COMP 5970/6970-004
Computational Biology: Genomics and Transcriptomics
Lecture notes 4: 1/25/2022

Haynes Heaton

Spring, 2022

Lecture Objectives

- Review distributions
- Normal distribution
    - Distribution of the estimate of the mean
- Beta distribution
- Compound distributions

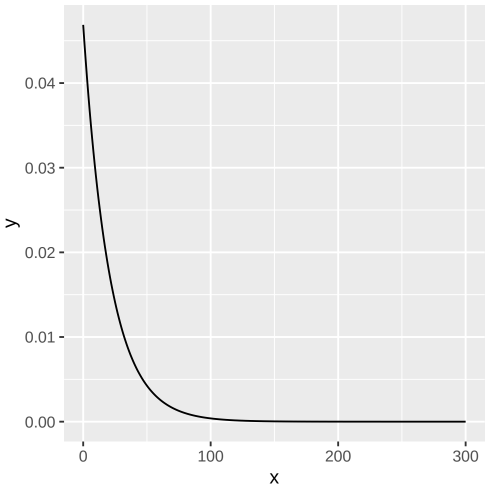
Review

There is an old adage that bad things happen in threes. Of course this is not true, but maybe the nature of random intervals versus what we expect as humans to be random can explain why people thought this.

The number of discrete time points between events is geometrically distributed. Let's plot the probability density function once again.

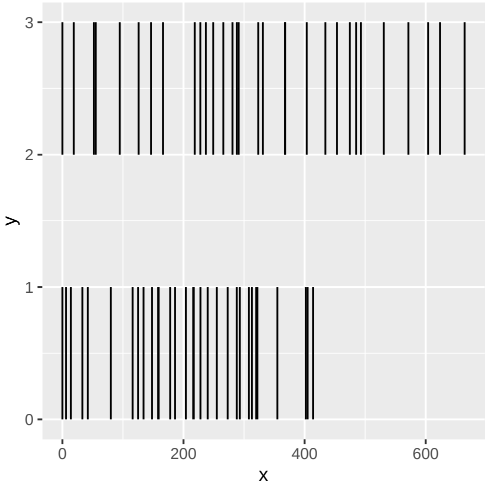
```
if(!require(ggplot2)) {
    install.packages("ggplot2", repos = "http://cran.us.r-project.org")
    library(ggplot2)
}

## Loading required package:  ggplot2
```

```
x = seq(0,300)
p = 3/64
ggplot(data.frame(x=x,y=(1-p)^x * p))+geom_line(aes(x=x,y=y))
```



So smaller intervals are more common than any given larger interval. This makes it fairly likely that we have clumps of short intervals by random chance. We can compare intervals sampled from the geometric distribution to intervals with the same mean length sampled from a normal distribution.

```
p = 3/64
x = c(0)
for (i in 1:30) {
  x = c(x, x[length(x)] + rgeom(1,3/64))
}
y = c(0)
for (i in 1:30) {
  y = c(y, y[length(y)]+rnorm(1,64/3,14))
}

ggplot()+geom_segment(data=data.frame(x=x), aes(x=x,xend=x,y=0,yend=1))+
  geom_segment(data=data.frame(x=y),aes(x=x,xend=x,y=2,yend=3))
```



So you can see that the lower set of intervals looks much clumpier than the upper set. So even if events—"bad things" in the adage—may seem to occur in 3's, or simply close together more often than what humans would intuitively think. But this can occur frequently just due to the nature of geometrically distributed intervals.

Narrowing down the distribution that describes your data

It can be confusing determining which distribution correctly models the random variable of interest from your data. I propose a series of steps that will aid you in this process. First identify and describe the output or observed variable, also known as the **random variable**. For the distributions we have discussed

1. The length, time, or number of negative events before a positive event
2. The number of positive events in a fixed number of trials
3. The number of positive events in a unit time or space given a continuous rate

First try to answer which distributions models each of these. After doing this, if the distribution is still not clear, try answering the two following questions.

1. Is the observed/random variable continuous or discrete?
2. What values can this random variable take on (ie. what is its support). Can it be negative? Can it be infinite or finite?

Then think about the distributions you know and if they are continuous or discrete and what values they can produce. This should dramatically reduce the possibilities of which distribution creates your data.

