HW5 - Common Probability Distributions

Madhu Peduri

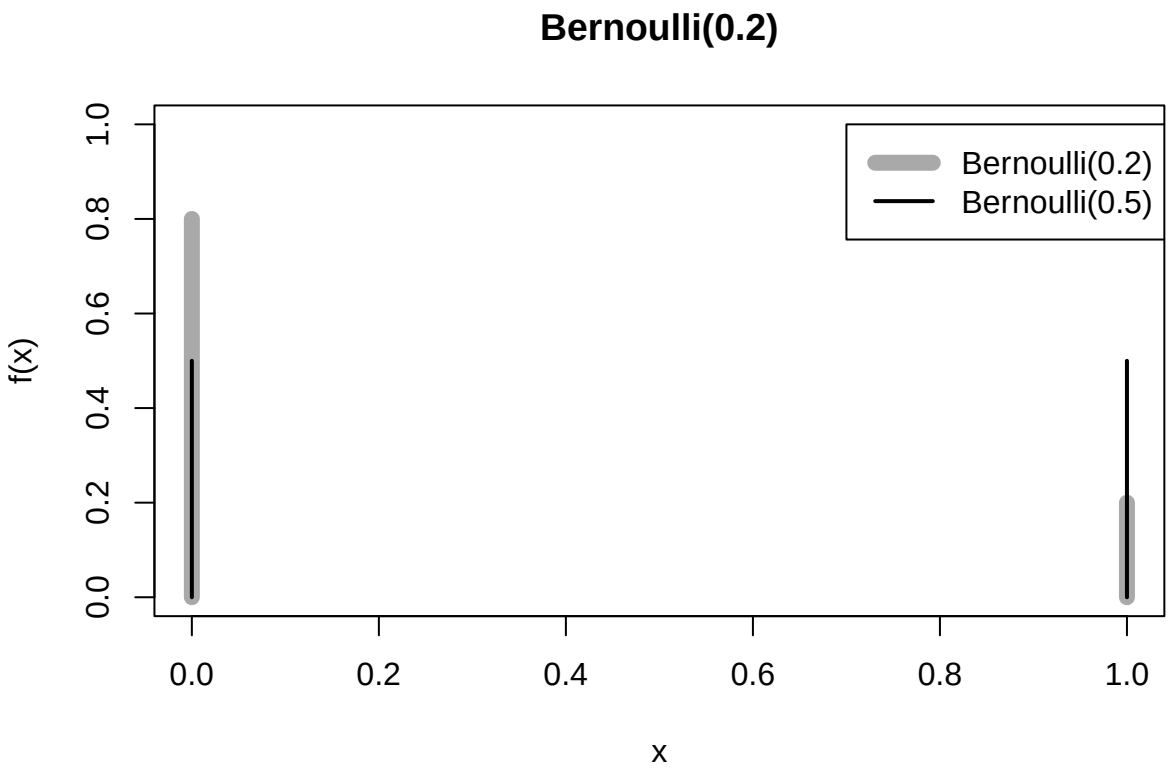
June 26, 2021

0.1 Discrete Distributions

0.1.1 Bernoulli

The **Bernoulli distribution**, named for Jacob Bernoulli, assigns probability to the outcomes of a single Bernoulli experiment—one where the only possible outcomes can be thought of as a "success" or a "failure" (e.g., a coin toss). Here, the random variable x can take on the values 1 (success) with probability p, or 0 (failure) with probability $q = 1 - p$. The plot below contains the pmf of two Bernoulli distributions. The first (in gray) has a probability of success $p = 0.2$ and the second (in black) has a probability of success $p = 0.5$.

```
x <- 0:1
plot(x, dbinom(x, 1, 0.2), type = "h", ylab = "f(x)", 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, c("Bernoulli(0.2)", "Bernoulli(0.5)"), col = c("dark gray",
    "black"), lwd = c(8, 2))
```

