MA 354: Data Analysis I – Fall 2021
Homework 3:

Complete the following opportunities to use what we've talked about in class. These questions will be graded for correctness, communication and succinctness. Ensure you show your work and explain your logic in a legible and refined submission.

0. **Complete weekly diagnostics.**

1. Consider data originally from a study of the nesting horseshoe crabs (Brockmann, 1996). Each female crab in the study had a male crab attached to her in her nest. The study investigated factors that affect whether the female crab had any other males, called satellites, residing nearby her. Explanatory variables thought possibly to affect this included the female crab's color, spine condition, weight, and carapace width. The response outcome for each female crab is her number of satellites. The sample is

Number of Satellites	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of Observations	62	16	9	19	19	15	13	4	6	3	3	1	1	1	1	1

It is believed that the distribution of the number of satellites for a female crab is distributed Poisson(λ) where the parameter λ is of interest.

(a) Calculate the method of moments estimator for λ.
Solution: We'll be using a nleqslv package Hasselman (2018) to solve a non-linear equation to find λ! Hopefully, you're excited!

```
library(nleqslv)
library(tidyverse)
#put data into the dataframe
crab.dat <- data.frame(x=c(rep(0,62), rep(1, 16), rep(2, 9), rep(3, 19),
                           rep(4, 19), rep(5, 15), rep(6, 13), rep(7, 4),
                           rep(8, 6), rep(9, 3), rep(10, 3), 11, 12,
                           13, 14, 15))




#MOM function
momfunc <- function(par, data){
  population.mean <- par
  sample.mean <- mean(data)
  return(sample.mean-population.mean)
}
#Solve the equation
answer <- nleqslv(x=1, #worst guess :3
        fn = momfunc,
        data=crab.dat$x)

paste("Lambda is ~", answer$x, sep="")

## [1] "Lambda is ~2.97701149425287"
```

(b) Find the maximum likelihood estimator for λ.
Solution: For this, we'll use a `dpois()` function from the base R package and `optim()` function to find MLE for our Poisson distribution.

```r
LLfunc <- function(par, data){
  answer <- sum(dpois(x=data, lambda = par, log=T))
  -answer #most likely
}

answer <- optim(par = 1, #worst guess c:
      fn = LLfunc,
      data = crab.dat$x,
      method = "Brent",
      lower = 0,
      upper = 100)

paste("Parameter:",answer$par)
```

```
## [1] "Parameter: 2.9770114489889"
```

So λ is 2.97! It will help us when plotting the data with fitted Poisson distribution over it!

(c) Plot the data with the Poisson distribution fit with the MLE estimates. How well does the distribution fit the data?

```r
        crab.dat.plot <- crab.dat %>%
          mutate(f = dpois(x=x, lambda=2.977011))


        sats <- 0:15
        ggplot(crab.dat.plot, aes(x=x))+
          geom_histogram(aes(y=..density..), fill="deepskyblue2", color="black",
                         binwidth=1)+
          geom_linerange(aes(ymin=0, ymax=f), color="dark red")+
          scale_x_continuous("Number of satellites",
                             labels = as.character(sats), breaks = sats)+
          labs(y="Density",
               title="Number of observations per satellite",
               subtitle="Poisson distribution. Lambda=2.977")+
  theme_bw()+
  geom_hline(yintercept=0)
```

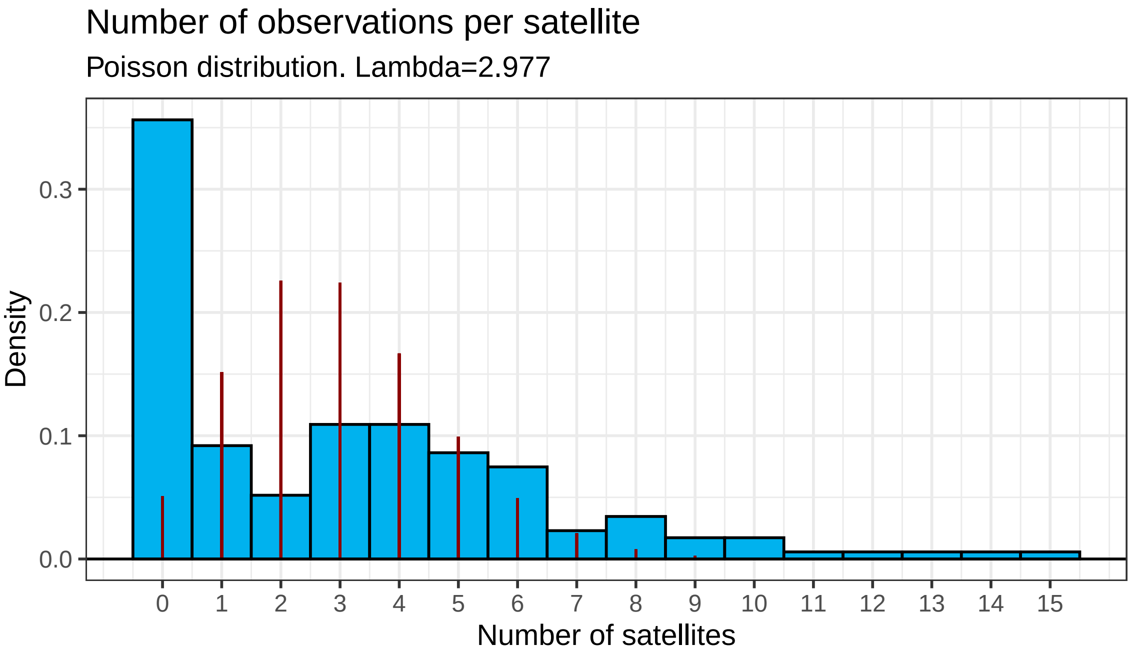


Figure 1: Poisson model fitted to our data

As you might notice, our model doesn't really fit the data that neatly — it doesn't take into account the number of zeros that we have in our dataset. This leads the distribution to either underestimate or overestimate values!

