Random Variables in R

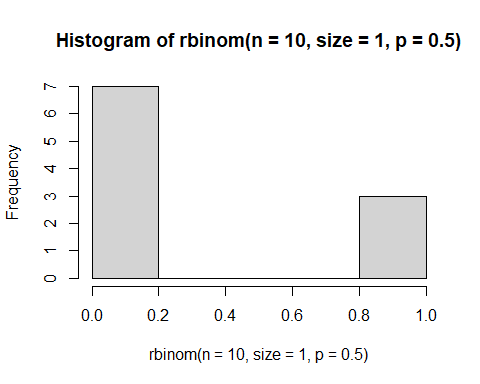
Chad Huntebrinker

#### Introduction

R has a suite of functions dealing with random variables.

1. The r functions generate random numbers drawn from several different probability distributions. In order to replicate the **exact** random numbers each time you run this document, you must set your seed using set.seed. Try running the chunk of code below, then uncomment the first line and rerun it with the seed set. See how things change for different values of set.seed.

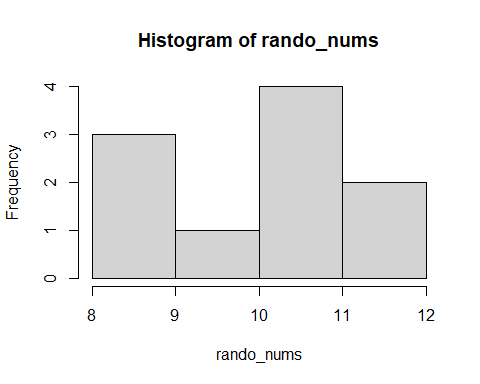
#set.seed(3)  
hist(rbinom(n=10,size=1,p=0.5)) # 10 Bernoulli random numbers with p = 0.5



##### Your Turn 1

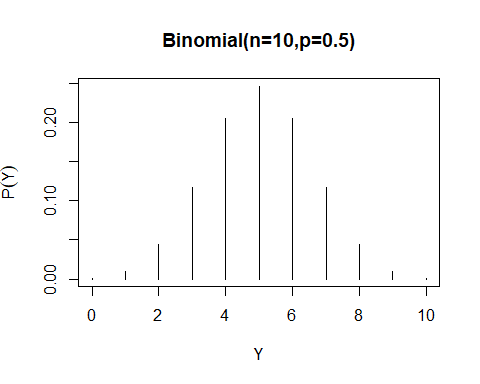
In the code chunk below, generate 10 random numbers from a normal distribution with mean = 10 and variance = 2, then use the hist function to display them. Use the RStudio search feature to pull up the help on the normal distribution if you need it. And don’t forget to set a seed so you can replicate your exact output later.

set.seed(3)  
  
rando\_nums <- rnorm(10, mean = 10, sd = sqrt(2))  
  
hist(rando\_nums)



1. The d functions generate the probability functions (height of function). Below, I generate the probability distribution for a binomial random variable with success probability and independent Bernoulli trials.

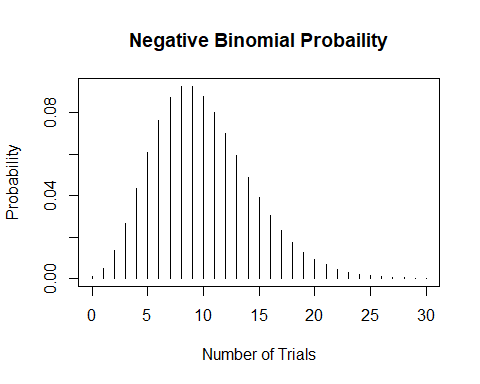
#set.seed(3)  
x <- seq(0,10,1) # All possible outcomes of the binomial random variable   
 # assuming n = 10 trials  
y <- dbinom(x,size=10,prob=0.5) # Binomial probability distribution with   
 # p = 0.5 in n = 10 trials  
plot(y~x, type="h", xlab="Y", ylab=expression(P(Y)), main="Binomial(n=10,p=0.5)")



##### Your Turn 2

In the code chunk below, use the vector x that I have created to generate the negative binomial probability distribution with probability of success = 0.5 and number of successes = 10, then plot the probability distribution. Make sure you accurately label all your axes and create an informative figure title. Use the RStudio search feature to pull up the help on the negative binomial distribution if you need it. And don’t forget to set a seed so you can replicate your exact output later. Remember, here the -axis represents the number of trials required to observe successes when the probability of a success in each trial is 0.5. Does the probability distribution make sense to you?

set.seed(3)  
x <- seq(0,30,1)  
  
y <- dnbinom(x,size=10,prob=0.5)  
  
plot(y~x, type="h", xlab="Number of Trials", ylab="Probability", main="Negative Binomial Probaility")

 Yes, the probability distribution makes sense.

1. The q functions generate quantiles from the probability distributions. Below, I am using the qt function to figure out what value of corresponds to the median, or 50th percentile of a distribution with 20 degrees of freedom. The 50th percentile is the value of that cuts the probability distribution exactly in half. Remember what the distribution looks like. Does my answer make sense?

qt(p=0.5,df=20)

## [1] 0

Yes, that does make sense. Since the t distribution is symmetric about zero, it would make sense for the median to be 0.

