Foundations of Probability in R
by David Robinson

statistical inference > the process where you have some observed data, and you use it to build an underlying model
probability > the study of how data can be generated from a model
first tool in R is the trusty rbinom
rbinom(1, 1, 0.5) #what we use to simulate a one fair flip of one coin
‘draw’ is a single outcome from a random variable
first argument is the number of random draws
second is the number of coins we are flipping on each draw
third is the probability of a ‘heads’ or a success
the rbinom function has two outcomes here 0 or 1

a probability distribution is a mathematical description of the possible outcomes of a random variable

Finding density with simulation
the fraction of outcomes equal to a value within the binomial distribution is called the density of the binomial at that point
using R:
flips <- rbinom(100000, 10, 0.5)
flips == 5 #compares each item in the vector to 5
mean(flips ==5) #this finds the fractin of comparisons that are TRUE (ie the values that are 1 which represents heads in our example)
output > 0.2463 (approx 24.6%) > meaning that 25% of the time 5 heads hit when the 10 coins are flipped

dbinom allows us to easily calculate the exact probability density
dbinom(5, 10, 0.5)
output > 0.2461 > *near equal to our manual calculation above

Cumulative density
P(X<=4)
manually using R:
flips <- rbinom(100000, 10, 0.5)
mean(flips <= 4)
output > 0.376
we can use the pbinom function in R to calculate this directly
pbinom(4, 10, 0.5)
output > 0.377

Example
# Calculate the probability that 2 are heads using dbinom
dbinom(2, 10, 0.3)

# Confirm your answer with a simulation using rbinom
mean(rbinom(10000, 10, 0.3) == 2)

Example
Calculate the probability that at least five coins are heads
1 - pbinom(4, 10, 0.3)

# Confirm your answer with a simulation of 10,000 trials
mean(rbinom(10000, 10, 0.3) >= 5)

Expected value
ie the mean of the distribution
this puts it right at the center of the distribution if visualized
example
mean(rbinom(100000, 10, .5))
output > 5.00196
mean(ribnom(1000000, 10, .2))
output > 2.001
*general rule > the expected value of a binomial distribution by multiplying the size (ie the number of coins) by the probability each is heads
E[X] = size*p

Variance
is the average squared distance of each value from the mean of the sample
with R
X <- rbinom(100000, 10, .5)
var(X)
output > 2.504
what this tells us > we saw above that the mean for this distribution was 5 > variance tells us that approx 2.5 is the average squared distance between 5 and one random draw
general rule > variance is the size times p times 1 - p
Var(X) = size * p * (1-p)