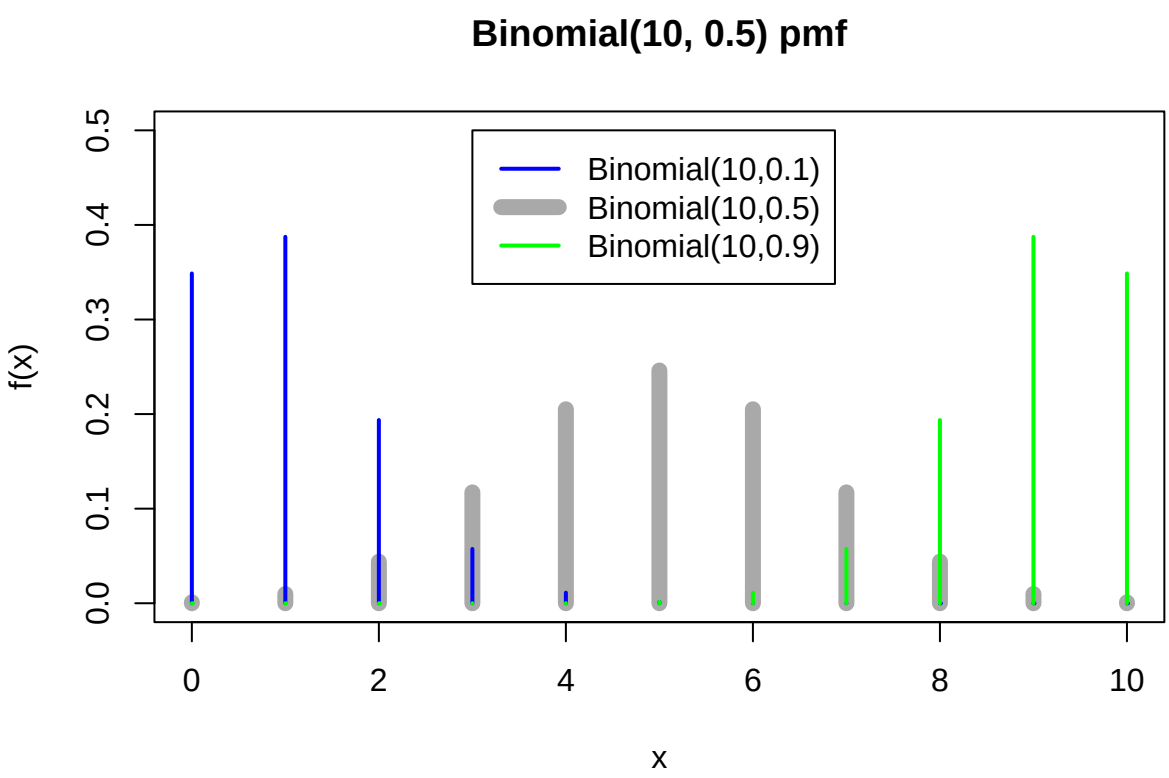


The Bernoulli experiment forms the foundation for many of the next discrete distributions.

0.1.2 Binomial

The **binomial distribution** applies when we perform n Bernoulli experiments and are interested in the total number of "successes" observed. The outcome here, $y = \sum x_i$, where $P(x_i = 1) = p$ and $P(x_i = 0) = 1 - p$. The plot below displays three binomial distributions, all for $n = 10$ Bernoulli trials: in gray, $p = 0.5$; in blue, $p = 0.1$; and in green, $p = 0.9$.

```
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 $p = 0.1$ to high values for $p = 0.9$. This makes sense, as it becomes more likely with $p = 0.9$ to observe a success for an individual trial. Thus, in 10 trials, more successes (e.g., 8, 9, or 10) are likely. For $p = 0.5$, the number of successes are likely to be around 5 (e.g., half of the 10 trials).

0.1.3 Hypergeometric

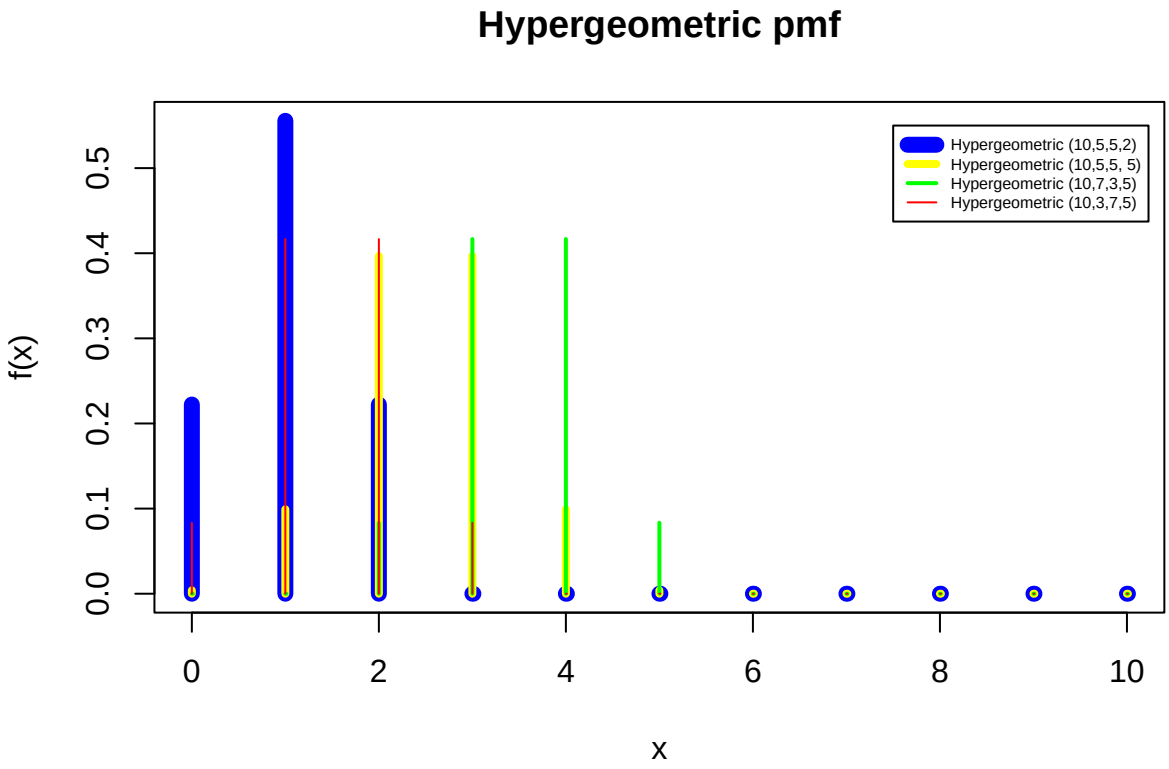
The **Hypergeometric distribution** is a discrete distribution that describes the probability of x successes in n draws without replacement wherein each draw is either success or failure. In contrast, the **binomial distribution** describes the probability of x successes in n draws with replacement. We can represent this using below notation

$$p(x) = \frac{choose(m, x).choose(n, k - x)}{choose(m + n, k)}$$

- In the example below, we have different set of parameters.
- If we choose 2 balls from the urn - $x = (0, 2), (1, 1), (2, 0)$.
- If we choose 5 balls from the urn with different number of colored balls - (7,3),$x = (2, 3)(3, 2), (4, 1), (5, 0)$.
- Probability distribution shifts as the number of success or failures changes.
- Probability distribution is symmetric.

