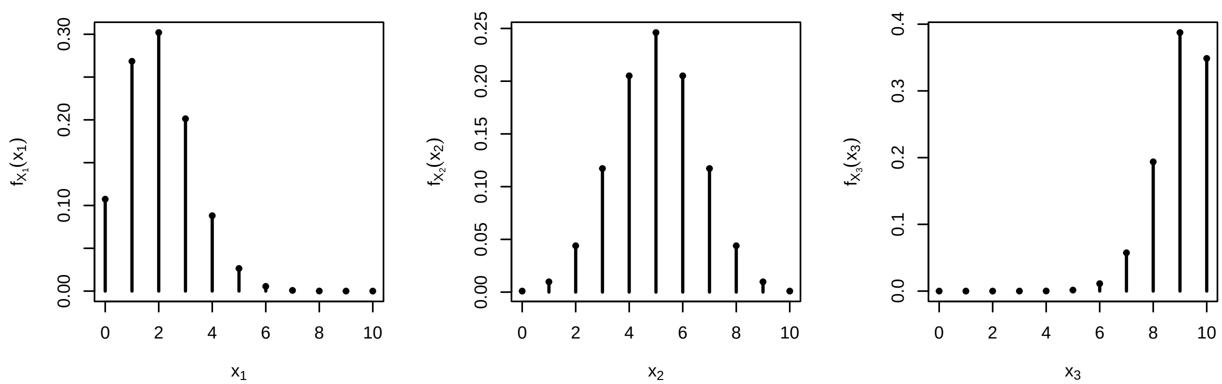
STAT 501 Fall 2022 Course Notes

Common Distributions in R

See the .Rmd file in the course Github repository for code to generate all figures.

Using R to further investigate the Binomial(n, p) distribution Make your best guess at the parameters (i.e., n and p) of the following Binomial(n, p) distributions.



Use R to find values for Binomial probabilities.

- To find $P(Y = y)$ use `dbinom(x = y, prob = p, size = n)`

```
# Example. Let Y ~ Binomial(n = 20, p = 0.2), find P(Y = 4)
dbinom(x = 4, size = 20, prob = 0.2)
## [1] 0.2181994
```

- To find $P(Y \leq y)$ use `pbinom(q = y, prob = p, size = n)`

```
# Example. Let Y ~ Binomial(n = 20, p = 0.2), find P(Y <= 4)
pbinom(q = 4, size = 20, prob = 0.2)
## [1] 0.6296483
```

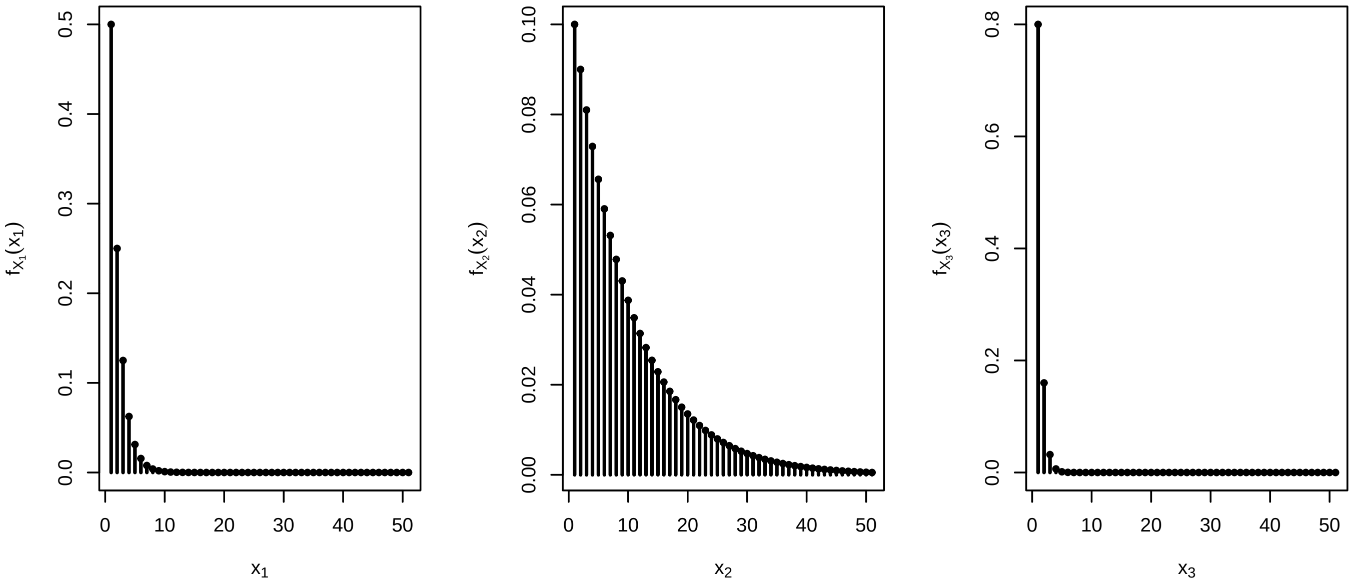
- To find the smallest y such that $P(Y \leq y) \geq c$ use `qbinom(p = c, size = n, prob = p)`

```
# Example. Let Y ~ Binomial(n = 20, p = 0.2), find y such that P(Y <= y) = 0.5
qbinom(p = 0.5, size = 20, prob = 0.2)
## [1] 4
```

