Types of error and probability distributions

Gov 2001 Section

February 4, 2015

Outline

Replication Paper

- Read *Publication, Publication* on Gary's website.
- Find a partner to coauthor with.
- Find a set of papers you would be interested in replicating.
  1. Recently published (in the last two years).
  2. From a top journal in your field.

     Think *American Political Science Review* or *American Economic Review*, not the Nordic Council for Reindeer Husbandry Research's *Rangifer*. Unless your field is reindeer husbandry.
  3. Use methods at least as sophisticated as in this class (more than just basic OLS).

Replication Paper

- Have a classmate approve your choice of article. They should make sure it fits the criteria listed in *Publication*, *Publication*.
- Each set of partners should submit a PDF of your article, a brief (2-3 sentence) explanation for why you picked it, and the name of the person who checked that it met the requirements in *Publication*, *Publication*.
- Begin to find the data. Some journals require authors to submit their data to the journal. Some authors you'll have to email directly for the data.

Canvas

- Use the discussion board on Canvas. You'll get an answer quicker than if you email us. Please don't paste big chunks of code though. You don't want people copying your work.

- Everyone should change their notification preferences in Canvas. We'll be using the "Announcements" feature as a way to email the class. The only way you'll get these messages is if you change your preferences.
  First you have to make sure your email address is linked to Canvas. Go to 'Settings' in the top right, then on the far right under 'Ways to Contact' you should see your email address. If it's not there, be sure to add it.
  Then change your notification settings by by clicking 'Settings' in the top right, and then 'Notifications' on the left, and choosing 'Notify me right away' for 'Announcement.'

This week's homework

- By later tonight there will be two assignments on the Quizzes section on Canvas
  1. The second problem set, which you'll complete exactly as you did last week
  2. An assessment problem, which you must complete **independently** and can only submit one time to Canvas
- We won't answer any questions about R code or anything on the assessment problem. If you have a clarifying question, email all three of us and we'll post an answer on Canvas if we think we need to clarify something about the question.

Debugging R code

```
OLS <- function(y,X){
  X <- as.matrix(cbind(1,X))
  betas <- solve(X %*% X) %*% t(X) %*% y
  return(betas)
}
OLS(y.sim, covs)
```

```
Error in X %*% X : non-conformable arguments
```

Debugging R code

First create objects in your workspace that are the same name as the arguments in your function

```
X <- covs
y <- y.sim
```

Debugging R code

Next, step through each line of code inside your function and check to make sure the results look like they're supposed to.

```
> X <- as.matrix(cbind(1,X))
> head(X)
     1        x1          x2        x3
[1,] 1 2.347927  -3.490413 14.657524
[2,] 1 4.104924 -11.292412  8.288408
[3,] 1 1.650112  -5.283454  9.886145
[4,] 1 3.428719  -1.956373 15.244391
[5,] 1 3.621538  -5.238201 11.896115
[6,] 1 5.347933  -5.765387 12.549702
```

Debugging R code

When you get an error, stop and figure out what's wrong

```
> betas <- solve(X %*% X) %*% t(X) %*% y
Error in X %*% X : non-conformable arguments
```

```
> solve(X %*% X)
Error in X %*% X : non-conformable arguments
> t(X) %*% y
           [,1]
1     35436.61
x1   105809.93
x2 -216456.24
x3   376985.23
```

Debugging R code

Once you've isolated the problem, figure out what you need to do to fix it

```
> solve(X %*% X)
Error in X %*% X : non-conformable arguments
> dim(X)
[1] 1000    4
```

```
> solve(t(X) %*% X)
                1            x1            x2            x3
1   0.0129365839 -7.370667e-04  5.463082e-04 -6.817348e-04
x1 -0.0007370667  2.725400e-04  4.691896e-07 -5.747609e-06
x2  0.0005463082  4.691896e-07  1.026750e-04 -2.867793e-06
x3 -0.0006817348 -5.747609e-06 -2.867793e-06  6.643738e-05
```

Debugging R code

Now rerun the corrected code

```
> X <- covs
> X <- as.matrix(cbind(1,X))
> betas <- solve(t(X) %*% X) %*% t(X) %*% y
```

```
>    betas
           [,1]
1    5.1839790
x1   0.4499720
x2  -3.8968102
x3   0.9001369
```

Outline

Types of uncertainty

Estimation uncertainty: "arises from not knowing [the parameters] perfectly, an unavoidable consequence of having fewer than an infinite number of observations"
Under the likelihood framework, we assume that there's a true value of the parameters that exists in the world and with an infinite amount of data we could estimate their values perfectly

Fundamental uncertainty: "results from innumerable chance events such as weather or illness that may influence Y but are not included in X"
No matter what we do, the world isn't deterministic. If we could re-run the world starting at midnight last night, things would come out pretty similar, but everything wouldn't be perfectly the same. Leaves or snow falling from trees would land in different places; some people would choose to read a book instead of watch TV, etc.

Uncertainty in the UPM framework

In the first problem set you simulated outcomes based on this model:

$$y_i \sim N(\mu, \sigma^2 = 36)$$
$$\mu = 4 + .5x_{i,1} - 4x_{i,2} + x_{i,3}$$

Where is the fundamental uncertainty in this model?

$$\sigma^2$$

Where is the estimation uncertainty in this model?