Normal distribution/Gaussian distribution

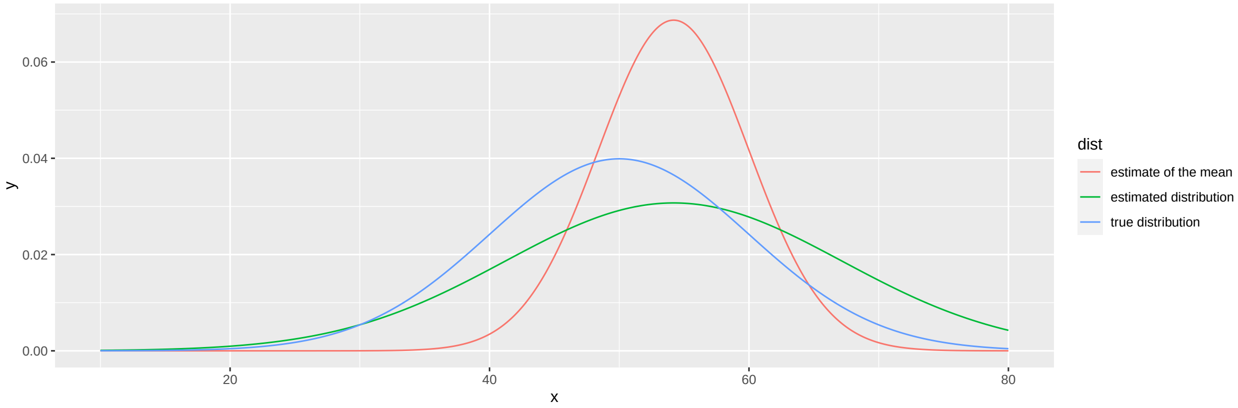
I don't have much to say about the Normal distribution as it will be covered in every other class covering distributions. I view it as the distribution of last resort to be used when the process you are measuring is complex and is the sum of many different random events. In fact, the **central limit theorem** states that the mean of many independent random variables tends toward a normal distribution. The normal distribution is parameterized by a mean μ and a standard deviation σ or a variance σ^2 and the probability density function is

$$\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{1}$$

which I do not expect you to memorize. Estimating the parameters of a normal distribution given a sample is easy. The mean is simply the average of the sample values and the variance is $\hat{\sigma}^2 = \frac{(x_i-\bar{x})^2}{n-1}$ and of course the standard deviation is the square root of that. When estimating the mean, we may also be interested in how accurate our estimate is. Our estimate naturally gets more accurate as the number of samples n increases. The estimate of the mean of a normal distribution is itself normally distributed with the center at the estimate of the mean and a variance of $\frac{\hat{\sigma}^2}{n}$ or a standard deviation of $\frac{\hat{\sigma}}{\sqrt{n}}$. So the standard deviation of the estimate of the mean goes down with the square root of the sample size. This is important to know how to create error bars. This distribution describes the probability density of the location of the true mean given the sample data.

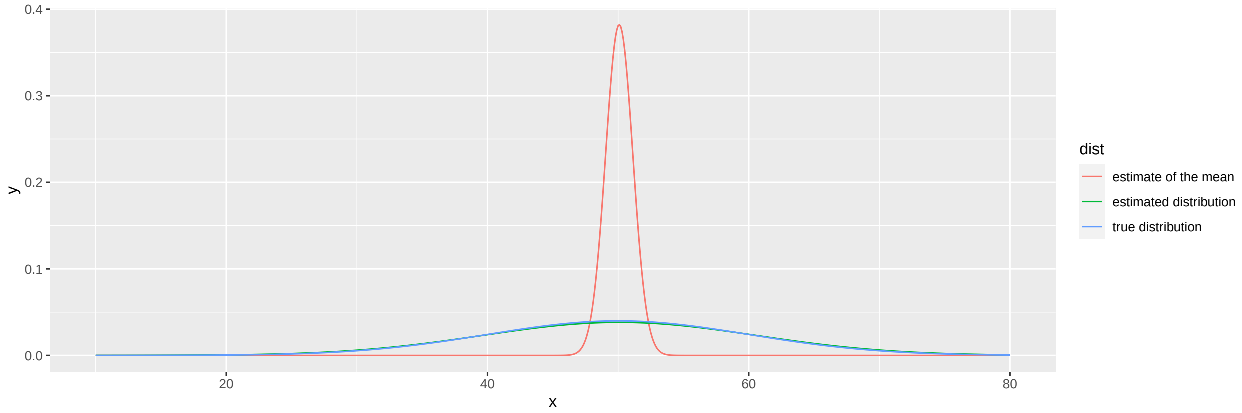
```
mu = 50
sd = 10
x = rnorm(5, mu, sd)
n = length(x)
estimate_of_mean = mean(x)
print(estimate_of_mean)

## [1] 54.19319

estimate_of_sd = sd(x) #sqrt(sum((x[i]-mean(x))^2)/(n-1))

sd_of_estimate_of_mean = estimate_of_sd/sqrt(n)
```

```
vals = seq(10,80,by=.1)
true_density = dnorm(vals, mu, sd)
estimate_of_dist = dnorm(vals, estimate_of_mean, estimate_of_sd)
estimate_of_mean_dist = dnorm(vals, estimate_of_mean, sd_of_estimate_of_mean)


data = data.frame(x=vals, y=true_density, dist="true distribution")
data = rbind(data, data.frame(x=vals, y=estimate_of_dist, dist="estimated distribution"))
data = rbind(data, data.frame(x=vals, y=estimate_of_mean_dist, dist="estimate of the mean"))
ggplot(data)+geom_line(aes(x=x,y=y,color=dist))
```