##### Your Turn 3

In the code chunk below, find the value of from an distribution with 2 and 32 degrees of freedom that cuts off the lower 95% of the probability distribution from the upper 5%. Use the RStudio search feature to pull up the help on the distribution if you need it.

#??"F Distribution"  
qf(0.95, 2, 32)

## [1] 3.294537

1. The p functions generate cumulative probability from the probability distributions. Below I am using the pt function to figure out using a distribution with 20 degrees of freedom. Remember what the distribution looks like. Does my answer make sense?

pt(q=0,df=20)

## [1] 0.5

Yes, you’re answer makes sense. It’s saying that there’s a 50% chance of a t-distribution to be less than 0.

##### Your Turn 4

In the code chunk below, find from a normal distribution with mean 0 and variance 1. How can you use one of the rules of probability to calculate from your first line of code? Write the code on the next line of the code chunk. Finally, use the RStudio documentation on pnorm to figure out how to use the function to get directly.

pnorm(-2.83)

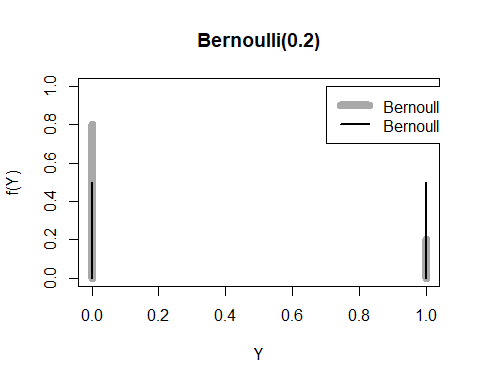
## [1] 0.0023274

### Discrete Distributions

#### Bernoulli

A **Bernoulli random variable** can take on the values 1 (success) with probability , or 0 (failure) with probability . The plot below contains the probability distribution of two Bernoulli distributions. The first (in gray) has a probability of success and the second (in black) has a probability of success .

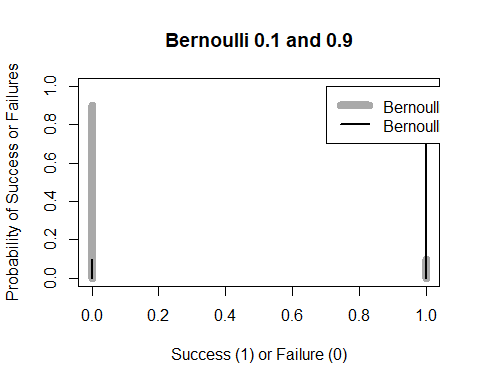
x <- 0:1  
plot(x, dbinom(x, 1, 0.2), type = "h", xlab= "Y", ylab = "f(Y)",   
 ylim = c(0,1), lwd = 8, col = "dark gray", main = "Bernoulli(0.2)")  
lines(x, dbinom(x, 1, 0.5), type = "h", lwd = 2, col = "black")  
legend(0.7, 1.0, c("Bernoulli(0.2)","Bernoulli(0.5)"),col=c("dark gray","black"), lwd=c(8,2))



##### Your Turn 5

Plot Bernoulli probability distributions for and . Make sure you accurately label all your axes and create an informative figure title.

x <- 0:1  
plot(x, dbinom(x, 1, 0.1), type = "h", xlab= "Success (1) or Failure (0)", ylab = "Probability of Success or Failures",   
 ylim = c(0,1), lwd = 8, col = "dark gray", main = "Bernoulli 0.1 and 0.9")  
lines(x, dbinom(x, 1, 0.9), type = "h", lwd = 2, col = "black")  
legend(0.7, 1.0, c("Bernoulli(0.1)","Bernoulli(0.9)"),col=c("dark gray","black"), lwd=c(8,2))

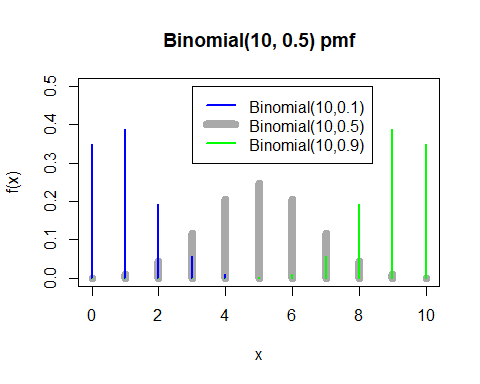


The Bernoulli experiment forms the foundation for many of the next discrete distributions.

