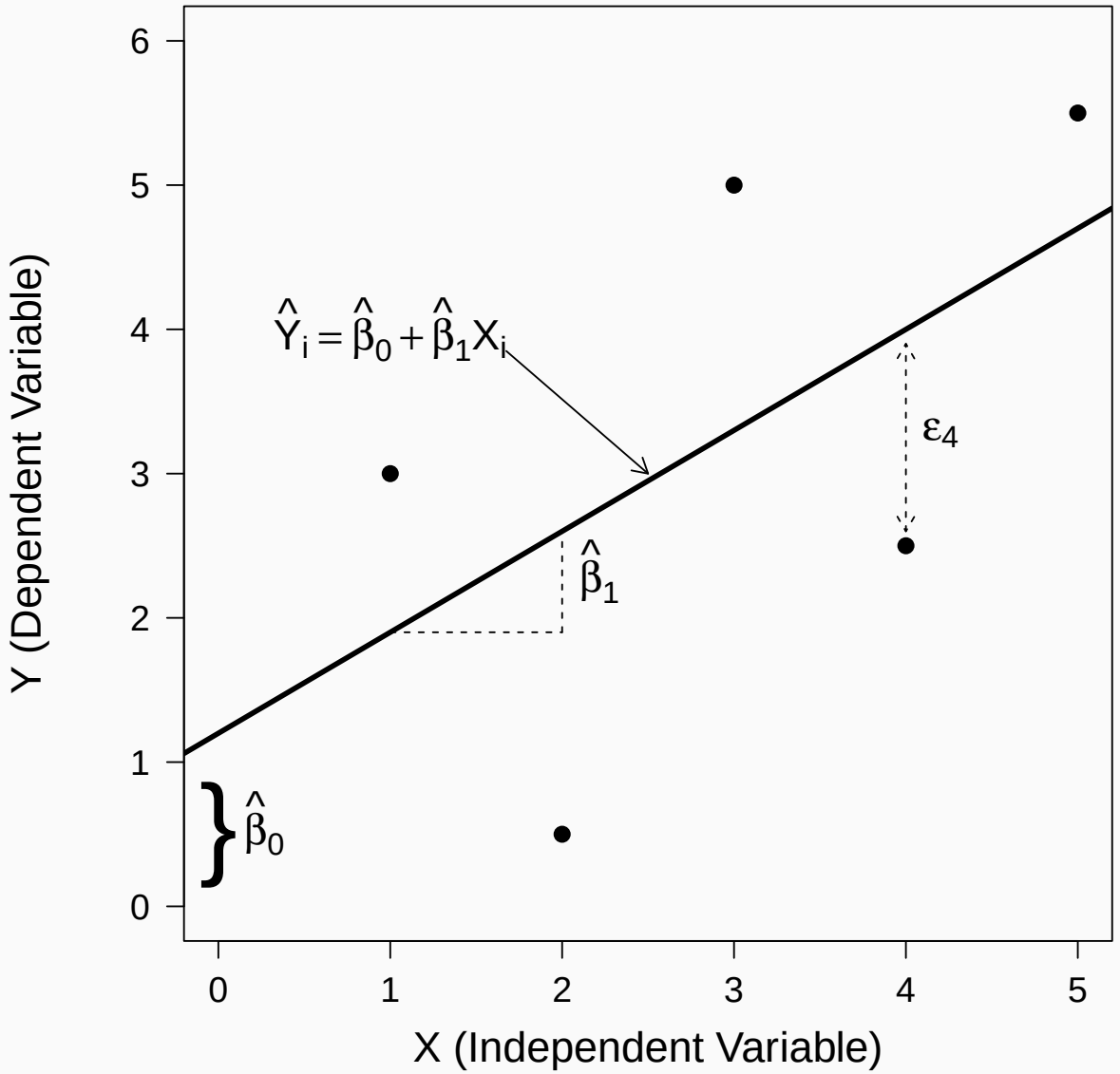


- Regression models give us the *conditional expectation* of Y given X, $E(Y|X)$. This should be more informed than the *unconditional expected value* $E(Y)$.
- Generally speaking, we try to model the *population regression function*, $E(Y|X_i) = f(X_i)$.

A linear population regression function

- A common specification is the linear population regression function: $E(Y|X_i) = \alpha + \beta_1 X_i$.
- Equivalent representation: $Y_i = \alpha + \beta_1 X_i + u_i$.
- It's important to emphasize that this is the *population* regression function. We usually (many would say always) have to estimate our models with a *sample* from the population.
- This means we need to put hats on the parameters (as they're estimated) and develop uncertainty measures to conduct inference (how likely is it that we obtained estimates of at least the magnitude we did given that the null hypothesis is true?)

The linear regression model



The meaning of the term “linear”

- More on all of this later (including how we estimate the *best* values for the unknown parameters, and what *best* even means)—but first: a fun trick!
- A model is linear in the *variables* if y is a linear function of every X variable.
- A model is linear in the *parameters* if each parameter is only raised to the power 1 and is not multiplied or divided by any other parameter.
- The **linear regression model** is linear in the parameters.
- **BUT**, a linear regression model need not be linear in the variables!
- Hence, the linear regression model can in fact produce a variety of nonlinear relationships!

Using simulation for resampling (bootstrapping)

- Another teaser: let's say we have a real dataset.
- Remember that this represents a sample from the population.
- How might we use simulation to reproduce the sampling distribution?
- We'll revisit this idea later.