DATA 252/DATA 551
Modeling and Simulation

Lecture 4: Continuous distributions

February 10, 2020

Distribution name reference in R

Distribution	R name	additional arguments
beta	beta	shape1, shape2, ncp
binomial	binom	size, prob
Cauchy	cauchy	location, scale
chi-squared	chisq	df, ncp
exponential	exp	rate
F	f	df1, df2, ncp
gamma	gamma	shape, scale
geometric	geom	prob
hypergeometric	hyper	m, n, k
log-normal	lnorm	meanlog, sdlog
logistic	logis	location, scale
negative binomial	nbinom	size, prob
normal	norm	mean, sd
Poisson	pois	lambda
signed rank	signrank	n
Student's t	t	df, ncp
uniform	unif	min, max
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m, n

rbinom → simulate
dbinom → prob.

runif

Continuous random variables

Recall, a random variable X is continuous if its sample space is continuous (and infinite). Can you think of some real life examples?

Waiting time
Weight
height

For discrete random variables, we calculate $P(X = x)$ for each possible value. We **cannot** do the same with continuous random variables. Why?

$P(\text{weight} = 121.5 \text{ lbs}) = \_\_\_$ $\quad P(\text{weight} = 121.6 \text{ lbs}) = \_\_\_$

no one weighs exactly 121.5 lbs; too many values inbetween

Important property of a continuous random variable X:
$P(X = x) = 0$ for any value x. As a result:

- EX: $P(1 < X < 3) = P(1 \le X \le 3) = P(1 < X \le 3)$
- Only makes sense to calculate $P(X \in A)$ for some subset $A \subset \mathbb{R}$.

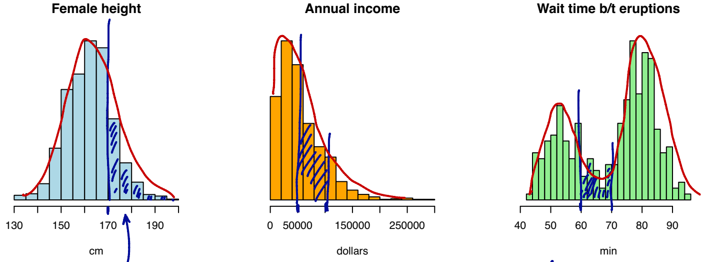
Don't do: $P(\text{weight} = 120 \text{ lbs})$

do: $P(\text{weight} < 130 \text{ lbs})$

Probability of continuous distributions

We can use **histograms** to visualize *univariate*, continuous distributions (the discrete equivalent is called a bar chart).

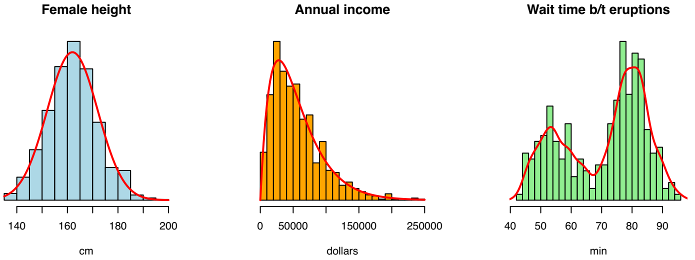
For the following continuous distributions, how would you calculate something like $P(a < X < b)$?



$P(x > 170 \text{ cm}) =$ $P(50k < X < 100k)$ $P(60 < X < 70)$

Probability density function

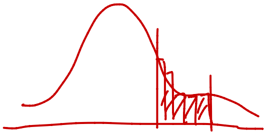
The probability mass function $f(x) = P(X = x)$ cannot be defined for continuous random variables. Instead, we use something called the probability density function (defined as a function such that $P(X \in A) = \int_A f(x)dx$).



A small digression: **density estimation** is a very big topic in statistics. We'll do a little demo in R...

Common continuous distributions

For an *arbitrary* continuous distribution, calculating probabilities involves integration (in closed form or with approximation techniques).



Think of:

$f(x) = mx + b$, w/ parameters m = slope
b = Intercept

R can handle common continuous distributions (i.e., their densities have certain functional forms, specified by some parameters); these are also used a lot in modeling.

- Uniform with range parameters a and b
- Exponential with rate λ
- Normal with mean μ and standard deviation σ
- Other distributions (t, F, chi-square, gamma, beta, etc.)

