Lognormal, Weibull, Beta Distributions

Engineering Statistics
Section 4.5

Josh Engwer

TTU

02 May 2016

PART I:

LOGNORMAL DISTRIBUTION

Lognormal Random Variables (Applications)

Lognormal random variables reasonably model:

- Blood pressures of adults
- File sizes of sound, music, and movie files available online
- Concentration of an air pollutant over a forest
- Ductile strength of a material
- Lifetimes of certain electronics that are **not memoryless**
- Particle sizes of a powder
- Power of received radio signals

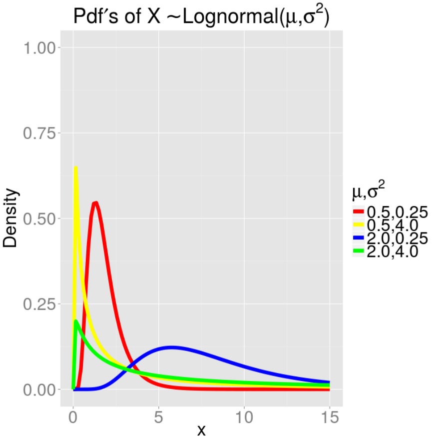
Lognormal Random Variables (Summary)

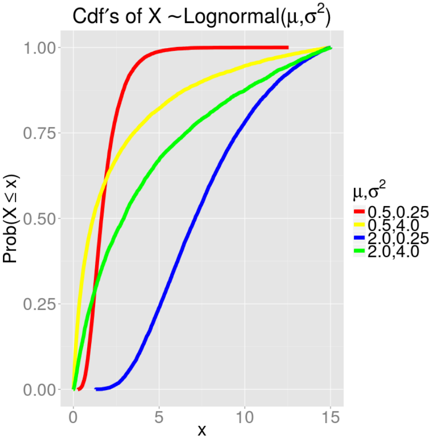
The natural logarithm of a **Lognormal** r.v. is a Normal r.v.:

Proposition

Notation	$X \sim \textit{Lognormal}(\mu, \sigma^2), \quad -\infty < \mu < \infty, \quad \sigma > 0$
Parameter(s)	$\mu \equiv$ *Mean of Normal r.v.* $\log X$ $\sigma^2 \equiv$ *Variance of Normal r.v.* $\log X$
Support	$\textit{Supp}(X) = (0, \infty)$
pdf	$f_X(x; \mu, \sigma^2) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-(\log(x)-\mu)^2/(2\sigma^2)}$
cdf	$F_X(x; \mu, \sigma^2) = \Phi\left(\frac{\log(x)-\mu}{\sigma}\right)$
Mean	$\mathbb{E}[X] = e^{\mu+\sigma^2/2}$
Variance	$\mathbb{V}[X] = e^{2\mu+\sigma^2} \cdot \left(e^{\sigma^2} - 1\right)$
Model(s)	*Blood pressure of adults* *Lifetimes of electronics* *File sizes of online multimedia*
Assumption(s)	*Natural logarithm is normally-distributed*

Lognormal Density Plots (pdf & cdf)





PART II:

WEIBULL DISTRIBUTION

Weibull Random Variables (Applications)

Weibull random variables reasonably model:

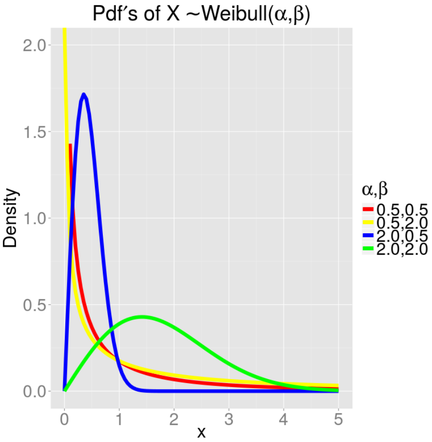
- **Lifetimes** that are **not memoryless**:
  - Lifetimes of certain electronics
  - Lifetimes of certain mechanical devices
  - Lifetimes of certain (biological) plants
  - Lifetimes of certain animals
- Sizes of particles resulting from grinding or crushing a material
- Wind speeds in weather forecasting
- Total time from ordering a defective product to returning it

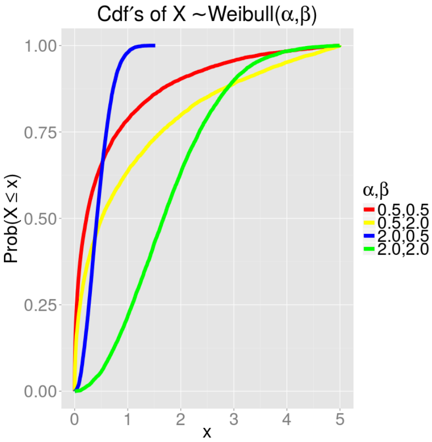
Weibull Random Variables (Summary)