And then we can compare that to having more samples.

```
mu = 50
sd = 10
x = rnorm(100, mu, sd)
n = length(x)
estimate_of_mean = mean(x)
print(estimate_of_mean)

## [1] 50.0845

estimate_of_sd = sd(x) #sqrt(sum((x[i]-mean(x))^2)/(n-1))

sd_of_estimate_of_mean = estimate_of_sd/sqrt(n)

vals = seq(10,80,by=.1)
true_density = dnorm(vals, mu, sd)
estimate_of_dist = dnorm(vals, estimate_of_mean, estimate_of_sd)
estimate_of_mean_dist = dnorm(vals, estimate_of_mean, sd_of_estimate_of_mean)


data = data.frame(x=vals, y=true_density, dist="true distribution")
data = rbind(data, data.frame(x=vals, y=estimate_of_dist, dist="estimated distribution"))
data = rbind(data, data.frame(x=vals, y=estimate_of_mean_dist, dist="estimate of the mean"))
ggplot(data)+geom_line(aes(x=x,y=y,color=dist))
```

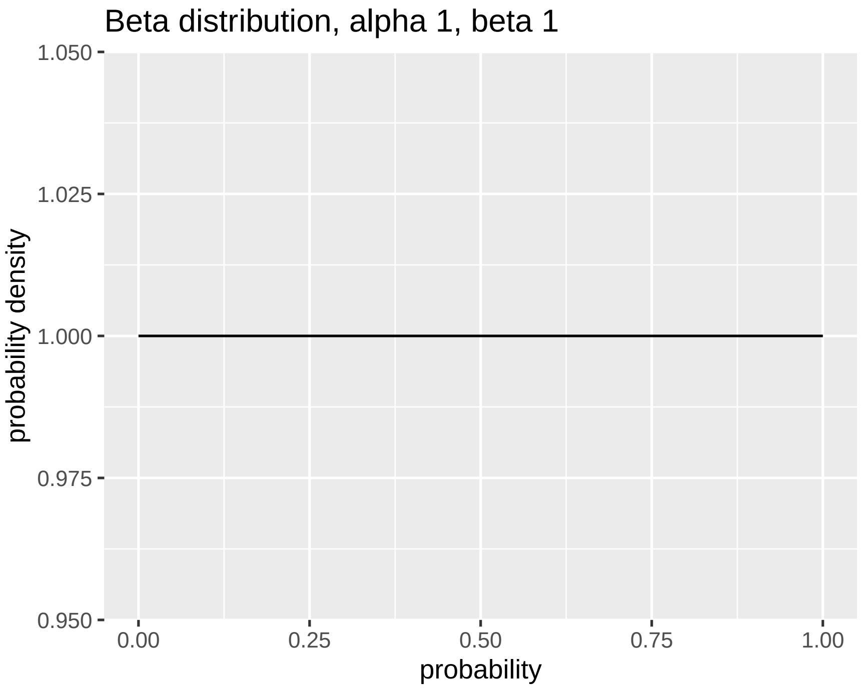


As you can see, with more samples, the estimate of the distribution more closely resembles the true distribution and the distribution of the estimate of the mean becomes much tighter as we are more confident in the location of the true mean.

