COMPSCIX 415.2 Homework 6

Sanatan Das

March 9, 2018

Contents

Code and Documents Git Repository

All the work can be found in the below Git repository location:

https://github.com/sanatanonline/compscix-415-2-assignments

Load packages (prerequisites to run the code in this document)

```
library(mosaicData)
library(tidyverse)
```

Analysis of Whickham dataset

Load the Whickham dataset (data(Whickham)). You will need to load the mosaicData package first, but I also included the data as an rds file on Canvas if you would rather download it there and load it with readRDS().

Look at the help file on this dataset to learn a bit about it.

1. What variables are in this data set?
2. How many observations are there and what does each represent?
3. Create a table (use the R code below as a guide) and a visualization of the relationship between smoking status and outcome, ignoring age. What do you see? Does it make sense?

```
library(mosaicData)
library(tidyverse)
Whickham %>% count( _________ , _________ )
```

4. Recode the age variable into an ordered factor with three categories: age <= 44, age > 44 & age <= 64, and age > 64. Now, recreate visualization from above, but facet on your new age factor. What do you see? Does it make sense?

Answer

```
# load Whickham.rds
whickham = readRDS("C:/view/opt/apps/git/R/compscix-415-2-assignments/Whickham.rds")

# glimpse whickham
glimpse(whickham)
```

```
## Observations: 1,314
## Variables: 3
## $ outcome <fct> Alive, Alive, Dead, Alive, Alive, Alive, Alive, Dead, ...
## $ smoker  <fct> Yes, Yes, Yes, No, No, Yes, Yes, No, No, No, No, Yes, ...
## $ age     <int> 23, 18, 71, 67, 64, 38, 45, 76, 28, 27, 28, 34, 20, 72...
```

1. There are three variables in the dataset: *outcome, smoker, age.*
2. This dataset has total 1314 observations. Each observation represents the survival status (Alive/Dead) of smoker or non-smoker women with some particular age in years.

- **outcome** survival status: a factor with levels Alive or Dead
- **smoker** smoking status at baseline: a factor with levels No or Yes

- **age** age (in years)

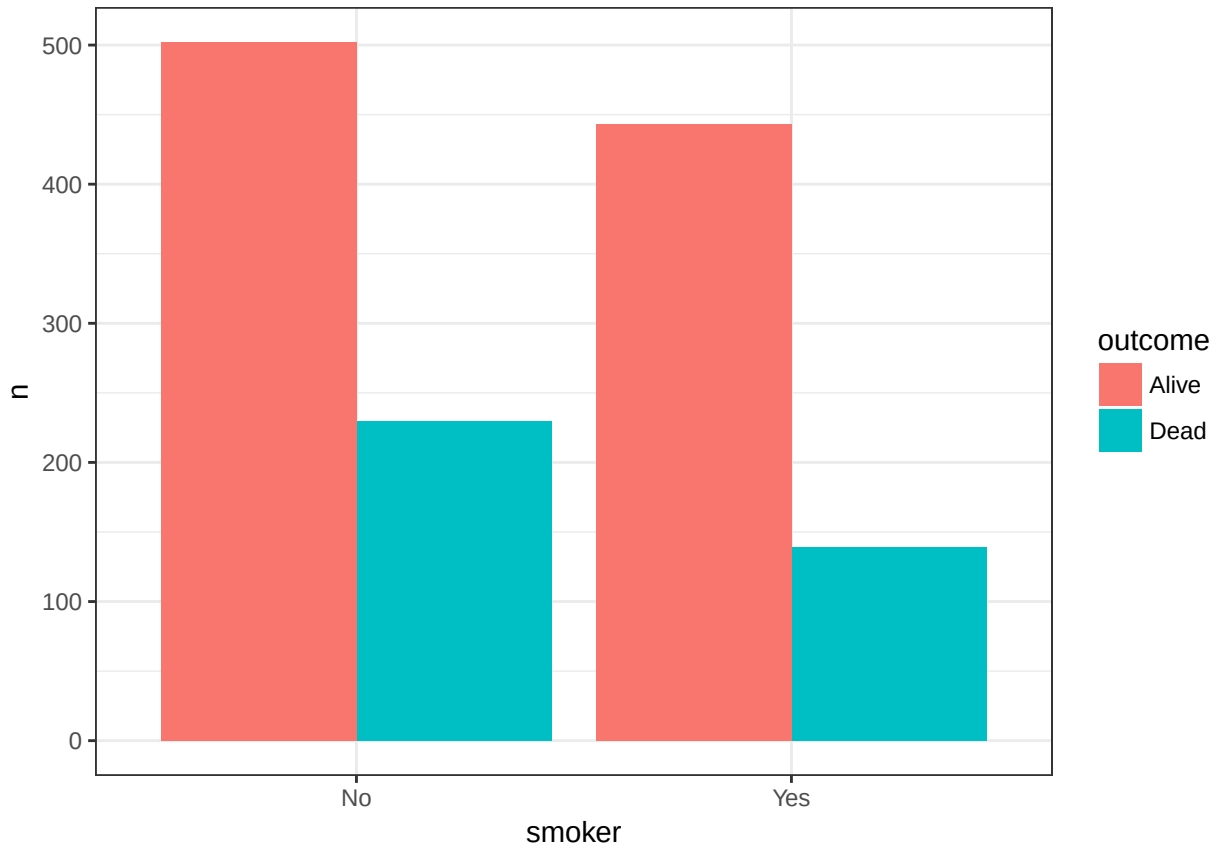
3. Here is the relationship between smoking status and outcome.

```
Whickham %>% count( smoker , outcome)
```

```
## # A tibble: 4 x 3
##   smoker outcome     n
##   <fct>  <fct>   <int>
## 1 No     Alive     502
## 2 No     Dead      230
## 3 Yes    Alive     443
## 4 Yes    Dead      139
```

