inman-callen-ada-homework-2

Callen Inman

3/28/2021

# CHALLENGE 1  
  
  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.5 v dplyr 1.0.3  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
f <- "https://raw.githubusercontent.com/difiore/ada-2021-datasets/master/IMDB-movies.csv"  
  
d <- read.csv(f)  
  
class(d)

## [1] "data.frame"

movies <- as\_tibble(d)  
  
class(movies)

## [1] "tbl\_df" "tbl" "data.frame"

new\_movies <- dplyr::filter(movies, startYear %in% (1920:1979) & runtimeMinutes%in%(60:180)) %>%  
dplyr::mutate("decade" = paste(floor(startYear/10)\*10))  
  
 library(ggplot2)  
  
library(mosaic)

## Warning: package 'mosaic' was built under R version 4.0.4

## Registered S3 method overwritten by 'mosaic':  
## method from   
## fortify.SpatialPolygonsDataFrame ggplot2

##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.

##   
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':  
##   
## mean

## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally

## The following object is masked from 'package:purrr':  
##   
## cross

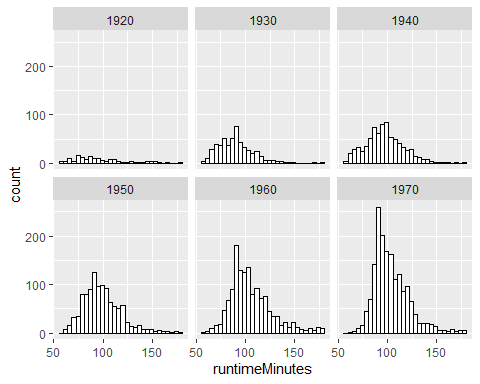
## The following object is masked from 'package:ggplot2':  
##   
## stat

## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,  
## quantile, sd, t.test, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum

m<-ggplot(new\_movies, aes(x=runtimeMinutes))+  
 geom\_histogram(color="black", fill="white")+  
 facet\_wrap(decade ~ .)  
  
m

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



results <- new\_movies %>% group\_by(decade) %>% summarise\_at(vars(runtimeMinutes), list(name = mean, name = sd))  
  
  
results

## # A tibble: 6 x 3  
## decade name..1 name..2  
## \* <chr> <dbl> <dbl>  
## 1 1920 96.3 26.2  
## 2 1930 90.3 17.3  
## 3 1940 97.2 19.1  
## 4 1950 98.9 19.2  
## 5 1960 106. 21.2  
## 6 1970 104. 18.0

# one sample of 20s films  
  
twentiesmovies <- new\_movies %>% filter(decade == 1920)  
  
  
  
k <- 1 # number of samples  
n <- 100 # size of each sample  
  
twentiesruntime <- list()  
  
for (i in 1:k) {  
 twentiesruntime[[i]] <- sample(twentiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
mean(unlist(twentiesruntime))

## [1] 97.01

twenties\_sd <- sd(unlist(twentiesruntime))  
  
twenties\_se <- twenties\_sd/(sqrt(n))  
   
  
   
   
  
# sampling from 30s films  
  
thirtiesmovies <- new\_movies %>% filter(decade == 1930)  
  
  
k <- 1 # number of samples  
n <- 100 # size of each sample  
  
thirtiesruntime <- list()  
  
for (i in 1:k) {  
 thirtiesruntime[[i]] <- sample(thirtiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
mean(unlist(thirtiesruntime))

## [1] 89.12

thirties\_sd <- sd(unlist(thirtiesruntime))  
  
thirties\_se <- thirties\_sd/(sqrt(n))  
  
  
  
# sampling from 40s films  
  
  
fortiesmovies <- new\_movies %>% filter(decade == 1940)  
  
  
k <- 1 # number of samples  
n <- 100 # size of each sample  
  
fortiesruntime <- list()  
  
for (i in 1:k) {  
 fortiesruntime[[i]] <- sample(fortiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
mean(unlist(fortiesruntime))

## [1] 96.35

forties\_sd <- sd(unlist(fortiesruntime))  
  
forties\_se <- forties\_sd/(sqrt(n))  
  
  
  
# sampling from 50s films  
  
  
fiftiesmovies <- new\_movies %>% filter(decade == 1950)  
  
k <- 1 # number of samples  
n <- 100 # size of each sample  
  
fiftiesruntime <- list()  
  
for (i in 1:k) {  
 fiftiesruntime[[i]] <- sample(fiftiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
mean(unlist(fiftiesruntime))

## [1] 98.97

fifties\_sd <- sd(unlist(fiftiesruntime))  
  
fifties\_se <- fifties\_sd/(sqrt(n))  
  
  
# sampling from 60s films  
  
sixtiesmovies <- new\_movies %>% filter(decade == 1960)  
  
k <- 1 # number of samples  
n <- 100 # size of each sample  
  
sixtiesruntime <- list()  
  
for (i in 1:k) {  
 sixtiesruntime[[i]] <- sample(sixtiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
mean(unlist(sixtiesruntime))

## [1] 105.12

sixties\_sd <- sd(unlist(sixtiesruntime))  
  
sixties\_se <- sixties\_sd/(sqrt(n))  
  
  
  
# sampling from 70s films  
  
seventiesmovies <- new\_movies %>% filter(decade == 1970)  
  
seventiesruntime <- list()  
  
for (i in 1:k) {  
 seventiesruntime[[i]] <- sample(seventiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
mean(unlist(seventiesruntime))

## [1] 103.06

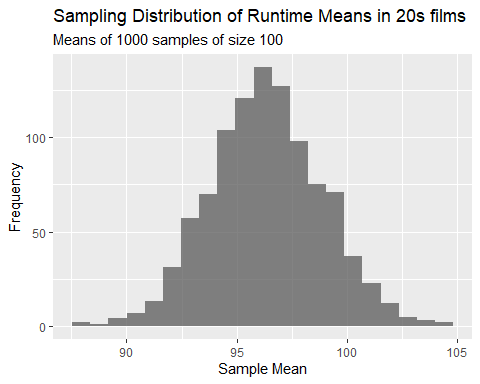
seventies\_sd <- sd(unlist(seventiesruntime))  
  
seventies\_se <- seventies\_sd/(sqrt(n))  
  