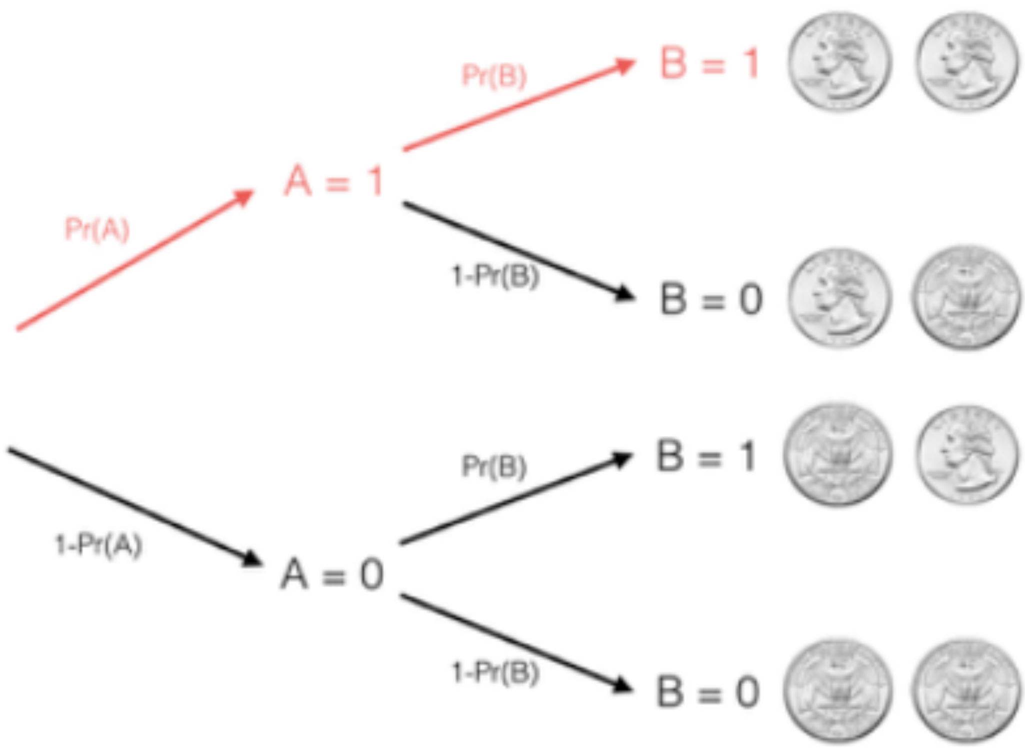
Example
Calculate the expected value using the exact formula
25 * .3

```
# Confirm with a simulation using rbinom
mean(rbinom(10000, 25, .3))

# Calculate the variance using the exact formula
25 * .3 * (1-.3)

# Confirm with a simulation using rbinom
var(rbinom(10000, 25, .3))
```

Probability of event A and event B
we have determined an event can be heads (1 or True) or tails (0 or False)
we want to know the probability that event A and event B are both heads
ie we want to know the probability that both flips end up tails
Visualizing the probability of A and B



*we represent our desired outcome by multiplying the probabilities
1/2 (prob of event A) * 1/2 (prob of event B) = 1/4

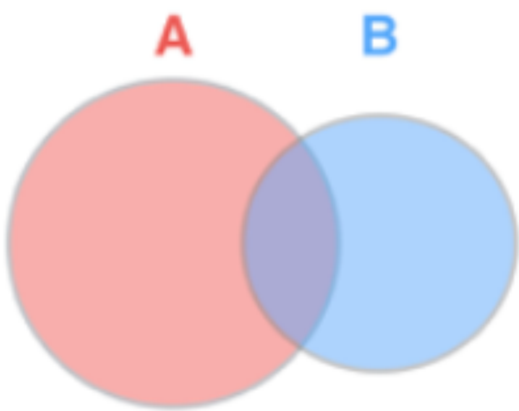
with R:

```
A <- rbinom(100000, 1, .5)
B <- rbinom(100000, 1, .5)
# the '&' allows you to compare these two variables
# *this will compare each corresponding flip in A and B, and result in true if and only if both A and B are true (heads for our example)
```

A & B
# then take the mean
mean(A & B)
ouput > 0.24959 (darn close to .5 * .5 or the prob of A * prob of B)

denoted as Pr(A and B) = Pr(A) * Pr(B)

Probability of A or B
Pr(A or B) = Pr(A) + Pr(B) - Pr(A and B)
also denoted as
Pr(A or B) = Pr(A) + Pr(B) - Pr(A) * Pr(B)
visualizing what this looks like



ie the prob of A plus the prob of B minus the overlap of both A and B

example if events are independent ie probability of 0.5 (50%)
Pr(A or B) = .5 + .5 - .5 * .5 = .75
with R we would create our random events as above (A and B)
instead of the & operator we use the or operator (|)
mean(A | B)
output > 0.7513
similar to our manual calculation

the advantage to simulation and R
can see how cumbersome the manual calculation can become >
Pr(A or B or C) = Pr(A) + Pr(B) + Pr(C) - Pr(A and B) - Pr(A and C) - Pr(A and B) - Pr(A and B and C)