Let's plot a bar graph on above data.

```
Whickham %>%
  count( smoker , outcome) %>%
  ggplot() +
  geom_bar(aes(x = smoker, y = n, fill = outcome), stat = 'identity', position = 'dodge') +
  theme_bw()
```



It looks like smoking status has not much impact on the outcome. Even more women died although they did not smoke.

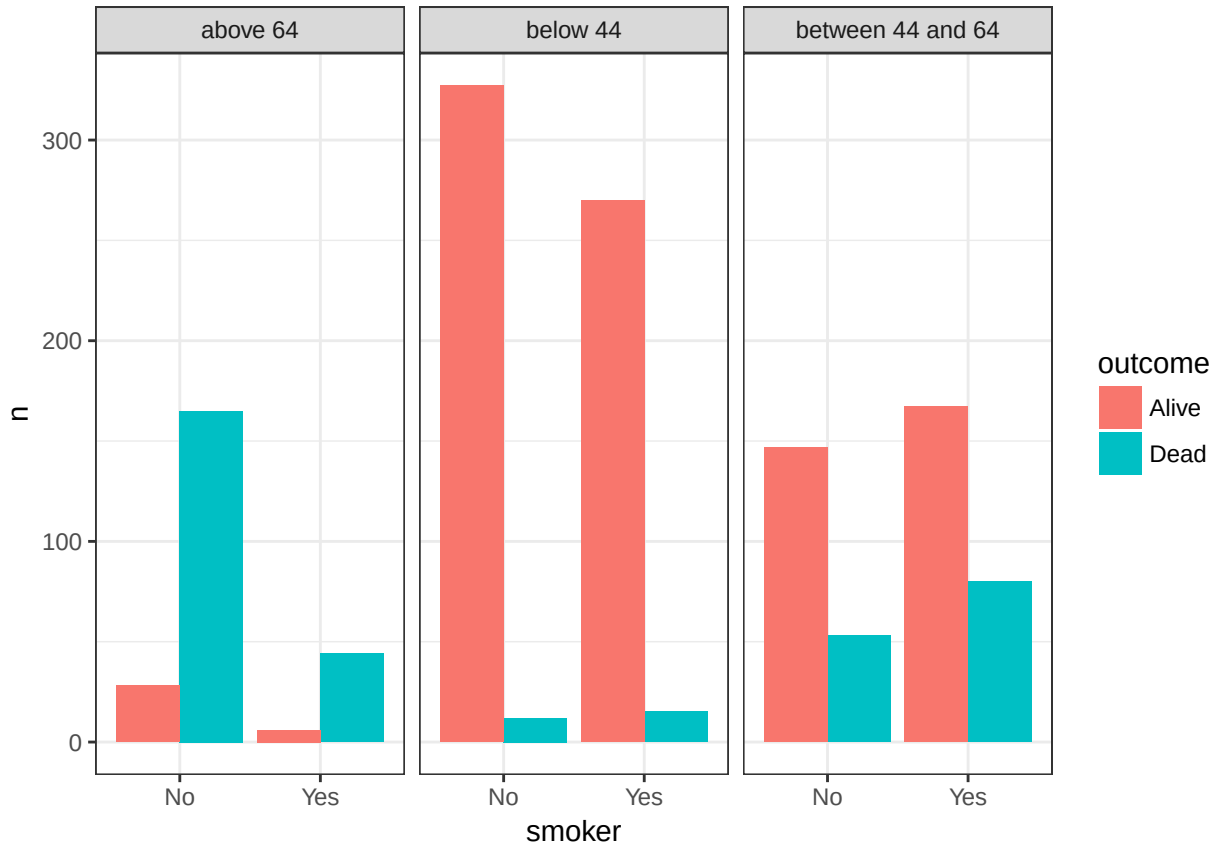
4. Let's recode the age variable with three categories: age <= 44, 44 < age <= 64, and age > 64 and plot again.

```
Whickham %>%
  mutate(age_cat = case_when(age <= 44 ~ 'below 44',
```

```
                           age > 44 & age <= 64 ~ 'between 44 and 64',
                           age > 64 ~ 'above 64')) %>%
  count( smoker , outcome, age_cat) -> whickham_categorized

# Print the categorized table
arrange(whickham_categorized, desc(age_cat))
```

```
## # A tibble: 12 x 4
##    smoker outcome age_cat              n
##    <fct>  <fct>   <chr>            <int>
##  1 No     Alive   between 44 and 64   147
##  2 No     Dead    between 44 and 64    53
##  3 Yes    Alive   between 44 and 64   167
##  4 Yes    Dead    between 44 and 64    80
##  5 No     Alive   below 44            327
##  6 No     Dead    below 44             12
##  7 Yes    Alive   below 44            270
##  8 Yes    Dead    below 44             15
##  9 No     Alive   above 64             28
## 10 No     Dead    above 64            165
## 11 Yes    Alive   above 64              6
## 12 Yes    Dead    above 64             44
```

```
whickham_categorized %>%
  ggplot() +
  geom_bar(aes(x = smoker, y = n, fill = outcome), stat = 'identity', position = 'dodge') +
  facet_wrap(~ age_cat) +
  theme_bw()
```