# comparing population means to sample means  
  
pop.means <- results$name..1  
  
pop.means

## [1] 96.25658 90.30000 97.20332 98.94820 105.58586 103.75000

# creating a sampling distribution for 20s films  
  
  
  
k <- 1000 # number of samples  
n <- 100 # size of each sample  
  
twenties\_samplingdist <- list()  
  
for (i in 1:k) {  
 twenties\_samplingdist[[i]] <- sample(twentiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
m <- vector(length = k)  
for (i in 1:k) {  
 m[[i]] <- mean(twenties\_samplingdist[[i]])  
}  
  
mean(m)

## [1] 96.33706

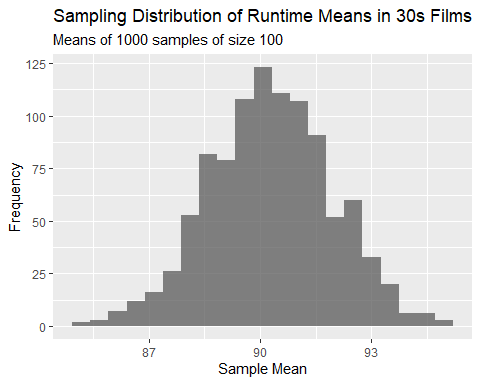
twentiessamplingdist\_sd <- sd(m)  
  
  
(  
 p <- ggplot(data = as.data.frame(m), aes(x = m)) +  
 geom\_histogram(binwidth = function(x) (max(m) - min(m)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Runtime Means in 20s films"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency"))



# The sampling distribution is roughly bell curve-shaped (normal), with a slight right skew.  
   
  
# creating a sampling distribution for 30s films  
  
k <- 1000 # number of samples  
n <- 100 # size of each sample  
  
thirties\_samplingdist <- list()  
  
for (i in 1:k) {  
 thirties\_samplingdist[[i]] <- sample(thirtiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
m <- vector(length = k)  
for (i in 1:k) {  
 m[[i]] <- mean(thirties\_samplingdist[[i]])  
}  
  
mean(m)

## [1] 90.28602

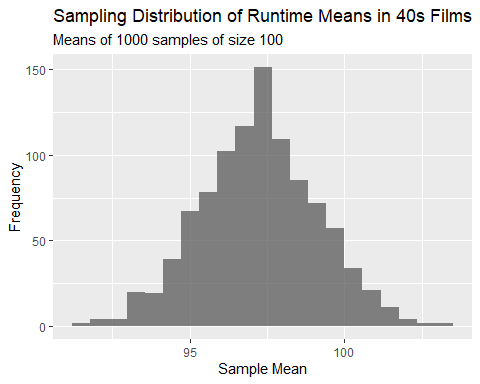
thirtiessamplingdist\_sd <- sd(m)  
  
  
(  
 p <- ggplot(data = as.data.frame(m), aes(x = m)) +  
 geom\_histogram(binwidth = function(x) (max(m) - min(m)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Runtime Means in 30s Films"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency"))



# This sampling distribution is normally distributed with an extremely slight left skew.  
  
# creating a sampling distribution for 40s films  
  
k <- 1000 # number of samples  
n <- 100 # size of each sample  
  
forties\_samplingdist <- list()  
  
for (i in 1:k) {  
 forties\_samplingdist[[i]] <- sample(fortiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
m <- vector(length = k)  
for (i in 1:k) {  
 m[[i]] <- mean(forties\_samplingdist[[i]])  
}  
  
mean(m)

## [1] 97.23572

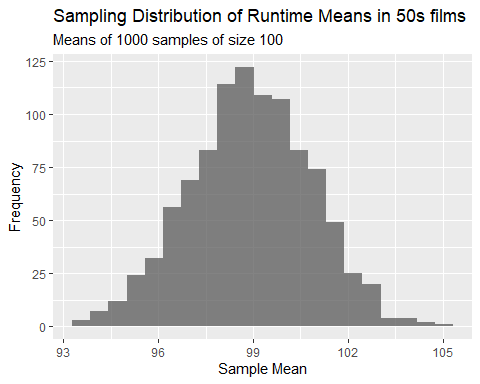
fortiessamplingdist\_sd <- sd(m)  
  
  
(  
 p <- ggplot(data = as.data.frame(m), aes(x = m)) +  
 geom\_histogram(binwidth = function(x) (max(m) - min(m)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Runtime Means in 40s Films"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency"))



# The sampling distribution is normally distributed with a slight left skew.  
  
# creating a sampling distribution for 50s films  
  
  
k <- 1000 # number of samples  
n <- 100 # size of each sample  
  
fifties\_samplingdist <- list()  
  
for (i in 1:k) {  
 fifties\_samplingdist[[i]] <- sample(fiftiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
m <- vector(length = k)  
for (i in 1:k) {  
 m[[i]] <- mean(fifties\_samplingdist[[i]])  
}  
  
mean(m)

## [1] 98.91439

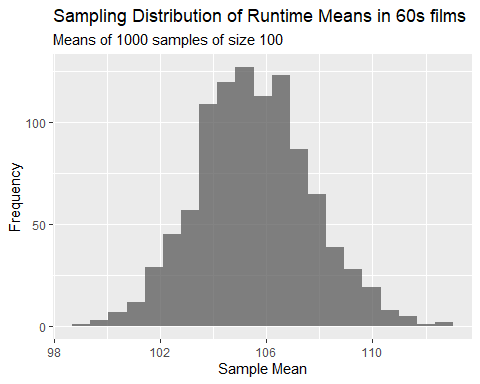
fiftiessamplingdist\_sd <- sd(m)  
  