In the standard errors of the four β coefficients and σ^2

Identifying uncertainty in regression results

```
Call:
lm(formula = y.sim ~ covs$x1 + covs$x2 + covs$x3)

Residuals:
     Min       1Q   Median       3Q      Max
-21.2804  -3.7465   0.0219   3.9893  17.5016

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.18398    0.68112   7.611 6.30e-14 ***
covs$x1      0.44997    0.09886   4.552 5.98e-06 ***
covs$x2     -3.89681    0.06068 -64.219  < 2e-16 ***
covs$x3      0.90014    0.04881  18.441  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 5.988 on 996 degrees of freedom
Multiple R-squared:  0.816, Adjusted R-squared:  0.8154
F-statistic:  1472 on 3 and 996 DF,  p-value: < 2.2e-16
```

Outline

How do we build or select a statistical model?

Imagine a friend comes to you with a bunch of data, and they want you to help build a model that helps predict some outcome of interest.

What are the first questions you should ask them?

1. What is the dependent variable?
2. What was the data generation process that created that dependent variable?

Fundamental Uncertainty and Stochastic Models

"If we played them ten times, they might win nine. But not this game. Not tonight."

- Coach Herb Brooks in *Miracle*

Or put another way:

"If $\mathbf{Y}$ is whether the Soviet team beats us, then $Y \sim Bern(0.9)$. But $y_{tonight} = 0$."

- Coach Herb Brooks in *Miracle*

Even the most certain things have fundamental uncertainty around them. We use the stochastic component of our models to capture this fact.

So why become familiar with probability distributions?

Several reasons:

- You can pick the probability distribution that corresponds with the data generation process of your dependent variable.
- You can fit models to a variety of data. Not just Normal data!
- You can help your friends or colleagues build a good statistical model for predictive or descriptive inference

What do we have to know about probability distributions in order to apply them to a particular problem?

- You have to know the "stories" behind them
- You know to pick a distribution which corresponds with the DGP for your dependent variable
- You have to understand the assumptions you're making them you pick that distribution

Outline

The Bernoulli Distribution

- **The story:** flipping a *single* coin
- The Bernoulli distribution has one parameter, π, which is the probability of "success".
- If $Y \sim Bern(\pi)$, then $y = 1$ with success probability π and $y = 0$ with failure probability $1 - \pi$.
- Ideal for modeling **one-time** yes/no (or success/failure) events.
- Other examples:
  - one voter voting yes/no
  - a patient living or dying in a cancer drug trial

The Bernoulli Distribution

$Y \sim \text{Bernoulli}(\pi)$

y can take on a value of either 0 or 1. Nothing else.

probability of success: $\pi \in [0, 1]$

$p(y|\pi) = \pi^y(1-\pi)^{(1-y)}$

$E(Y) = \pi$

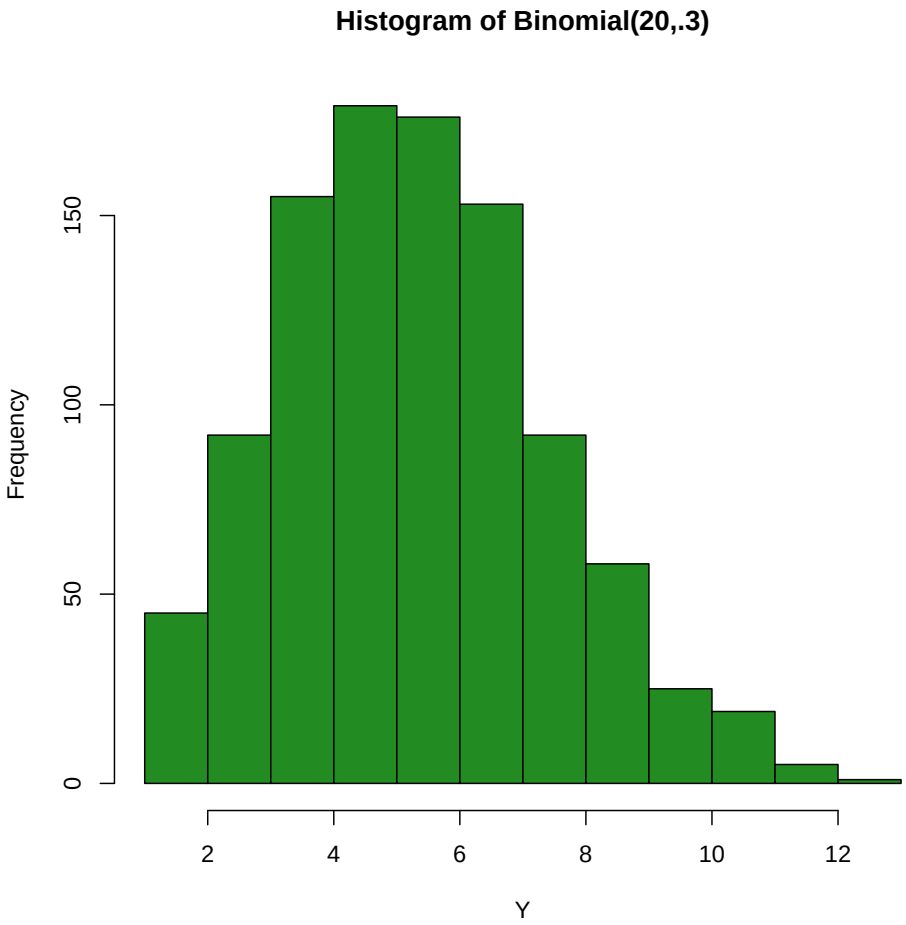
$\text{Var}(Y) = \pi(1-\pi)$

```
rbinom(100, size = 1, prob = .7)
[1] 0 0 1 1 1 1 1 0 1 1 1 1 1...
```