Now, we see that the death rate is more for the age category between 44 and 64 compared to other two actegories. At the same time, above 64 category, the death count is higher even for non-smokers.

Analysis of The Gamma Distribution

The Central Limit Theorem states that the sampling distribution of sample means is approximately Normal, regardless of the distribution of your population. For this exercise our population distribution will be a Gamma(1,2) distribution, and we'll show that the sampling distribution of the mean is in fact normally distributed.

1. Generate a random sample of size n = 10000 from a gamma(1,2) distribution and plot a histogram or density curve. Use the code below to help you get your sample.

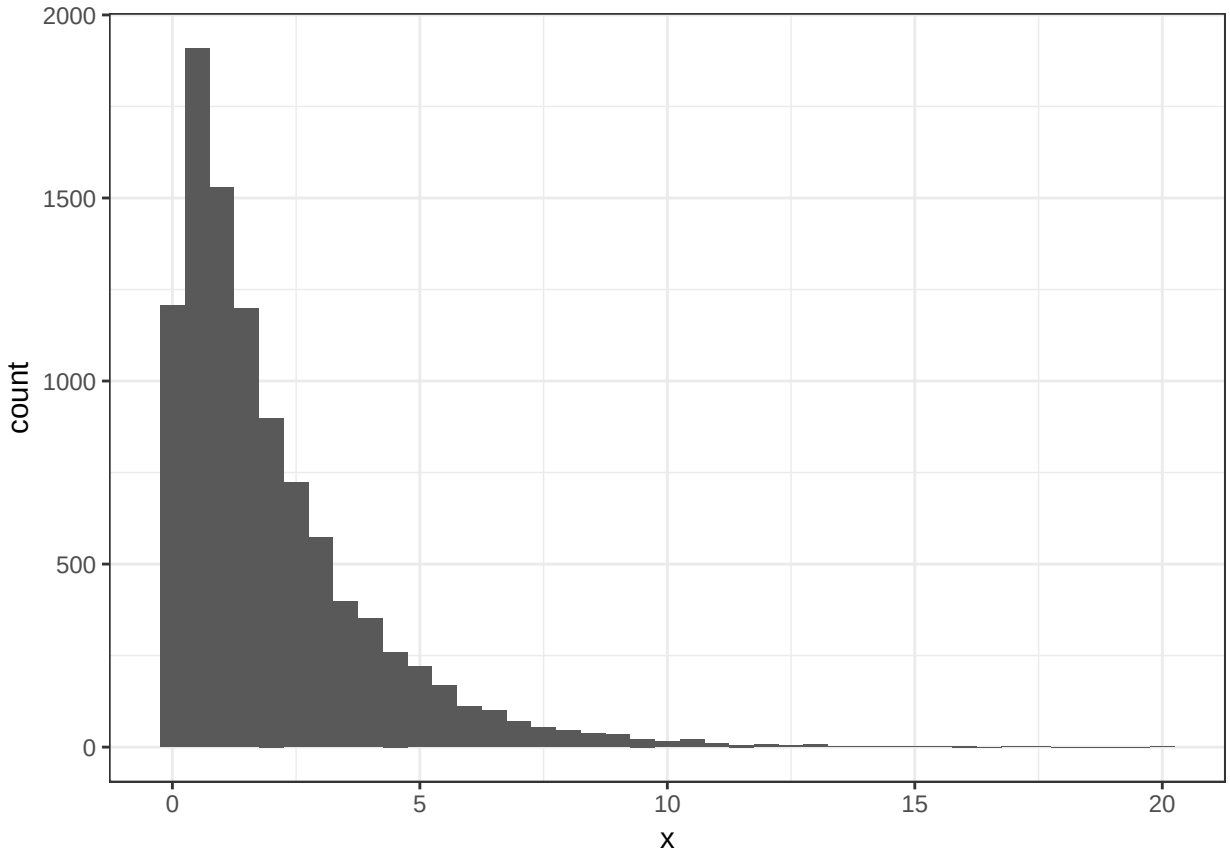
```
n <- 10000
# look at ?rgamma to read about this function
gamma_samp <- tibble(x = rgamma(n, shape = 1, scale = 2))
```