  
(  
 p <- ggplot(data = as.data.frame(m), aes(x = m)) +  
 geom\_histogram(binwidth = function(x) (max(m) - min(m)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Runtime Means in 50s films"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency"))



# The sampling distribution is normally distributed with a slight left skew.  
  
# creating a sampling distribution for 60s films  
  
k <- 1000 # number of samples  
n <- 100 # size of each sample  
  
sixties\_samplingdist <- list()  
  
for (i in 1:k) {  
 sixties\_samplingdist[[i]] <- sample(sixtiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
m <- vector(length = k)  
for (i in 1:k) {  
 m[[i]] <- mean(sixties\_samplingdist[[i]])  
}  
  
mean(m)

## [1] 105.5315

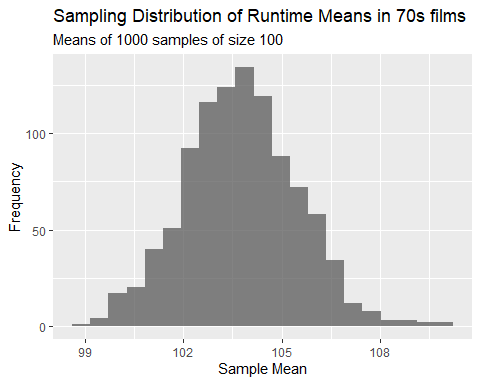
sixtiessampling\_distsd<- sd(m)  
  
  
(  
 p <- ggplot(data = as.data.frame(m), aes(x = m)) +  
 geom\_histogram(binwidth = function(x) (max(m) - min(m)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Runtime Means in 60s films"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency"))



# The sampling distribution is normally distributed with a slight left skew.  
  
# creating a sampling distribution for 70s films  
  
k <- 1000 # number of samples  
n <- 100 # size of each sample  
  
seventies\_samplingdist <- list()  
  
for (i in 1:k) {  
 seventies\_samplingdist[[i]] <- sample(seventiesmovies$runtimeMinutes, size = n, replace = TRUE )  
}  
  
m <- vector(length = k)  
for (i in 1:k) {  
 m[[i]] <- mean(seventies\_samplingdist[[i]])  
}  
  
mean(m)

## [1] 103.7715

seventiessamplingdist\_sd <- sd(m)  
  
  
(  
 p <- ggplot(data = as.data.frame(m), aes(x = m)) +  
 geom\_histogram(binwidth = function(x) (max(m) - min(m)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Runtime Means in 70s films"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency"))



# The sampling distribution is normally distributed and slightly right skewed.  
  
# comparing SEs and sampling distribution SDs  
  
# for the twenties:  
  
print(twenties\_se)

## [1] 2.302984

print(twentiessamplingdist\_sd)

## [1] 2.527792

# for the thirties:  
  
print(thirties\_se)

## [1] 1.780777

print(thirtiessamplingdist\_sd)

## [1] 1.65772

# for the forties:  
  
print(forties\_se)

## [1] 1.85567

print(fortiessamplingdist\_sd)

## [1] 1.894593

# for the fifties:  
  
print(fifties\_se)

## [1] 1.972357

print(fiftiessamplingdist\_sd)

## [1] 1.923264

# for the sixties:  
  
print(sixties\_se)

## [1] 1.907205

print(sixtiessampling\_distsd)

## [1] 2.138269

# for the seventies:  
  
print(seventies\_se)

## [1] 1.6378

print(seventiessamplingdist\_sd)

## [1] 1.713901

# CHALLENGE 2  
  
library(ggplot2)  
library(mosaic)  
  
  
ppois(9, lambda=12)

## [1] 0.2423922

ppois(0, lambda = 12)

## [1] 6.144212e-06

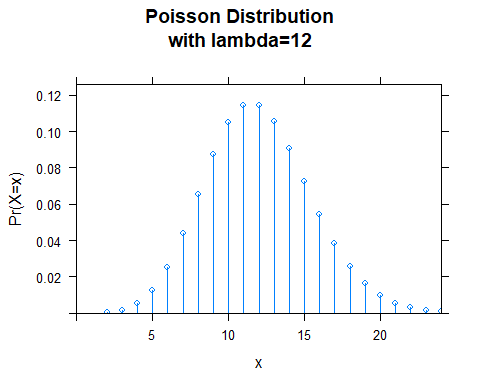
dpois(5, lambda=12)

## [1] 0.01274064

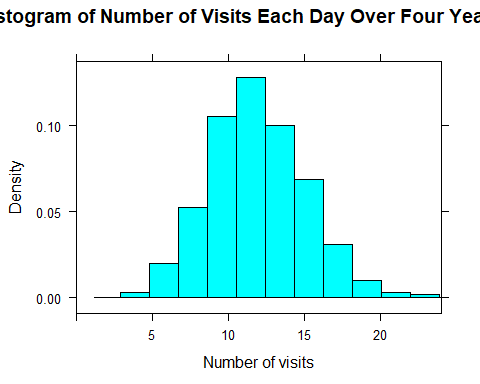
1- (ppois(18, lambda = 12))

## [1] 0.03741649

l <- 12  
p1 <-  
 plotDist(  
 "pois",  
 lambda = 12,  
 main = paste0("Poisson Distribution\nwith lambda=", 12),  
 xlab = "x",  
 ylab = "Pr(X=x)", xlim = c(0, 24)  
 )  
  
print(p1)



fouryearsof\_bees <- rpois(n = 1460, lambda = 12)  
  
histogram(fouryearsof\_bees, main = paste0("Histogram of Number of Visits Each Day Over Four Years"), xlab = "Number of visits", ylab = "Density", xlim = c(0, 24))