- To simulate m random draws from a Binomial(n, p) distribution use `rbinom(n = m, size = n, prob = p)`

```
# Example. Let Y ~ Binomial(n = 20, p = 0.2), generate a RS of size 5
rbinom(n = 5, size = 20, prob = 0.2)
## [1] 1 3 7 2 2
```

Using R to further investigate Geom(p) RVs Make your best guess at the parameter value (i.e., p).



What parameterization does R use for the geometric RV?

Details

The geometric distribution with `prob` = *p* has density

$$p(x) = p\ (1-p)^\wedge x$$

for $x = 0, 1, 2, \ldots, 0 < p \leq 1$.

If an element of `x` is not integer, the result of `dgeom` is zero, with a warning.

The quantile is defined as the smallest value *x* such that $F(x) \geq p$, where *F* is the distribution function.

To use R to compute quantities of interst from the same Geometric(p) distribution provided in the text, do the following:

- To find $P(Y = y)$ use `dgeom(x = y - 1, prob = p)`

```
# Example. Let Y ~Geom(p = 0.2), find P(Y = 4)
dgeom(x = 4 - 1, prob = 0.2)
## [1] 0.1024
```

- To find $P(Y \leq y)$ use `pgeom(q = y - 1, prob = p)`

```
# Example. Let Y ~Geom(p = 0.2), find P(Y <= 4)
pgeom(q = 4 - 1, prob = 0.2)
## [1] 0.5904
```

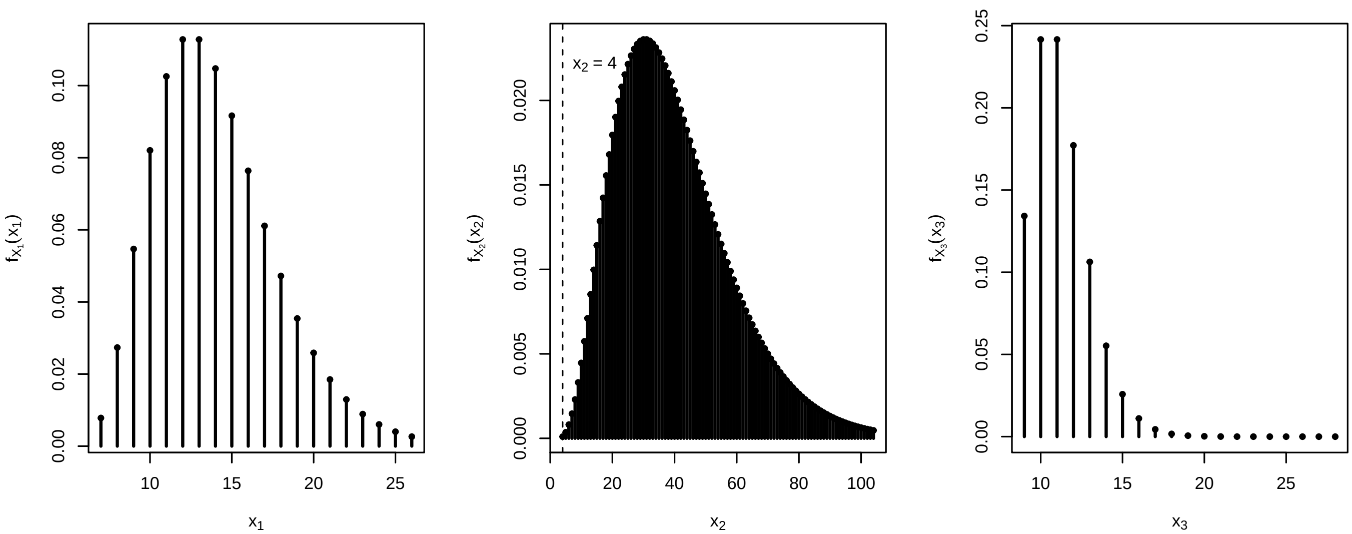
- To find the smallest y such that $P(Y \leq y) \geq c$ use `qgeom(p = c, prob = p) + 1`

```
# Example. Let Y ~Geom(p = 0.2), find y such that P(Y <= y) = 0.5
qgeom(p = 0.5, prob = 0.2) + 1
## [1] 4
```

- To simulate m random draws from a Geom(p) distribution use `rgeom(n = m, size = n) + 1`

```
# Example. Let Y ~ Geom(p = 0.2), generate a RS of size 5
rgeom(n = 5, prob = 0.2) + 1
## [1]  1  1  1 11  4
```

Using R to further investigate NegBinom(r, p) RVs Make your best guess at the parameter values (i.e., $r =?$, $p = ?$) for the following plots.