(d) Let's try another distribution - the zero-inflated Poisson distribution. Now, it is believed that the distribution of the number of satellites for a female crab is distributed $\text{Poisson}_0(\lambda, \sigma)$ where the parameters λ and σ are of interest. Find the method of moments estimators for both σ and λ.

$$f_X(x|\lambda,\sigma) = (1-\sigma)\frac{\lambda^x e^{-\lambda}}{x!} I(x \geq 1) + (\sigma + (1-\sigma)e^{-\lambda}) I(x = 0)$$

Hint: Noticing that this is a function of the Poisson PMF and rewriting it will help remarkably.
Solution: Let's use our favorite nleqslv package Hasselman (2018) to calculate the parameters that we are looking for for our zero-inflated Poisson model. We need to find λ and σ!
When calculating E(X) for ZIP, I noticed that the only difference between the regular Poisson distribution and this one is an additional $(1-\sigma)$, so I assumed that $\text{E(X)}=(1-\sigma)\lambda$. While I managed to calculate it, I struggle with Var(X), so I found cardinal (2011) discussion on StackOverflow really helpful for my calculations!

```
zip.pois <- function(par, data){
  lambda <- par[1]
  pi <- par[2]

  #Applying my knowledge
  EX1 <- (1-pi)*lambda
  EX2 <- (1-pi)*(lambda^2+lambda)

  xbar1 <- mean(data)
  xbar2 <- mean(data^2)

  c(EX1-xbar1, EX2-xbar2)
}

#solving
answerZIP<-nleqslv(x = c(1,0), #best guess at parameter
        fn = zip.pois,
        data = crab.dat$x)

paste("Lambda:",answerZIP$x[1],"Sigma:",answerZIP$x[2])
```

```
## [1] "Lambda: 5.46332046334072 Sigma: 0.455091182323775"
```

It's impossible to assess how accurate we are in our calculations without graphing the results. I am going to use `dzipois()` from VGAM package Yee et al. (2015).

```
library("VGAM")
crab.dat.zim <- crab.dat %>%
          mutate(f = dzipois(x=x, lambda=5.4633205, pstr0=0.4550912))
#make a true density function

ggplot(crab.dat.zim, aes(x=x))+
          geom_histogram(aes(y=..density..), fill="deepskyblue2", color="black",
                         binwidth=1)+
          geom_linerange(aes(ymin=0, ymax=f), color="dark red")+
          scale_x_continuous("Number of satellites",
                             labels = as.character(sats), breaks = sats)+
          labs(y="Density",
               title="Number of observations per satellite",
               subtitle="ZIP distribution. Lambda=5.4633205, Sigma=0.4550912")+
  theme_bw()+
  geom_hline(yintercept=0)
```

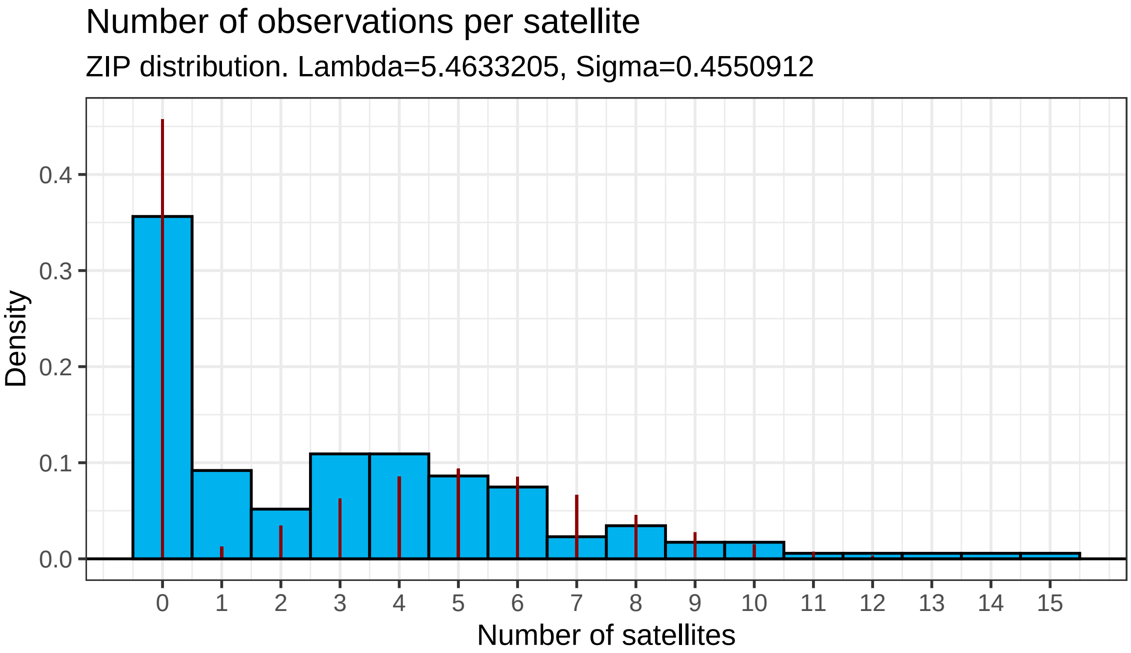


Figure 2: ZIP with parameters that we calculated via MOM

It would seem MOM inflates zero values a bit too much and underestimates the other values. I would suggest that doing MLE would be a superior way of calculating parameters! Let's see if I am correct about it.

(e) Find the maximum likelihood estimator for λ and σ.
Solution:

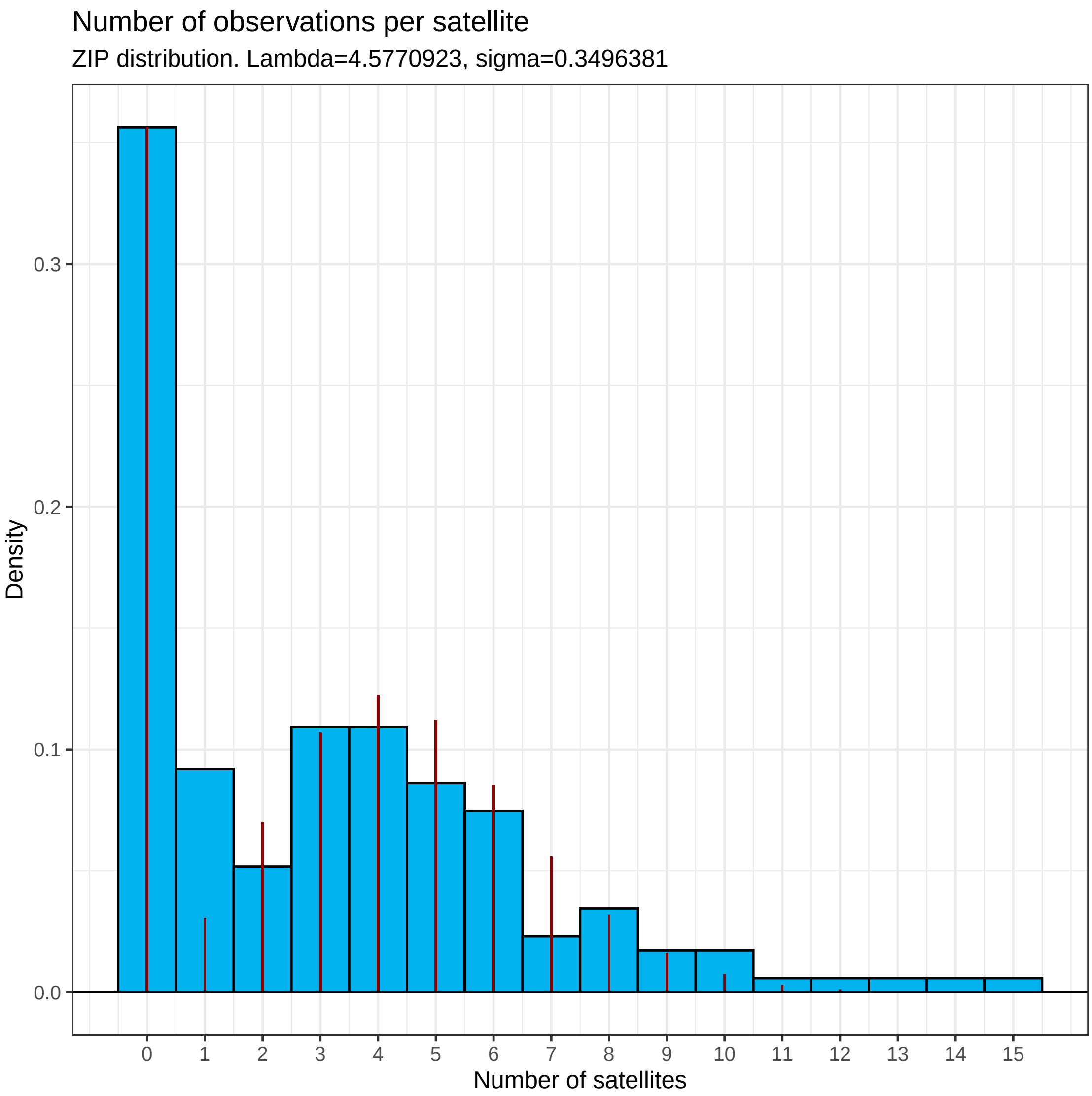
```
#MLE
dpois.ll<-function(par, data){
  lambda <- par[1]
  pi <- par[2]
  ll <- sum(dzipois(x=data, lambda=lambda, pstr0=pi, log=T))
  -ll
}
answerZIPll<-optim(par = c(1,0),
      fn = dpois.ll,
      data=crab.dat$x)
paste("Lambda:",answerZIPll$par[1],"Sigma:",answerZIPll$par[2])
```

```
## [1] "Lambda: 4.57709217492355 Sigma: 0.349638133663682"
```