#### Binomial

The **binomial distribution** applies when we perform independent Bernoulli trials and are interested in the total number of “successes” observed. The outcome here, , where and . The plot below displays three binomial distributions, all for Bernoulli trials: in gray, ; in blue, ; and in green, .

x <- seq(0, 10, 1)  
plot(x, dbinom(x, 10, 0.5), type = "h", ylab = "f(x)", lwd = 8, col = "dark gray", ylim = c(0,0.5),  
 main = "Binomial(10, 0.5) pmf")  
lines(x, dbinom(x, 10, 0.1), type = "h", lwd = 2, col = "blue")  
lines(x, dbinom(x, 10, 0.9), type = "h", lwd = 2, col = "green")  
legend(3, 0.5, c("Binomial(10,0.1)","Binomial(10,0.5)","Binomial(10,0.9)"), col=c("blue","dark gray","green"),lwd=c(2,8,2))

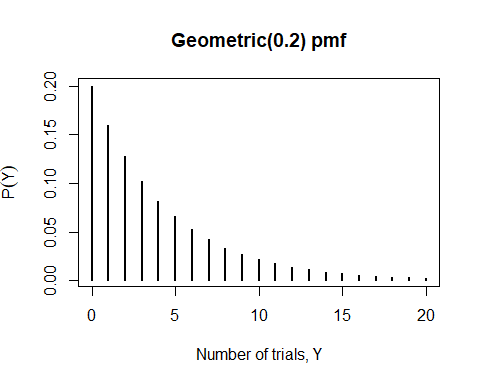


We can see the shifting of probability from low values for to high values for . This makes sense, as it becomes more likely with to observe a success for an individual trial. Thus, in 10 trials, more successes (e.g., 8, 9, or 10) are likely. For , the number of successes are likely to be around 5 (e.g., half of the 10 trials).

#### Geometric

The **geometric distribution** applies when we are interested in the number of independent Bernoulli trials that are required to observe a single “success.” The plot below displays the probability distribution for the number of trials required to observe 1 “success” with the probability of success is .

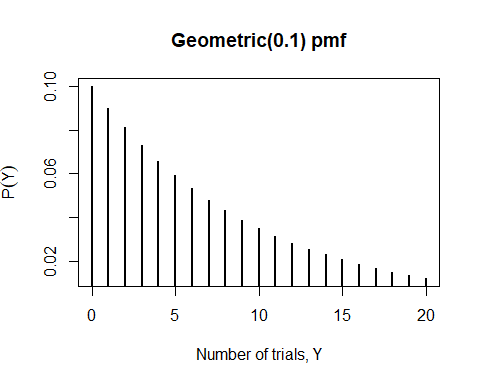
x <- seq(0, 20, 1)  
plot(x, dgeom(x, 0.2), type = "h", lwd = 2,  
 ylab = expression(P(Y)), xlab = "Number of trials, Y",  
 main = "Geometric(0.2) pmf")



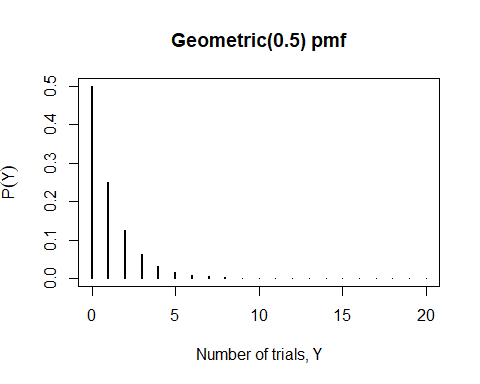
##### Your Turn 6

What happens to the geometric distribution if you vary ? Show me a few plots and explain.

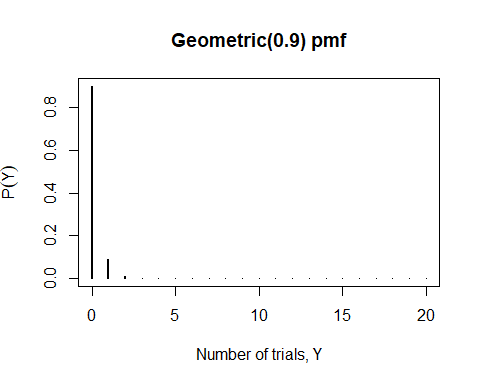
x <- seq(0, 20, 1)  
plot(x, dgeom(x, 0.1), type = "h", lwd = 2,  
 ylab = expression(P(Y)), xlab = "Number of trials, Y",  
 main = "Geometric(0.1) pmf")



plot(x, dgeom(x, 0.5), type = "h", lwd = 2,  
 ylab = expression(P(Y)), xlab = "Number of trials, Y",  
 main = "Geometric(0.5) pmf")



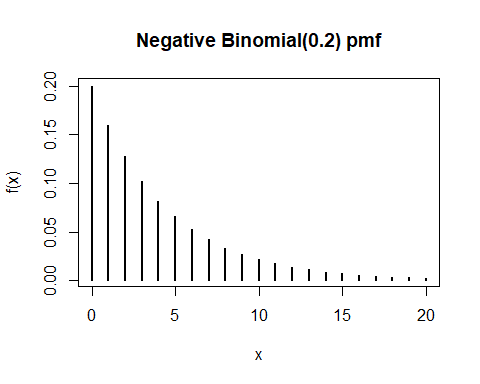
plot(x, dgeom(x, 0.9), type = "h", lwd = 2,  
 ylab = expression(P(Y)), xlab = "Number of trials, Y",  
 main = "Geometric(0.9) pmf")

 What we see is that as increases (from 0.1 -> 0.5 -> 0.9), we see the curve gets steeper and steeper. As a result, we can conclude that as we increase the probability of success, the less amount of trials are needed.

#### Negative Binomial

The **negative binomial distribution** generalizes the geometric distribution for 1 or more successes. I have below has set , so it’s identical to the geometric above.