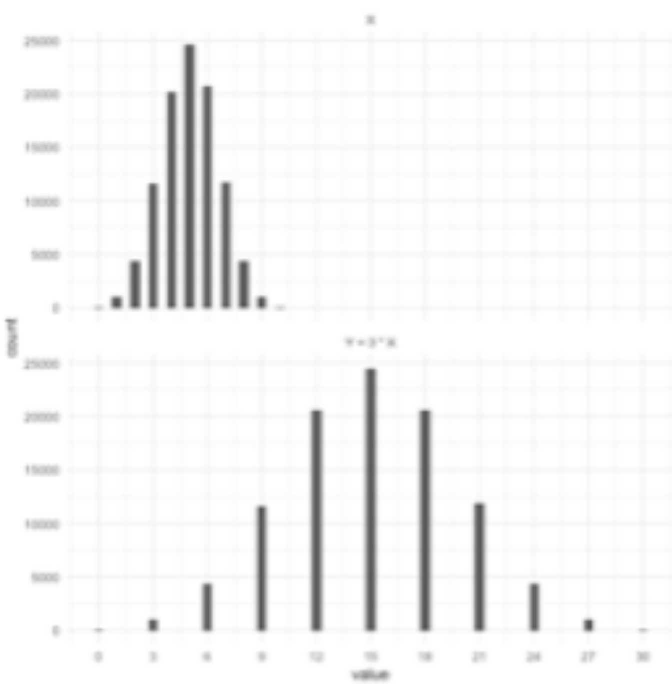
Example
# Use rbinom to simulate 100,000 draws from each of X and Y
X <- rbinom(100000, 10, .6)
Y <- rbinom(100000, 10, .7)

```
# Estimate the probability either X or Y is <= to 4
mean(X <= 4 | Y <= 4)

# Use pbinom to calculate the probabilities separately
prob_X_less <- pbinom(4, 10, .6)
prob_Y_less <- pbinom(4, 10, .7)

# Combine these to calculate the exact probability either <= 4
prob_X_less + prob_Y_less - prob_X_less * prob_Y_less
```

Multiplying random variables
imagine random variables as algebraic symbols
we have x that flips a coin 5x with a fair probability
we can then multiply this random variable
say by 3 and name it y
so x is 5 draws and y is 15 draws
we can then visualize both of these variables



*shape is the same but y is both larger and more spread out
*so we would expect both the expected value and the variance to increase

Let’s look at this in more detail
we have >

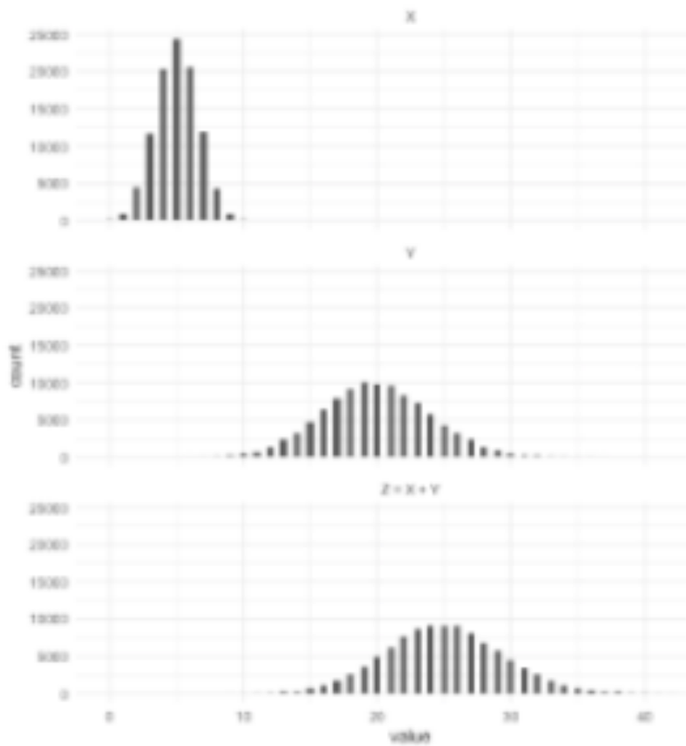
```
X <- rbinom(100000, 10, .5)
mean(X)
```

output > approx 5
#we want Y to be three times this
Y <- 3 * X
#this literally multiplies every individual value by 3
mean(Y)
output > approx 15
*key point is that by doing this we are also multiplying the expected value by 3
we can see this in our visualization that X and Y are the same shape, but Y is 3x bigger
general rule > E[k*X] = k*E[X]