Just like `_geom`, `_nbinom` is parameterized differently in R!

Details

r

The negative binomial distribution with `size` = *n* and `prob` = *p* has density

Γ(x+n)/(Γ(n) x!) p^n (1-p)^x

for *x = 0, 1, 2, …, n > 0* and *0 < p ≤ 1*.

This represents the number of failures which occur in a sequence of Bernoulli trials before a target number of successes is reached. The mean is *μ = n(1-p)/p* and variance *n(1-p)/p^2*.

To use R to compute quantities of interst from the same parameterization of the NegativeBinomial(r, p) distribution that is provided in the text, do the following:

- To find $P(Y = y)$ use `dnbinom(x = y - r, size = r, prob = p)`

```
# Example. Y ~ NegBinom(r = 4, p = 0.2), find P(Y = 8)
dnbinom(x = 8 - 4, size = 4, prob = 0.2)
## [1] 0.0229376
```

- To find $P(Y \leq y)$ use `pgeom(q = y, size = r, prob = p)`

```
# Example. Let Y ~ NegBinom(r = 4, p = 0.2), find P(Y <= 8)
pnbinom(q = 8 - 4, size = 4, prob = 0.2)
## [1] 0.0562816
```

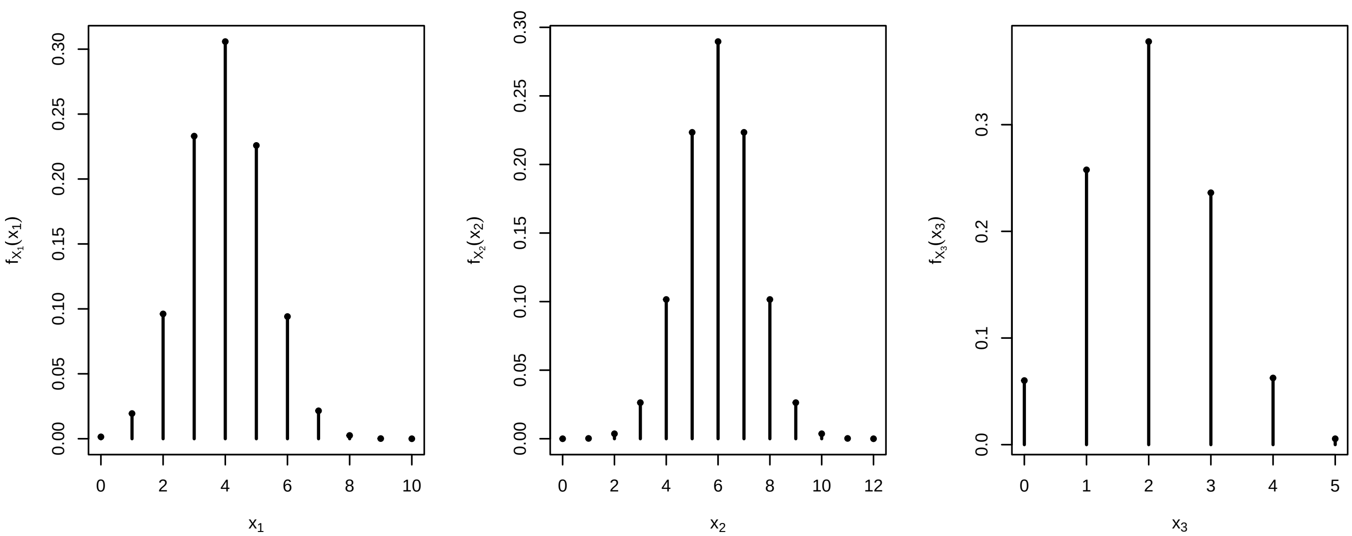
- To find the smallest y such that $P(Y \leq y) \geq c$ use `qnbinom(p = c, size = r, prob = p) + r`

```
# Example. Let Y ~ NegBinom(r = 4, p = 0.2), find y such that P(Y <= y) = 0.5
qnbinom(p = 0.5, size = 4, prob = 0.2) + 4
## [1] 19
```

- To simulate m random draws from a NegBinom(r, p) distribution use `rnbinom(n = m, size = r, prob = p) + r`

```
# Example. Let Y ~ NegBinom(r = 4, p = 0.2), generate a RS of size 5
rnbinom(n = 5, size = 4, prob = 0.2) + 4
## [1] 17 35 26 17 26
```

Using R to further investigate Hypergeometric(N, r, n) **RVs** Make your best guess at the parameter values (i.e., $N = 30$, $r = 18$, $n = ?$) for the following plots.



The help file for `_hyper` is shown below, what does each argument provide for the conventional notation n = sample size, N = total in population, M = total number of successes.

