COMPSCIX 415.2 Homework 6

Vishnu Vardhan

3/9/2018

Contents

Exercise 1

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.

Q1.

What variables are in this data set?

Answer.

The variables are 'outcome', 'smoker' and 'age'

```
glimpse(Whickham)
```

```
## Observations: 1,314
## Variables: 3
## $ outcome <fctr> Alive, Alive, Dead, Alive, Alive, Alive, Alive, Dead,...
## $ smoker  <fctr> 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...
```

Q2.

How many observations are there and what does each represent?

Answer.

There are 1314 observations. Each represents an individual. The data set indicates if the individual is a smoker or not, his current age, and if he is alive or dead.

Q3.

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(_____ , _____)

Answer.

We can do this two ways.

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

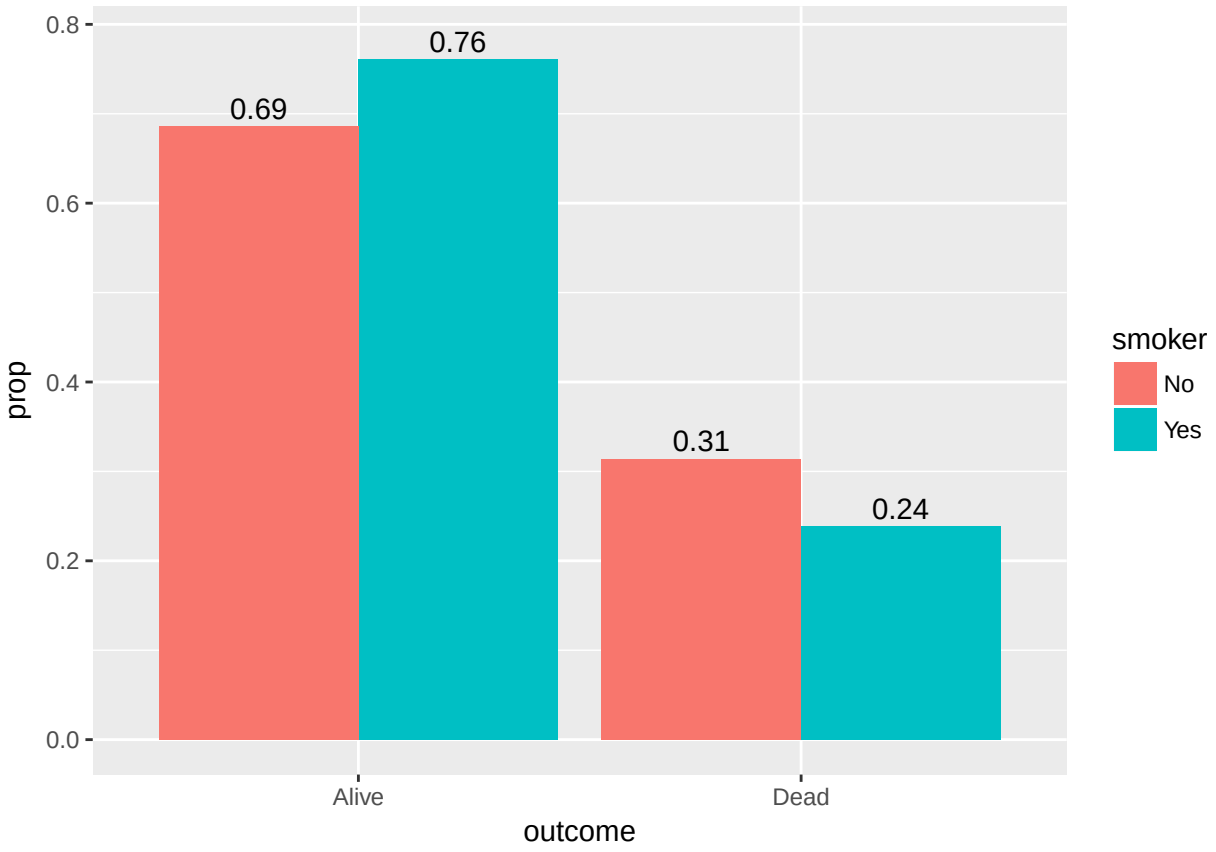
```
## # A tibble: 4 x 3
##   smoker outcome     n
##   <fctr>  <fctr> <int>
## 1     No   Alive   502
## 2     No    Dead   230
## 3    Yes   Alive   443
## 4    Yes    Dead   139
```