Proposition

Notation	$X \sim \textit{Weibull}(\alpha, \beta), \quad \alpha, \beta > 0$
Parameter(s)	$\alpha \equiv$ *Shape parameter* $\beta \equiv$ *Scale parameter*
Support	$\textit{Supp}(X) = [0, \infty)$
pdf	$f_X(x; \alpha, \beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha}$
cdf	$F_X(x; \alpha, \beta) = 1 - e^{-(x/\beta)^\alpha}$
Mean	$\mathbb{E}[X] = \beta\Gamma\left(1 + \frac{1}{\alpha}\right)$
Variance	$\mathbb{V}[X] = \beta^2 \left\{\Gamma\left(1 + \frac{2}{\alpha}\right) - \left[\Gamma\left(1 + \frac{1}{\alpha}\right)\right]^2\right\}$
Model(s)	*Certain lifetimes* *Wind speeds*
Assumption(s)	*Memoryless Property does not hold*

Weibull Density Plots (pdf & cdf)





PART III:

BETA DISTRIBUTION

Beta Random Variables (Applications)

Beta random variables reasonably model **proportions**/**percentages**:

- **Proportions** of some quantity in different samples:
  - Proportion of some ingredient in a mixture
  - Proportion of some mineral in a rock
  - Proportion of some nutrient in a plot of land soil
  - Relative frequency of some gene in an animal population
  - Time allocation of tasks in a managerial project

The **beta function** is a **special function** that routinely shows up in higher mathematics, combinatorics, physics and (of course) statistics:

Definition

(Beta Function)

The **beta function** is defined to be:

$$B(\alpha, \beta) := \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \quad \text{where } \alpha, \beta > 0$$

Moreover, the beta function possesses several useful properties:

Corollary

(Useful Properties of the Beta Function)

- $B(\alpha+1, \beta) = \left(\frac{\alpha}{\alpha+\beta}\right) B(\alpha, \beta)$
- $B(\alpha, \beta+1) = \left(\frac{\beta}{\alpha+\beta}\right) B(\alpha, \beta)$

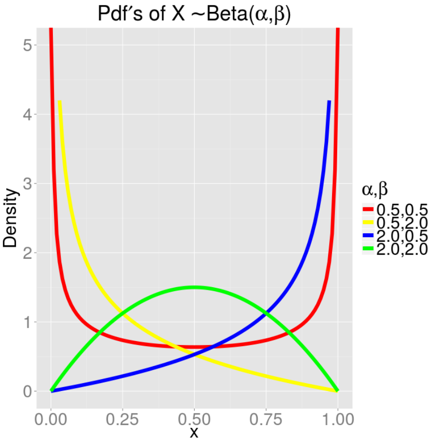
Beta Random Variables (Summary)

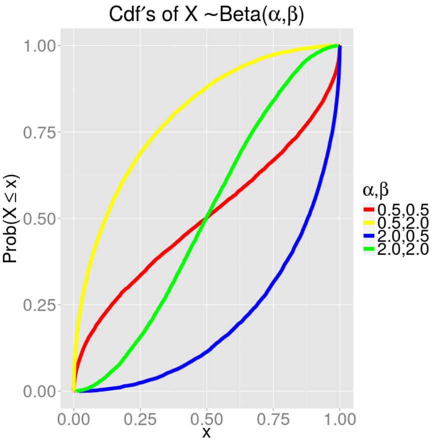
The **beta distribution** is a continuous dist. whose support is a **finite interval**:

Proposition

Notation	$X \sim \textit{Beta}(\alpha, \beta), \quad \alpha, \beta > 0$
Parameter(s)	$\alpha \equiv 1^{st}$ *Shape parameter* $\beta \equiv 2^{nd}$ *Shape parameter*
Support	$\textit{Supp}(X) = (0, 1)$
pdf	$f_X(x; \alpha, \beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$
cdf	$B_I(x; \alpha, \beta) = \frac{1}{B(\alpha,\beta)} \int_0^x \xi^{\alpha-1}(1-\xi)^{\beta-1}\, d\xi$
Mean	$\mathbb{E}[X] = \frac{\alpha}{\alpha+\beta}$
Variance	$\mathbb{V}[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
Model(s)	*Proportions or percentages*
Assumption(s)	*Support is the finite interval* $(0, 1)$ *Convert* $(a, b) \to (0, 1)$ *by* $Y = \frac{X-a}{b-a}$

Beta Density Plots (pdf & cdf)





Computing Probabilities involving Beta r.v.'s

Let $X \sim \text{Beta}(\alpha, \beta) \iff$ pdf $f_X(x) = \dfrac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$

Then, $\text{Prob}(a \le X \le b) = \displaystyle\int_a^b f_X(x)\,dx = \frac{1}{B(\alpha,\beta)} \underbrace{\int_a^b x^{\alpha-1}(1-x)^{\beta-1}\,dx}_{\text{Nonelementary integral!!}}$

and the cdf is

$$F_X(x) = \text{Prob}(X \le x) = B_I(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} \underbrace{\int_0^x \xi^{\alpha-1}(1-\xi)^{\beta-1}\,d\xi}_{\text{Nonelementary integral!!}}$$

The problem with the Beta distribution is the resulting integrals have no finite closed-form anti-derivative when α, β are not integers!!
If α, β are integers, tedious use of **polynomial expansion** is necessary!!