(f) Plot a histogram of the data with the zero-inflated Poisson distribution fit with the MLE estimates. How well does the distribution fit the data?
Solution: Now, let's assess if we were right about the ultimate superiority of MLE compared to MOM. Yet again, we're going to use VGAM package Yee et al. (2015) to create a true density function.

```
crab.dat.zim <- crab.dat %>%
  mutate(f = dzipois(x=x, lambda=4.5770923, pstr0=0.3496381))

ggplot(crab.dat.zim, aes(x=x))+
  geom_histogram(aes(y=..density..), fill="deepskyblue2", color="black",
                 binwidth=1)+
  geom_linerange(aes(ymin=0, ymax=f), color="dark red")+
  scale_x_continuous("Number of satellites",
                     labels = as.character(sats), breaks = sats)+
  labs(y="Density",
       title="Number of observations per satellite",
       subtitle="ZIP distribution. Lambda=4.5770923, sigma=0.3496381") +
  theme_bw()+
  geom_hline(yintercept=0)
```



Now, that's definitely something to write home about! While I don't really like that this model undervalues some values (look at 1, for example), but it does really well when it comes to zero and most other values!

Case closed: MLE is superior!

2. The time to death for rats injected with a toxic substance, denoted by Y (measured in days), follows an exponential distribution with $\lambda = 1/5$. That is,

$$Y \sim \text{exponential}(\lambda = 1/5).$$

This is the population distribution. It describes the time to death for all individual rats in the population.

The **exponential distribution** serves as a very good model for measurements like waiting times or lifetimes.

$$\lambda \in \mathbb{R}^{+} \qquad \textbf{[Parameters]}$$

$$\mathcal{X} = \{\omega : \omega \in \mathbb{R}^{+}\} \qquad \textbf{[Support]}$$

$$f_X(x|\mu, \sigma) = \lambda e^{-\lambda x}\ I(x \in \mathbb{R}^{+}) \qquad \textbf{[PDF]}$$

$$F_X(x|\mu, \sigma) = \left(1 - e^{-\lambda x}\right) I(x \in \mathbb{R}^{+} \qquad \textbf{[CDF]}$$

$$F_X^{-1}(p|\mu, \sigma) = \frac{-\ln(1-p)}{\lambda} \qquad \textbf{[Inverse CDF]}$$

$$E(X) = \frac{1}{\lambda} \qquad \textbf{[Expected Value]}$$

$$var(X) = \frac{1}{\lambda^2} \qquad \textbf{[Population Variance]}$$

(a) Plot the exponential($\lambda = 1/5$) population distribution.
Solution: I am going to use ggplot2 package Wickham (2016) to plot this exponential distribution.

```
x<-seq(0, 100, 0.1)
y<-dexp(x, 1/5)
ggdat<-data.frame(x,y)

ggplot(ggdat, aes(x=x, y=y))+
  geom_line(color="dark red")+
  labs(y="Density",
  x="x",
  title="Exponential distribution",
  subtitle="Lambda=0.2")+
  theme_bw()
```

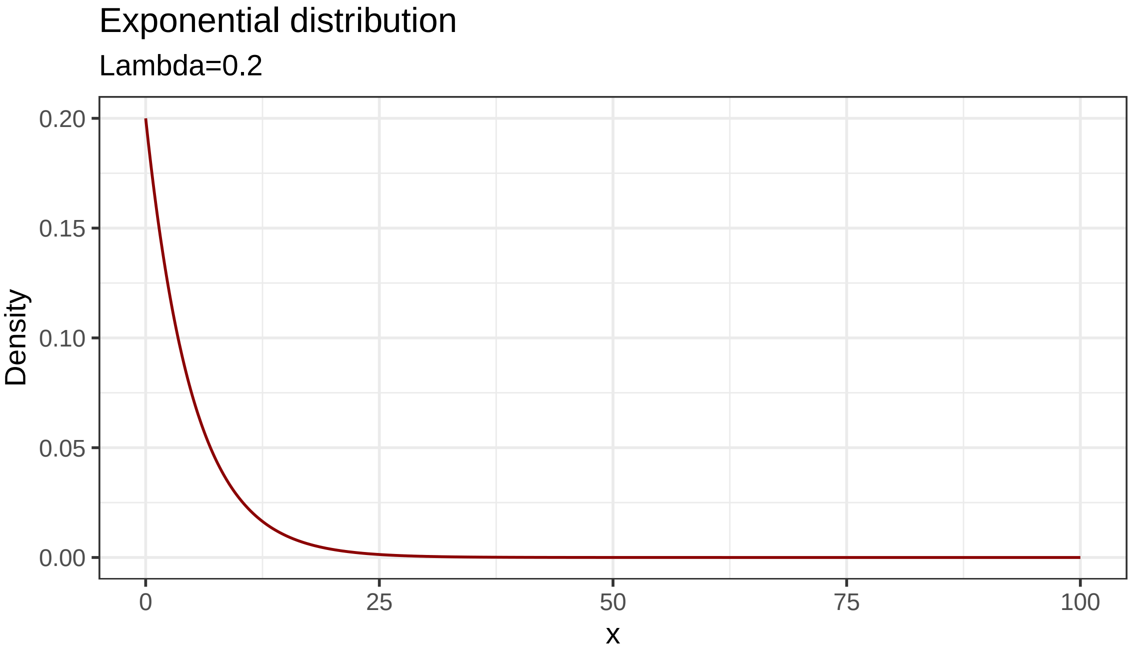


Figure 3: Exponential distribution via Wickham (2016)