or more interestingly, as outlined here https://stackoverflow.com/questions/24576515/relative-frequencies-proportions-with-dply

```
w_3 <- Whickham %>% group_by(smoker,outcome) %>% summarize(n = n()) %>% mutate ( prop = n/sum(n))
w_3
```

```
## # A tibble: 4 x 4
## # Groups:   smoker [2]
##   smoker outcome     n      prop
##   <fctr>  <fctr> <int>     <dbl>
## 1     No   Alive   502 0.6857923
## 2     No    Dead   230 0.3142077
## 3    Yes   Alive   443 0.7611684
## 4    Yes    Dead   139 0.2388316
```

```
ggplot(data = w_3, mapping = aes(x=outcome, y = prop, fill = smoker, label = round(prop,2) )) +
  geom_col(position = "dodge") +
  geom_text(mapping = aes(y = prop+0.02),position = position_dodge(width = 0.9))
```



The data shows that 68% of non-smokers are alive, but 76% of smokers are alive. I.e more smokers are alive.

Unless you believe smoking is healthy, and want to stop further analysis, this does not make sense.

It is an interesting anamoly, that is likely caused by how the smoker population is distributed accross the population, versus the nature of the population itself, where the non-smokers live longer.

Q4.

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?

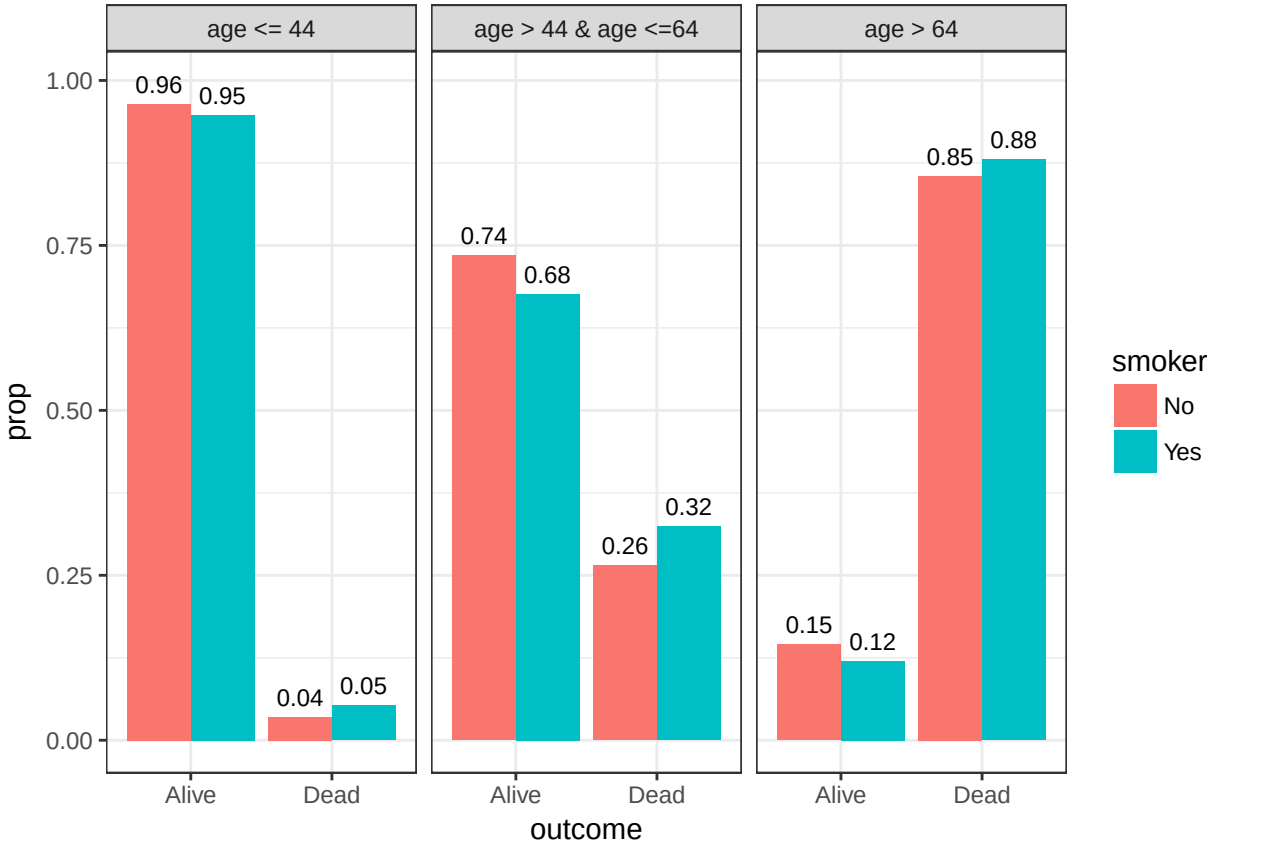
Answer.