For variance
var(X)
output > approx 2.5
var(Y)
output > approx 22.5
*an increase by a factor of 9
why 9 > that is 3 squared and variance is the average squared distance of values from the mean
Var[k*X] = k^2*Var[X]

**these properties hold true no matter what the distribution the random variable follows

Adding two random variables together
X + Y = Z



the bottom is both larger and more spread out

Expected Value General Rule
E[X+Y] = E[X] + E[Y]
expected value of X + Y is the expected value of X plus the expected value of Y
*works even if X and Y aren’t independent

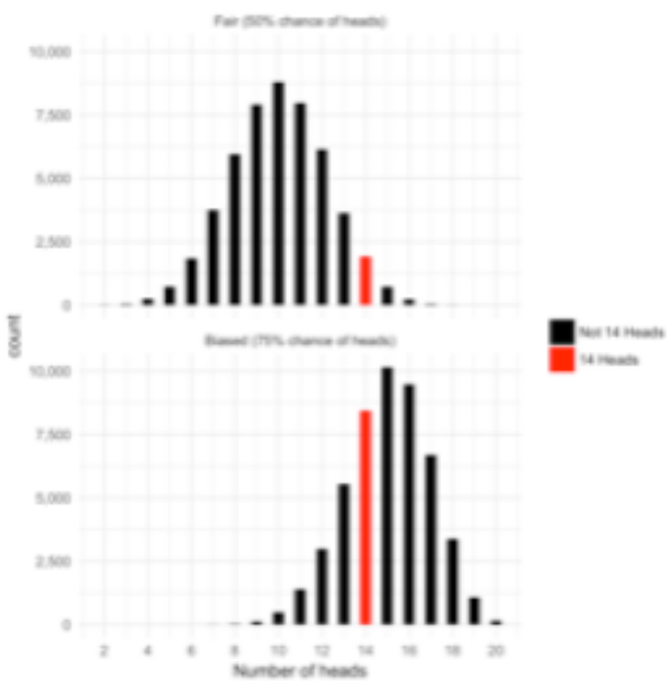
Variance General Rule
Var[X+Y] = Var[X] + Var[Y]
the variance of the sum of two independent random variables is the sum of their variances
*only works if X and Y are independent

Example
```
# Simulation from last exercise of 100,000 draws from X and Y
X <- rbinom(100000, 20, .3)
Y <- rbinom(100000, 40, .1)

# Find the variance of X + Y
var(X+Y)

# Find the variance of 3 * X + Y
var(3*X+Y)
```

Updating with evidence
the process of updating our beliefs after seeing evidence lies at the heart of Bayesian statistics
example two piles of 50,000 coins one with a fair coin (.5%) and one pile with a biased coin (.75%)
fair <- rbinom(50000, 20, .5)
#we want to know the chances of getting 14 heads out of 20
sum(fair == 14)
output > 1888
biased <- rbinom(50000, 20, .75)
sum(biased == 14)
output > 8372
a visual of expected histograms



#we now add up the 2 red bars
1888 + 8372 > 10260
Now we can get the conditional probability
ie the probability the coin is biased given the condition that we got 14 heads
Pr(Biased | 14 Heads) =
here '|' means 'given'
#biased w/14 Heads / #total w/14 Heads = 8372 / 1888 + 8372 = 82%
now we can say there is an 82% chance the coin is biased
a large difference from our original thought of 50%