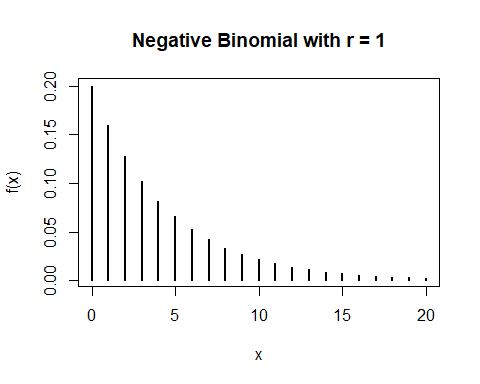
x <- seq(0, 20, 1)  
plot(x, dnbinom(x,1,0.2), type = "h", ylab = "f(x)", lwd = 2,  
 main = "Negative Binomial(0.2) pmf")



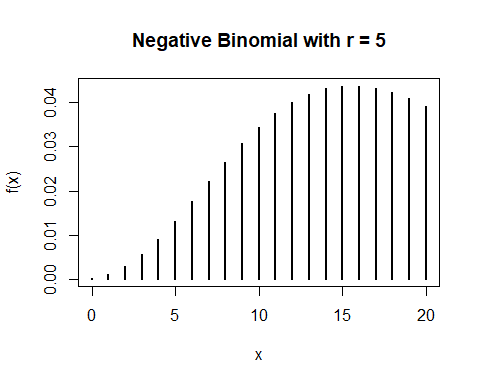
##### Your Turn 7

Play around with the number of successes and see how the negative binomial distribution changes.

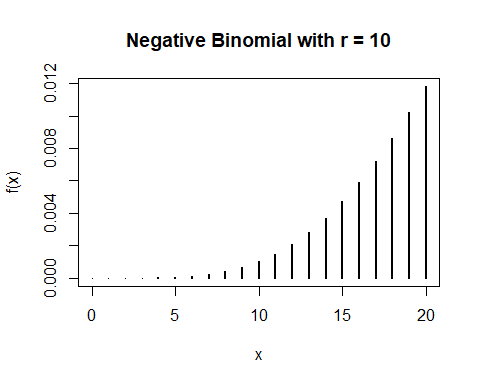
x <- seq(0, 20, 1)  
plot(x, dnbinom(x,1,0.2), type = "h", ylab = "f(x)", lwd = 2,  
 main = "Negative Binomial with r = 1")



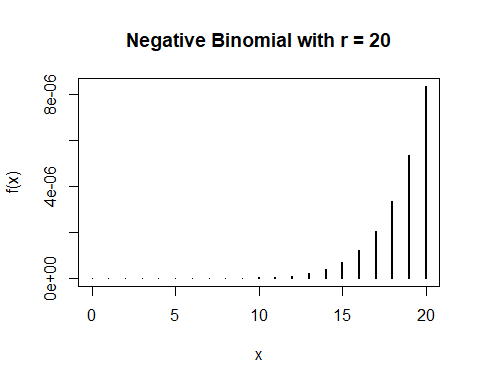
plot(x, dnbinom(x,5,0.2), type = "h", ylab = "f(x)", lwd = 2,  
 main = "Negative Binomial with r = 5")



plot(x, dnbinom(x,10,0.2), type = "h", ylab = "f(x)", lwd = 2,  
 main = "Negative Binomial with r = 10")



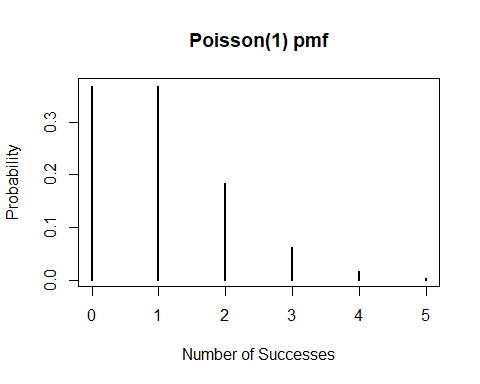
plot(x, dnbinom(x,20,0.2), type = "h", ylab = "f(x)", lwd = 2,  
 main = "Negative Binomial with r = 20")



#### Poisson

The **Poisson distribution** describes the number of “successes” that occur over a fixed period of time. The rate parameter represents the expected number of successes during the time period. For example, the distribution below could be describing the chances that 0, 1, 2, 3, 4, or 5 parties will be seated at a restaurant in the next 15 minutes if the average number of parties seated every 15 minutes is only .

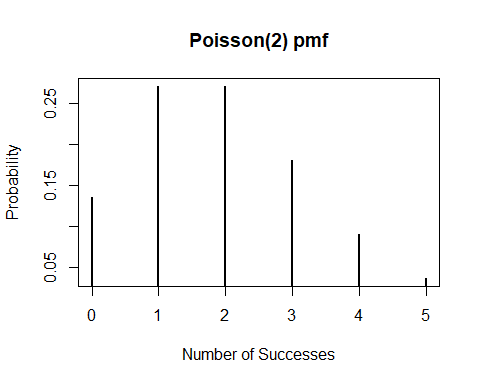
x <- seq(0, 5, 1)  
plot(x, dpois(x, lambda=1), type = "h", xlab = "Number of Successes", ylab = "Probability", main = "Poisson(1) pmf", lwd = 2)