The Binomial Distribution

- **The story:** flipping a coin a bunch of times and counting how many times it came up heads
- The Binomial distribution is the total of a bunch of Bernoulli trials.
- Two parameters: probability of "success", π, and number of trials, n
  Generally you *must* know the number of trials that generated your data. If you don't, or if there's a huge number of trials, you might want to use a different distribution.
- Examples:
  - You flip a coin three times and count the total number of heads you got. (The order doesn't matter.)
  - The number of women in a group of 10 Harvard students
  - The number of snowy days in the seven-day week

The Binomial Distribution



$$Y \sim \text{Binomial}(n, \pi)$$

$$y = 0, 1, \ldots, n$$

number of trials: $n \in \{1, 2, \ldots\}$

probability of success: $\pi \in [0, 1]$

$$p(y|\pi) = \binom{n}{y} \pi^y (1 - \pi)^{(n-y)}$$

$$E(Y) = n\pi; \text{Var}(Y) = n\pi(1 - \pi)$$

```
rbinom(100, size = 5,
prob = .7)
4 4 5 5 3 5 4 2 3 5...
```

The Multinomial Distribution

- **The story:** rolling a dice (even or unevenly weighted) a bunch of times and counting how many times each number comes up
- Generalization of the binomial, which is just a special case of the multinomial
- Two parameters: number of trials, n, and a vector of probabilities of each of the K outcomes, $\mathbf{p} = \{p_1, p_2, ...p_K\}$
- Multinomial assumes that you have mutually exclusive outcomes.

- Examples:
  - you toss a die 15 times and count how many times 1, 2,...6 show up
  - election vote totals where there's 3+ candidates to choose from

The Multinomial Distribution

$Y \sim \text{Multinomial}(n, \pi_1, \ldots, \pi_K)$

y_i is a vector of counts of successes for each outcome, where each count ranges from 0 to n; the sum of this vector must equal n

number of trials: $n \in \{1, 2, \ldots\}$

probability of success for outcome k: $\pi_k \in [0, 1]$; $\sum_{k=1}^{K} \pi_k = 1$

The Multinomial Distribution

$$p(\mathbf{y}|n,\boldsymbol{\pi}) = \frac{n!}{y_1!y_2!\ldots y_K!}\pi_1^{y_1}\pi_2^{y_2}\ldots\pi_K^{y_K}$$

$$E(Y) = \{n\pi_1, n\pi_2, \ldots n\pi_K\}$$

$$\text{Var}(Y) = \{n\pi_1(1-\pi_1), n\pi_2(1-\pi_2), \ldots n\pi_K(1-\pi_K)\}$$

```
rmultinom(100, size = 5, prob = c(.2,.4,.3,.1))
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    0    0    2    0    0    4    0    3
[2,]    2    2    1    2    2    1    2    1
[3,]    2    2    2    2    3    0    2    0
[4,]    1    1    0    1    0    0    1    1
```

What is this?

It's a Prussian soldier getting kicked in the head by a horse.

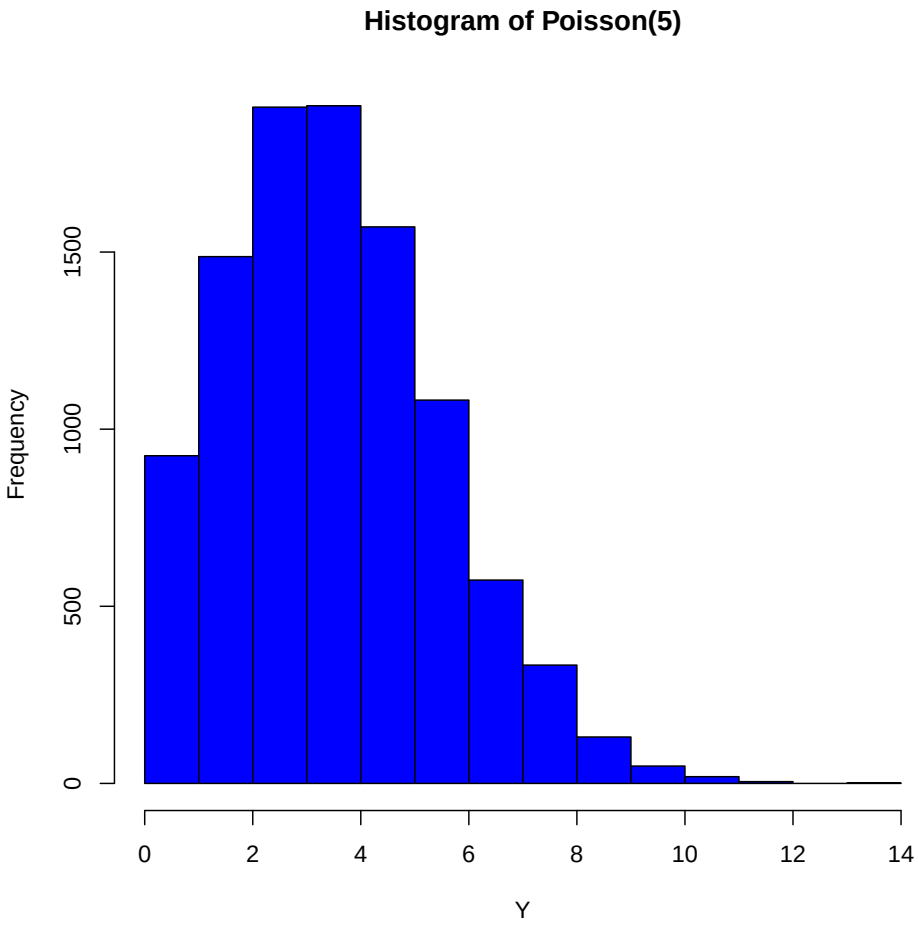
Also, it's the logo for Stata Press

The Poisson Distribution

- **The story:** count the number of times an (uncommon) event happens
- One parameter: the rate of occurrence, usually called λ
- Represents the number of events occurring in a fixed period of time or in a specific distance, area or volume.
- Can never be negative – so, good for modeling events.
- Assumes that you have lots of trials (or lots of "opportunities" for an event to happen)
- Assumes the probability of a "success" on any particular trial is tiny
- Makes a potentially strong assumption about the mean and variance