Arguments

`x, q`	vector of quantiles representing the number of white balls drawn without replacement from an urn which contains both black and white balls.
`m`	the number of white balls in the urn.
`n`	the number of black balls in the urn.
`k`	the number of balls drawn from the urn.
`p`	probability, it must be between 0 and 1.
`nn`	number of observations. If `length(nn) > 1`, the length is taken to be the number required.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.

- To find $P(Y = y)$ use `dhyper(x = y, m = M, n = N-M, k = n)`

```
# Example. Y ~ Hyper(N = 10, r = 4, n = 5), find P(Y = 3)
dhyper(x = 3, m = 4, n = 6, k = 5)
## [1] 0.2380952
```

- To find $P(Y \leq y)$ use `phyper(q = y, m = M, n = N-M, k = n)`

```
# Example. Y ~ Hyper(N = 10, r = 4, n = 5), find P(Y <= 3)
phyper(q = 3, m = 4, n = 6, k = 5)
## [1] 0.9761905
sum(dhyper(x = 0:3, m = 4, n = 6, k = 5))
## [1] 0.9761905
```

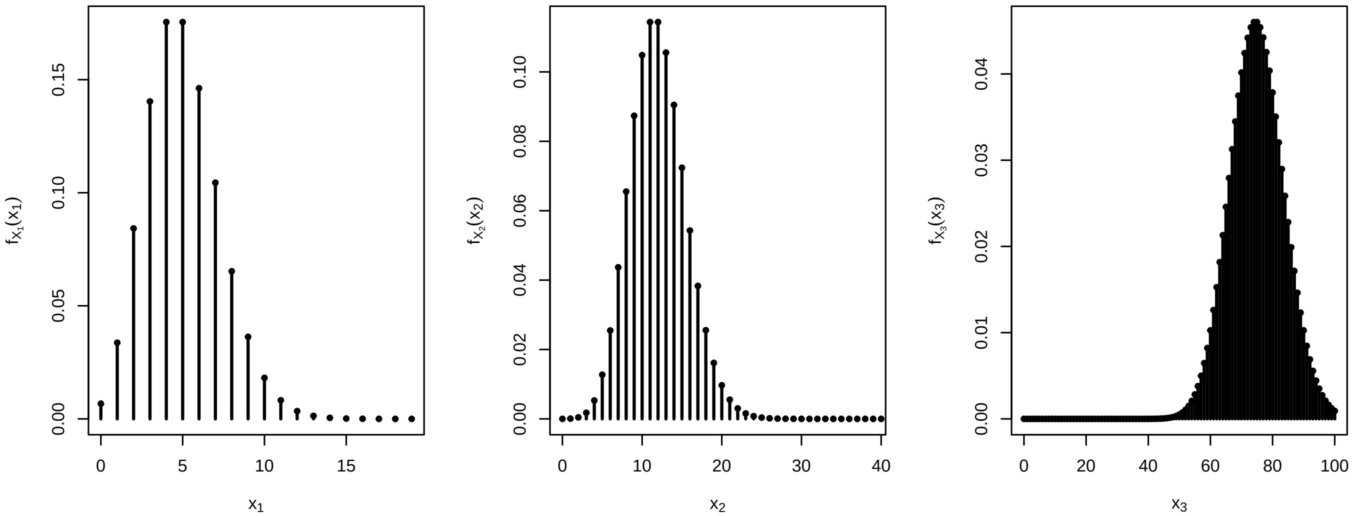
- To find the smallest y such that $P(Y \leq y) \geq c$ use `qhyper(p = c, m = M, n = N-M, k = n)`

```
# Example. Let Y ~ Hyper(N = 10, r = 4, n = 5), find y such that P(Y <= y) = 0.5
qhyper(p = 0.5, m = 4, n = 6, k = 5)
## [1] 2
phyper(q = 2, m = 4, n = 6, k = 5)
## [1] 0.7380952
phyper(q = 1, m = 4, n = 6, k = 5)
## [1] 0.2619048
```

- To simulate m random draws from a Hypergeometric(N, M, n) distribution use `rhyper(n = m, m = M, n = N-M, k = n)`

```
# Example. Let  Y ~ Hyper(N = 10, r = 4, n = 5), generate a RS of size 5
rhyper(nn = 5, m = 4, n = 6, k = 5)
## [1] 2 0 2 3 2
```

Using R to further investigate Poisson(λ) RVs Make your best guess at the parameter value, λ for the following plots.