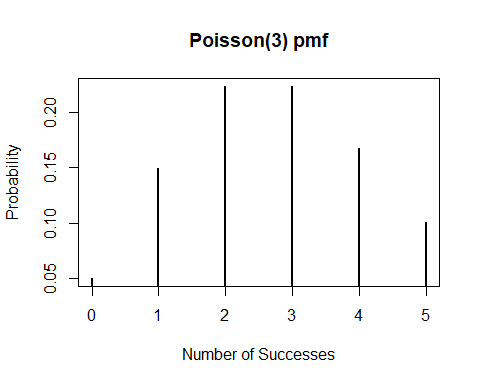
##### Your Turn 8

What happens if you increase ? To 2? To 3? to 5? How do these changes impact the chances that 5 parties will be seated in the next 15 minutes?

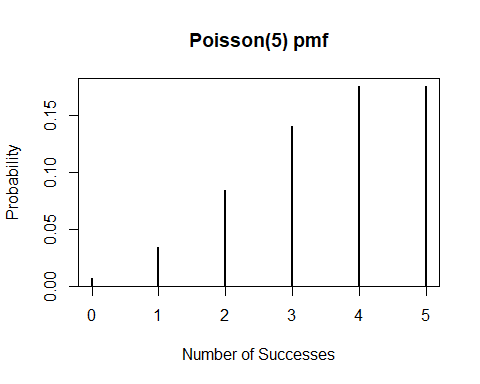
x <- seq(0, 5, 1)  
plot(x, dpois(x, lambda=2), type = "h", xlab = "Number of Successes", ylab = "Probability", main = "Poisson(2) pmf", lwd = 2)



plot(x, dpois(x, lambda=3), type = "h", xlab = "Number of Successes", ylab = "Probability", main = "Poisson(3) pmf", lwd = 2)



plot(x, dpois(x, lambda=5), type = "h", xlab = "Number of Successes", ylab = "Probability", main = "Poisson(5) pmf", lwd = 2)

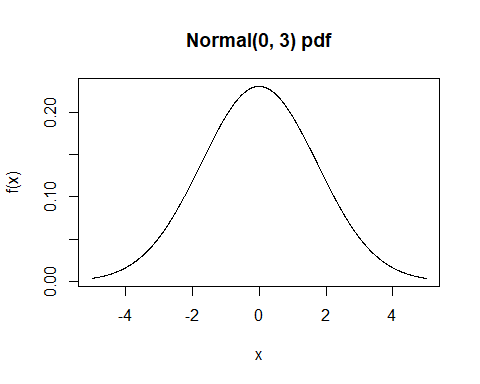
 We see that the impact is in the number of successes. As we increase r from 2 -> 3 -> 5, we see that the number of success increase as well.

### Continuous Distributions

#### Normal

The **Normal distribution** has two parameters: which is its mean, and which is its variance. The Normal distribution functions in R use , the standard deviation, instead of . Below, I have plotted the probability distribution of a normal random variable with mean 0 and variance 3.

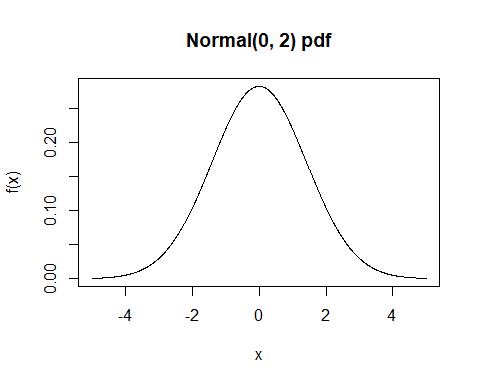
x <- seq(-5, 5, 0.01)  
plot(x, dnorm(x, 0, sqrt(3)), type = "l", ylab = "f(x)", main = "Normal(0, 3) pdf")



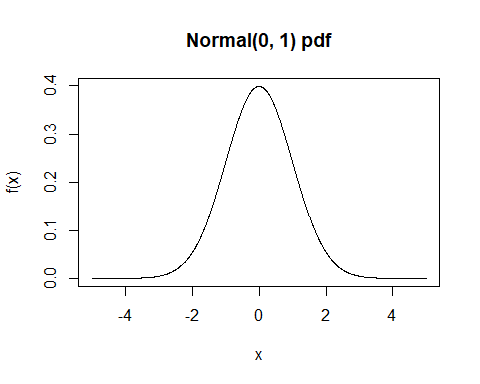
##### Your Turn 9

What happens if you decrease to 2? To 1? How do these changes impact the probability that will be bigger than 3? Use the pnorm function to calculate for my normal distribution and these distributions.

x <- seq(-5, 5, 0.01)  
plot(x, dnorm(x, 0, sqrt(2)), type = "l", ylab = "f(x)", main = "Normal(0, 2) pdf")



plot(x, dnorm(x, 0, sqrt(1)), type = "l", ylab = "f(x)", main = "Normal(0, 1) pdf")



1 - pnorm(3, sd = sqrt(3))

## [1] 0.04163226

1 - pnorm(3, sd = sqrt(2))

## [1] 0.01694743

1 - pnorm(3, sd = sqrt(1))

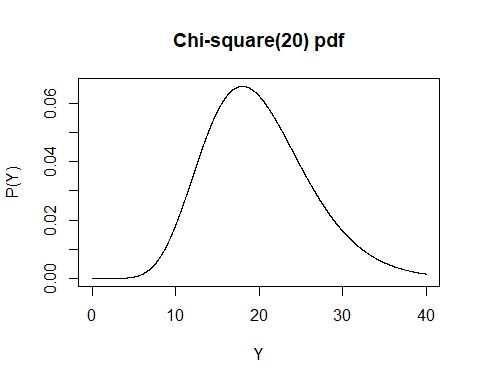
## [1] 0.001349898

We see that as we decrease , there is a higher a amount around the mean 0 (thus a thinner and taller peak).