- Examples:
  - Number of snowflakes that hit a 1" x 1" square on the sidewalk during a snowstorm
  - Number of executive orders a president issues in a week
  - Number Prussian solders who died each year by being kicked in the head by a horse (Bortkiewicz, 1898)

The Poisson Distribution



$Y \sim \text{Poisson}(\lambda)$

$y = 0, 1, \ldots$

rate of occurrence parameter, λ, is always greater than zero

$$p(y|\lambda) = \frac{e^{-\lambda}\lambda^y}{y!}$$

$E(Y) = \lambda$; $\text{Var}(Y) = \lambda$

```
rpois(100, lambda = 2)
0 4 1 3 2 0 4 3 3 2...
```

Outline

The Univariate Normal Distribution

- **The story:** any outcome that can take any real number as a value
- Describes data that cluster in a bell curve around the mean.
- It's tough to think of examples of things are are *truly* unbounded, so be careful not to extrapolate your results outside of the range of valid outcomes.
  If we're using a normal distribution to model vote outcomes, don't tell me that you predict a candidate to get 110% of the vote

The Univariate Normal Distribution

Unless you're studying Pakistan

Pakistan elections: 49 polling stations had more than 100 percent voting, say observers

World | Indo-Asian News Service | Updated: May 14, 2013 15:19 IST

ISLAMABAD: Forty-nine polling stations in Pakistan achieved the impossible - the votes polled were more than the total number of voters.

Data gathered by Free and Fair Election Network (FAFEN) observers at polling stations have shown voter turnout greater than 100 percent, reported News International.