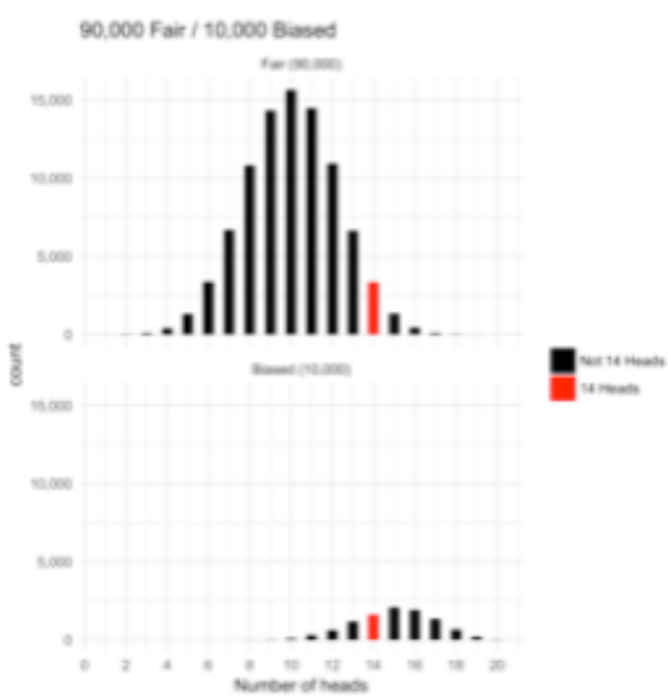
Example

```
# Simulate 50000 cases of flipping 20 coins from fair and from biased
fair <- rbinom(50000, 20, .5)
biased <- rbinom(50000, 20, .75)

# How many fair cases, and how many biased, led to exactly 11 heads?
fair_11 <- sum(fair == 11)
biased_11 <- sum(biased == 11)

# Find the fraction of fair coins that are 11 out of all coins that were 11
fair_11 / (fair_11 + biased_11)
```

‘Prior probability’ is an important part of Bayesian statistics
this is judgment or belief prior to the experiment
example same as above
but instead now we believe that the chance of getting a biased coin isn’t 50/50 its 10/90
this assessment is made off of elements of the environment (ie in this example we trust the person more who is on the other side of the bet)
how does this change the experiment?
now instead of having two equal piles of 50k coins each
we now have one pile with 90k coins where we believe the probability is .5
and the second pile has 10k coins where we believe the probability is .75
let’s visualize this



simulate this with R

fair = 3410
biased = 1706
next we find our conditional probability >
1706 / 1706 + 3410 = .333
*this is interesting > we thought there was a 10% chance the coin was biased but given the chances of 14 heads out of 20 on the fair coin we need to update our probability and expectation to 33%

Example

```
# Simulate 8000 cases of flipping a fair coin, and 2000 of a biased coin
fair_flips <- rbinom(8000, 20, .5)
biased_flips <- rbinom(2000, 20, .75)

# Find the number of cases from each coin that resulted in 14/20
fair_14 <- sum(fair_flips == 14)
biased_14 <- sum(biased_flips == 14)

# Use these to estimate the posterior probability
fair_14 / (fair_14 + biased_14)

# Simulate 80,000 draws from fair coin, 10,000 from each of high and low coins
flips_fair <- rbinom(80000, 20, .5)
flips_high <- rbinom(10000, 20, .75)
flips_low <- rbinom(10000, 20, .25)

# Compute the number of coins that resulted in 14 heads from each of these piles
fair_14 <- sum(flips_fair == 14)
high_14 <- sum(flips_high == 14)
low_14 <- sum(flips_low == 14)

# Compute the posterior probability that the coin was fair
fair_14 / (fair_14 + high_14 + low_14)
```