(b) Mathematical statisticians can show the exact sampling distributions of $\bar{Y}$ are gamma; i.e.,

$$\bar{Y} \sim \text{gamma}(\alpha = n, \beta = \frac{1}{n\lambda}).$$

The gamma distribution is described below.

$$\begin{aligned} \alpha > 0, \beta > 0 & & \textbf{[Parameters]} \\ \mathcal{X} = (0, \infty) & & \textbf{[Support]} \\ f_X(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta} I(x > 0) & & \textbf{[PDF]} \\ E(X) = \alpha\beta & & \textbf{[Population Mean]} \\ var(X) = \alpha\beta^2 & & \textbf{[Population Variance]} \end{aligned}$$

You can ask `R` for the gamma distribution PDF using `dgamma(x=x,shape=alpha,scale=beta)`. Plot the exact sampling distribution for $n = 2$, $n = 10$, $n = 35$, and $n = 50$.
Solution: Now, in order to plot our gamme distribution, we need to use `dgamma()` function within the base R package! Let's see how the graphs are going to vary based on the changes to n!

```r
library(patchwork)
#plotting distributions based on the info you provided!
ggdat <- data.frame(x1=seq(0, 100, 0.1))%>%
  mutate(y1 = dgamma(x=x1, shape=2, scale=1/(2*0.2)),
         y2 = dgamma(x=x1, shape=10, scale=1/(10*0.2)),
         y3 = dgamma(x=x1, shape=35, scale=1/(35*0.2)),
         y4 = dgamma(x=x1, shape=50, scale=1/(50*0.2)))

#You would ask me:
#Chris, but why do you limit your x from 0 to 30.
#And I would tell you:
#Because it's cool!
plot1<-ggplot(ggdat, aes(x=x, y=y1))+
  geom_line(color="green")+
  xlim(0,30)+
  theme_bw()+
  labs(y="Density",
       title="Gamma distribution for n = 2",
       subtitle="Alpha = 2, Beta = 2.5")

plot2<-ggplot(ggdat, aes(x=x, y=y2))+
  geom_line(color="dark red")+
  xlim(0,30)+
  theme_bw()+
  labs(y="Density",
       title="Gamma distribution for n = 10",
       subtitle="Alpha = 10, Beta = 0.5")

plot3<-ggplot(ggdat, aes(x=x, y=y3))+
  geom_line(color="blue")+
  xlim(0,30)+
  theme_bw()+
  labs(y="Density",
       title="Gamma distribution for n = 35",
       subtitle="Alpha = 35, Beta = 0.14")

plot4<-ggplot(ggdat, aes(x=x, y=y4))+
  geom_line(color="black")+
  xlim(0,30)+
  theme_bw()+
  labs(y="Density",
       title="Gamma distribution for n = 50",
       subtitle="Alpha = 50, Beta = 0.1")

(plot1|plot2)/(plot3|plot4)
```

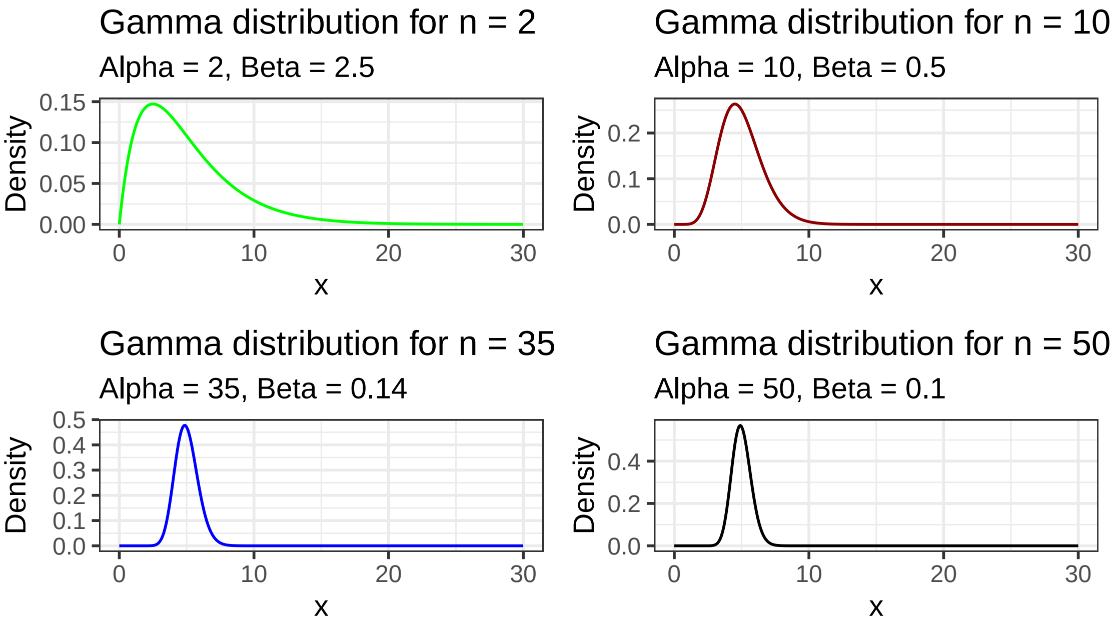


Figure 4: Gamma distributions for $n = 2$, $n = 10$, $n = 35$, and $n = 50$

It seems to me that the bigger n gets, the more normal gamma distribution becomes (yet another proof that CLT is correct!). To prove that I am correct, I am going to superimpose a normal distribution function.