```
x <- seq(0, 10, 1)
plot(x, dhyper(x, 5, 5, 2), type = "h", ylab = "f(x)", lwd = 8,
    col = "blue", main = "Hypergeometric pmf")
lines(x, dhyper(x, 5, 5, 5), type = "h", ylab = "f(x)", lwd = 4,
    col = "yellow")
lines(x, dhyper(x, 7, 3, 5), type = "h", ylab = "f(x)", lwd = 2,
```

```
    col = "green")
lines(x, dhyper(x, 3, 7, 5), type = "h", ylab = "f(x)", lwd = 1,
    col = "red")
par(cex = 0.5)
legend(7.5, 0.55, c("Hypergeometric (10,5,5,2)", "Hypergeometric (10,5,5, 5)",
    "Hypergeometric (10,7,3,5)", "Hypergeometric (10,3,7,5)"),
    col = c("blue", "yellow", " green", "red"), lwd = c(8, 4,
        2, 1))
```



0.1.4 Poisson

The **Poisson distribution** expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event. The Poisson distribution can also be used for the number of events in other specified intervals such as distance, area or volume.

We use below function to derive the poisson distribution $p(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad for \quad x = 0, 1, 2 \ldots n$

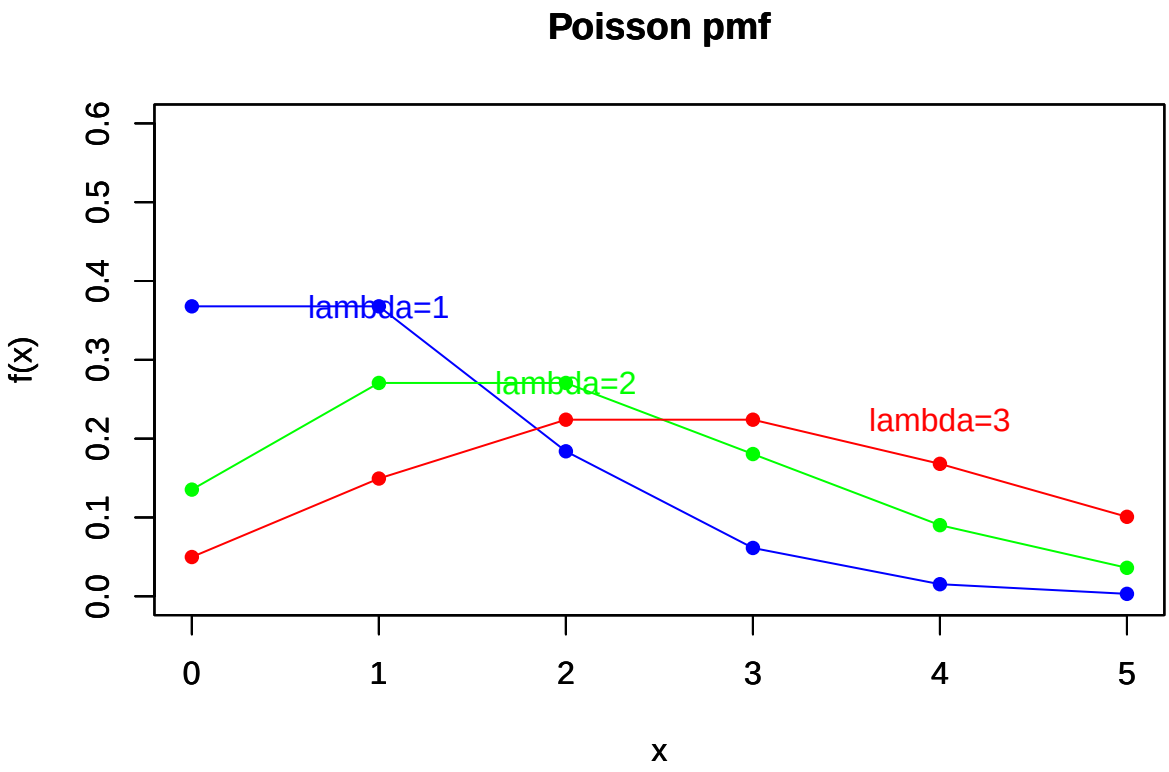
- For poission distribution, we use lambda as the parameter. It denotes the number of successes that we get with in the given time frame.
- We can observe from the below distributions that, as the lamda increases, we get the maximum probability towards the right. That is, to have more number of success events, number of trials should be more.

```
x <- seq(0, 5, 1)
poisl <- dpois(x, 1)
plot(x, poisl, ylab = "f(x)", main = "Poisson pmf", pch = 16,
```