# The histogram and the Poisson distribution plotted above look nearly identical in shape.

library(tidyverse)  
  
f <- "https://raw.githubusercontent.com/difiore/ada-2021-datasets/master/zombies.csv"  
d <- read\_csv(f, col\_names = TRUE)

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## id = col\_double(),  
## first\_name = col\_character(),  
## last\_name = col\_character(),  
## gender = col\_character(),  
## height = col\_double(),  
## weight = col\_double(),  
## zombies\_killed = col\_double(),  
## years\_of\_education = col\_double(),  
## major = col\_character(),  
## age = col\_double()  
## )

# calculating population mean and standard deviation for height  
  
(height\_sigma <- sqrt(sum(((d$height) - mean(d$height))^2) / length(d$height)))

## [1] 4.30797

(height\_mu <- mean(d$height))

## [1] 67.6301

# calculating population mean and standard deviation for weight  
  
(weight\_sigma <- sqrt(sum(((d$weight) - mean(d$weight))^2) / length(d$weight)))

## [1] 18.39186

(weight\_mu <- mean(d$weight))

## [1] 143.9075

# calculating population mean and standard deviation for age  
  
(age\_sigma <- sqrt(sum(((d$age) - mean(d$age))^2) / length(d$age)))

## [1] 2.963583

(age\_mu <- mean(d$age))

## [1] 20.04696

# calculating population mean and standard deviation for number of zombies killed  
  
(zombieskilled\_sigma <- sqrt(sum(((d$zombies\_killed) - mean(d$zombies\_killed))^2) / length(d$zombies\_killed)))

## [1] 1.747551

(zombieskilled\_mu <- mean(d$zombies\_killed))

## [1] 2.992

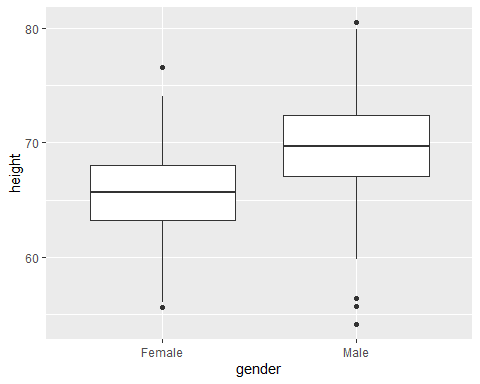
# calculating population mean and standard deviation for years of education  
  
(yearsofeducation\_sigma <- sqrt(sum(((d$years\_of\_education) - mean(d$years\_of\_education))^2) / length(d$years\_of\_education)))

## [1] 1.675704

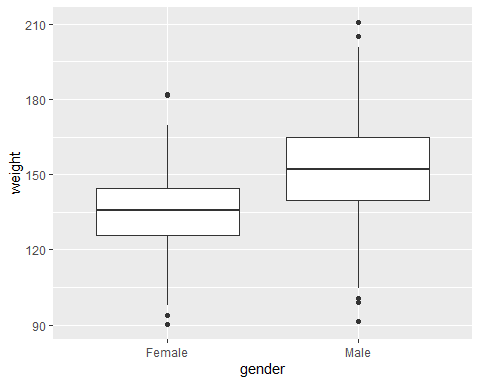
(yearsofeducation\_mu <- mean(d$years\_of\_education))

## [1] 2.996

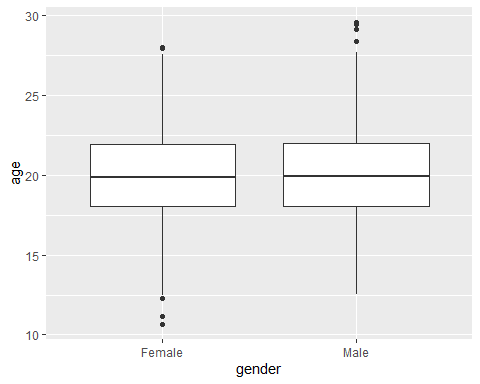
# boxplot of height by gender  
  
p\_height <- ggplot(d, aes(x = gender, y = height)) +  
 geom\_boxplot()  
  
p\_height



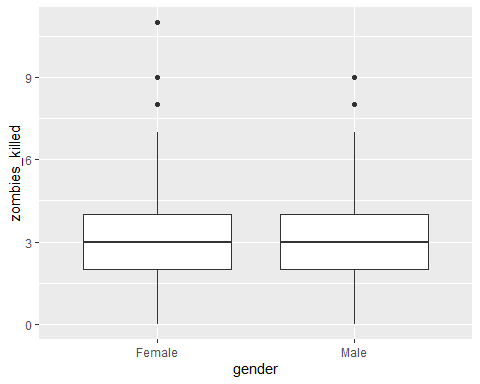
# boxplot of weight by gender  
  
p\_weight <- ggplot(d, aes(x = gender, y = weight)) +  
 geom\_boxplot()  
  
print(p\_weight)



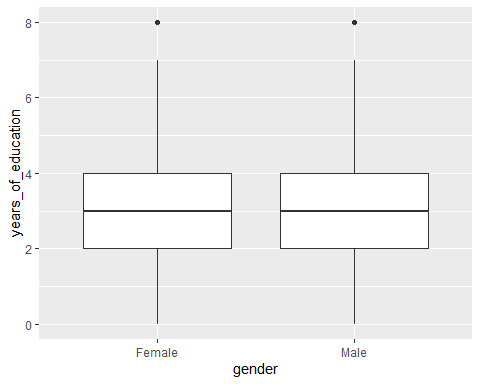
# boxplot of age by gender  
  
p\_age <- ggplot(d, aes(x = gender, y = age)) +  
 geom\_boxplot()  
  
print(p\_age)