Bayes' Theorem
what we are really looking at in our simulations so far is probability densities
we look at this more directly in R with dbinom
example
instead of simulating with rbinom for a count of true values as such

```
fair <- rbinom(90000, 20, .5)
sum(fair == 14)
```

output > 3140
we calculate for the probability where .9 will represent the proportion we are seeking out

dbinom(14, 20, .5) * .9
output > 0.033
denoted as Pr(14 Heads|Fair) * Pr(Fair)
#finish the rest of the proportion
biased <- rbinom(10000, 20, .75)
sum(biased == 14)
output > 1706
dbinom(14, 20, .75) * .1
output > 0.016
denoted as Pr(14 Heads|Biased) * Pr(Biased)

This is what we are talking about

Conditional probability

$$\Pr(\text{Biased}|\text{14 Heads}) = \frac{\Pr(\text{14 Heads and Biased})}{\Pr(\text{14 Heads and Biased}) + \Pr(\text{14 Heads and Fair})}$$

$$= \frac{\Pr(\text{14 Heads}|\text{Biased})\Pr(\text{Biased})}{\Pr(\text{14 Heads}|\text{Biased})\Pr(\text{Biased}) + \Pr(\text{14 Heads}|\text{Fair})\Pr(\text{Fair})}$$

```
prob_14_fair <- dbinom(14, 20, .5) * .9
prob_14_biased <- dbinom(14, 20, .75) * .1

prob_14_biased / (prob_14_fair + prob_14_biased)
```

Above helps show what the numerator and denominator really represent in Bayes' Theorem
by imagining what fraction of all coins resulting in 14 heards were biased
Bayes >
Pr(A|B) = Pr(B|A)Pr(A) / Pr(B|A)Pr(A) + Pr(B|not A)Pr(not A)
*for our example A = Biased and B = 14 Heads
what this all means? >
finding the probability of event A given event B when you knew the probability of event B given event A
*the key point (and the trickery, also Bayes beauty) is we knew the probability of getting 14 heads given that the coin is biased, but we needed to convert it to the probability that a coin is biased given that it resulted in 14 heads

Example
# Use dbinom to calculate the probability of 11/20 heads with fair or biased coin
probability_fair <- dbinom(11, 20, .5)

probability_biased <- dbinom(11, 20, 0.75)

Calculate the posterior probability that the coin is fair
probability_fair / (probability_fair + probability_biased)

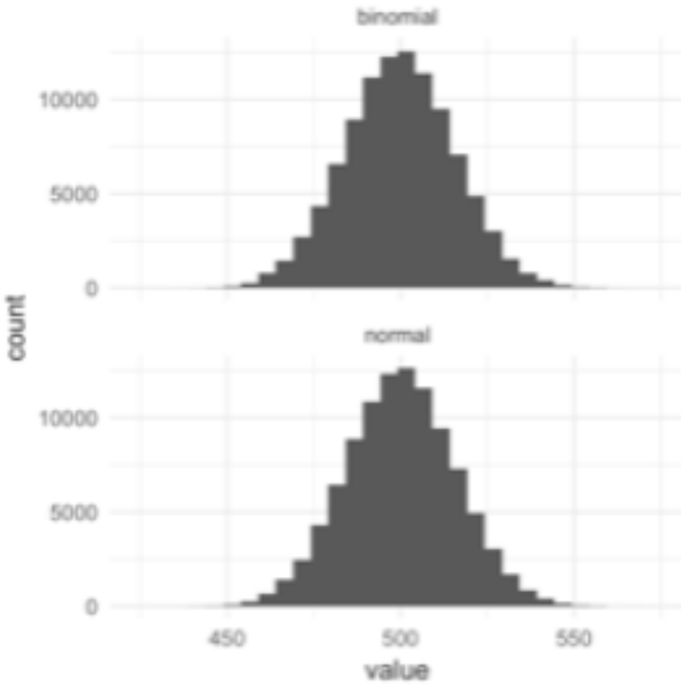
# Find the probability that a coin resulting in 14/20 is fair
fair_prob_14 <- dbinom(14, 20, .5)
biased_prob_14 <- dbinom(14, 20, .75)
fair_prob_14 / (fair_prob_14 + biased_prob_14)