(c) The Central Limit Theorem says that as n increases, the sampling distribution of $\bar{Y}$ can be well approximated with a Gaussian distribution. Superimpose the approximate sampling distribution of $\bar{Y}$ for $n = 2$, $n = 10$, $n = 35$, and $n = 50$.
Solution: So in order to apply CLT here, we need to keep in mind that $\mu_{\bar{x}} = \mu_x$ and $\sigma_{\bar{x}} = sqrt(var(\bar{X}))$. Let's apply it when plotting!

```
plot1 <- plot1+
  geom_function(fun=dnorm, args=list(mean=2*(1/(2*0.2)), sd=sqrt(2*(1/(2*0.2)^2))),
                color="red")

plot2 <- plot2+
  geom_function(fun=dnorm, args=list(mean=10*(1/(10*0.2)), sd=sqrt(10*(1/(10*0.2)^2))),
                color="red")

plot3 <- plot3+
  geom_function(fun=dnorm, args=list(mean=35*(1/(35*0.2)), sd=sqrt(35*(1/(35*0.2)^2))),
                color="red")

plot4 <- plot4+
  geom_function(fun=dnorm, args=list(mean=50*(1/(50*0.2)), sd=sqrt(50*(1/(50*0.2))^2)),
                color="red")

(plot1|plot2)/(plot3|plot4)
```

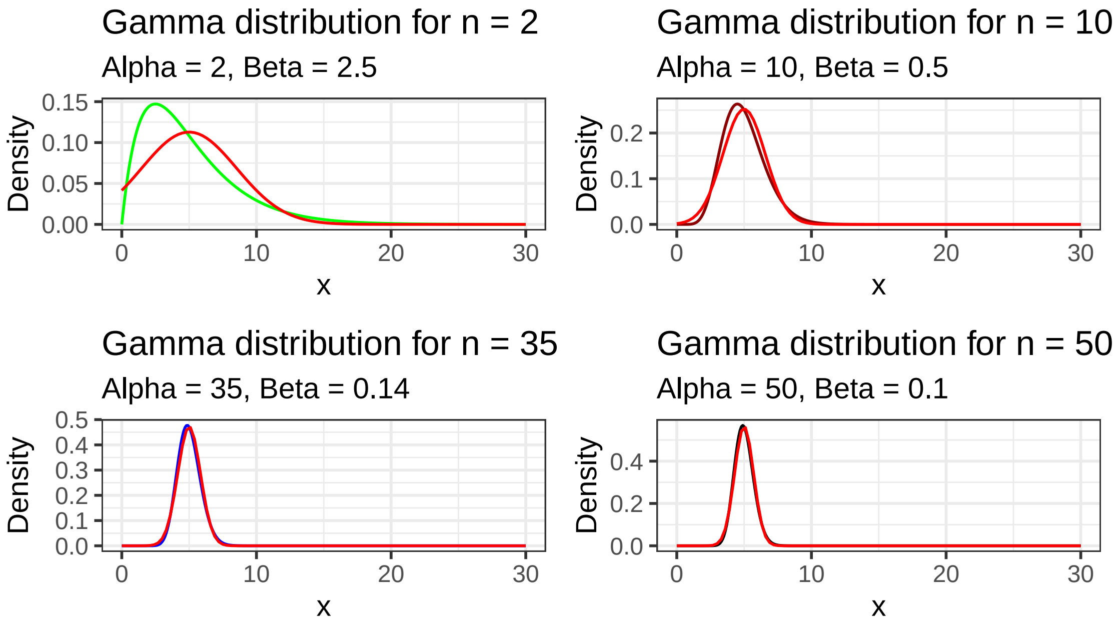


Figure 5: Gamma distributions for $n = 2$, $n = 10$, $n = 35$, and $n = 50$ with superimposed Gaussian distribution

(d) Find the probability that a randomly selected rat injected with the toxic substance lives between 1 and 3 days.
Solution:

```
# 1-P(x<=1)-(1-P(x>3))
1-pgamma(q=1, shape=1, scale=1/0.2)-(1-pgamma(q=3, shape=1, scale=1/0.2))

## [1] 0.2699191
```

(e) Find the exact probability, using the exact sampling distribution, that two randomly selected rats injected with the toxic substance live between 1 and 3 days on average.
Solution:

```
# 1-P(x<=1)-(1-P(x>3))
1-pgamma(q=1, shape=2, scale=1/(2*0.2))-
  (1-pgamma(q=3, shape=2, scale=1/(2*0.2)))

## [1] 0.2758208
```

(f) Find the approximate probability, using the Central Limit Theorem, that two randomly selected rats injected with the toxic substance live between 1 and 3 days on average. Comment on connection between the results and the assumptions of Central Limit Theorem.
Solution:

```
1-pnorm(1, mean=2*(1/(2*0.2)), sd=sqrt(2*(1/(2*0.2)^2)))-
  (1-pnorm(3, mean=2*(1/(2*0.2)), sd=sqrt(2*(1/(2*0.2)^2))))

## [1] 0.1568543
```

We can see that the probability is far off. The explanation is simple: here, we are using only two rats (so $n = 2$), and one of CLT assumptions is that $n > 30$. So no wonder that the result is off!

(g) Under what conditions would the approximate probability calculated in part (f) better match the exact probability in part (e)?
Solution: According to the CLT, the bigger the n, the closer our AG distribution will get to the actual distribution.

3. Below you will load and summarize a dataset containing 575 observations of drug treatments. The data includes the following

   - ID – Identification Code (1 - 575)
   - AGE – Age at Enrollment (Years)
   - BECK – Beck Depression Score (0.000 - 54.000)
   - HC – Heroin/Cocaine Use During 3 Months Prior to Admission (1 = Heroin & Cocaine; 2 = Heroin Only, 3 = Cocaine Only; 4 = Neither Heroin nor Cocaine)
   - IV – History of IV Drug Use (1 = Never; 2 = Previous; 3 = Recent)
   - IV3 – Recent IV use (1 = Yes; 0 = No)
   - NDT – Number of Prior Drug Treatments (0 - 40)
   - RACE – Subject's Race (0 = White; 1 = Non-White)
   - TREAT – Treatment Randomization (0 = Short Assignment; 1 = Long Assignment)
   - SITE – Treatment Site (0 = A; 1 = B)
   - LEN.T – Length of Stay in Treatment (Days Admission Date to Exit Date)
   - TIME – Time to Drug Relapse (Days Measured from Admission Date)
   - CENSOR – Event for Treating Lost to Follow-Up as Returned to Drugs (1 = Returned to Drugs or Lost to Follow-Up; 0 = Otherwise)
   - etc.

   (a) Load the data provided in the "quantreg" package for R (Koenker, 2021).