Till the age of 44, non-smokers have only a 1% advantage compared to smokers, but this gap increases dramatically between the ages 44 & 65 where 6% more non-smokers are alive than smokers. After 65, the difference drops to 3% with non-smokers still being alive more often than smokers.

```
m_p <- Whickham %>%
  mutate ( cat =
             factor (case_when( age <= 44 ~ "age <= 44", age > 44 & age <= 64 ~ "age > 44 & age <=64",
                                age > 64 ~ "age > 64"))) %>%
  group_by(cat,smoker,outcome) %>%
  summarise( n = n()) %>%
  mutate   (prop = n/sum(n))

ggplot(data = m_p, aes(x=outcome, y = prop, label = round(prop,2), fill=smoker)) +
```

```
  geom_col(position = "dodge") +
  geom_text( aes(y=prop+0.03), position = position_dodge(width = 0.9), size = 3) +
  facet_wrap( ~ cat) +
  theme_bw()
```



Referenced this awesome stackoverflow response on positions: https://stackoverflow.com/questions/34889766/what-is-the-width-argument-in-position-dodge

Exercise 2

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.

Q1.

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.

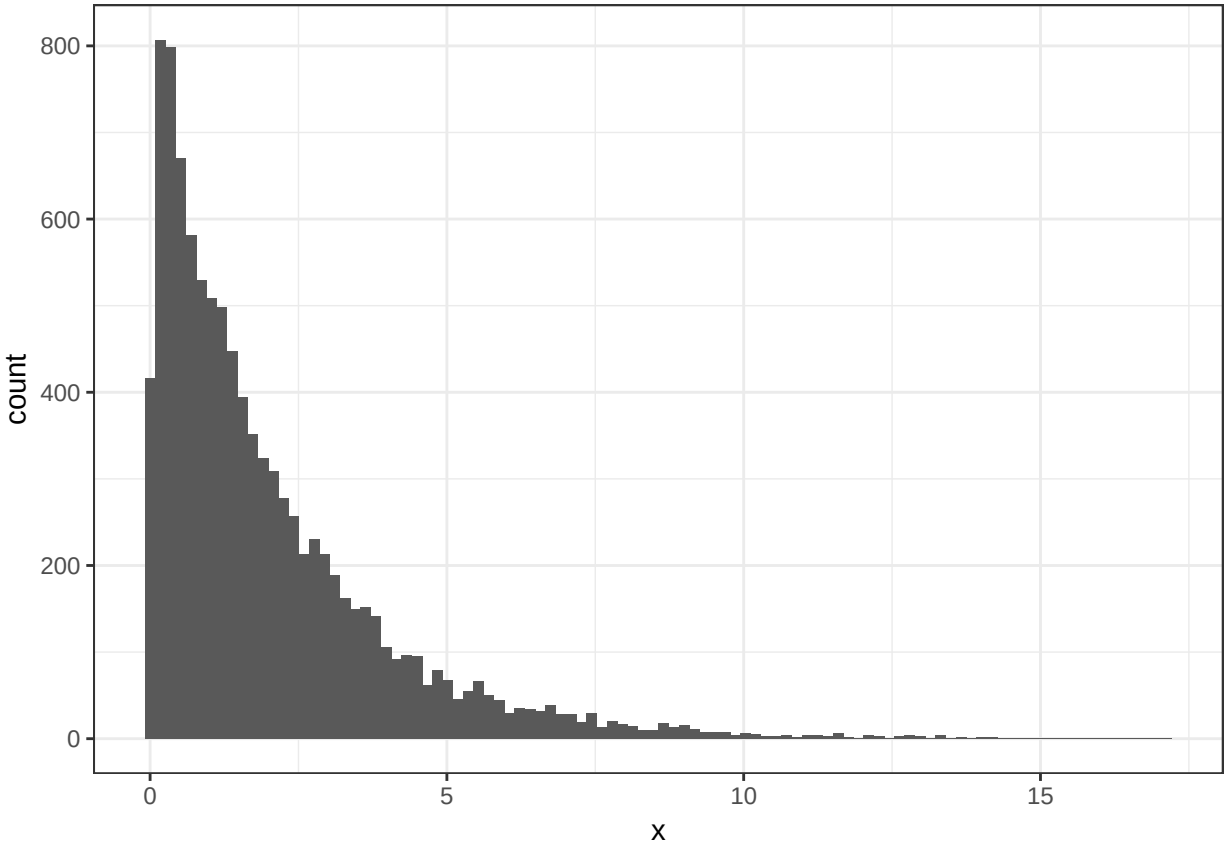
library(tidyverse) n <- 10000

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

Answer.