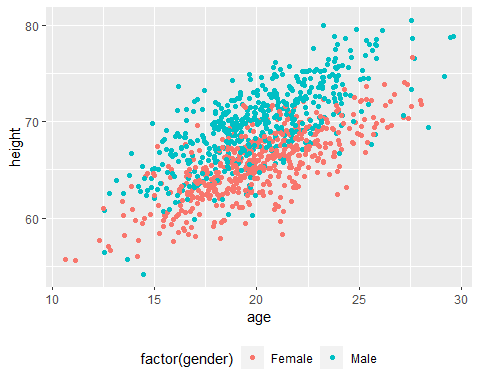
# boxplot of zombies killed by gender  
  
p\_zombieskilled <- ggplot(d, aes(x = gender, y = zombies\_killed)) +  
 geom\_boxplot()  
  
print(p\_zombieskilled)



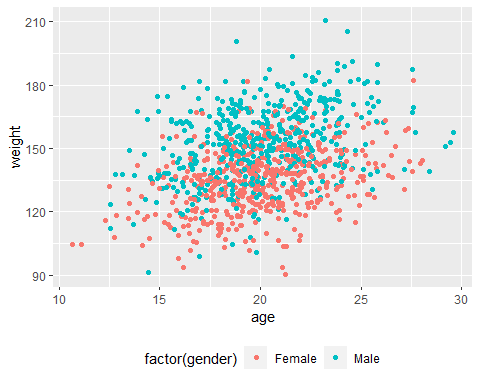
# boxplot of years of education by gender  
  
p\_yearsofeducation <- ggplot(d, aes(x = gender, y = years\_of\_education)) + geom\_boxplot()  
  
print(p\_yearsofeducation)



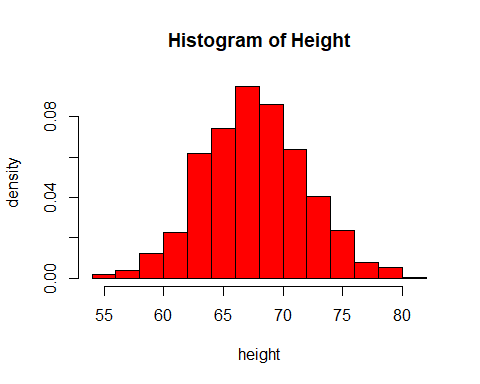
# making a scatterplot of height by age  
  
p <- ggplot(data = d, aes(  
 x = age,  
 y = height,  
 color = factor(gender)  
))  
p <- p + xlab("age") + ylab("height")  
p <- p + geom\_point(na.rm = TRUE)  
p <- p + theme(legend.position = "bottom", legend.title = element\_blank() + geom\_point())  
  
print(p)



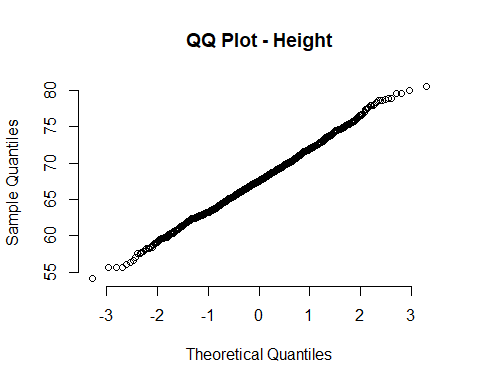
# making a scatterplot of weight by age  
  
p <- ggplot(data = d, aes(  
 x = age,  
 y = weight,  
 color = factor(gender)  
))  
p <- p + xlab("age") + ylab("weight")  
p <- p + geom\_point(na.rm = TRUE)  
p <- p + theme(legend.position = "bottom", legend.title = element\_blank() + geom\_point())  
  
p



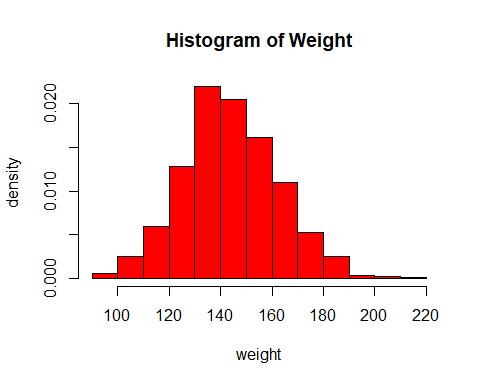
# Histogram of height  
  
attach(d)  
  
hist(  
 height,  
 freq = FALSE,  
 col = "red",  
 main = "Histogram of Height",  
 xlab = "height",  
 ylab = "density"  
 )



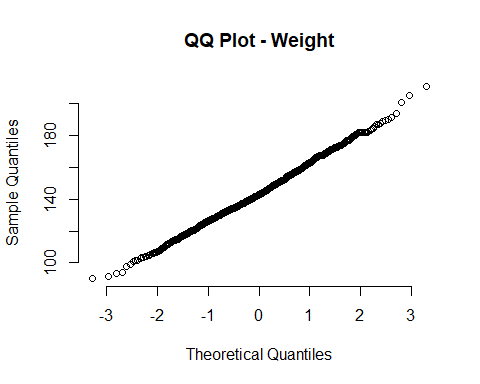
library(ggplot2)  
  
qqnorm(d$height, main = "QQ Plot - Height", frame = FALSE)



# Height is roughly normally distributed with a slight right skew.  
  
hist(  
 weight,  
 freq = FALSE,  
 col = "red",  
 main = "Histogram of Weight",  
 xlab = "weight",  
 ylab = "density"  
)



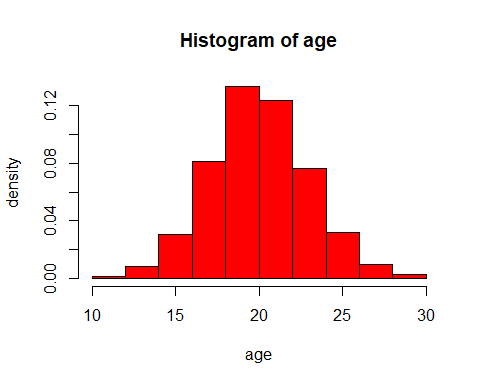
qqnorm(d$weight, main = "QQ Plot - Weight", frame = FALSE)