# Find the probability that a coin resulting in 18/20 is fair
fair_prob_18 <- dbinom(18, 20, .5)
biased_prob_18 <- dbinom(18, 20, .75)
fair_prob_18 / (fair_prob_18 + biased_prob_18)

Example
# Use dbinom to find the probability of 16/20 from a fair or biased coin
probability_16_fair <- dbinom(16, 20, .5)
probability_16_biased <- dbinom(16, 20, .75)

Use Bayes' theorem to find the posterior probability that the coin is fair
(probability_16_fair * 0.99) / ((probability_16_fair * 0.99) + (probability_16_biased * 0.01))

The normal distribution
normal approximation to the binomial
mean or mew or expected value = size * p
variance = size * p * (1 - p)
std = sqrt(variance)
for normal distribution we use rnorm(#of draws, expected value, std)
*normal distribution is a good approximation to the binomial
to visualize this >



Example
# Draw a random sample of 100,000 from the Binomial(1000, .2) distribution
binom_sample <- rbinom(100000, 1000, .2)

# Draw a random sample of 100,000 from the normal approximation
normal_sample <- rnorm(100000, 200, sqrt(160))

# Compare the two distributions with the compare_histograms function
compare_histograms(binom_sample, normal_sample)

# Simulations from the normal and binomial distributions
binom_sample <- rbinom(100000, 1000, .2)
normal_sample <- rnorm(100000, 200, sqrt(160))

# Use binom_sample to estimate the probability of <= 190 heads
mean(binom_sample <= 190)

# Use normal_sample to estimate the probability of <= 190 heads
mean(normal_sample <= 190)

# Calculate the probability of <= 190 heads with pbinom
pbinom(190, 1000, .2)

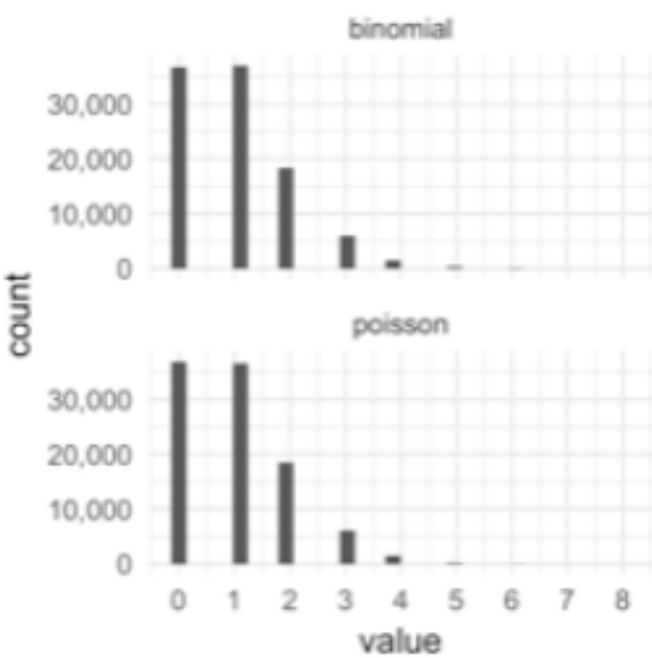
# Calculate the probability of <= 190 heads with pnorm
pnorm(190, 200, sqrt(160))

Example
# Draw a random sample of 100,000 from the Binomial(10, .2) distribution
binom_sample <- rbinom(100000, 10, .2)

# Draw a random sample of 100,000 from the normal approximation
normal_sample <- rnorm(100000, 2, sqrt(1.6))

# Compare the two distributions with the compare_histograms function
compare_histograms(binom_sample, normal_sample)

The Poisson distribution
defines where n is large and p is small
R example, flipping a coin where the probability of heads is only one in a thousand
rbinom(100000, 1000, 1/1000)
Poisson distribution is described only by one parameter > the mean
Poisson's mean is often referred to as lambda
E[X] = lambda
simulating Poisson in R
rpois(100000, 1,)
compare these two created distributions
compare_histograms(binomial, poisson)
we get >