#### Chi-square

The **Chi-square distribution** is formed by squaring and summing up independent standard normal random variables. This distribution has one parameters: which is its degrees of freedom. We will talk more about degrees of freedom later. Below, I have plotted the probability distribution of a chi-square random variable with 20 degrees of freedom.

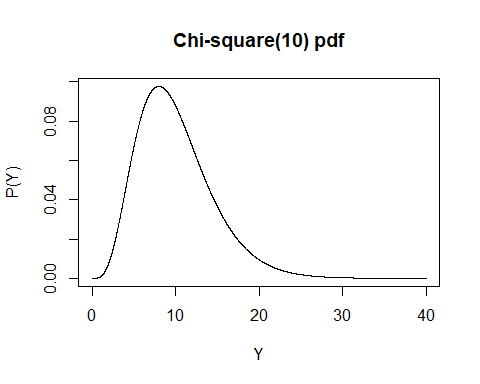
x <- seq(0, 40, 0.01)  
plot(x, dchisq(x, 20), type = "l", xlab = "Y", ylab = "P(Y)", main = "Chi-square(20) pdf")



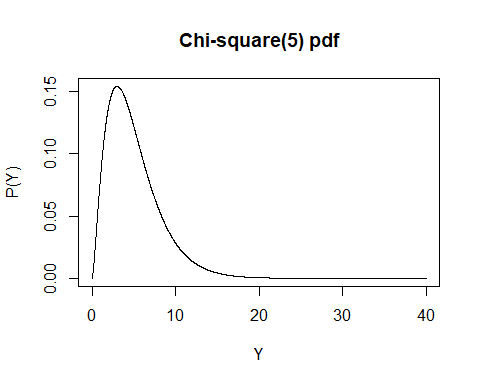
##### Your Turn 10

What happens if you decrease to 10? To 5? To 1? Plot these new distributions below and use the pchisq function to calculate and the dchisq function to determine the height of the function at for each distribution.

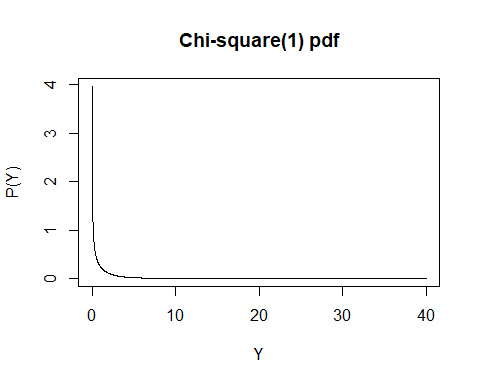
x <- seq(0, 40, 0.01)  
plot(x, dchisq(x, 10), type = "l", xlab = "Y", ylab = "P(Y)", main = "Chi-square(10) pdf")



plot(x, dchisq(x, 5), type = "l", xlab = "Y", ylab = "P(Y)", main = "Chi-square(5) pdf")



plot(x, dchisq(x, 1), type = "l", xlab = "Y", ylab = "P(Y)", main = "Chi-square(1) pdf")



1 - pchisq(30, 20)

## [1] 0.06985366

dchisq(30, 20)

## [1] 0.01620358

1 - pchisq(30, 10)

## [1] 0.0008566412

dchisq(30, 10)

## [1] 0.0003226314

1 - pchisq(30, 5)

## [1] 1.474858e-05

dchisq(30, 5)

## [1] 6.684262e-06

1 - pchisq(30, 1)

## [1] 4.320463e-08

dchisq(30, 1)

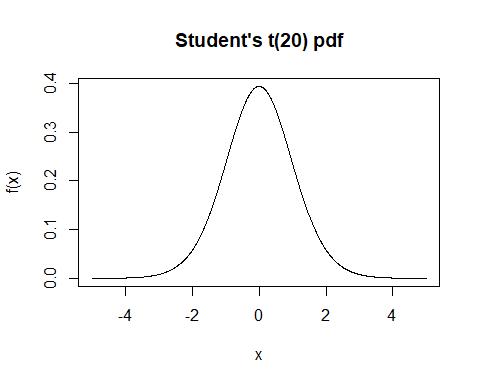
## [1] 2.228087e-08

As we decrease , we see that not only does the slope of the graph move back, but it also gets thinner and taller too; thus showing the mean and varaince decreases

#### Students

The **Student’s distribution** is formed using a standard normal random variable and an independent chi-square random variable. This distribution has one parameter that results from its dependence on a chi-square: which is also its degrees of freedom. Below, I have plotted the probability distribution of a random variable with 20 degrees of freedom.