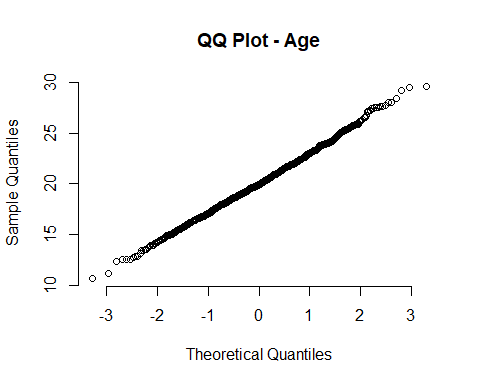
# Weight is roughly normally distributed with a slight right skew.  
  
attach(d)

## The following objects are masked from d (pos = 3):  
##   
## age, first\_name, gender, height, id, last\_name, major, weight,  
## years\_of\_education, zombies\_killed

hist(  
 age,  
 freq = FALSE,  
 col = "red",  
 main = "Histogram of age",  
 xlab = "age",  
 ylab = "density"  
)



qqnorm(d$age, main = "QQ Plot - Age", frame = FALSE)

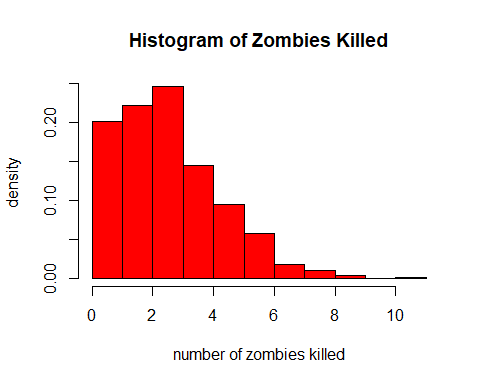


# Age is roughly normally distributed with a slight right skew.  
  
attach(d)

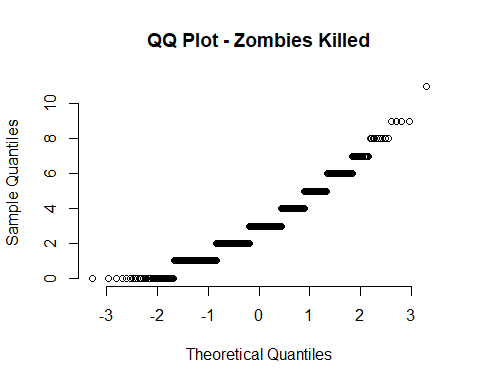
## The following objects are masked from d (pos = 3):  
##   
## age, first\_name, gender, height, id, last\_name, major, weight,  
## years\_of\_education, zombies\_killed

## The following objects are masked from d (pos = 4):  
##   
## age, first\_name, gender, height, id, last\_name, major, weight,  
## years\_of\_education, zombies\_killed

hist(  
 zombies\_killed,  
 freq = FALSE,  
 col = "red",  
 main = "Histogram of Zombies Killed",  
 xlab = "number of zombies killed",  
 ylab = "density",  
)



qqnorm(d$zombies\_killed, main = "QQ Plot - Zombies Killed", frame = FALSE)

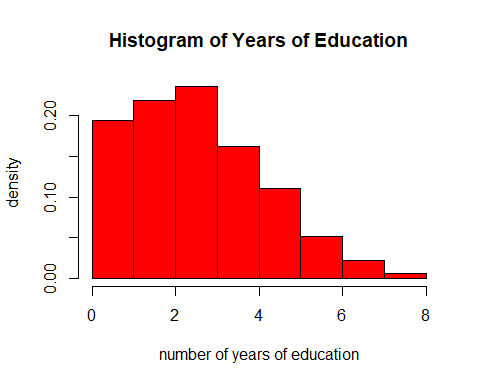


# Zombies killed is not roughly normally distributed; it is heavily right skewed.  
  
attach(d)

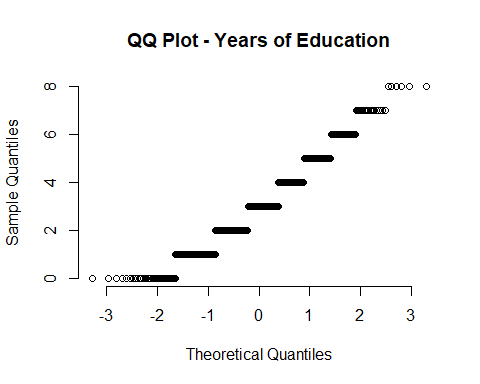
## The following objects are masked from d (pos = 3):  
##   
## age, first\_name, gender, height, id, last\_name, major, weight,  
## years\_of\_education, zombies\_killed  
##   
## The following objects are masked from d (pos = 4):  
##   
## age, first\_name, gender, height, id, last\_name, major, weight,  
## years\_of\_education, zombies\_killed

## The following objects are masked from d (pos = 5):  
##   
## age, first\_name, gender, height, id, last\_name, major, weight,  
## years\_of\_education, zombies\_killed

hist(  
 years\_of\_education,  
 freq = FALSE,  
 col = "red",  
 main = "Histogram of Years of Education",  
 xlab = "number of years of education",  
 ylab = "density",  
)



qqnorm(d$years\_of\_education, main = "QQ Plot - Years of Education", frame = FALSE)



# Years of education is moderately right skewed.  
  
  
# one sample of 50  
n <- 50  
s <- sample\_n(d, size = n, replace = FALSE)  
  
# mean, standard deviation, standard error, and confidence interval for each variable  
  
# mean, standard deviation, standard error, and confidence interval for height  
  
mean(s$height)

## [1] 67.52833

sd(s$height)