Answer

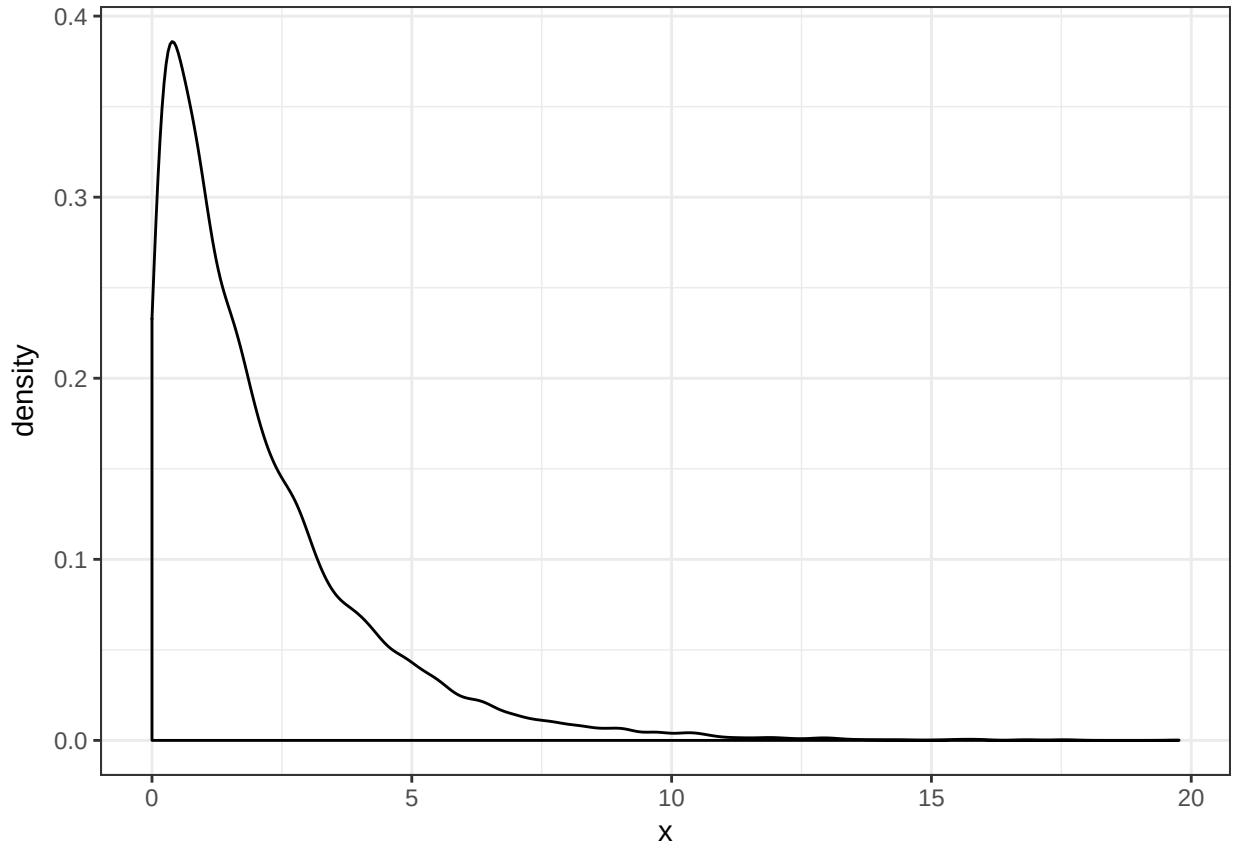
Let's plot the graph.

```
n <- 10000
# look at ?rgamma to read about this function
gamma_samp <- tibble(x = rgamma(n, shape = 1, scale = 2))

# Histogram
ggplot(data=gamma_samp) +
```

```
  geom_histogram(aes(x = x), binwidth = 0.5) +
  theme_bw()
```




```
# Density curve
ggplot(data=gamma_samp) +
  geom_density(aes(x = x)) +
  theme_bw()
```



2. What is the mean and standard deviation of your sample? They should both be close to 2 because for a gamma distribution: mean = shape x scale variance = shape x scale^2

```
mean_samp <- gamma_samp %>% .[['x']] %>% mean()
```