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

ggplot(data = gamma_samp) +
  geom_histogram(aes(x=x), bins=100) +
  theme_bw()
```



Q2.

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.

The mean is 1.98, and the standard deviation is 2.00

```
sprintf ("Mean = %f",sapply(gamma_samp, mean, na.rm = TRUE))
```

```
## [1] "Mean = 1.995663"
```

```
sprintf ("Standard deviation = %f", sapply(gamma_samp, sd, na.rm = TRUE))
```

```
## [1] "Standard deviation = 2.025641"
```

Q3.

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.

The sample still looks like a gamma distribution.

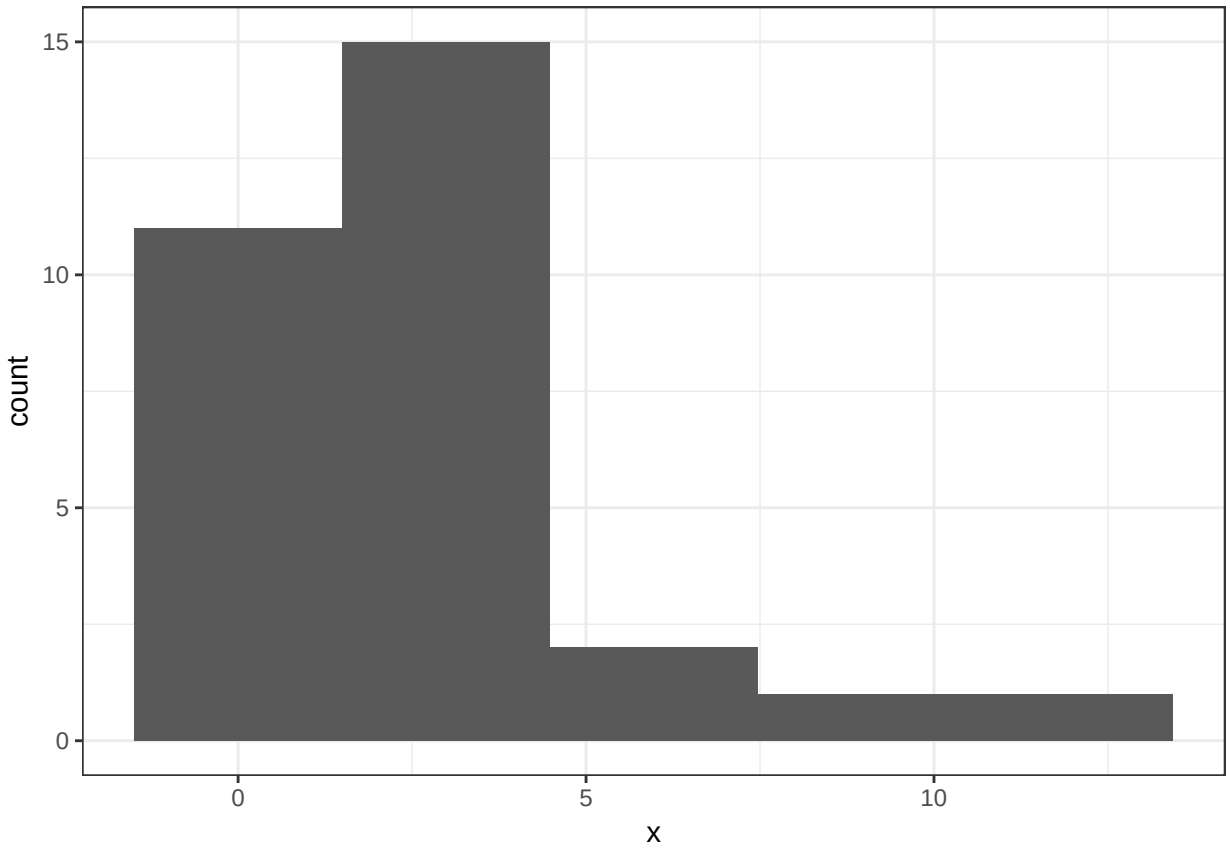
```
s <- gamma_samp %>% sample_n(30, replace = TRUE)
sprintf("Mean = %f", sapply(s, mean, na.rm = TRUE))
```