   ```
   library("quantreg")
   data("uis")
   ```

   (b) Is there evidence that patients receiving drug treatments are at least mildly depressed on average? That is, is there evidence that the average BECK depression score is greater than 13, $\mu > 13$?

      i. What is the null hypothesis for this test?
      Solution: $H_0 : \mu = 13$

      ii. What is the alternative hypothesis for this test?
      Solution: $H_a : \mu > 13$

      iii. What is the sample mean BECK score for these data?
      Solution:

      ```
      mu <- mean(uis$BECK)
      mu0 <- 13
      paste("The mean of BECK is", mean(mu))
      ## [1] "The mean of BECK is 17.367427826087"
      ```

      iv. What is the test statistics for this test?
      Solution: Since we are working with the means here, going for the T-score would be the most reasonable solution, so we are going to use T-test.

      v. At what value of $\bar{X}$ does the rejection region start for $\alpha = 0.05$?

$$se = \frac{s}{\sqrt{n}} \tag{Formula for finding t-value}$$

$$t = \frac{\bar{X} - \mu}{se} \tag{Formula for finding t-value}$$

$$\bar{X} = t * se + \mu \tag{Solving for $\bar{X}$}$$

```
n<-nrow(uis)
se = sd(uis$BECK)/sqrt(n)
(value<-(qt(.95, n-1)*se)+mu0)
```

```
## [1] 13.64123
```

vi. What is the p value for this test?

```
n=nrow(uis)
tstat <- t.test(x=uis$BECK,
       mu = 13,
       alternative = "greater")
paste("P-value:", tstat$p.value)
```

```
## [1] "P-value: 7.41212268149766e-27"
```

vii. Graph the results of this test.
Solution: Since we are using T-test for this, let's create a T-distribution with 574 degrees of freedom. Then, we are going to map our rejection region onto the graph (it's going to be at $t = 1.647513$). Finally, I am going to put our value on X-axis and assess how far away it is from the rejection region (keep in mind the p-value — we already have in mind enough evidence to reject null hypothesis).

```r
ggdat <- data.frame(t=seq(-5,5,length=500))%>%
  mutate(f=dt(x=t, df=n-1))

ggplot(data=ggdat, aes(x=t, y=f))+
  geom_line() +
  geom_hline(yintercept=0)+
  geom_point(aes(x=tstat$statistic, y=0), color="red")+
  geom_vline(xintercept=qt(p=0.95, df=n-1),
             linetype="dashed", color="red")+
  annotate("text", x=1.2*qt(p=0.95, df=n-1), y=0.30,
           label="Rejection Region", angle="270",
           color="red")+
  annotate("text", x=tstat$statistic-1.8, y=0.02,
           label="P-value = 7.41 * 10^-27",
           size=3)+
  theme_bw()+
  labs(x="T",
       y="Density",
       title="T-distribution to assess a null hypothesis")
```

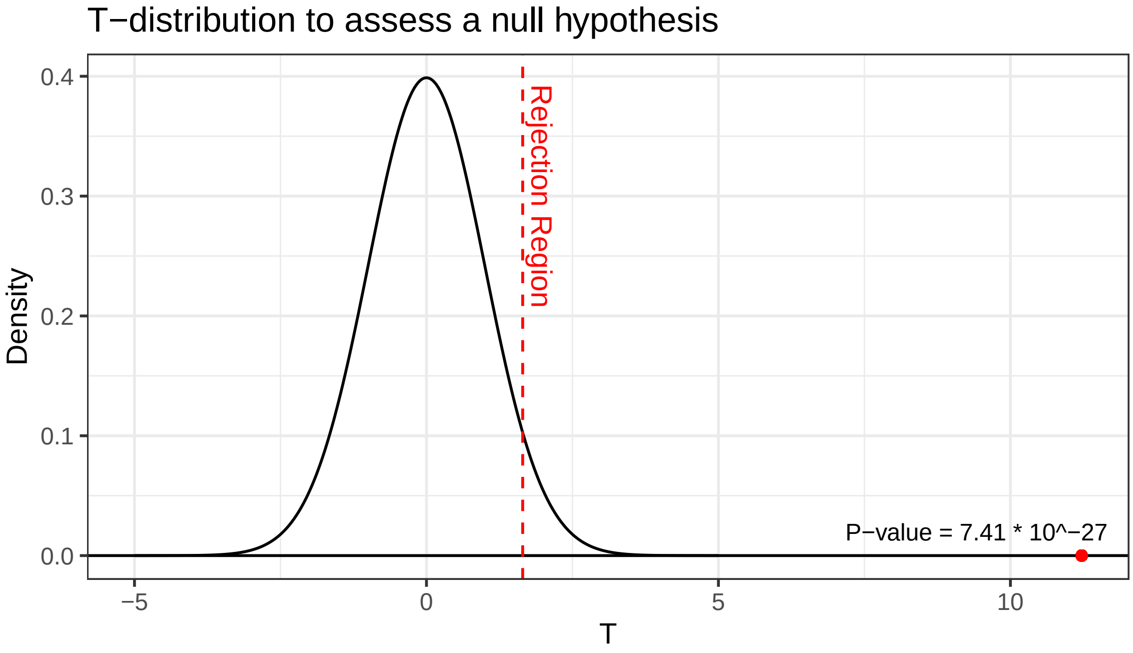


Figure 6: T-distribution with 574 degrees of freedom

viii. Report a 95% confidence interval for the average BECK depression score and interpret it in the context of this question.
Solution:

```r
ci<-t.test(x=uis$BECK,
       alternative = "two.sided")

paste("The confidence interval is [", ci$conf.int[1], ", ",
      ci$conf.int[2],"]", sep="")
## [1] "The confidence interval is [16.6029755170859, 18.131880135088]"
```

(c) Is there a significant difference in the length of stay in treatment by treatment site?
Solution: First, let's establish our hypothesis.

$$H_0 : \mu_0 - \mu 1 = 0$$
$$H_a : \mu_0 - \mu 1 \neq 0$$

```
res <- t.test(uis$LEN.T ~ uis$SITE,
              data = uis, var.equal = F)
paste("P-value:",res$p.value)

## [1] "P-value: 4.07192587806883e-09"
```

Based on this two-sample t-test, we have enough evidence to suggest that (with 95% confidence) our true mean is not equal to 0! Therefore, we have statistically significant results showing that there's a difference in the length of stay between two sites!

4. Below you will load and summarize a dataset containing 53 observations of prostate cance patients. In this research, a number of possible predictor variables were measured before surgery. The patients then had surgery to determine nodal involvement.

   - r – Nodal Involvement (0=No, 1=Yes)
   - aged – Age Group (0=Less than 60, 1=At least 60)
   - stage – Palpitation Result Severity (0=Less severe, 1=More severe)
   - grade – Biopsy Result Severity (0=Less severe, 1=More severe)
   - xray – X-ray Result Severity (0=Less severe, 1=More severe)
   - acid – the level of acid phosphatase in the blood serum

   The treatment strategy for a patient diagnosed with cancer of the prostate depend highly on whether the cancer has spread to the surrounding lymph nodes (nodal involvement). It is common to operate on the patient to get samples from the nodes which can then be analysed under a microscope.

   (a) Load the data provided in the "boot" package for `R` (Canty and Ripley, 2021).