The Univariate Normal Distribution

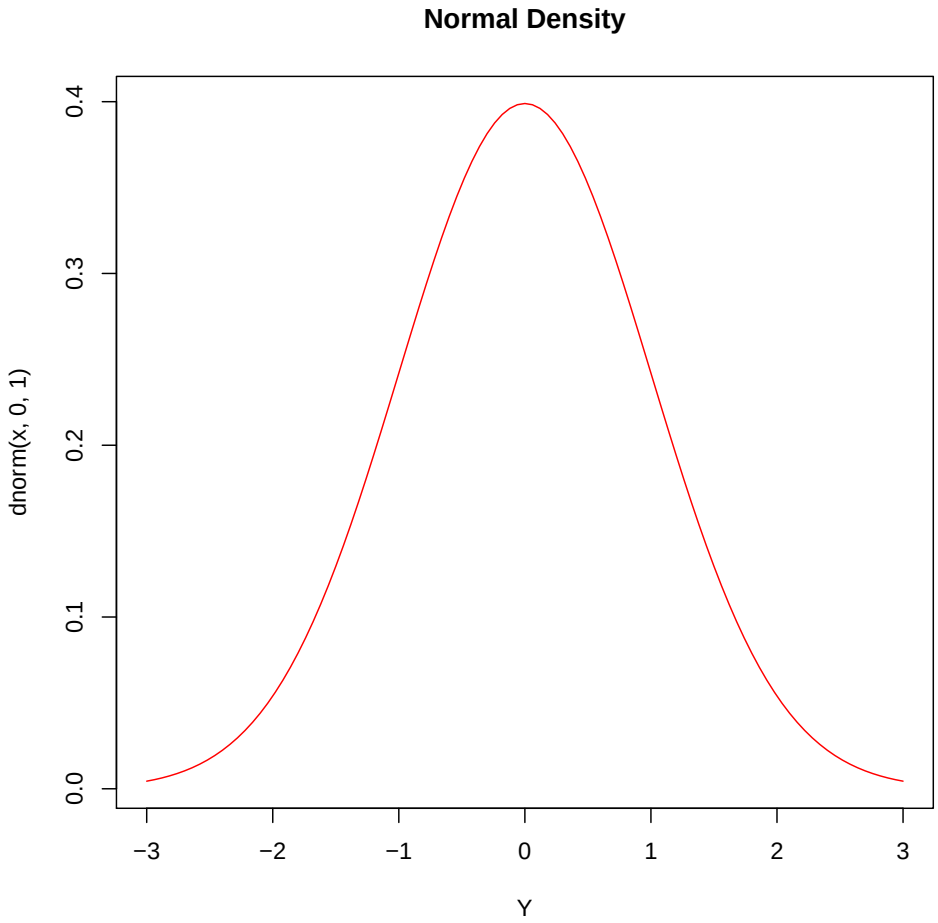
Or Vladimir Putin

Putin Clings to Victory as Russia's Voter Turnout Exceeds 146%

Seth Abramovitch
Filed to: RUSSIA 12/04/11 9:15pm

64,062 2 ★

The Univariate Normal Distribution



$$Y \sim \text{Normal}(\mu, \sigma^2)$$

$y \in \mathbb{R}$

mean: $\mu \in \mathbb{R}$

variance: $\sigma^2 > 0$

$$p(y|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

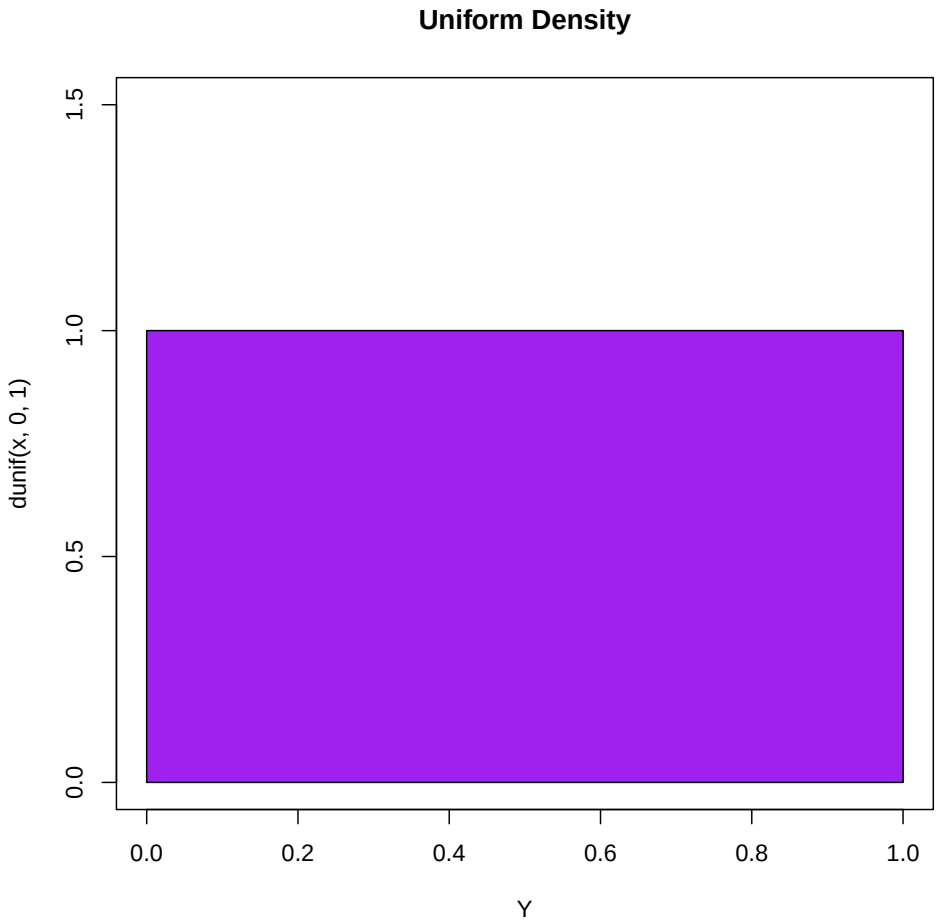
$$E(Y) = \mu; \text{Var}(Y) = \sigma^2$$

```
rnorm(100, mean = 0, sd = 1)
-1.2167436558 -0.8748603823
-0.2917406643...
```

The Uniform Distribution

- **The story:** Any value in the interval you chose is equally probable.
- Two parameters: a and b (or α and β), lower and upper bounds of the range of possible results
- Intuitively easy to understand, but often examples are discrete
- Examples:
  - the degree of longitude you're pointing to if you stop a spinning globe with your finger
  - the bib number of a person who comes in first in a race (discrete)
  - drawing a ball from a tumbler in a lottery (also discrete)

The Uniform Distribution



$$Y \sim \text{Uniform}(\alpha, \beta)$$

$$y \in [\alpha, \beta]$$

Interval: $[\alpha, \beta]$; $\beta > \alpha$

$$p(y|\alpha, \beta) = \frac{1}{\beta - \alpha}$$

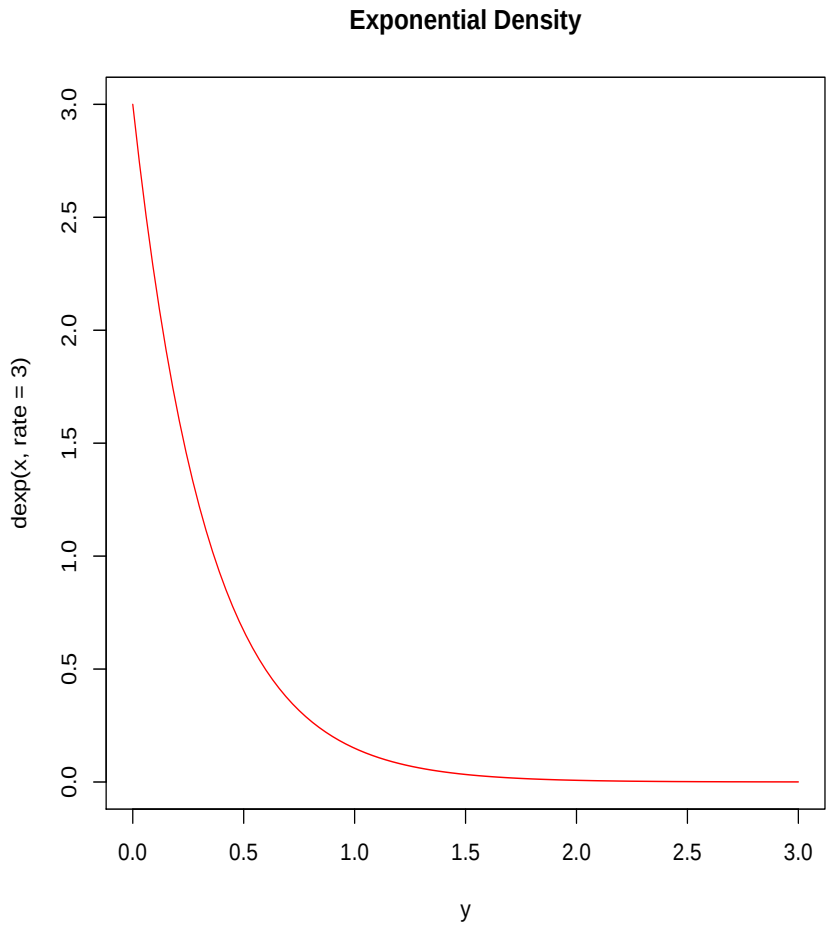
$$E(Y) = \frac{\alpha+\beta}{2}; \text{Var}(Y) = \frac{(\beta-\alpha)^2}{12}$$

```
runif(100, min=-5, max=10)
-3.4090615 8.7924703
6.8907343...
```