- To find $P(Y = y)$ use `dpois(x = y, lambda = lambda)`

```
# Example. Y ~ Poisson(lambda = 5), find P(Y = 3)
dpois(x = 3, lambda = 5)
## [1] 0.1403739
```

- To find $P(Y \leq y)$ use `ppois(q = y, lambda = lambda)`

```
# Example. Y ~ Poisson(lambda = 5), find P(Y <= 3)
ppois(q = 3, lambda = 5)
## [1] 0.2650259
sum(dpois(x = 0:3, lambda = 5))
## [1] 0.2650259
```

- To find the smallest y such that $P(Y \leq y) \geq c$ use `qpois(p = c, lambda = lambda)`

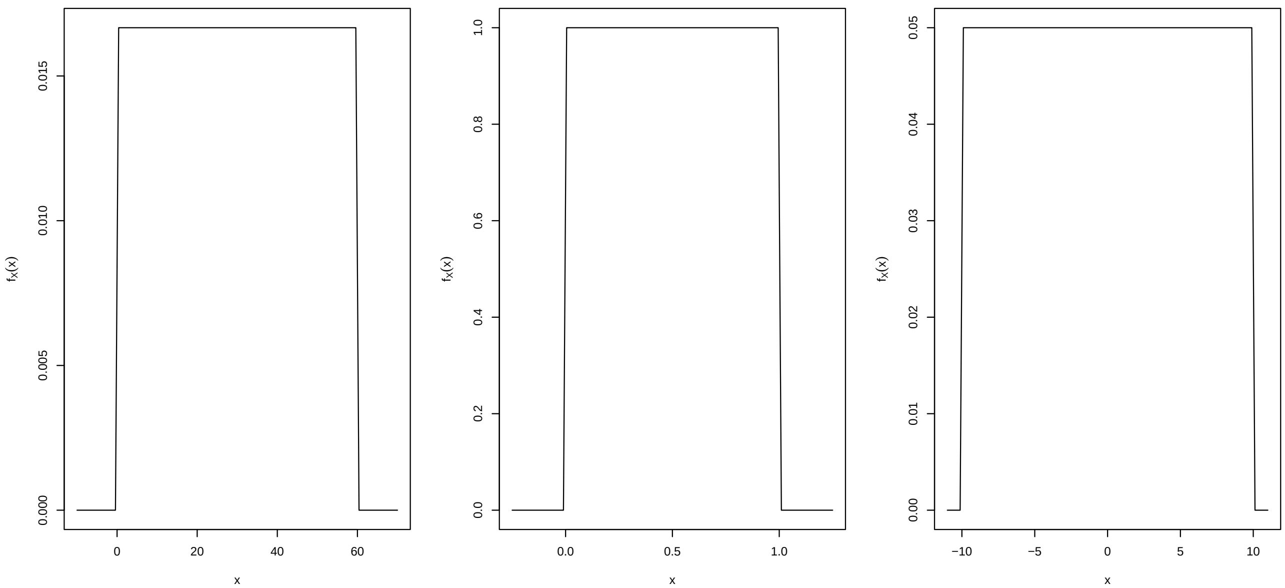
```
# Example. Let Y ~ Poisson(lambda = 5), find y such that P(Y <= y) = 0.5
qpois(p = 0.5, lambda = 5)
## [1] 5
ppois(q = 5, lambda = 5)
## [1] 0.6159607
ppois(q = 4,lambda = 5)
## [1] 0.4404933
```

- To simulate m random draws from a Poisson(λ) distribution use `rpois(n = m, lambda = lambda)`

```
# Example. Let  Y ~ Poisson(lambda = 5), generate a RS of size 5
rpois(n = 5, lambda = 5)
## [1] 7 3 8 4 7
```

Some common continuous distributions

Using R to further investigate the uniform distribution Find the parameters of the following Uniform(θ_1, θ_2) distributions. That is, find θ_1 and θ_2. Come up with an example of a RV that could be modeled by one of the distributions below.



Using R to find values for uniform probabilities is a bit overkill due to the shape of the density, but here's how you do it. Let $Y \sim Uniform(\theta_1, \theta_2)$

- To find the *density* of the pdf (i.e., the height of the pdf) at a particular value for y, use `dunif(x = y, min = theta1, max = theta2)`. What is $P(Y = y)$?

```
# Example. Let Y ~ Uniform(0,1), find density at y = 0.5
dunif(x = 0.5, min = 0, max = 1)
## [1] 1
```

- To find $P(Y \leq y)$ use `punif(q = y, min = theta1, max = theta2)`

```
# Example.  Let Y ~ Uniform(0,1), find P(Y <= 0.5)
punif(q = 0.5, min = 0, max = 1)
## [1] 0.5
```