   ```
   library("boot")
   data("nodal")
   ```

   (b) Is there evidence that less than half of prostate cancer patients have nodal involvment?

      i. What is the null hypothesis for this test?
      Solution: $H_0 : \hat{P} = 0.5$

      ii. What is the alternative hypothesis for this test?
      Solution: $H_0 : \hat{P} < 0.5$

      iii. What is the sample proportion of patients with nodal involvement for these data?
      Solution: It's 37%!

      ```
      table(nodal$r)[2]/nrow(nodal) #37%
      ```

      ```
      ##         1
      ## 0.3773585
      ```

      iv. What is the test statistics for this test?
      Solution: We are working with proportions, so we are going to use a Z-value.

      v. At what value of $\hat{P}$ does the rejection region start for $\alpha = 0.05$?
      Solution:

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

$$\hat{p} = \frac{\sqrt{n} z \sqrt{p_0(1-p_0)}}{n} + p_0$$

      ```
      p0 <- 0.5
      phat <- table(nodal$r)[2]/nrow(nodal)
      n <- nrow(nodal)
      z1 <- qnorm(.05, 0, 1)

      answerProp<-(sqrt(n)*z1*sqrt(p0*(1-p0)))/(n) + p0

      paste("Rejection starts with", answerProp, "but we already have a lower number:",
            table(nodal$r)[2]/nrow(nodal))
      ```

      ```
      ## [1] "Rejection starts with 0.38703098909445 but we already have a lower number: 0.377358
      ```

      Having a number lower than the rejection region is the first sign that true $\hat{P}$ is not 0.5!

vi. What is the p value for this test?

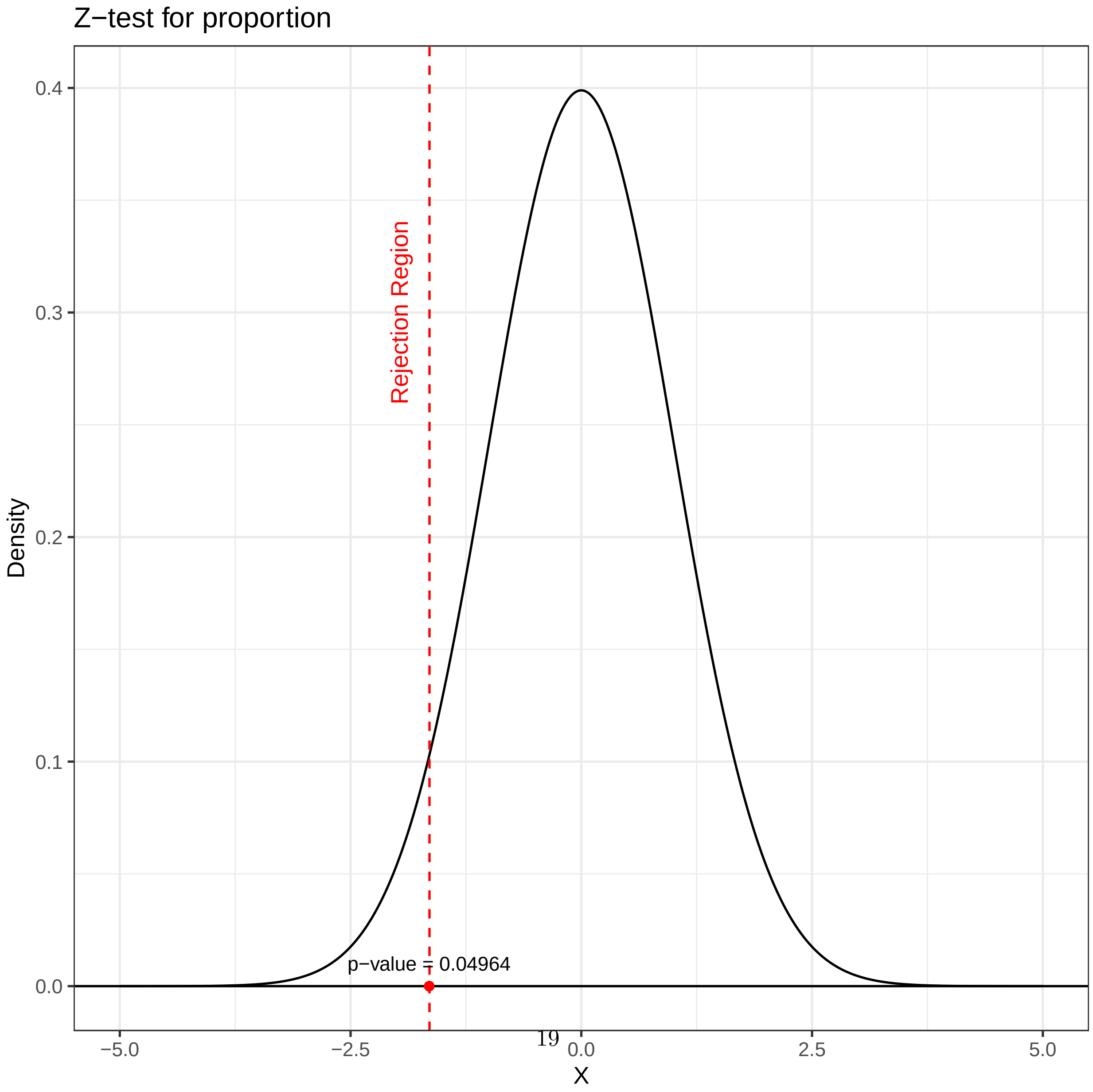
```r
answerPropTest <- prop.test(20, 53, p=0.5, alternative="less")
paste("P-value:",answerPropTest$p.value)
## [1] "P-value: 0.0496428173856788"
```

We have enough evidence to reject null hypothesis in favor of the alternative!

vii. Graph the results of this test.

```r
ggdat <- data.frame(t=seq(-5,5,length=500))%>%
  mutate(f=dnorm(x=t, mean=0, sd=1))

ggplot(data=ggdat, aes(x=t, y=f))+
  geom_line() +
  geom_hline(yintercept=0)+
   geom_point(aes(x=-sqrt(answerPropTest$statistic), y=0), color="red")+
  geom_vline(xintercept=qnorm(p=0.05, mean=0, sd=1),
               linetype="dashed", color="red")+
  annotate("text", x=1.2*qnorm(p=0.05, mean=0, sd=1), y=0.30,
           label="Rejection Region", angle="90",
           color="red")+
  annotate("text", -sqrt(answerPropTest$statistic), y=0.01,
           label="p-value = 0.04964",
           size=3)+
  theme_bw()+
  labs(x="X",
       y="Density",
       title="Z-test for proportion")
```



It's one of those times when it's really hard to tell by the graph alone if the dot within the rejection region or outside of it. So, as I've done before, we can use `prop.test()` in order to check the p-value!

viii. Report a 95% confidence interval for the proportion of prostate cancer patients with nodal involvement and interpret it in the context of this question.

```
propCI<-prop.test(20, 53, alternative="two.sided")
paste("Confidence interval is", propCI$conf.int)

## [1] "Confidence interval is 0.251167223511435"
## [2] "Confidence interval is 0.521281347288513"
```

(c) Clearly, it would be preferable if an accurate assessment of nodal involvement could be made without surgery. Is there a significant difference in the nodal involvement of patients with any of the severity indicators?