## [1] 3.942552

height\_se <- sd(s$height)/(sqrt(n))  
  
height\_se

## [1] 0.557561

m\_height <- mean(s$height)  
  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- m\_height + qnorm(alpha / 2) \* height\_se  
upper <- m\_height + qnorm(1 - alpha / 2) \* height\_se  
(ci <- c(lower, upper))

## [1] 66.43553 68.62113

ci

## [1] 66.43553 68.62113

# We are 95% sure that the population mean of height is between 67.12372 and 69.23766.  
  
# mean, standard deviation, standard error, and confidence interval for weight  
  
mean(s$weight)

## [1] 145.8179

sd(s$weight)

## [1] 17.26049

weight\_se <- sd(s$weight)/(sqrt(n))  
  
weight\_se

## [1] 2.441001

m\_weight <- mean(s$weight)  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- m\_weight + qnorm(alpha / 2) \* weight\_se  
upper <- m\_weight + qnorm(1 - alpha / 2) \* weight\_se  
(ci <- c(lower, upper))

## [1] 141.0336 150.6022

ci

## [1] 141.0336 150.6022

# We are 95% sure that the population mean of weight is between 142.6283 and 152.0481.  
  
# mean, standard deviation, standard error, and confidence interval for age  
  
mean(s$age)

## [1] 20.01807

sd(s$age)

## [1] 3.029315

age\_se <- sd(s$age)/(sqrt(n))  
  
age\_se

## [1] 0.4284099

m\_age <- mean(s$age)  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- m\_age + qnorm(alpha / 2) \* age\_se  
upper <- m\_age + qnorm(1 - alpha / 2) \* age\_se  
(ci <- c(lower, upper))

## [1] 19.17840 20.85773

ci

## [1] 19.17840 20.85773

# We are 95% sure that the population mean age is between 18.98103 and 20.58677.  
  
# Mean, standard deviation, standard error, and confidence interval of zombies killed.  
  
mean(s$zombies\_killed)

## [1] 3.3

sd(s$zombies\_killed)

## [1] 1.775686

zombieskilled\_se <- sd(s$zombies\_killed)/(sqrt(n))  
  
zombieskilled\_se

## [1] 0.2511199

m\_zombieskilled <- mean(s$zombies\_killed)  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- m\_zombieskilled + qnorm(alpha / 2) \* zombieskilled\_se  
upper <- m\_zombieskilled + qnorm(1 - alpha / 2) \* zombieskilled\_se  
(ci <- c(lower, upper))

## [1] 2.807814 3.792186

ci

## [1] 2.807814 3.792186

# We are 95% sure that the population mean number of zombies killed is between 2.157106 and 3.282894.  
  
# Mean, standard deviation, standard error, and confidence interval for number of years of education  
  
mean(s$years\_of\_education)

## [1] 2.56

sd(s$years\_of\_education)

## [1] 1.514016

yearsofeducation\_se <- sd(s$years\_of\_education)/(sqrt(n))  
  
yearsofeducation\_se

## [1] 0.2141142

m\_yearsofeducation <- mean(s$years\_of\_education)  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- m\_yearsofeducation + qnorm(alpha / 2) \* yearsofeducation\_se  
upper <- m\_yearsofeducation + qnorm(1 - alpha / 2) \* yearsofeducation\_se  
(ci <- c(lower, upper))

## [1] 2.140344 2.979656

ci

## [1] 2.140344 2.979656

# We are 95% sure that the population mean number of years of education is between 2.39075 and 3.28925  
  
# Drawing 99 samples  
  
# 99 samples: height  
  
k <- 99  
n <- 50  
s <- list()  
for (i in 1:k) {  
 s[[i]] <- sample(d$height, size = n, replace = FALSE)  
}  
  
  
  
m\_height <- vector(length = k)   
for (i in 1:k) {  
 m\_height[[i]] <- mean(s[[i]])  
}  
  
# each sample mean  
  
m\_height

## [1] 67.79940 66.82095 69.05894 67.09340 66.31890 67.78161 66.79385 66.87612  
## [9] 68.05918 66.80452 67.86850 68.43037 67.07400 67.16106 66.60269 67.31257  
## [17] 67.20092 67.47008 67.56820 67.92998 67.91478 67.97898 67.63349 67.53289  
## [25] 66.64024 68.47686 67.64669 67.74082 68.37006 67.49438 67.18941 67.54420  
## [33] 67.34280 68.08147 67.24018 67.84959 66.68303 67.34189 67.26469 67.51181  
## [41] 67.48551 67.62428 67.89677 67.34598 68.56345 66.76851 67.41068 66.70013  
## [49] 67.54201 67.53213 67.62945 67.17304 67.60982 66.05697 67.53099 67.57088  
## [57] 68.05234 68.20424 68.03866 67.94113 67.10307 67.97364 67.42610 67.48927  
## [65] 66.92441 68.63134 68.79154 67.44731 67.91670 66.76171 68.17339 68.15558  
## [73] 67.37760 67.59720 68.23324 67.30135 67.43421 66.59566 68.31660 67.28857  
## [81] 67.72814 68.70586 67.38542 66.89236 67.87104 67.62683 68.46611 67.54947  
## [89] 67.93922 67.68194 67.73438 67.84534 68.30986 68.60046 68.55348 67.17121  
## [97] 67.14644 67.29787 67.26345

# mean of the sampling distribution  
  
mean(m\_height)

## [1] 67.58473

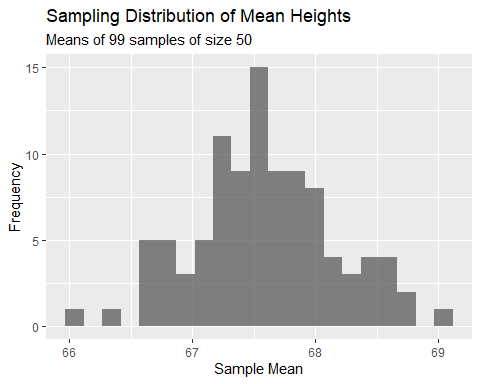
# standard deviation of the sampling distribution  
  
sd(m\_height)