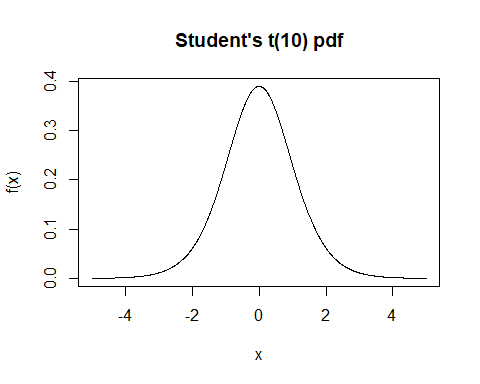
x <- seq(-5, 5, 0.01)  
plot(x, dt(x, 20), type = "l", ylab = "f(x)", main = "Student's t(20) pdf")



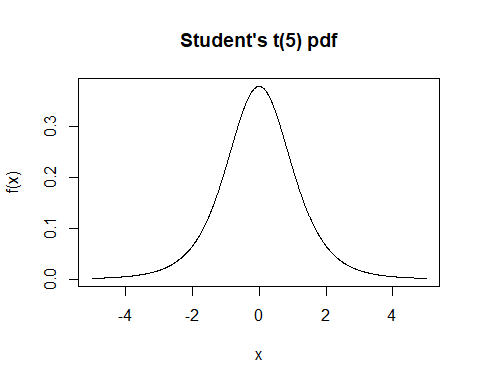
##### Your Turn 11

What happens if you decrease to 10? To 5? To 1? Plot these new distributions below and use the pt function to calculate and the dt function to determine the height of the function at for each distribution.

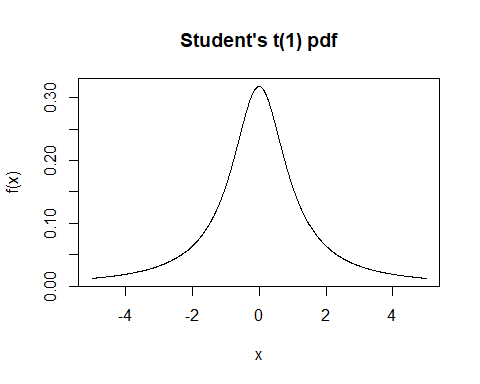
x <- seq(-5, 5, 0.01)  
plot(x, dt(x, 10), type = "l", ylab = "f(x)", main = "Student's t(10) pdf")



plot(x, dt(x, 5), type = "l", ylab = "f(x)", main = "Student's t(5) pdf")



plot(x, dt(x, 1), type = "l", ylab = "f(x)", main = "Student's t(1) pdf")



1 - pt(3.5, 20)

## [1] 0.001127562

dchisq(3.5, 20)

## [1] 3.68582e-05

1 - pt(3.5, 10)

## [1] 0.002863253

dchisq(3.5, 10)

## [1] 0.03395437

1 - pt(3.5, 5)

## [1] 0.008642216

dchisq(3.5, 5)

## [1] 0.1513128

1 - pt(3.5, 1)

## [1] 0.08858553

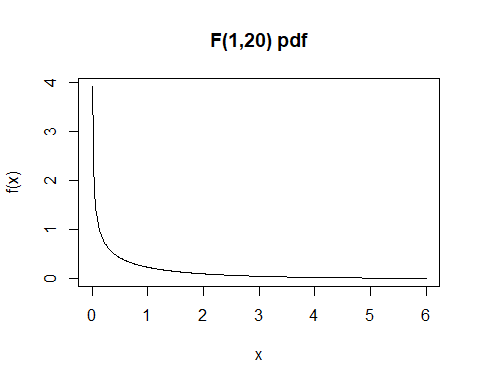
dchisq(3.5, 1)

## [1] 0.03705618

We see that as we decrease the shape get smaller and thinner (meaning the probability of more values being outside of the mean increases).

An  **random variable** is created by taking the ratio of two independent chi-square random variables each dividing by its corresponding degrees of freedom. So, the distribution has two parameters that result from its dependence on the two chi-squares: and , or the numerator and denominator degrees of freedom. Below, I have plotted the probability distribution of an random variable with 1 and 20 degrees of freedom.

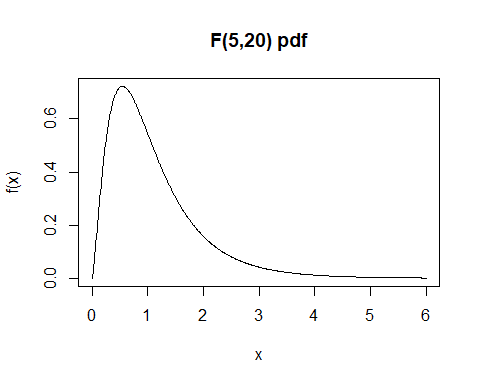
x <- seq(0, 6, 0.01)  
plot(x, df(x, 1, 20), type = "l", ylab = "f(x)", main = "F(1,20) pdf")



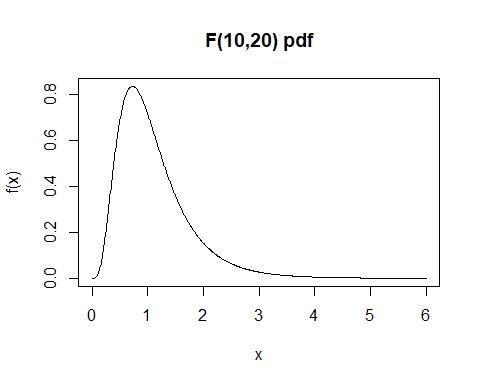
##### Your Turn 12

How do the degrees of freedom (numerator and/or denominator) change the shape? Increase the numerator degrees of freedom to 19 and plot. Explain the changes you see.