The fix to this is to **numerically** approximate the Beta **cdf** via a **table**.

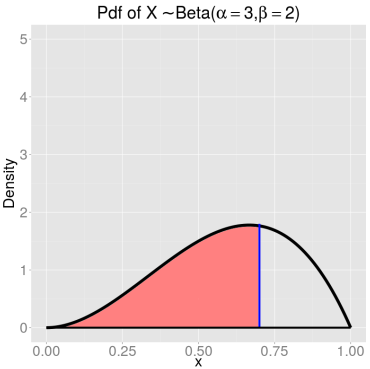
Hence, use the table for the **Beta cdf**, which is called the **incomplete Beta function** and is denoted $B_I(x;\alpha,\beta)$.

Computing Incomplete Beta Fcn $B_I(x;\alpha,\beta)$ via Table

$\beta = $ **2.0**	1^{st} **Shape Parameter** (α)						
x	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**	**3.5**
0.00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.05	0.32982	0.09750	0.02711	0.00725	0.00189	0.00048	0.00012
0.10	0.45853	0.19000	0.07431	0.02800	0.01028	0.00370	0.00131
0.15	0.55190	0.27750	0.13217	0.06075	0.02723	0.01198	0.00520
0.20	0.62610	0.36000	0.19677	0.10400	0.05367	0.02720	0.01360
0.25	0.68750	0.43750	0.26563	0.15625	0.08984	0.05078	0.02832
0.30	0.73943	0.51000	0.33685	0.21600	0.13556	0.08370	0.05102
0.35	0.78388	0.57750	0.40895	0.28175	0.19024	0.12648	0.08307
0.40	0.82219	0.64000	0.48067	0.35200	0.25298	0.17920	0.12548
0.45	0.85530	0.69750	0.55091	0.42525	0.32262	0.24148	0.17880
0.50	0.88388	0.75000	0.61872	0.50000	0.39775	0.31250	0.24307
0.55	0.90848	0.79750	0.68322	0.57475	0.47672	0.39098	0.31772
0.60	0.92952	0.84000	0.74361	0.64800	0.55771	0.47520	0.40155
0.65	0.94732	0.87750	0.79917	0.71825	0.63868	0.56298	0.49264
0.70	0.96216	0.91000	0.84921	0.78400	0.71744	0.65170	0.58830
0.75	0.97428	0.93750	0.89309	0.84375	0.79160	0.73828	0.68504

Computing Prob$(X \leq x) = B_I(x; \alpha, \beta)$ by Table/Software

$\beta = 2.0$	1^{st} Shape Parameter (α)						
x	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**	**3.5**
0.00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.05	0.32982	0.09750	0.02711	0.00725	0.00189	0.00048	0.00012
0.10	0.45853	0.19000	0.07431	0.02800	0.01028	0.00370	0.00131
0.15	0.55190	0.27750	0.13217	0.06075	0.02723	0.01198	0.00520
0.20	0.62610	0.36000	0.19677	0.10400	0.05367	0.02720	0.01360
0.25	0.68750	0.43750	0.26563	0.15625	0.08984	0.05078	0.02832
0.30	0.73943	0.51000	0.33685	0.21600	0.13556	0.08370	0.05102
0.35	0.78388	0.57750	0.40895	0.28175	0.19024	0.12648	0.08307
0.40	0.82219	0.64000	0.48067	0.35200	0.25298	0.17920	0.12548
0.45	0.85530	0.69750	0.55091	0.42525	0.32262	0.24148	0.17880
0.50	0.88388	0.75000	0.61872	0.50000	0.39775	0.31250	0.24307
0.55	0.90848	0.79750	0.68322	0.57475	0.47672	0.39098	0.31772
0.60	0.92952	0.84000	0.74361	0.64800	0.55771	0.47520	0.40155
0.65	0.94732	0.87750	0.79917	0.71825	0.63868	0.56298	0.49264
0.70	0.96216	0.91000	0.84921	0.78400	0.71744	0.65170	0.58830
0.75	0.97428	0.93750	0.89309	0.84375	0.79160	0.73828	0.68504
0.80	0.98387	0.96000	0.93020	0.89600	0.85865	0.81920	0.77851
0.85	0.99110	0.97750	0.95999	0.93925	0.91590	0.89048	0.86345



TI-8x	(No built-in function)	
TI-36X Pro	(No built-in function)	
MATLAB	`betacdf(0.7,3,2)`	(Stats Toolbox)
R	`pbeta(q=0.7,shape1=3,shape2=2)`	
Python	`scipy.stats.beta.cdf(0.7,3,2)`	(Need SciPy)

Fin.