```
    ylim = c(0, 0.6), col = "blue")
lines(x, poisl, col = "blue")
text(x = which(poisl == max(poisl))[1], y = poisl[poisl == max(poisl)][1],
    "lambda=1", col = "blue")

par(new = TRUE)
poisl <- dpois(x, 2)
plot(x, poisl, ylab = "f(x)", main = "Poisson pmf", pch = 16,
    ylim = c(0, 0.6), col = "green")
lines(x, poisl, col = "green")
text(x = which(poisl == max(poisl))[1], y = poisl[poisl == max(poisl)][1],
    "lambda=2", col = "green")

par(new = TRUE)
poisl <- dpois(x, 3)
plot(x, poisl, ylab = "f(x)", main = "Poisson pmf", pch = 16,
    ylim = c(0, 0.6), col = "red")
lines(x, poisl, col = "red")
text(x = which(poisl == max(poisl))[1], y = poisl[poisl == max(poisl)][1],
    "lambda=3", col = "red")
```



0.1.5 Geometric

The **geometric distribution** gives the probability that the first occurrence of success requires k independent trials, each with success probability p. We use below function to derive the geometric distribution.

$Pr(X = k) = p(1-p)^{k-1}$

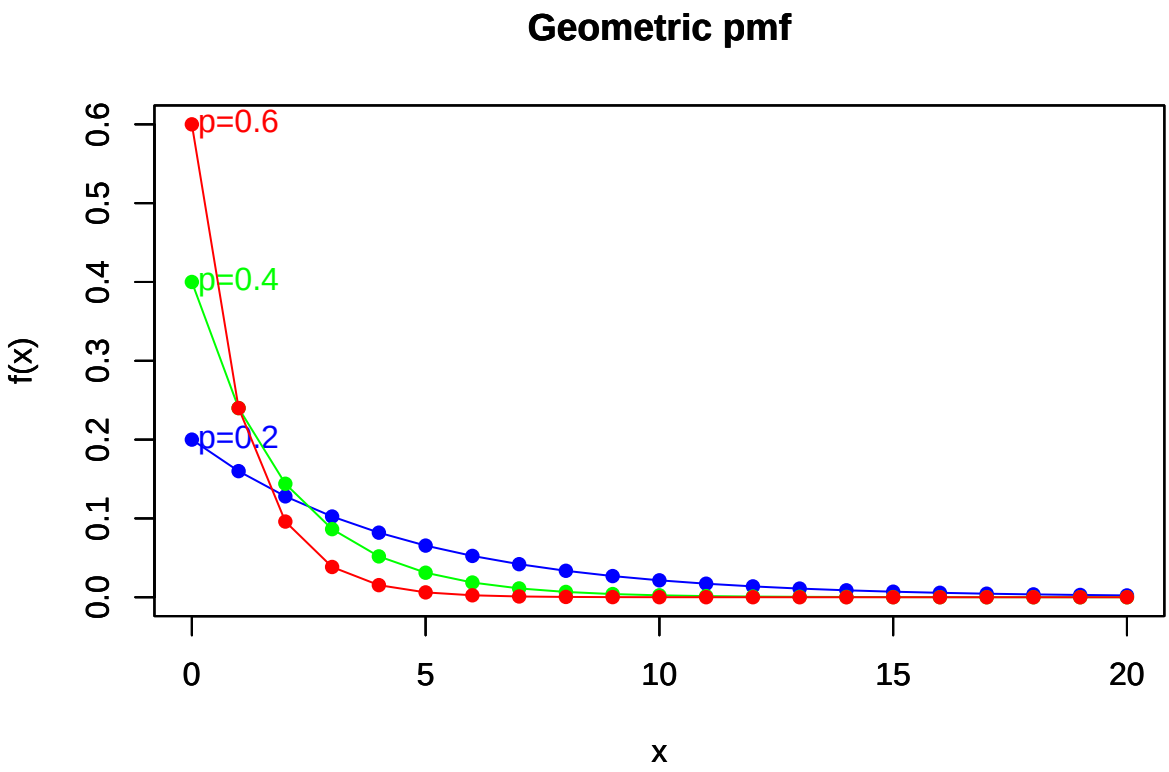
We can make below observations from below plots,

- As the initial success probability p is high, the probability of getting a success event of certian failures decreases.
- For example, after 3 falilures (x=3), the probablity of observing a success event decreases from 0.14 to 0.05 as the initial probability p increases from 0.2 to 0.6.
- For a given success probability, as the number of failure trials in creases, the probability of observing a success decreases.

```
x <- seq(0, 20, 1)
# plot(x, dgeom(x, 0.2), type = 'h', ylab = 'f(x)', lwd =
# 2, main = 'Geometric(0.2) pmf')
geom <- dgeom(x, 0.2)
plot(x, geom, ylab = "f(x)", main = "Geometric pmf", pch = 16,
    ylim = c(0, 0.6), col = "blue")
lines(x, geom, col = "blue")
text(x = which(geom == max(geom))[1], y = geom[geom == max(geom)][1],
    "p=0.2", col = "blue")

par(new = TRUE)
geom <- dgeom(x, 0.4)
plot(x, geom, ylab = "f(x)", main = "Geometric pmf", pch = 16,
    ylim = c(0, 0.6), col = "green")
lines(x, geom, col = "green")
text(x = which(geom == max(geom))[1], y = geom[geom == max(geom)][1],
    "p=0.4", col = "green")