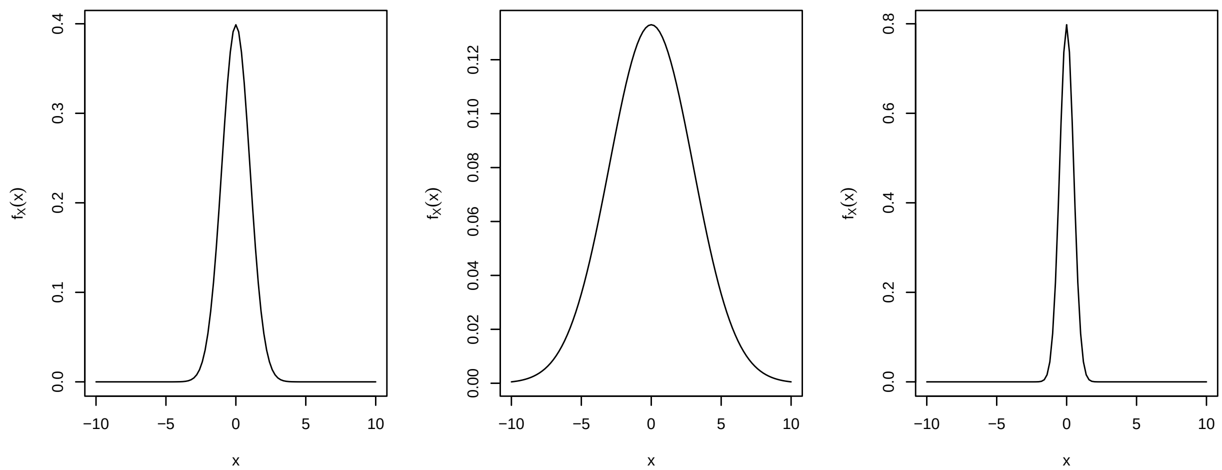
- To find the smallest y such that $P(Y \leq y) \geq c$ use `qunif(p = c, min = theta1, max = theta2)`, or ϕ_c

```
# Example. Let Y ~ Uniform(0,1), y such that P(Y <= y) = 0.5
qunif(p = 0.5, min = 0, max = 1)
## [1] 0.5
```

- To simulate m random draws from a Uniform(θ_1, θ_2) distribution use `runif(n = m, min = theta1, max = theta2)`. This is the most useful function for the uniform distribution!

```
# Example. Let Y ~ Uniform(0,1), generate a RS of size 5
runif(n = 5, min = 0, max = 1)
## [1] 0.52396710 0.11412051 0.05080103 0.55860619 0.73613647
```

The normal distribution



Using R to find normal probabilities can save you from transforming to a standard normal and then using a probability table!

- To find the *density* of the pdf (i.e., the height of the pdf) at a particular value for y, use `dnorm(x = y,  mean = mu, sd = sigma)`. Recall, $P(Y = y) = 0 \; \forall y \in \mathbb{R}$!

```
# Example. Let Y ~ Normal(10,4), find density at y = 5
dnorm(x = 5, mean = 10, sd = 4)
## [1] 0.04566227
```

- To find $P(Y \leq y)$ use `pnorm(q = y, mean = mu, sd = sigma)`

```
# Example. Let Y ~ Normal(10,4), P(Y <=5)
pnorm(q = 5, mean = 10, sd = 4)
## [1] 0.1056498
# convert to z-score
z <- (5 - 10)/4
z
## [1] -1.25
pnorm(q = z, mean = 0, sd = 1)
## [1] 0.1056498
```

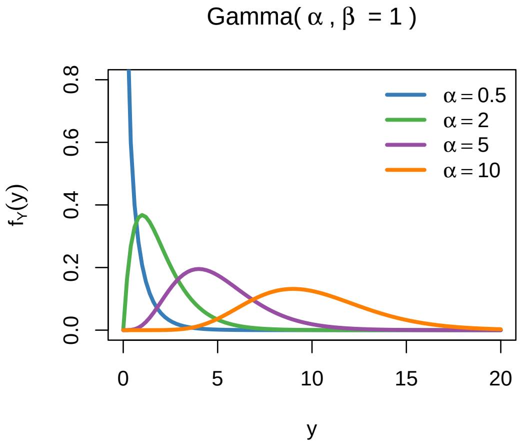
- To find the smallest y such that $P(Y \leq y) \geq c$ use `qnorm(p = c, mean = mu, sd = sigma)`, or ϕ_c

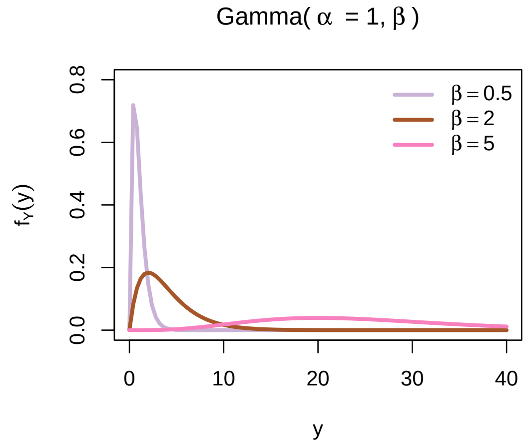
```
# Example. Let Y ~ Normal(10,4), y such that P(Y <= y) = 0.25
qnorm(p = 0.25, mean = 10, sd = 4)
## [1] 7.302041
```

- To simulate m random draws from a Normal(μ, σ) distribution use `rnorm(n = m, mean = mu, sd = sigma)`. This can be incredibly useful for simulation studies!

```
# Example. Let Y ~ Normal(10,4), generate a RS of size 5
rnorm(n = 5, mean = 10, sd = 4)
## [1]  7.975917  7.638915 12.611480 13.469484  6.202663
```

The gamma family of distributions Gamma distributions are incredibly flexible in shape.