The Exponential Distribution

- **The story:** how long do you have to wait until an event occurs?
- One parameter: λ, arrival rate of the event
- The distribution assumes that your process is memoryless. The expected time until the event happens is constant, regardless of how much time has passed since the last event.
  $E(y) = E(y|y > 1) = E(y|y > 10)...etc. = 1/\lambda$
- Examples:
  - How long until the bus arrives?
  - Time between bombings in a war-torn country

The Exponential Distribution



$$Y \sim \text{Expo}(\lambda)$$

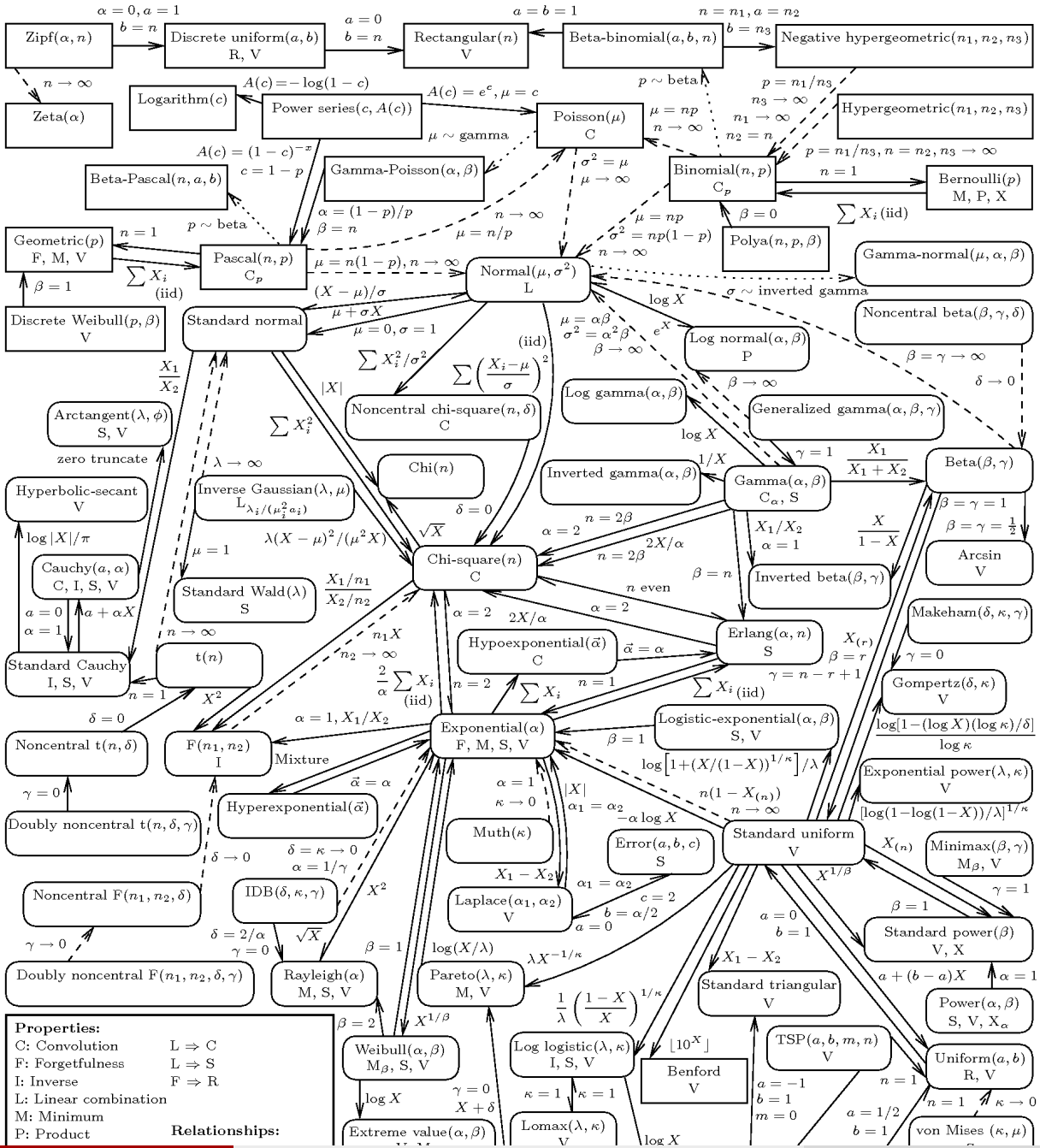
$$y \in [0, \infty]$$

$$\lambda > 0$$

$$p(y|\lambda) = \lambda e^{-\lambda y}$$

$$E(Y) = \frac{1}{\lambda}; \text{Var}(Y) = \frac{1}{\lambda^2}$$

```
rexp(100, rate = 3)
0.0062472728 0.4941267226
0.1309825860...
```