Answer

```
n <- 10000
gamma_samp <- tibble(x = rgamma(n, shape = 1, scale = 2))

mean_samp <- gamma_samp %>% .[['x']] %>% mean()
mean_samp
```

```
## [1] 2.014898
```

```
sd_samp <- gamma_samp %>% .[['x']] %>% sd()
sd_samp
```

```
## [1] 2.001393
```

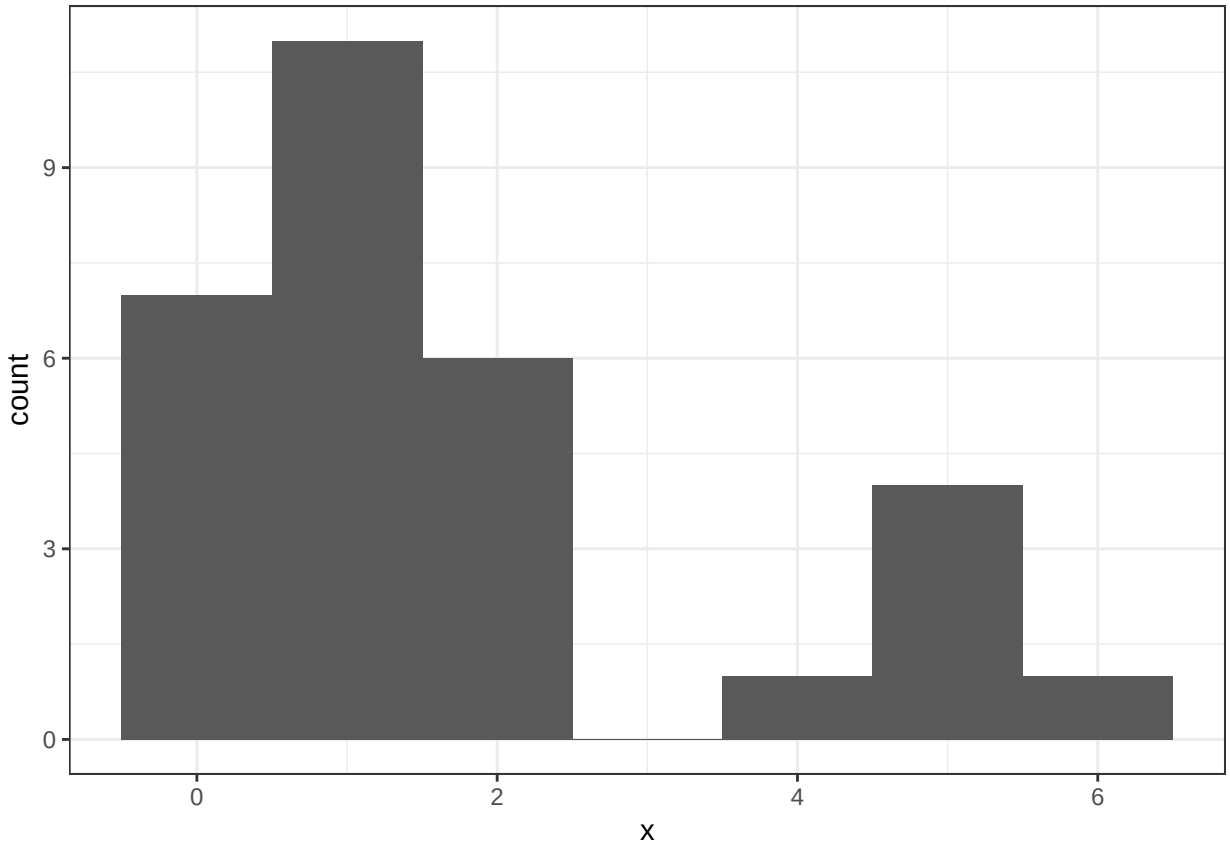
True, both mean and standard deviation are close to 2.

3. Pretend the distribution of our population of data looks like the plot above. Now take a sample of size n = 30 from a Gamma(1,2) distribution, plot the histogram or density curve, and calculate the mean and standard deviation.