```
## [1] "Mean = 2.415919"
```

```
sprintf("Standard deviation = %f", sapply(s, sd, na.rm = TRUE))
```

```
## [1] "Standard deviation = 2.469043"
```

```
ggplot(data =s, mapping = aes(x=x)) + geom_histogram(bins=5) + theme_bw()
```



Q4.

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.

I wrote the code slightly differently to use the distribution we already created, and also look at the standard deviations at the same time

```
no_mean_samples <-  10000
mean_samp <- rep(NA, no_mean_samples)
mean_sd   <- rep(NA, no_mean_samples)
for (i in 1:no_mean_samples) {
  g_samp <- gamma_samp %>% sample_n(30, replace = TRUE)
  mean_samp[i] <- mean(g_samp$x)
  mean_sd[i]   <- sd(g_samp$x)
}
mean_dist <- bind_cols(tibble(mean_samp),tibble(mean_sd))
m_d <- bind_rows(tibble(key="mean",value=mean_samp), tibble(key="sd",value=mean_sd))
```

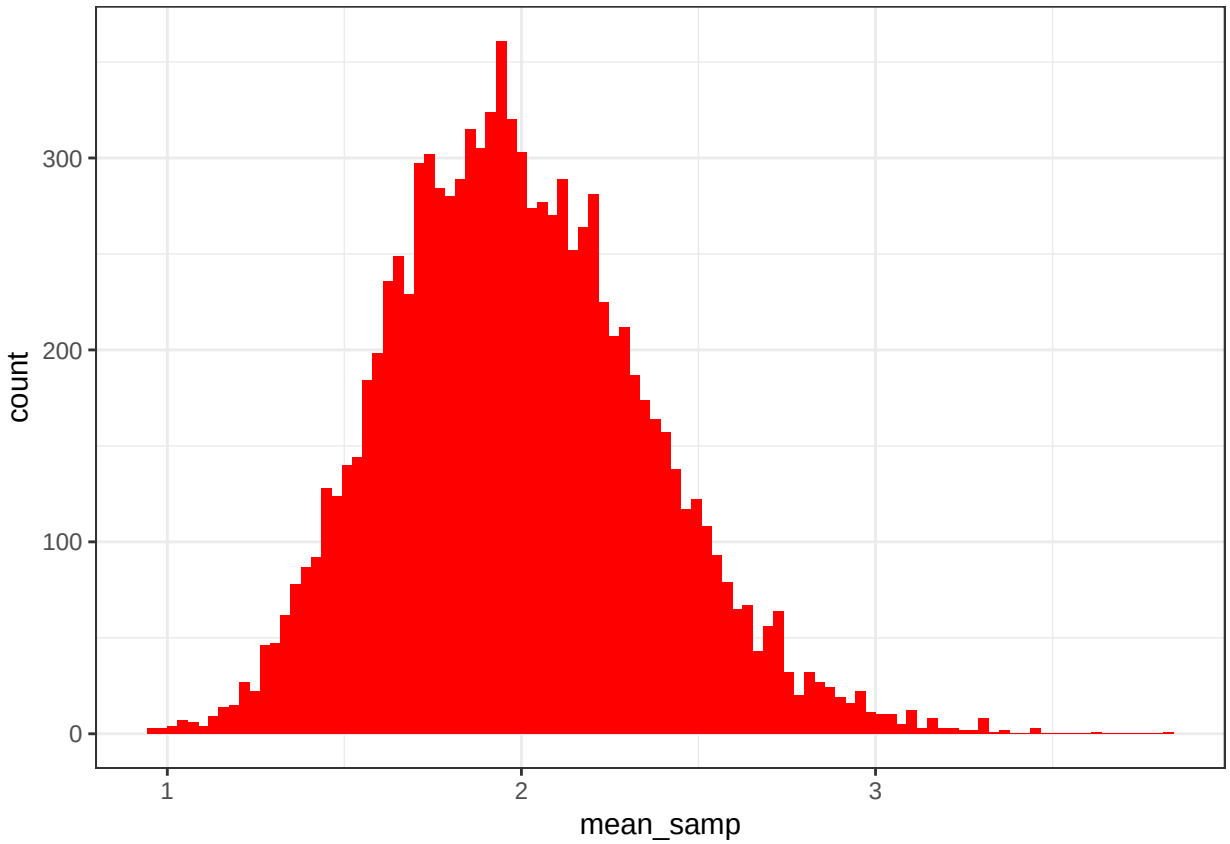
Q5.

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

Answer.

Lets plot the histogram