par(new = TRUE)
geom <- dgeom(x, 0.6)
plot(x, geom, ylab = "f(x)", main = "Geometric pmf", pch = 16,
    ylim = c(0, 0.6), col = "red")
lines(x, geom, col = "red")
text(x = which(geom == max(geom))[1], y = geom[geom == max(geom)][1],
    "p=0.6", col = "red")
```



0.1.6 Negative Binomial

A **Negative binomial distribution** (also called the Pascal Distribution) is a discrete probability distribution, for random variables in a negative binomial experiment, that models the number of successes in a sequence of independent and identically distributed Bernoulli trials before a specified (non-random) number of failures (denoted r) occurs. We use below formula to derive this,

$f(x) = (x + r - 1)_{C_r}(1 - p)^r p^x \quad for \quad x = 0, 1, \ldots$

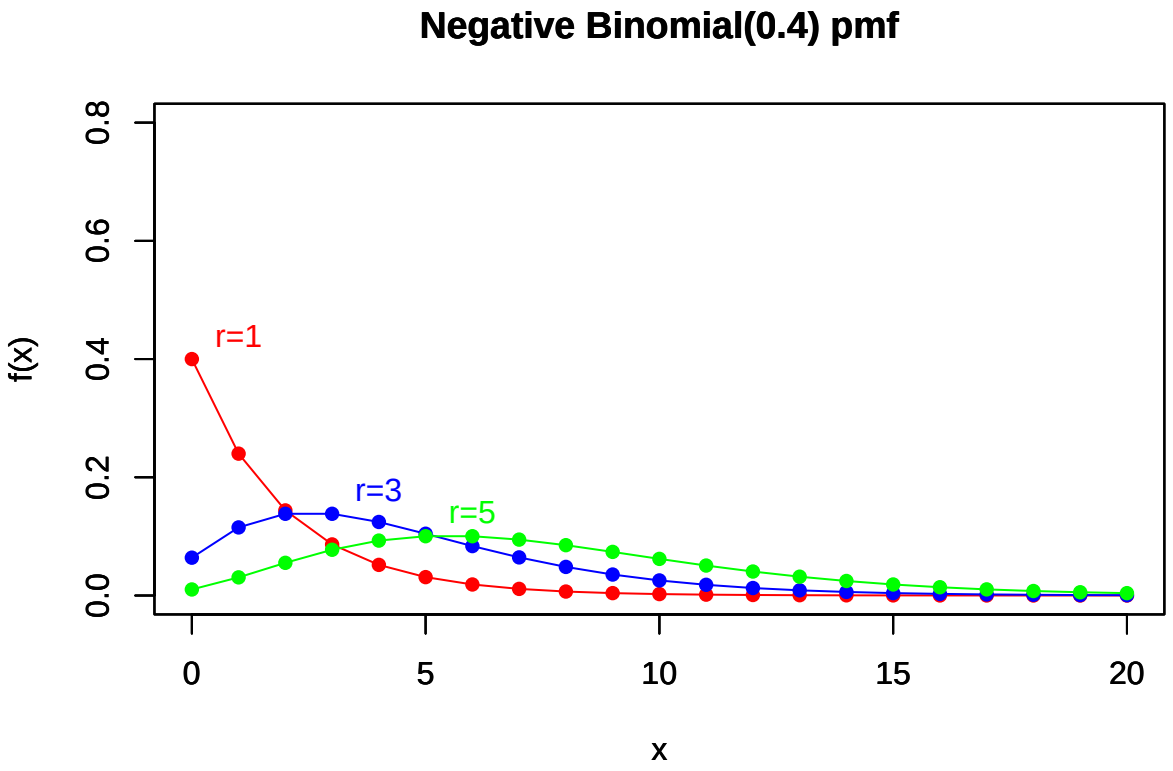
We make below observations from below plots,

- We keep the initial probability of success, p=0.4, as constant and vary the number of successes r to observe the pattern as number of trials are increased.
- As the number of successes r increases, the highest probability decreases for a given number of trials. For example, for 20 trials as r increases from 1 to 5, the highest probability $f(x)$ decreases from 0.4 to 0.08.
- When $r = 1$, it is a special case of geometric distribution.
- As the number of successes r increases, the number of trials should be more to get the highest probability. For example, As r increases from 1 to 5, the number of trials needed increases from 0 to 5 to get the highest probability $f(x)$

```
x <- seq(0, 20, 1)
nbinom <- dnbinom(x, 1, 0.4)
plot(x, nbinom, type = "p", ylab = "f(x)", pch = 16, ylim = c(0,
    0.8), main = "Negative Binomial(0.4) pmf", col = "red")
lines(x, nbinom, col = "red")
text(x = which(nbinom == max(nbinom))[1], y = nbinom[nbinom ==
    max(nbinom)][1] + 0.04, "r=1", col = "red")
```

```r
par(new = TRUE)
nbinom <- dnbinom(x, 3, 0.4)
plot(x, nbinom, type = "p", ylab = "f(x)", pch = 16, ylim = c(0,
    0.8), main = "Negative Binomial(0.4) pmf", col = "blue")
lines(x, nbinom, col = "blue")
text(x = which(nbinom == max(nbinom))[1], y = nbinom[nbinom ==
    max(nbinom)][1] + 0.04, "r=3", col = "blue")