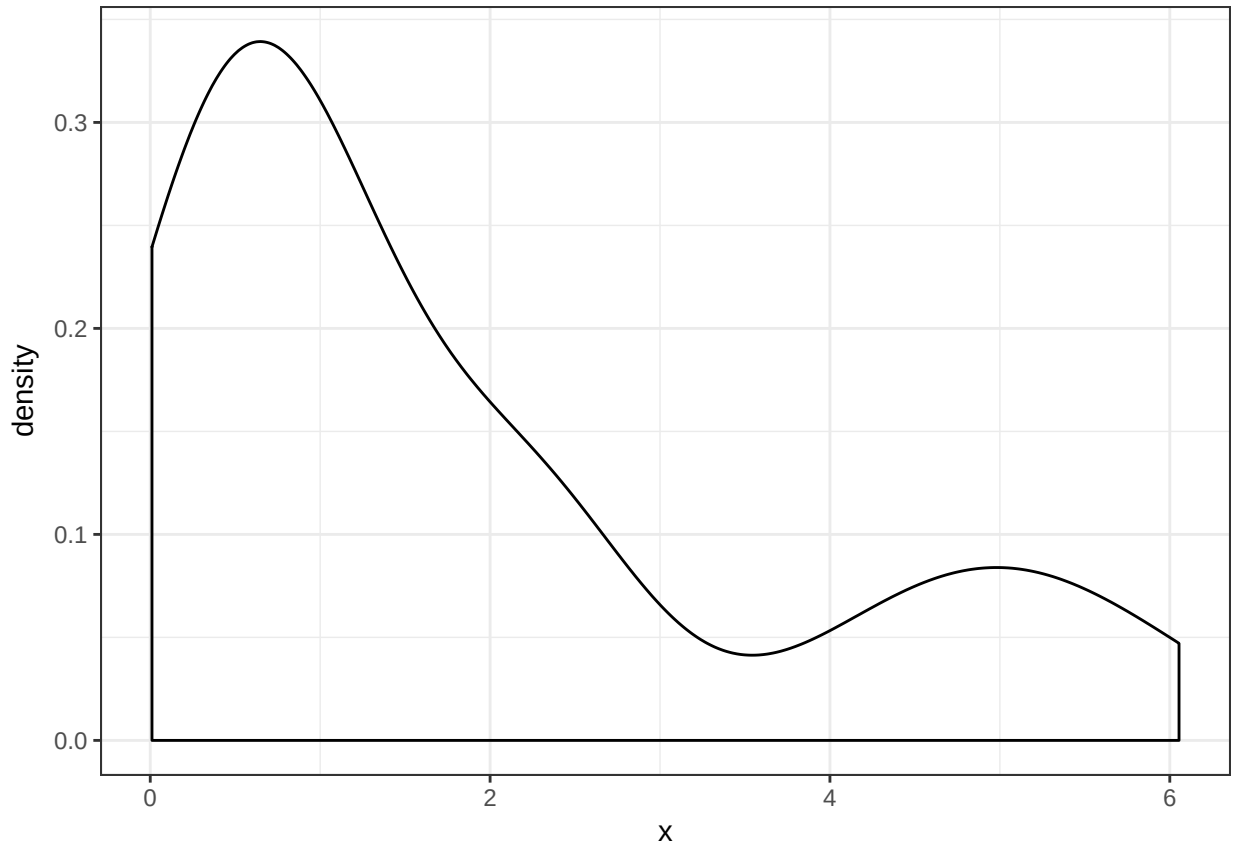
Answer

```
n <- 30
gamma_samp <- tibble(x = rgamma(n, shape = 1, scale = 2))
```

```
ggplot(data=gamma_samp) +
  geom_histogram(aes(x = x), binwidth = 1) +
  theme_bw()
```




```
ggplot(data=gamma_samp) +
  geom_density(aes(x = x)) +
  theme_bw()
```




```
mean_samp <- gamma_samp %>% .[['x']] %>% mean()
mean_samp
```

```
## [1] 1.798704
```

```
sd_samp <- gamma_samp %>% .[['x']] %>% sd()
sd_samp
```

```
## [1] 1.796487
```

4. Take a sample of size n = 30, again from the Gamma(1,2) distribution, calculate the mean, and assign it to a vector named mean_samp. Repeat this 10000 times!!!! The code below might help.

```
# create a vector with 10000 NAs
mean_samp <- rep(NA, 10000)

# start a loop
for(i in 1:10000) {
g_samp <- rgamma(30, shape = 1, scale = 2)
mean_samp[i] <- mean(g_samp)
}

#Convert vector to a tibble
mean_samp <- tibble(mean_samp)
```

Answer

Let's execute the code.