Solution:

$H_0 : \hat{p}_1 - \hat{p}_2 = 0$

$H_0 : \hat{p}_1 - \hat{p}_2 \neq 0$

```
#H0: p1-p2 = 0
#Ha: p1-p2 != 0
nodal.c <- nodal %>%
  mutate(sev = case_when(stage==1 ~ 1,
                         grade==1 ~ 1,
                         xray==1 ~ 1,
                         TRUE ~ 0)) #put all severity indicators into one
#non severe
x1<-1
n1<-16
#Non severe: 1 involvement out of 16 cases

#severe
x2<-19
n2<-37
#Severe: 19 involvements out of 37 cases

finalProp <- prop.test(x=c(x1,x2),
          n=c(n1,n2))
paste("P-value of two-sample test is", finalProp$p.value)

## [1] "P-value of two-sample test is 0.00509373051672405"
```

Based on the p-value, we have enough evidence to reject the null hypothesis in favor of alternative. Therefore, there's actually a significant in the nodal involvement in patients with severity indicators.

Bonus 1: Use the gganimate package (Pedersen and Robinson, 2020) for `R` to create a plot that demonstrates the Central Limit Theorem for the Poisson, Binomial, Exponential, and Gaussian distributions in a 2×2 grid as here.

Note that we can't add GIFs to the .pdf document, so you'll have to email me your code for this part. You'll find `transition_time()` helpful for creating your animation and `gganimate_save()` helpful for saving your animation.

Solution:

```
library(gganimate)
library(gifski)
library(transformr)

#Binom
#E(X)=np
#Var(X)=np*(1-p)
#100*0.5*(1-0.5)

final1.df <- c()
for(num in 1:150){
  generate <- rnorm(num, mean=num*0.3, sd=sqrt(num*0.3*0.7))
  num.df <- data.frame(values=generate, n=num)
  final1.df<-bind_rows(final1.df, num.df)
}
p1<-ggplot(final1.df, aes(x=values))+
  geom_histogram(aes(y=..density..),
                 color="white",
                 fill="dark red")+
  geom_function(fun=dnorm, args=list(mean=num*0.3, sd=sqrt(num*0.3*0.7)),
                color="red")+
  labs(x="X",
       y="Density",
       title="Poisson distribution",
       subtitle="Lambda: 5")+
  transition_states(n,
                    transition_length = 1,
                    state_length = 1)+
  ease_aes(’sine-in-out’)

anim1 <- animate(p1, renderer = gifski_renderer())
anim_save("poisson.gif", anim1)


#Poisson
#E(X)=lambda
#Var(X)=lambda
final2.df <- c()
for(num in 1:150){
  generate <- rnorm(num, mean=5, sd=sqrt(5))
  num.df <- data.frame(values=generate, n=num)
  final2.df<-bind_rows(final2.df, num.df)
}
```

```
p2<-ggplot(final2.df, aes(x=values))+
  geom_histogram(aes(y=..density..),
                 color="white",
                 fill="dark red")+
  geom_function(fun=dnorm, args=list(mean=5, sd=sqrt(5)),
                color="red")+
  labs(x="X",
       y="Density",
       title="Poisson distribution",
       subtitle="Lambda: 5")+
  transition_states(n,
                    transition_length = 1,
                    state_length = 1)+
  ease_aes(’sine-in-out’)

anim2 <- animate(p2, renderer = gifski_renderer())
anim_save("poisson.gif", anim2)


#Exponential
#E(X)=1/lambda
#Var(X)=1/lambda^2

final3.df <- c()
for(num in 1:150){
  generate <- rnorm(num, mean=1/5, sd=sqrt(1/25))
  num.df <- data.frame(values=generate, n=num)
  final3.df<-bind_rows(final3.df, num.df)
}

p3<-ggplot(final3.df, aes(x=values))+
  geom_histogram(aes(y=..density..),
                 color="white",
                 fill="dark red")+
  geom_function(fun=dnorm, args=list(mean=1/5, sd=sqrt(1/25)),
                color="red")+
  labs(x="X",
       y="Density",
       title="Exponential distribution",
       subtitle="Lambda: 5")+
transition_states(n,
                    transition_length = 1,
                    state_length = 1)+
  ease_aes(’sine-in-out’)

anim3 <- animate(p3, renderer = gifski_renderer())
anim_save("exponential.gif", anim3)

#Normal
#E(X)=mu
#Var(X)=sigma^2
final4.df <- c()
for(num in 1:150){
  generate <- rnorm(num, mean=0, sd=1)
  num.df <- data.frame(values=generate, n=num)
  final4.df<-bind_rows(final4.df, num.df)
}
p4<-ggplot(final4.df, aes(x=values))+
  geom_histogram(aes(y=..density..),
                 color="white",
                 fill="dark red")+
  geom_function(fun=dnorm, args=list(mean=1/5, sd=sqrt(1/25)),
                color="red")+
  labs(x="X",
       y="Density",
       title="Normal distribution",
       subtitle="Mean: 0, SD: 1")+
  transition_states(n,
                    transition_length = 1,
                    state_length = 1)+
  ease_aes('sine-in-out')

anim4 <- animate(p4, renderer = gifski_renderer())
anim_save("normal.gif", anim4)
```

Bonus 2: Compare the effectiveness of the t-interval with the bootstrap interval. In a loop, generate 1000 datasets, evaluate a t and bootstrapping confidence interval for each set of data, and track whether you've captured the true population mean. An effective answer here would evaluate this several times varying sample size and the data generating distribution.

References

Brockmann, H. J. (1996). Satellite male groups in horseshoe crabs, limulus polyphemus. *Ethology*, 102(1):1–21.

Canty, A. and Ripley, B. D. (2021). *boot: Bootstrap R (S-Plus) Functions.* R package version 1.3-28.

cardinal (2011). Mean and variance of a zero-inflated poisson distribution.

Hasselman, B. (2018). *nleqslv: Solve Systems of Nonlinear Equations.* R package version 3.3.2.

Koenker, R. (2021). *quantreg: Quantile Regression.* R package version 5.86.

Pedersen, T. L. and Robinson, D. (2020). *gganimate: A Grammar of Animated Graphics.* R package version 1.0.7.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York.

Yee, T. W., Stoklosa, J., and Huggins, R. M. (2015). The vgam package for capture-recapture data using the conditional likelihood. *Journal of Statistical Software*, 65(5):1–33.