par(new = TRUE)
nbinom <- dnbinom(x, 5, 0.4)
plot(x, nbinom, type = "p", ylab = "f(x)", pch = 16, ylim = c(0,
    0.8), main = "Negative Binomial(0.4) pmf", col = "green")
lines(x, nbinom, col = "green")
text(x = which(nbinom == max(nbinom))[1], y = nbinom[nbinom ==
    max(nbinom)][1] + 0.04, "r=5", col = "green")
```



0.2 Continuous Distributions

0.2.1 Normal

The **Normal distribution** is a type of continuous probability distribution for a real-valued random variable. Below are some of the properties of this distribution,

- The mean, mode and median are all equal.
- The curve is symmetric at the center (i.e. around the mean).
- Exactly half of the values are to the left of center and exactly half the values are to the right.

- The total area under the curve is 1.

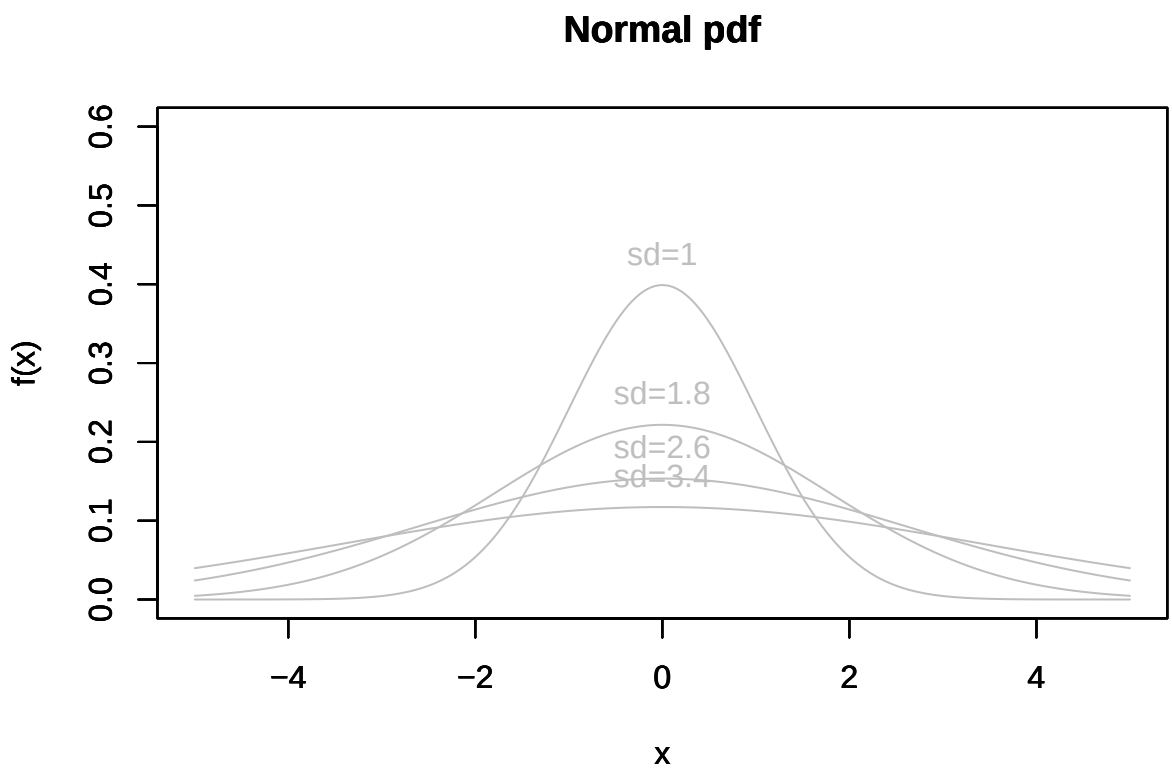
We use below function to dervive this distribution,

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad for \quad -\infty < x < \infty$$

We can make below observations,