```
ggplot(data = mean_dist, mapping = aes(x=mean_samp)) +
  geom_histogram(fill = 'red', bins=100) +
  theme_bw()
```



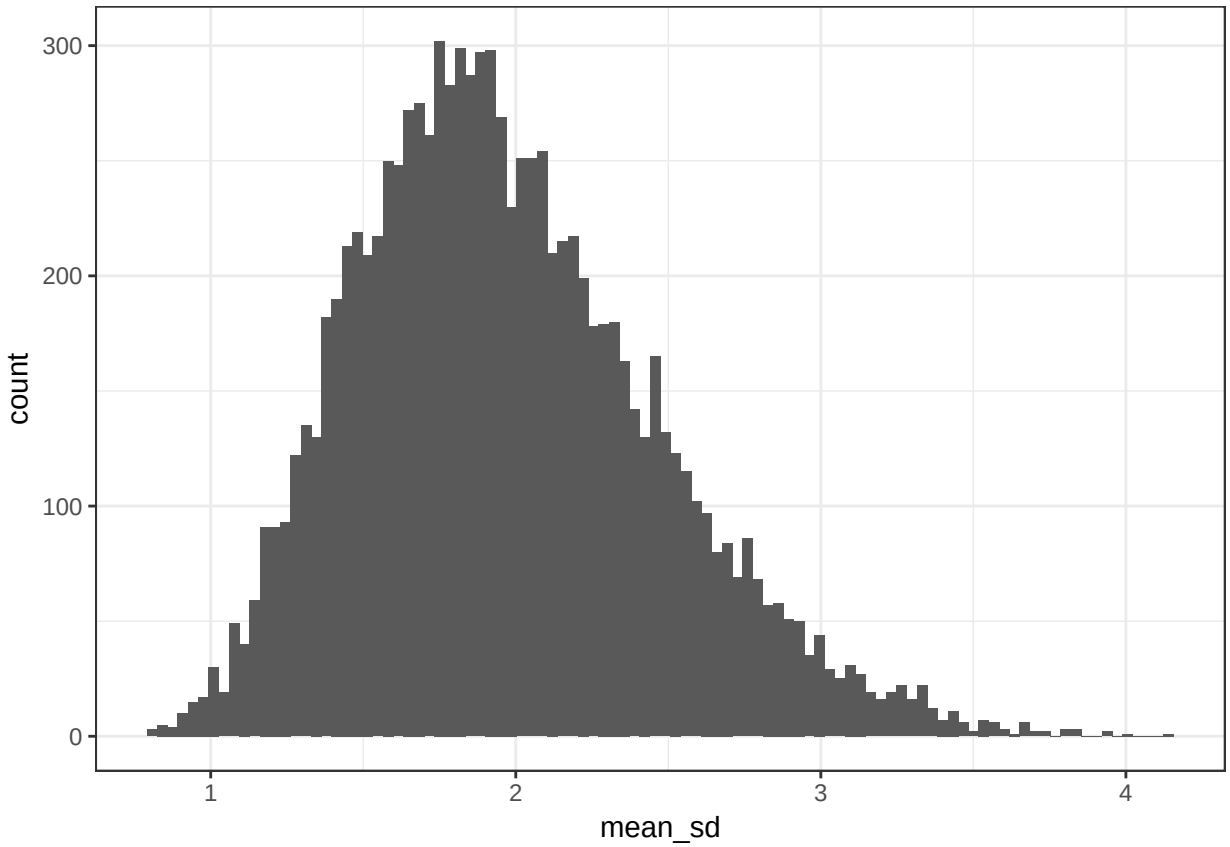
Q6.

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

Answer.

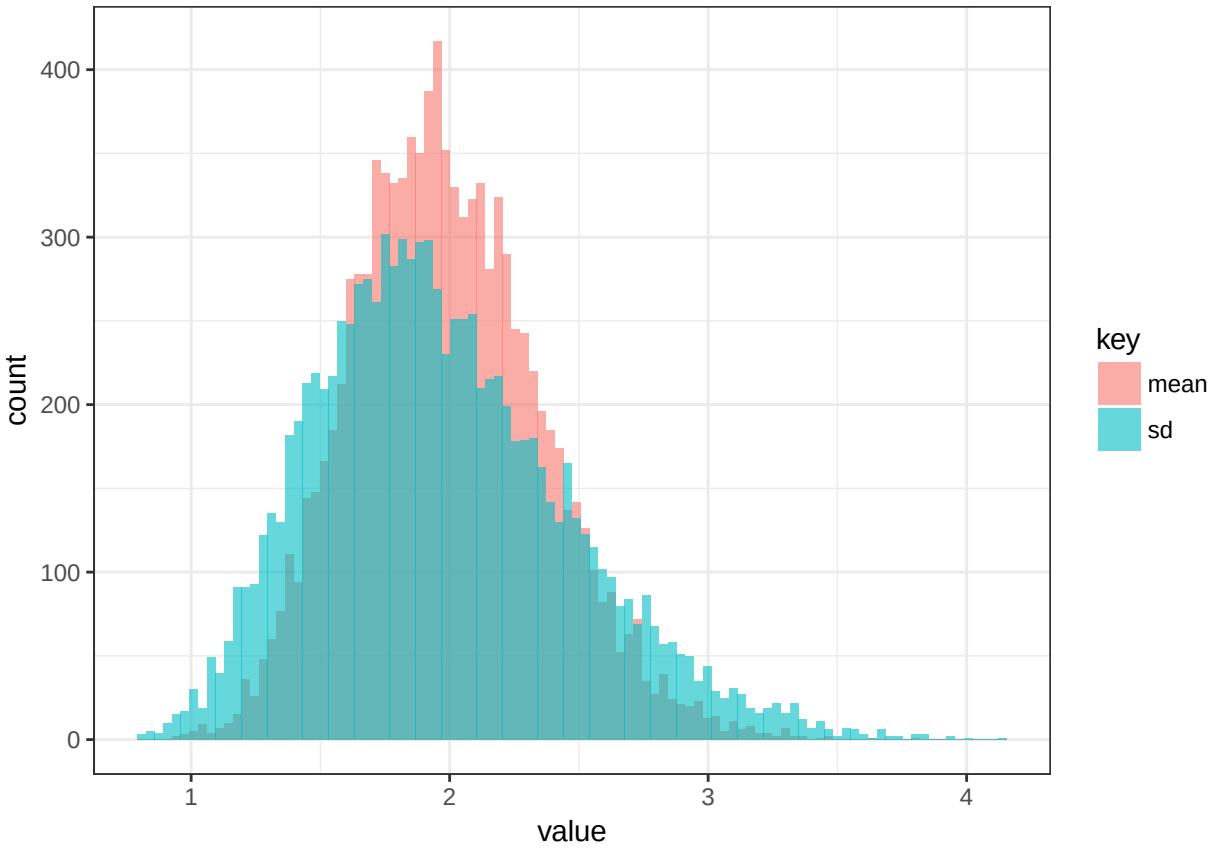
Lets plot a histogram of the standard deviations (the means is already shown above). I am not sure if this is following a normal or a gamma distribution. Looks more gamma to me.