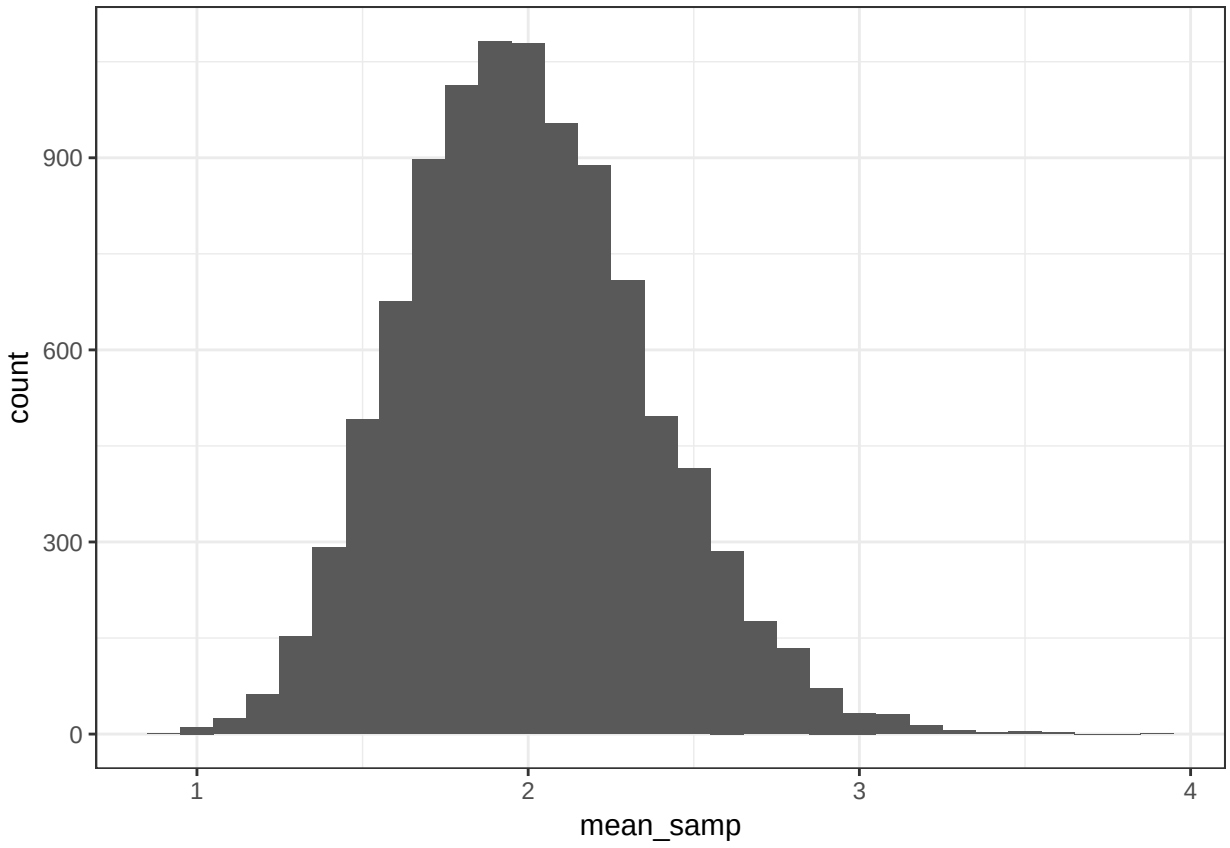
```r
# create a vector with 10000 NAs
mean_samp <- rep(NA, 10000)

# start a loop
for(i in 1:10000) {
g_samp <- rgamma(30, shape = 1, scale = 2)
mean_samp[i] <- mean(g_samp)
}

#Convert vector to a tibble
mean_samp <- tibble(mean_samp)

mean_samp
```

```
## # A tibble: 10,000 x 1
##    mean_samp
##        <dbl>
##  1      2.27
##  2      2.23
##  3      2.01
##  4      1.55
##  5      2.56
##  6      2.44
##  7      1.91
##  8      2.35
##  9      2.64
## 10      1.82
## # ... with 9,990 more rows
```