- As the standard deviation σ increases, the curve flattens along the x-axis. suggesting, the values are scattered away from the mean.
- As the σ increases, the probability $f(x)$ is decreased.

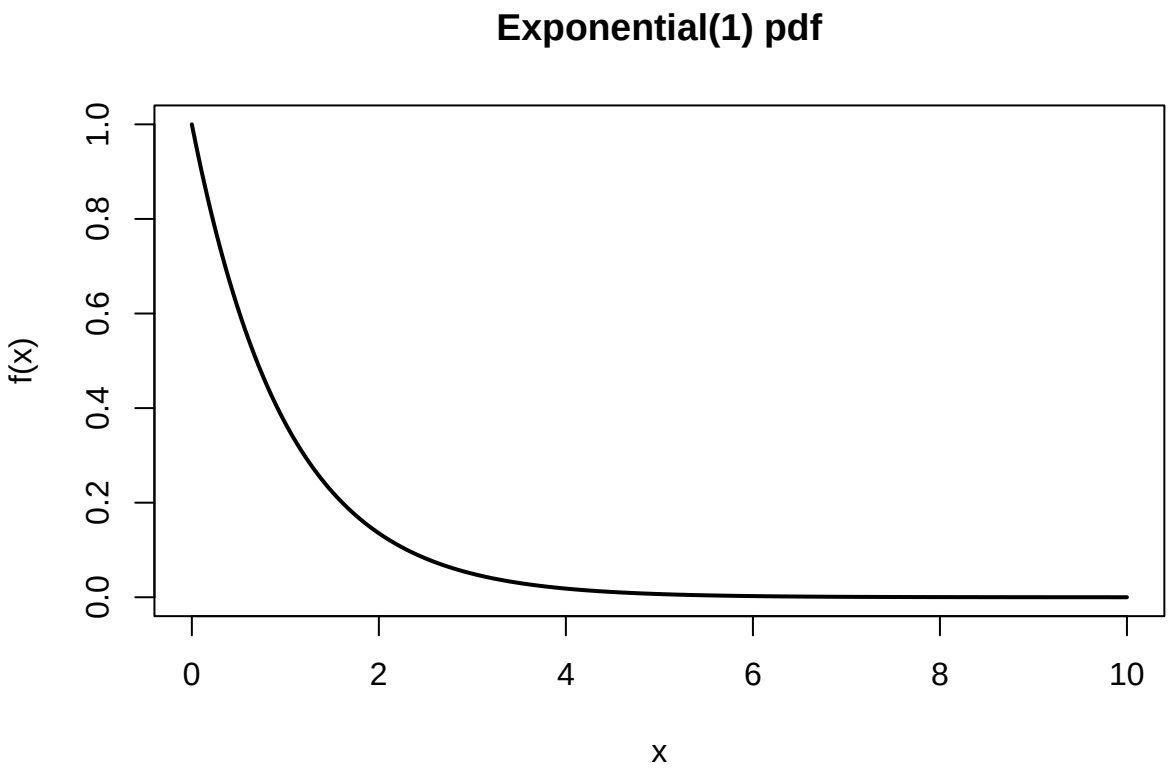
```
x <- seq(-5, 5, 0.01)
for (i in seq(1, 4, 0.8)) {
    nnorm <- dnorm(x, 0, i)
    plot(x, nnorm, type = "l", xlim = c(-5, 5), ylim = c(0, 0.6),
        ylab = "f(x)", main = "Normal pdf", col = "gray")
    text(x = x[nnorm == max(nnorm)][1], y = nnorm[nnorm == max(nnorm)][1] +
        0.04, paste("sd=", i, sep = ""), col = "gray")
    par(new = TRUE)
}
```



0.2.2 Exponential

Vary λ and describe.

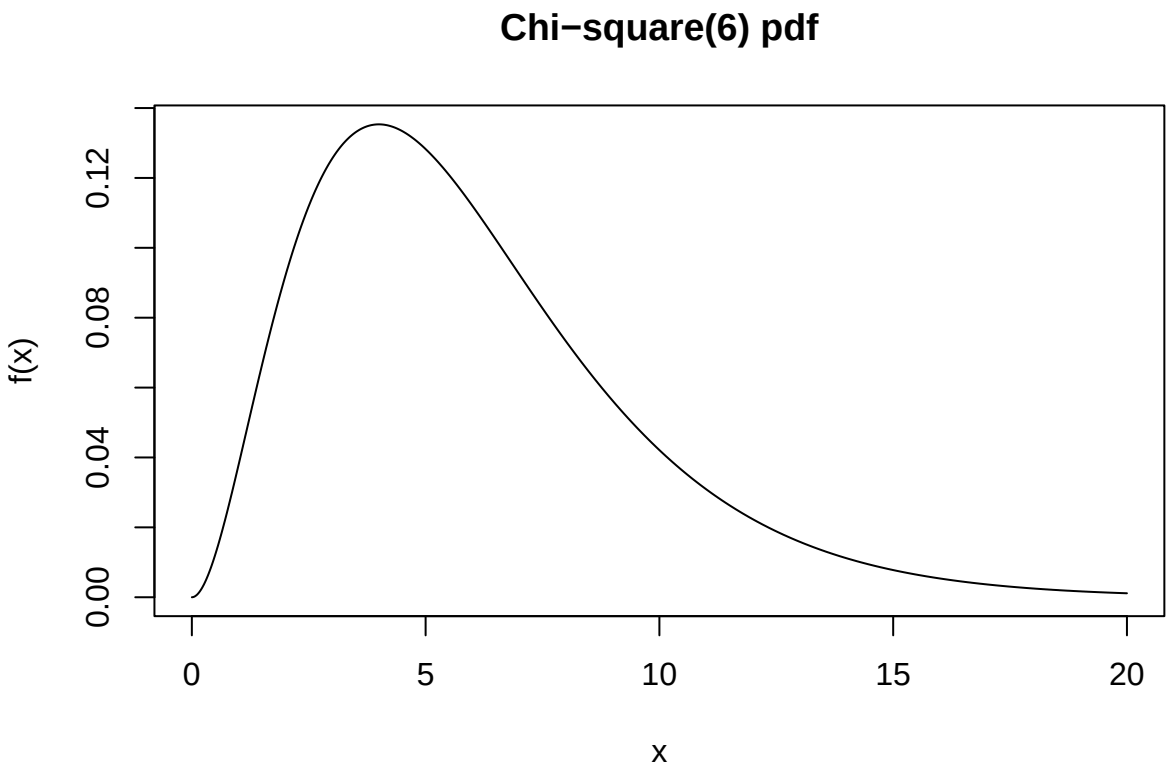
```
x <- seq(0, 10, 0.01)
plot(x, dexp(x, 1), type = "l", ylab = "f(x)", lwd = 2, main = "Exponential(1) pdf")
```



0.2.3 Chisquare

How do the degrees of freedom change the shape? Plot a few and explain.

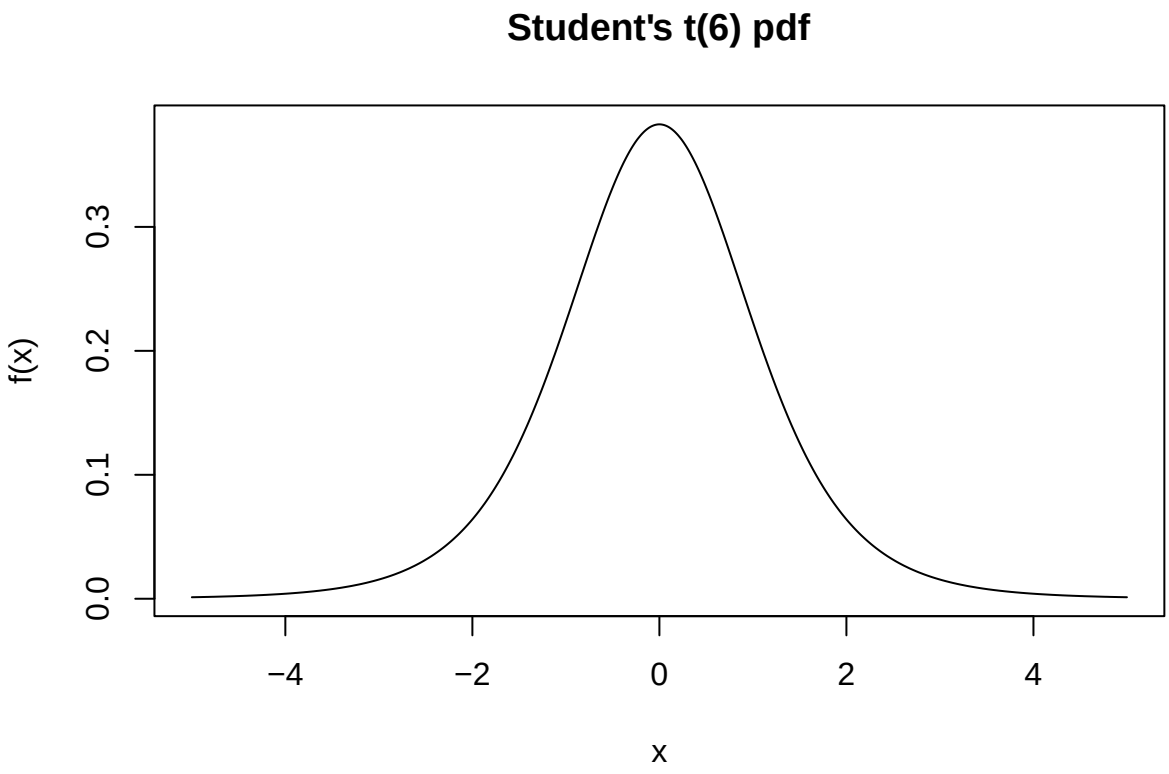
```
x <- seq(0, 20, 0.01)
plot(x, dchisq(x, 6), type = "l", ylab = "f(x)", main = "Chi-square(6) pdf")
```



0.2.4 Students t