```
ggplot(data = mean_dist) + geom_histogram(mapping = aes(x=mean_sd), bins=100) + theme_bw()
```



Plotting both on the same scales show that the distribution of the mean-of-samples is different from the distribution of the standard deviation-of-samples. Not sure if there is any insight here..

```
ggplot(data = m_d, mapping = aes(x=value, fill = key)) +
  geom_histogram(bins=100, position = "identity",,alpha = 0.6) +
  theme_bw()
```



Q7.

Did anything surprise you about your answers to #6?

Answer.

I am marginally suprised that the standard deviation also follows a distribution that looks "normal". I am not sure if it is. Should the standard deviation of the samples not resemble the original curve (which was a gamma distribution)? I.e it would seem more intuitive to me if the standard-deviation distribution was the same as the original distribution (gamma)

Q8.

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.

Answer.

The mean of all the sample means is 1.987. The standard deviation is 0.361, which is the same as the "popoulation standard deviation" / SqRt(no-of-samples=30)

```
sapply(mean_dist,mean, na.rm = TRUE)
```

```
## mean_samp   mean_sd
##  1.991094  1.968450
```

```
sapply(mean_dist, sd,  na.rm = TRUE)
```

```
## mean_samp   mean_sd
## 0.3672395 0.4926258
```

Q9.

Repeat #4-#6, but now with a sample of size n = 300 instead. Do your results match up well with the theorem?

Answer.

The mean of the sample-means is 1.989, and the standard deviation is 0.115, which is the same as (pop_sd = 2)/(sqrt(sample_size=300))