Beta distribution

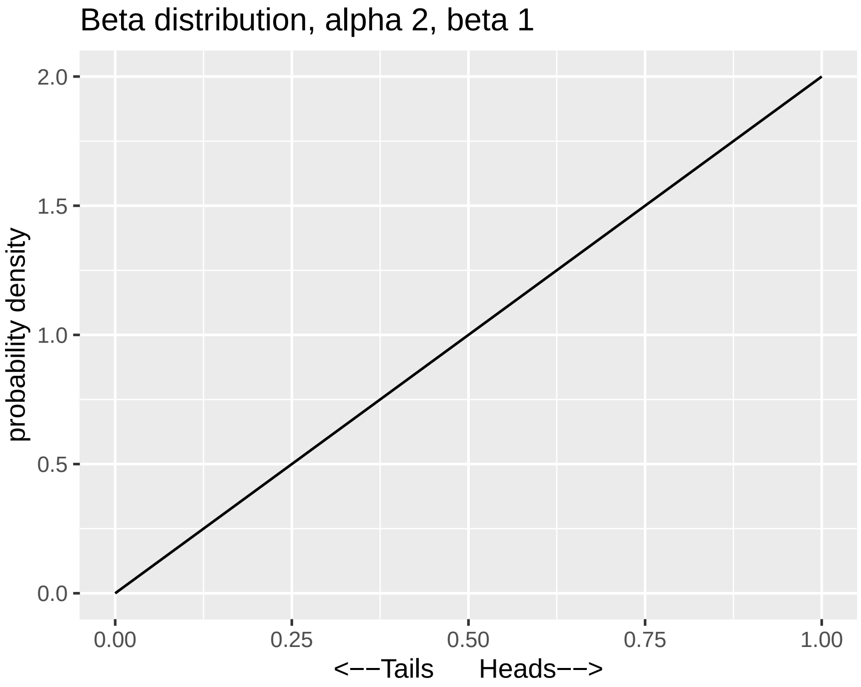
Of course there are many distributions—some of which could be argued to be more important than the ones we have covered—but I wanted to cover simple distributions which arise from simple but common random processes. The Beta distribution is not in this class, but is a beautiful, intuitive, and useful distribution that leads us into the idea of compound distributions AKA statistical models. The Beta distribution is a distribution over probabilities. Consider our first example of the Bayes theorem. We had two possible hypotheses—one in which the rate of GC's were a background rate and one in which they were significantly elevated. But what if we have no hypothesis? We don't know the potential underlying probabilities, we just see data. It is easiest to explain this with coin flips and I apologize for not giving it a more practical example. It is just easier to think about with the simplest possible example. So before we flip a coin, we do not know what percentage heads or tails it will produce, so we give it a flat prior. The Beta distribution is parameterized with α and β. These can thought of as counts of heads and tails and a uniform prior starts with a psuedocount of 1 for each.

```
x = seq(0,1,by=0.001)
alpha = 1
beta = 1
y = dbeta(x, alpha, beta)
ggplot(data.frame(x=x,y=y))+
  geom_line(aes(x=x,y=y))+ggtitle("Beta distribution, alpha 1, beta 1")+
  ylab("probability density")+xlab("probability")
```



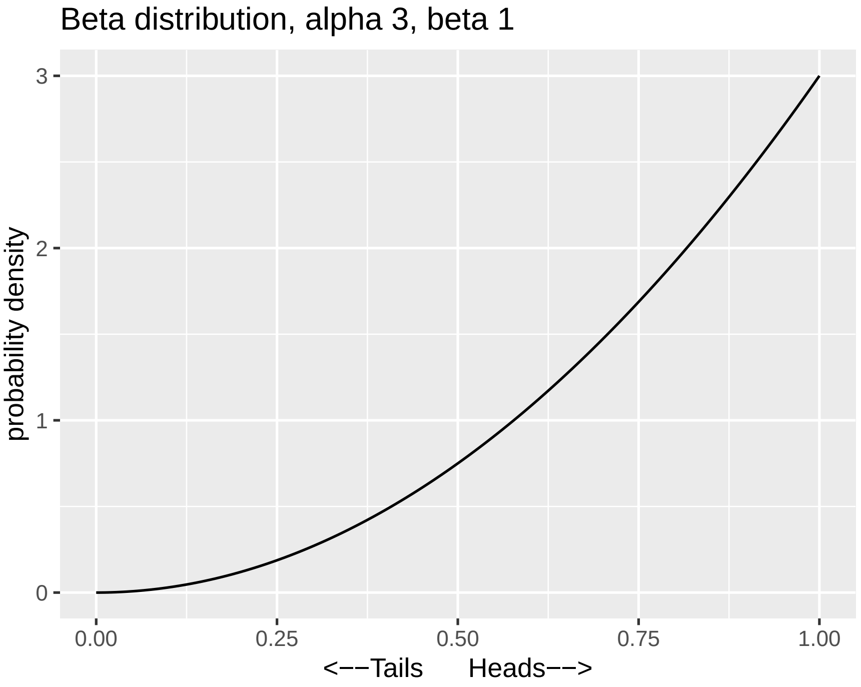
Now lets flip the coin for the first data point. We get heads.

```
alpha = 2
beta = 1
y = dbeta(x, alpha, beta)
ggplot(data.frame(x=x,y=y))+
  geom_line(aes(x=x,y=y))+ggtitle("Beta distribution, alpha 2, beta 1")+
  ylab("probability density")+xlab("<--Tails      Heads-->")
```