*interestingly what this shows us is that for the Poisson distribution the variance is equal to the mean
Poisson can have any mean as long as its positive
Poisson is used when modeling rare events as counts
and when we dont care about the total in the way we would with the binomial distribution
real world examples >

- counting the number of people that walk into a bookstore over the course of an hour
- or how many cells seen under a microscope
- how may whales spotted in a section of ocean

*we don't so much care about how many whales there are in the world, what we care about is how many we see in that section of the ocean

Example
# Draw a random sample of 100,000 from the Binomial(1000, .002) distribution
binom_sample <- rbinom(100000, 1000, .002)

# Draw a random sample of 100,000 from the Poisson approximation
poisson_sample <- rpois(100000, 2)

# Compare the two distributions with the compare_histograms function
compare_histograms(binom_sample, poisson_sample)

# Simulate 100,000 draws from Poisson(2)
poisson_sample <- rpois(100000, 2)

# Find the percentage of simulated values that are 0
mean(poisson_sample <= 0)

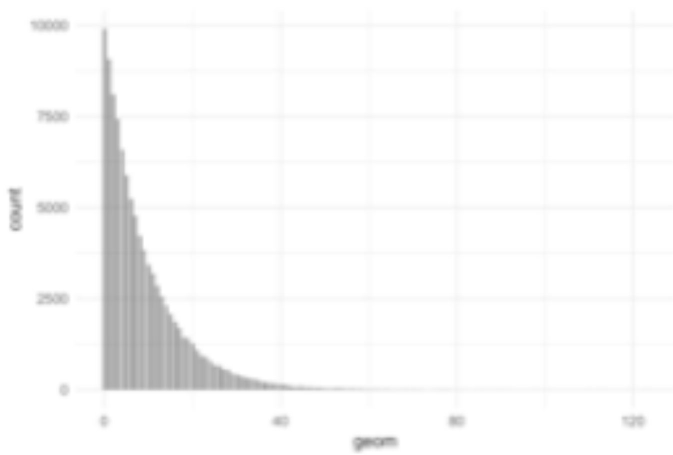
# Use dpois to find the exact probability that a draw is 0
dpois(0, 2)

Adding Poisson distributions to themselves results in a Poisson distribution

The geometric distribution
represents a random variable where you are waiting for particular event with some probability
idea > I have a coin that lands on heads 10% of the time > What can I expect here?
simulating this
flips <- rbinom(100, 1, .1)
#we can use the 'which' function which returns the indices that fit a particular condition

which(flips ==1)
#how do we get the first heads? which is the answer to our first question
which(flips ==1)[1]

We want to test this out multiple times
replicate(10, which(rbinom(100, 1, .1) == 1)[1])
this gives us the geometric distribution
we can also create one more directly >
geom <- rgeom(100000, .1)
what it looks like >



the most likely value for our example is 0
for our example what is the expected value or the mean
mean(geom)
output > approx 9
E[X] = 1/p - 1
*the minus 1 comes from the fact that R defines the geometric distribution as the number of tails before the first heads (or success)
real world example >

- give factories an idea of when they might need to repair a machine

Example
Existing code for finding the first instance of heads
which(rbinom(100, 1, 0.2) == 1)[1]

# Replicate this 100,000 times using replicate()
replications <- replicate(100000, which(rbinom(100, 1, 0.2) ==1)[1])

# Histogram the replications with qplot
qplot(replications)

Example

```
# Find the probability the machine breaks on 5th day or earlier
pgeom(4, .1)

# Find the probability the machine is still working on 20th day
1 - pgeom(19, .1)

# Calculate the probability of machine working on day 1-30
still_working <- 1 - pgeom(0:29, .1)

# Plot the probability for days 1 to 30
qplot(1:30, still_working)
```