```
no_mean_samples <-  10000
mean_samp <- rep(NA, no_mean_samples)
mean_sd   <- rep(NA, no_mean_samples)
for (i in 1:no_mean_samples) {
  g_samp <- gamma_samp %>% sample_n(300, replace = TRUE)
  mean_samp[i] <- mean(g_samp$x)
  mean_sd[i]   <- sd(g_samp$x)
}
mean_dist <- bind_cols(tibble(mean_samp),tibble(mean_sd))
m_d <- bind_rows(tibble(key="mean",value=mean_samp), tibble(key="sd",value=mean_sd))
sapply(mean_dist,mean, na.rm = TRUE)
```

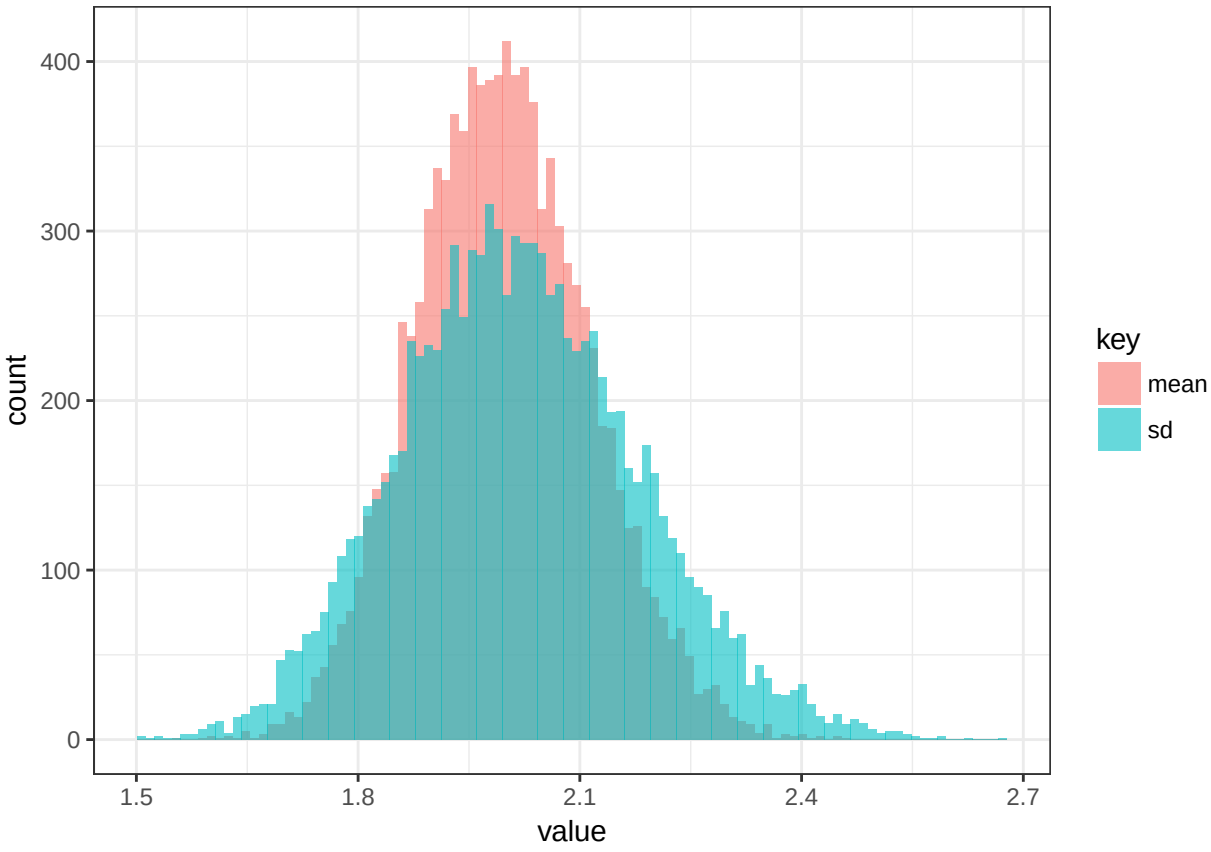
```
## mean_samp   mean_sd
##  1.996332  2.018644
```

```
sapply(mean_dist, sd,  na.rm = TRUE)
```

```
## mean_samp   mean_sd
## 0.1170526 0.1621238
```

Plotting an overlapping distribution shows that both graphs tend to normal distributions, with the mean having a lower standard deviation.

```
ggplot(data = m_d, mapping = aes(x=value, fill = key)) +
  geom_histogram(bins=100, position = "identity", alpha = 0.6) +
  theme_bw()
```


400
300
200
100
0
count
1.5
1.8
2.1
2.4
2.7
value
key
mean
sd