Using R to find gamma probabilities The functions `_gamma()`, `_exp()` and `_chisq()` can be used with `d`, `p`, `q`, and `r` to find values of the *pdf*, *CDF*, quantiles, and generate RS from the distributions, respectively. Just like the geometric and negative binomial distributions, the parameterization in R is different than what is in the text. Let $Y \sim Gamma(\alpha, \beta)$.

- To find the *density* of the pdf (i.e., the height of the pdf) at a particular value for y, use `dgamma(x = y,  shape = alpha, rate = 1/beta)`. Recall, $P(Y = y) = 0 \ \forall y \in \mathbb{R}$!

```
# Example. Let Y ~ Gamma(4,5), find density at y = 5
dgamma(x = 5, shape = 4, rate = 1/5)
# for Y ~ Exp(5)
dexp(x = 5, rate = 1/5)
# for Y ~ chisquared(nu = 5)
dchisq(x = 5, df = 5)
```

- To find $P(Y \leq y)$ use `pgamma(q = y, shape = alpha, rate = 1/beta)`

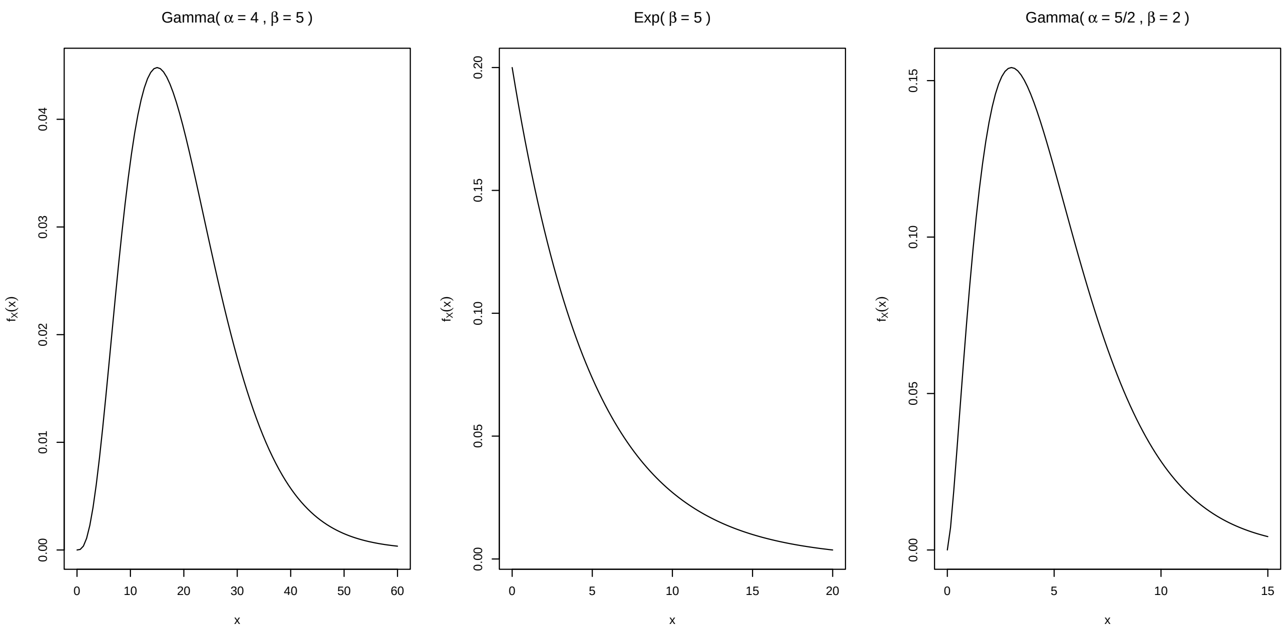
```
# Example. Let Y ~ Gamma(4,5), P(Y <=5)
pgamma(q = 5, shape = 4, rate = 1/5)
# for Y ~ Exp(5)
pexp(q = 5, rate = 1/5)
# for Y ~ chisquared(nu = 5)
pchisq(q = 5, df = 5)
```

- To find the smallest y such that $P(Y \leq y) \geq c$ use `qgamma(p = c, shape = alpha, rate = 1/beta)`, or ϕ_c