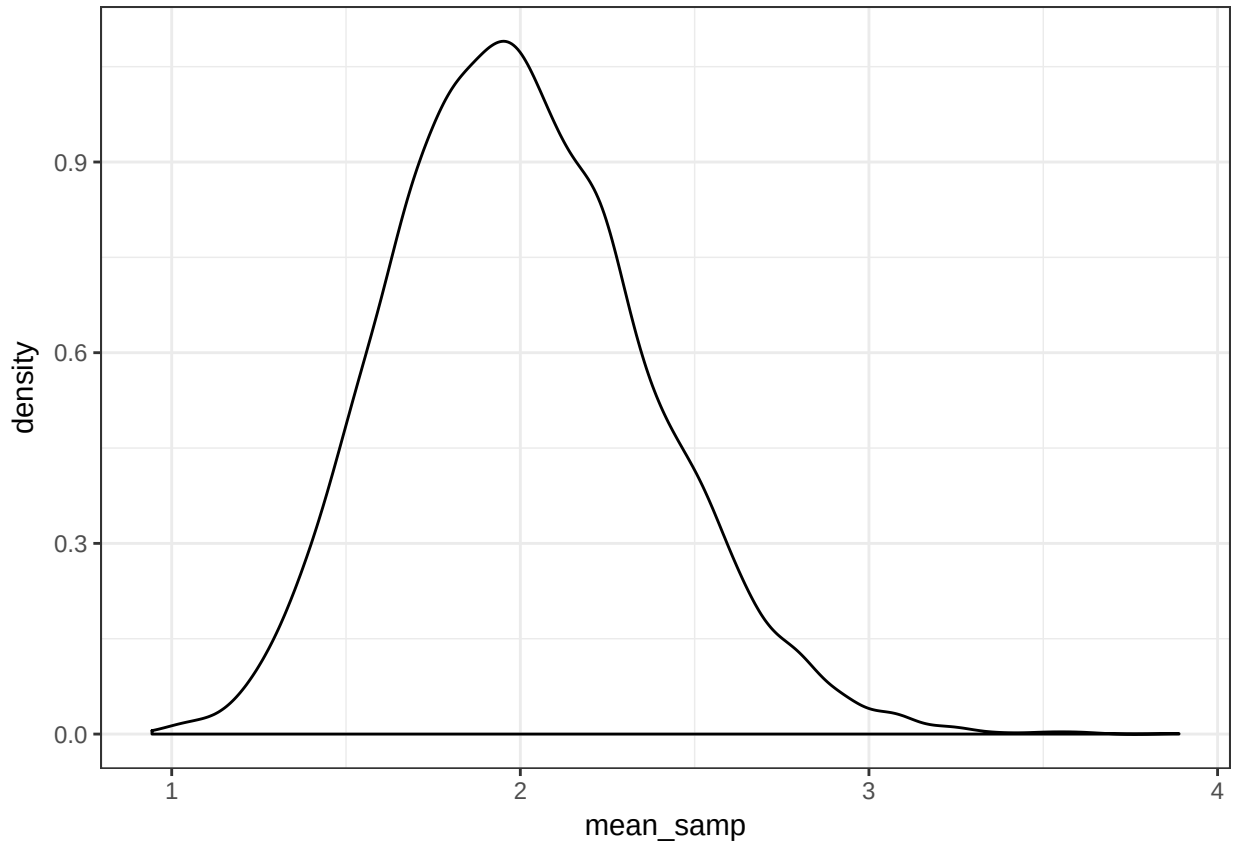
5. Make a histogram of your collection of means from above (mean_samp).

Answer

```r
# create a vector with 10000 NAs
mean_samp <- rep(NA, 10000)

# start a loop
for(i in 1:10000) {
g_samp <- rgamma(30, shape = 1, scale = 2)
mean_samp[i] <- mean(g_samp)
}

#Convert vector to a tibble
mean_samp <- tibble(mean_samp)

ggplot(data=mean_samp) +
  geom_histogram(aes(x = mean_samp), binwidth = 0.1) +
  theme_bw()
```




```
ggplot(data=mean_samp) +
  geom_density(aes(x = mean_samp)) +
  theme_bw()
```



6. Calculate the mean and standard deviation of all of your sample means.

Answer

```
# create a vector with 10000 NAs
mean_samp <- rep(NA, 10000)

# start a loop
for(i in 1:10000) {
g_samp <- rgamma(30, shape = 1, scale = 2)
mean_samp[i] <- mean(g_samp)
}

mean_of_mean_samp <- mean(mean_samp)
mean_of_mean_samp
```

```
## [1] 1.999661
```

```
sd_of_mean_samp <- sd(mean_samp)
sd_of_mean_samp
```

```
## [1] 0.3665746
```

7. Did anything surprise you about your answers to #6?

Answer

Yes, when we took a 10000 mean and calculated mean and standard deviation, mean remain same (~2) but standard deviation decreases (makes sense).

8. According to the Central Limit Theorem, the mean of your sampling distribution should be very close to 2, and the standard deviation of your sampling distribution should be close to 0.365. Repeat #4-#6, but now with a sample of size n = 300 instead. Do your results match up well with the theorem?

Answer

Let's find the mean and standard deviation for n=300

```
# create a vector with 10000 NAs
mean_samp <- rep(NA, 10000)

# start a loop
for(i in 1:10000) {
g_samp <- rgamma(300, shape = 1, scale = 2)
mean_samp[i] <- mean(g_samp)
}

mean_of_mean_samp <- mean(mean_samp)
mean_of_mean_samp
```

```
## [1] 1.998334
```

```
sd_of_mean_samp <- sd(mean_samp)
sd_of_mean_samp
```

```
## [1] 0.1145439
```

I see the mean remains same ~2 but the statndard deviation is decreasing. Let me take sample size n=3000.

n=3000

```
# create a vector with 10000 NAs
mean_samp <- rep(NA, 10000)

# start a loop
for(i in 1:10000) {
g_samp <- rgamma(3000, shape = 1, scale = 2)
mean_samp[i] <- mean(g_samp)
}

mean_of_mean_samp <- mean(mean_samp)
mean_of_mean_samp
```

```
## [1] 1.99959
```

```
sd_of_mean_samp <- sd(mean_samp)
sd_of_mean_samp
```

```
## [1] 0.0370315
```

We see standard deviation decreases.

End of Homework 6