How do the degrees of freedom change the shape? Plot a few and explain.

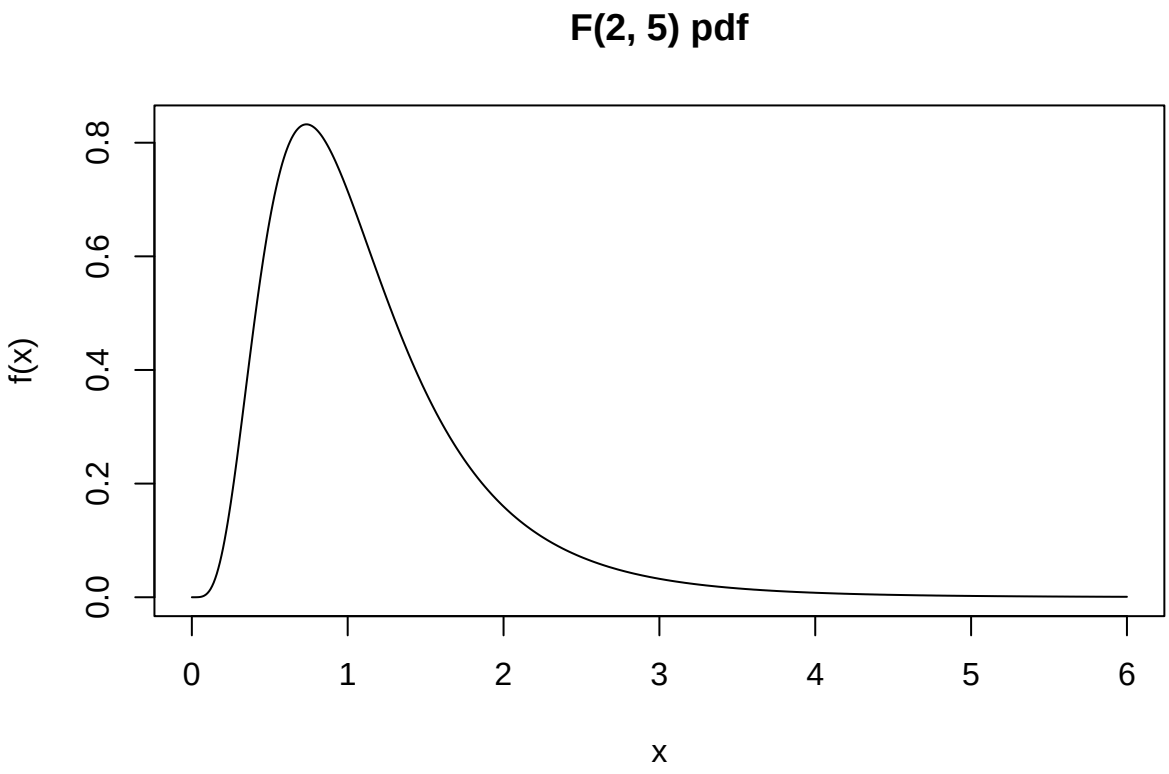
```
x <- seq(-5, 5, 0.01)
plot(x, dt(x, 6), type = "l", ylab = "f(x)", main = "Student's t(6) pdf")
```



0.2.5 F

How do the degrees of freedom (numerator and/or denominator) change the shape? Plot a few and explain.

```
x <- seq(0, 6, 0.01)
plot(x, df(x, 12, 15), type = "l", ylab = "f(x)", main = "F(2, 5) pdf")
```



0.3 Document Information.

All of the statistical analyses in this document will be performed using R version 4.1.0 (2021-05-18). R packages used will be maintained using the packrat dependency management system.

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] rmarkdown_2.8 knitr_1.33
##
## loaded via a namespace (and not attached):
```

```
##  [1] compiler_4.1.0    magrittr_2.0.1    formatR_1.11      tools_4.1.0
##  [5] htmltools_0.5.1.1 yaml_2.2.1        stringi_1.6.1     highr_0.9
##  [9] stringr_1.4.0     xfun_0.23         digest_0.6.27     rlang_0.4.11
## [13] evaluate_0.14
```