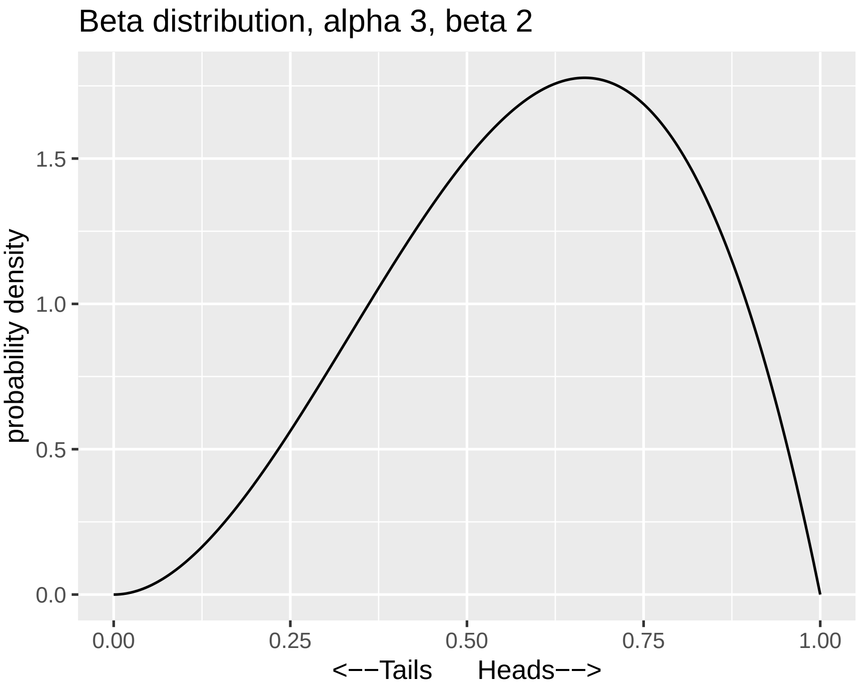
This is both strange but also intuitive. We saw heads, so we now know that there is 0 probability density that this coin produces 100% tails. Other than that, we don't know anything, so we have a linear relationship between 0 tails and 100% heads. We flip the coin again, and get another heads.

```r
alpha = 3
beta = 1
y = dbeta(x, alpha, beta)
ggplot(data.frame(x=x,y=y))+
  geom_line(aes(x=x,y=y))+ggtitle("Beta distribution, alpha 3, beta 1")+
  ylab("probability density")+xlab("<--Tails      Heads-->")
```



With another heads, we are more likely to believe this coin is weighted towards heads. Though we will have plenty of possibility in the form of probability density that it lies somewhere in between. The next coin flip is tails...

```r
alpha = 3
beta = 2
y = dbeta(x, alpha, beta)
ggplot(data.frame(x=x,y=y))+
  geom_line(aes(x=x,y=y))+ggtitle("Beta distribution, alpha 3, beta 2")+
  ylab("probability density")+xlab("<--Tails      Heads-->")
```



Now that we have seen tails, we see a dramatic change to our distribution over probabilities of this coin. We now know for certain that it is not 100% heads. As you can see, this will eventually tighten in on the correct underlying probability while being completely hypothesis free.

Compound distributions AKA Statistical Models

We now see that if there are distributions over probabilities, a Beta distribution may model the probability parameter of another distribution such as a *Bernoulli*, binomial, or geometric distribution. A Poisson distribution may model the n parameter for a binomial distribution. We can also have data coming from multiple different distributions. What I mean by this is that any given data point may come from distribution 1 or distribution 2 etc. This is known as a mixture model. We can use the marginal likelihood to model this mixture.

$$p(D|dist_1)p(dist_1) + p(D|dist_2)p(dist_2) \tag{2}$$

And even if we don't know the parameters of these distributions, if we know the types of distributions, we can solve for their parameters using **maximum likelihood estimation (MLE)** which is to say we find the parameters which maximize the likelihood of the data. This can be done with **expectation maximization (EM)** or through other numerical optimization strategies such as gradient descent.