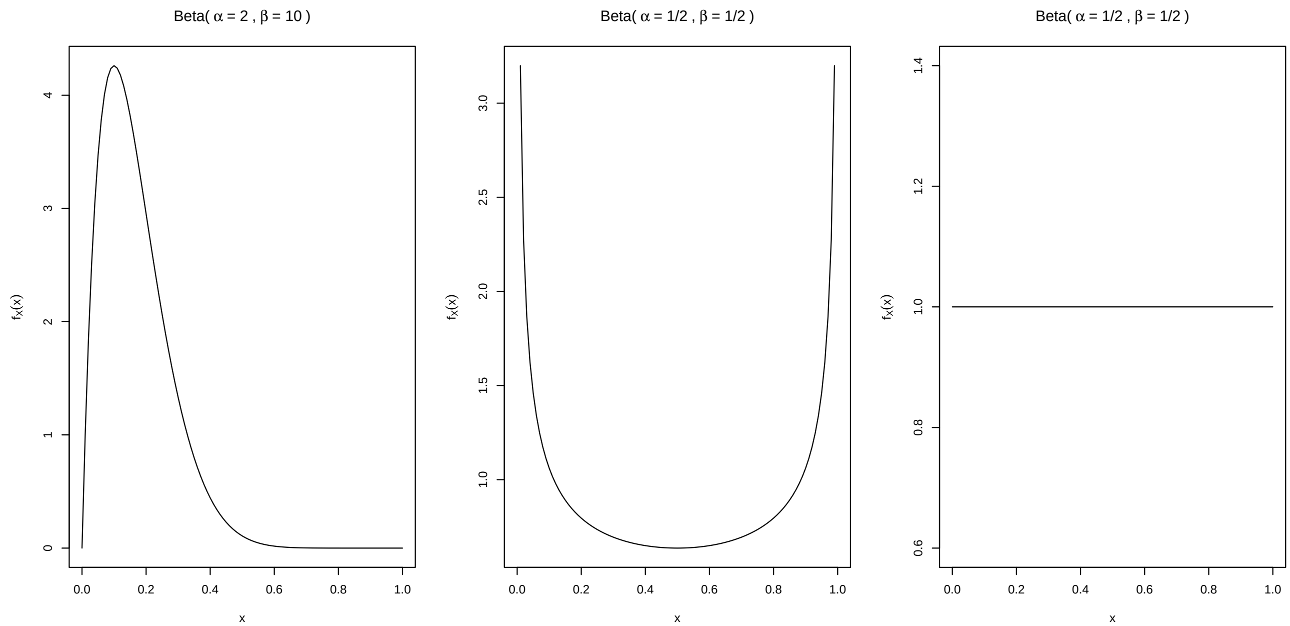
```
# Example. Let Y ~ Gamma(4,5), y such that P(Y <= y) = 0.25
qgamma(p = 0.25, shape = 4, rate = 1/5)
# for Y ~ Exp(5)
qexp(p = 0.25, rate = 1/5)
# for Y ~ chisquared(nu = 5)
qchisq(p = 0.25, df = 5)
```

- To simulate m random draws from a Gamma distribution use `rgamma(n = m, shape = alpha, rate = 1/beta)`.

```
# Example. Let Y ~ Gamma(4,5), generate a RS of size 5
rgamma(n = 5, shape = 4, rate = 1/5)
# for Y ~ Exp(5)
rexp(n = 5, rate = 1/5)
# for Y ~ chisquared(nu = 5)
rchisq(n = 5, df = 5)
```



The beta distribution in R



The functions `_beta` can be used with `d`, `p`, `q`, and `r` to find values of the *pdf*, *CDF*, quantiles, and generate RS from the beta distribution, respectively. Let $Y \sim Beta(\alpha, \beta)$.

- To find the *density* of the pdf (i.e., the height of the pdf) at a particular value for y, use `dbeta(x = y,  shape1 = alpha, shape2 = beta)`. Recall, $P(Y = y) = 0 \; \forall y \in \mathbb{R}$!

```
# Example. Let Y ~ Beta(2,10), find density at y = 0.5
dbeta(x = 0.5, shape1 = 2, shape2 = 10)
## [1] 0.1074219
```

- To find $P(Y \leq y)$ use `pbeta(q = y,  shape1 = alpha, shape2 = beta)`

```
# Example. Let Y ~ Beta(2,10), P(Y <=0.5)
pbeta(q = 0.5, shape1 = 2, shape2 = 10)
## [1] 0.9941406
```

- To find the smallest y such that $P(Y \leq y) \geq c$ use `qbeta(p = c,  shape1 = alpha, shape2 = beta)`, or ϕ_c

```
# Example. Let Y ~ Beta(2,10), y such that P(Y <= y) = 0.25
qbeta(p = 0.25,  shape1 = 2, shape2 = 10)
## [1] 0.08760995
```

- To simulate m random draws from a Gamma distribution use `rbeta(n = m,  shape1 = alpha, shape2 = beta)`.

```
# Example. Let Y ~ Beta(2,10), generate a RS of size 5
rbeta(n = 5,  shape1 = 2, shape2 = 10)
## [1] 0.23733831 0.09517714 0.30426535 0.14042085 0.16918364
```