Zipf(α, n)
Discrete uniform(a, b)
R, V
Rectangular(n)
V
Beta-binomial(a, b, n)
Negative hypergeometric(n1, n2, n3)
Zeta(α)
Logarithm(c)
Power series(c, A(c))
Poisson(μ)
C
Hypergeometric(n1, n2, n3)
Beta-Pascal(n, a, b)
Gamma-Poisson(α, β)
Binomial(n, p)
Cp
Bernoulli(p)
M, P, X
Geometric(p)
F, M, V
Pascal(n, p)
Cp
Polya(n, p, β)
Gamma-normal(μ, α, β)
Normal(μ, σ²)
L
Discrete Weibull(p, β)
V
Standard normal
Noncentral beta(β, γ, δ)
Log normal(α, β)
P
Arctangent(λ, φ)
S, V
Noncentral chi-square(n, δ)
C
Log gamma(α, β)
Generalized gamma(α, β, γ)
Chi(n)
Inverted gamma(α, β)
Gamma(α, β)
Cα, S
Beta(β, γ)
Hyperbolic-secant
V
Inverse Gaussian(λ, μ)
Chi-square(n)
C
Arcsin
V
Cauchy(a, α)
C, I, S, V
Standard Wald(λ)
S
Inverted beta(β, γ)
Makeham(δ, κ, γ)
Standard Cauchy
I, S, V
t(n)
Hypoexponential(α⃗)
C
Erlang(α, n)
S
Gompertz(δ, κ)
V
Noncentral t(n, δ)
F(n1, n2)
I
Exponential(α)
F, M, S, V
Logistic-exponential(α, β)
S, V
Exponential power(λ, κ)
V
Doubly noncentral t(n, δ, γ)
Hyperexponential(α⃗)
Muth(κ)
Error(a, b, c)
S
Standard uniform
V
Minimax(β, γ)
Mβ, V
Noncentral F(n1, n2, δ)
IDB(δ, κ, γ)
Laplace(α1, α2)
V
Standard power(β)
V, X
Doubly noncentral F(n1, n2, δ, γ)
Rayleigh(α)
M, S, V
Pareto(λ, κ)
M, V
Standard triangular
V
Power(α, β)
S, V, Xα
Weibull(α, β)
Mβ, S, V
Log logistic(λ, κ)
I, S, V
Benford
V
TSP(a, b, m, n)
V
Uniform(a, b)
R, V
Extreme value(α, β)
Lomax(λ, κ)
V
von Mises (κ, μ)
Properties:
C: Convolution
F: Forgetfulness
I: Inverse
L: Linear combination
M: Minimum
P: Product
L ⇒ C
L ⇒ S
F ⇒ R
Relationships:

Outline

Coding a density function from scratch

In later problem sets, you're going to have to code likelihood functions in R.

Often you'll be able to use canned functions like `rnorm()` or `rpois()`, but sometimes your likelihood function won't look like a distribution you're familiar with and you'll have to code the density from scratch.

We're going to get practice coding from scratch by programming the PDF of the normal distribution.

To do this we have to first write a function which takes the arguments y, μ, and σ.

Coding the PDF of the normal distribution

Recall that the PDF of a normal distribution is:

$$p(y|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

If we want to code this into R, we to set up a new function which takes the arguments y, μ, and σ:

```
normal <- function(y, mu, sigma){ # y must go first!
  exp(-(y - mu) ^ 2 / (2 * sigma^2)) / (sigma * sqrt(2 * pi))
}
```

Coding the PDF of the normal distribution

We know that, by definition, PDFs must integrate to 1 across their support

Let's check that we coded our function correctly by using the `integrate()` function in R:

```
integrate(normal, lower = -1000, upper = 1000,
          mu = 0, #extra arguments needed for your function
          sigma = 1)
1 with absolute error < 9e-05
```

Notice that we didn't need to integrate from $-\infty$ to ∞ to get the right answer.

Simulating from a coded PDF

Now we want to simulate from our function and plot its PDF. How could we do this?

To do this we first calculate the density at a bunch of different values of y:

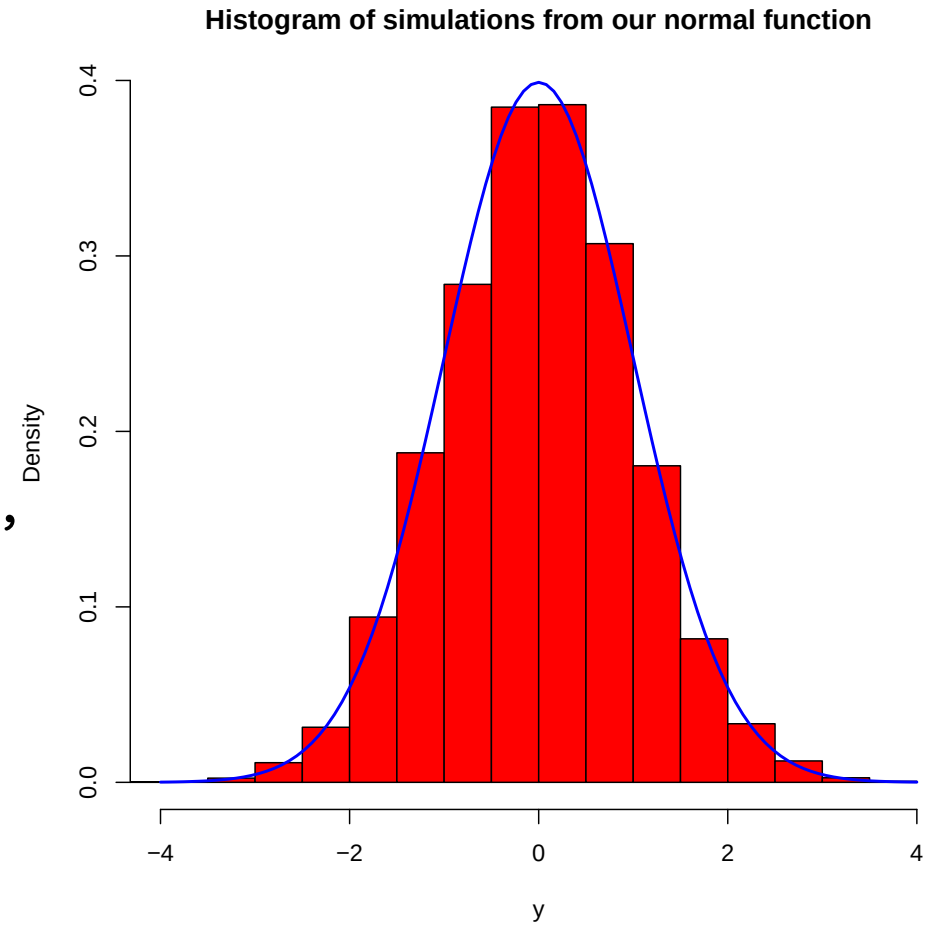
```
values <- seq(-10, 10, .001)
weights <- normal(values,
                  mu = 0,
                  sigma=1)
```

Simulating from a coded PDF

Now we draw samples from our `values` vector, where the probability of drawing each value is weighted by the densities (`weights`)

```
draws <- sample(values,
                size = 10000,
                prob = weights,
                replace = T)
```

We now have a vector of 10000 draws from the PDF we coded. We use them to plot the PDF



Using simulations to integrate

The draws we took from our function can also be used to integrate

We could use the `integrate()` function, but for some complicated likelihoods it might be very slow or not work

```
integrate(normal, lower = -1.96, upper = 1.96,
          mu = 0,
          sigma = 3)
0.9500042 with absolute error < 1e-11
```

Or we can use the draws we took:

```
mean(draws > -1.96 & draws < 1.96)
[1] 0.9506
```

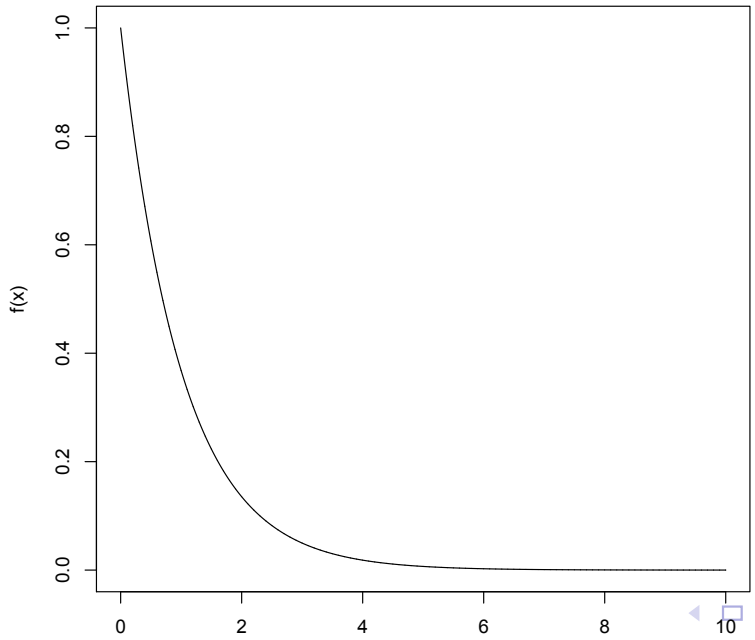
Outline

Transforming Distributions

- $X \sim p(x|\theta)$
- $y = g(x)$
- How is y distributed?

For example, if $X \sim Exponential(\lambda = 1)$ and $y = log(x)$

$$y \sim ?$$



Transforming Distributions

- It is NOT true that $p(y|\theta) \sim g(p(x|\theta))$. Why?

Transforming Distributions

The Rule

- $X \sim p_x(x|\theta)$
- $y = g(x)$

$$p_y(y) = p_x(g^{-1}(y)) \left| \frac{dg^{-1}}{dy} \right|$$

- What is $g^{-1}(y)$? The inverse of y=g(x).
- What is $\left| \frac{dg^{-1}}{dy} \right|$? The Jacobian.

Transforming Distributions – the log-Normal Example

For example,

- $X \sim Normal(x|\mu = 0, \sigma = 1)$
- $y = g(x) = e^x$
- what is $g^{-1}(y)$?

$$g^{-1}(y) = x = log(y)$$

- What is $\frac{dg^{-1}}{dy}$?

$$\frac{d(log(y))}{dy} = \frac{1}{y}$$

Transforming Distributions – the log-Normal Example

- Put it all together

$$p_y(y) = p_x(log(y)) \left| \frac{1}{y} \right|$$

- Notice we don't need the absolute value because $y > 0$.

$$p_y(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(log(y))^2} \frac{1}{y}$$

- $Y \sim$ log-Normal$(0, 1)$
- Challenge: derive the chi-squared distribution.
  $X \sim N(\mu, \sigma^2)$
  $Y = X^2$