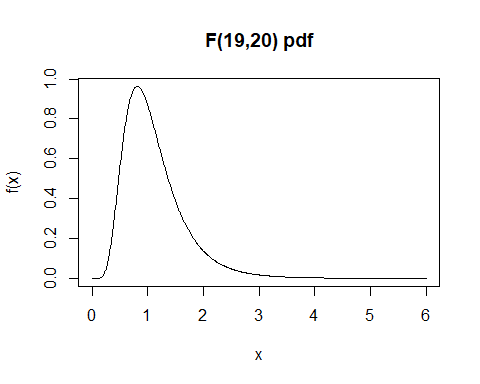
x <- seq(0, 6, 0.01)  
plot(x, df(x, 5, 20), type = "l", ylab = "f(x)", main = "F(5,20) pdf")



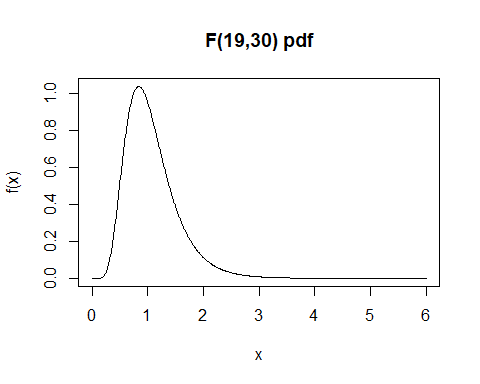
plot(x, df(x, 10, 20), type = "l", ylab = "f(x)", main = "F(10,20) pdf")



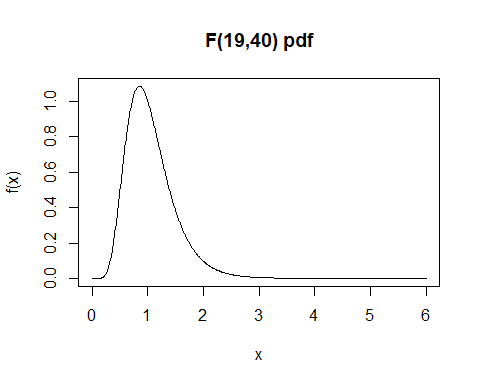
plot(x, df(x, 19, 20), type = "l", ylab = "f(x)", main = "F(19,20) pdf")



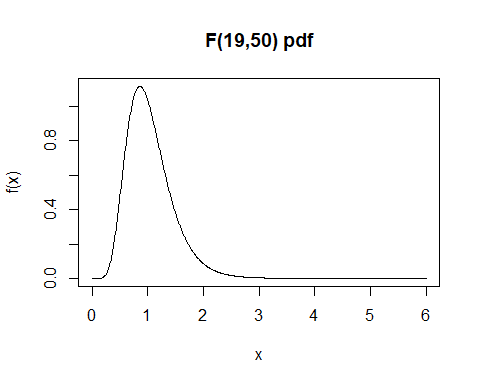
plot(x, df(x, 19, 30), type = "l", ylab = "f(x)", main = "F(19,30) pdf")



plot(x, df(x, 19, 40), type = "l", ylab = "f(x)", main = "F(19,40) pdf")



plot(x, df(x, 19, 50), type = "l", ylab = "f(x)", main = "F(19,50) pdf")

 We see that as we increase the numerator, the distribution becomes more concentrated around the peak (taller, thinner peak). We see the same thing as we increase the denominator as well.

### Document Information

All of the statistical analyses in this document will be performed using R version 4.4.1 (2024-06-14 ucrt). R packages used will be maintained using the [packrat](http://rstudio.github.io/packrat/) dependency management system.

sessionInfo()

## R version 4.4.1 (2024-06-14 ucrt)  
## Platform: x86\_64-w64-mingw32/x64  
## Running under: Windows 11 x64 (build 22631)  
##   
## Matrix products: default  
##   
##   
## locale:  
## [1] LC\_COLLATE=English\_United States.utf8   
## [2] LC\_CTYPE=English\_United States.utf8   
## [3] LC\_MONETARY=English\_United States.utf8  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United States.utf8   
##   
## time zone: America/Chicago  
## tzcode source: internal  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] rmarkdown\_2.28 knitr\_1.48   
##   
## loaded via a namespace (and not attached):  
## [1] compiler\_4.4.1 fastmap\_1.2.0 cli\_3.6.3 tools\_4.4.1   
## [5] htmltools\_0.5.8.1 rstudioapi\_0.16.0 yaml\_2.3.10 highr\_0.11   
## [9] xfun\_0.46 digest\_0.6.36 rlang\_1.1.4 evaluate\_0.24.0