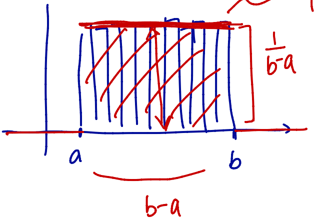
Uniform

Scenario: X is a random draw from the interval (a, b)

Parameter: a and b, end points of the range of X

Probability calculation:

$f(x) = \frac{1}{b-a} \quad x \in (a,b)$



For unif(0,1), density function is

$f(x) = \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{o.w.} \end{cases}$

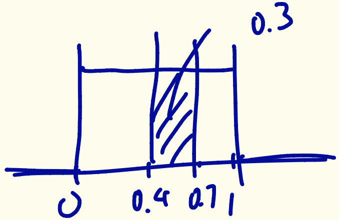
use punif(···)

Simulating from uniform:

runif(   )

$X \sim \text{Unif}(0,1)$

$f_X(x) = 1 \quad \text{for} \quad x \in (0,1)$



$P(0.4 < X < 0.7) = \underline{0.3}$

$$= \int_{0.4}^{0.7} 1 \, dx = x\Big]_{0.4}^{0.7} = 0.3$$

In R: dunif gives the density function, i.e.,

dunif(0.5) ⇒ 1, dunif(0.6) = 1, dunif(-2) = 0

This does not give the probability !!!

Instead: punif(x) = $P(X \leq x)$

EX: $P(X \leq 0.5)$ → punif(0.5) $\quad$ $P(X \geq 0.3)$ → 1 - punif(0.3)

$P(0.4 \leq X \leq 0.7)$ → punif(0.7) - punif(0.4)

Exponential

Scenario: Exponential is often used to model *Poisson waiting time*: if the number of times an event occurs during a fixed period follows a Poisson distribution, then the waiting time between two events follows an exponential distribution.

Q: if you stand in front of a window and spot taxis at a constant *rate* of 12.5 taxis per hour, how long would you *expect* to wait for the next taxi?

60 min/12.5 = 4.8 min

Parameter: λ = rate of occurrence (i.e., $1/\lambda$ = expected or average waiting time)

Probability calculation:

pexp(x , rate = __) = $P(X \le x)$

Simulating from exponential: rexp(1000, rate = 12.5)

12.5, 2.7, 3.86 ....

Exercise

- Suppose speeding cars occur at a constant rate. A policeman stands at a certain location and catches one speeding car per 12.5 minutes, on average. What is the chance that he has to wait for less than 5 minutes to catch the next speeding car? More than 15 minutes? Between 5 and 10 minutes?

$X =$ waiting time b/t speeding cars $\sim exp(\lambda = \frac{1}{12.5})$

- $P(X \leq 5)$ pexp(5, rate = $\frac{1}{12.5}$) 0.32968
- $P(X \geq 15)$ 1- pexp(15, rate = $\frac{1}{12.5}$) 0.3011942
- $P(5 \leq X \leq 10)$ 0.220991

pexp(15, rate = $\frac{1}{12.5}$, lower.tail = F)

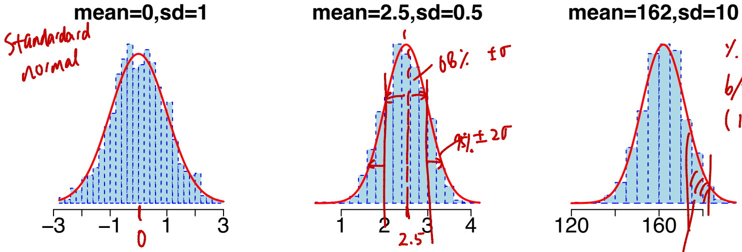
changes the output to $P(X \geq 15)$

- Simulate from the above distribution; plot a histogram to see its shape (note: the actual density function is given by dexp.)

Normal

Scenario: Normal distribution is mostly used to model the sum, or average, of "identical and independent" random variables (e.g., distribution of the sample mean), due to the *Central Limit Theorem*.

Parameter: μ = mean, σ = standard deviation



% of people b/t 162 ±20
(142, 182): 95%

Integration of the normal density is not available in closed form.

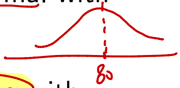
Probability calculation: pnorm

Simulating from normal: rnorm

Simulating Central Limit Theorem

1. Suppose each of you is teaching a class with $n = 25$ students. Simulate 25 exam scores from uniform $(0, 100)$ and calculate the class average of your simulated class (you can use sample code on RStudio Cloud).

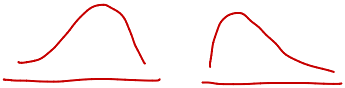
2. Repeat 1, but simulating 25 exam scores from normal with $mean = \underline{80}$ and $sd = \underline{5}$.



3. Repeat 1, but simulating 25 exam scores from beta with $shape1 = \underline{6}$ and $shape2 = \underline{3}$. (Does the choice of parameters make sense? Draw its density to verify.)

range (0,1)

rbeta( ) × 100



Do as homework

Simulating Central Limit Theorem

Key result of Central Limit Theorem: if you have a random sample of n scores, then the *sampling distribution* of the **average** of those n scores is approximately normal for large n.

Important question: How large does n have to be for CLT to work well? Design a simulation to recommend a sample size n under different population distributions.

Do as homework