## [1] 0.5756644

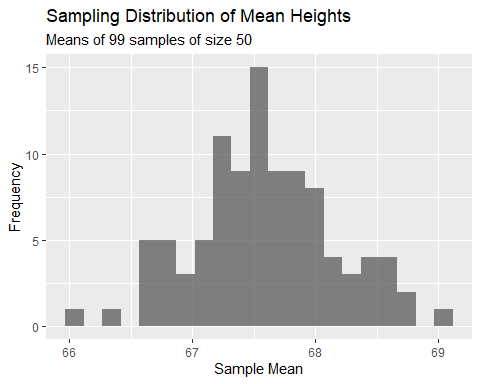
# Standard deviation of sampling distribution is a bit more than 0.1 off from the standard error calculated from one sample, which was 0.71.  
  
  
# confidence interval for the sampling distribution  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- mean(m\_height) + qnorm(alpha / 2) \* sd(m\_height)  
upper <- mean(m\_height) + qnorm(1 - alpha / 2) \* sd(m\_height)  
(ci <- c(lower, upper))

## [1] 66.45644 68.71301

# The confidence interval of the sampling distribution has overall higher values, indicating that the sampling distribution captures a lot more variation in weight than the sample does not capture.  
  
(  
 p <- ggplot(data = as.data.frame(m\_height), aes(x = m\_height)) +  
 geom\_histogram(binwidth = function(x) (max(m\_height) - min(m\_height)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Mean Heights"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency")  
   
)



print(p)



# The sampling distribution for heights is roughly normally distributed, with some gaps in the frequency of certain values that prevent the distribution from being perfectly bell curve-shaped. This might simply indicate that a larger number of sample means should be sampled from. The single sample was actually more clearly normally distributed than the sampling distribution.  
  
  
# 99 samples: weight   
k <- 99  
n <- 50  
s <- list()  
for (i in 1:k) {  
 s[[i]] <- sample(d$weight, size = n, replace = FALSE)  
}  
  
  
  
m\_weight <- vector(length = k)   
for (i in 1:k) {  
 m\_weight[[i]] <- mean(s[[i]])  
}  
  
# each sample mean  
  
m\_weight

## [1] 140.2324 140.9026 147.5400 140.1738 143.9771 144.3292 145.4327 142.0933  
## [9] 146.5530 142.6749 148.8276 143.0909 144.0457 139.1166 145.3525 143.4222  
## [17] 141.3063 144.9843 146.6120 141.4478 139.7179 139.5749 142.5991 146.3238  
## [25] 143.8241 144.0195 140.8384 141.2216 145.2365 141.8885 143.8286 148.5927  
## [33] 143.2611 143.3288 141.2860 145.5377 141.1648 144.2399 146.4926 146.7206  
## [41] 146.2756 143.7985 140.8746 142.1535 151.6855 143.8248 142.9946 147.7604  
## [49] 144.7221 141.9285 146.8092 141.3709 141.9973 144.2157 143.1293 141.6085  
## [57] 148.5149 147.3731 141.6159 146.0796 146.7729 148.7740 145.7734 137.3696  
## [65] 143.6222 143.2562 139.8307 145.6587 143.2618 146.8820 145.3512 147.0358  
## [73] 142.8608 145.0475 147.0193 145.2554 144.6499 148.6184 144.7126 146.6006  
## [81] 149.2587 144.7240 141.8014 146.5686 142.8295 143.0646 141.7778 140.6531  
## [89] 142.3391 144.1613 142.3783 138.7028 145.9429 147.3168 141.6645 142.5158  
## [97] 142.9570 145.3527 143.7833

# mean of the sampling distribution  
  
mean(m\_weight)

## [1] 144.007

# standard deviation of the sampling distribution  
  
sd(m\_weight)

## [1] 2.674833

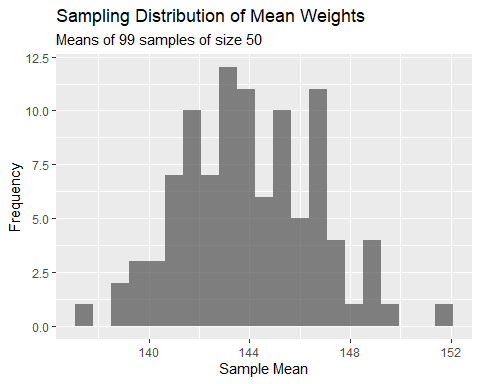
# The standard deviation calculated from the sampling distribution is almost exactly 0.1 off from the standard error calculated from the initial sample. However, this is fairly close since these are larger numbers (2.732722 for SE and 2.832534 for SD).  
  
  
# confidence interval for the sampling distribution  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- mean(m\_weight) + qnorm(alpha / 2) \* sd(m\_weight)  
upper <- mean(m\_weight) + qnorm(1 - alpha / 2) \* sd(m\_weight)  
(ci <- c(lower, upper))

## [1] 138.7644 149.2495

ci

## [1] 138.7644 149.2495

# The confidence interval of the sampling distribution is overall similar to the confidence interval of the sample. However, it is shifted to 4 lbs lower for both the lower and the upper bound of the interval.  
  
(  
 p <- ggplot(data = as.data.frame(m\_weight), aes(x = m\_weight)) +  
 geom\_histogram(binwidth = function(x) (max(m\_weight) - min(m\_weight)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Mean Weights"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency")  
)



# The sampling distribution of mean weights is roughly normally distributed, which is similar to the distribution of values calculated from the original sample.  
  
# 99 samples for age  
  
k <- 99  
n <- 50  
s <- list()  
for (i in 1:k) {  
 s[[i]] <- sample(d$age, size = n, replace = FALSE)  
}  
  
  
  
m\_age <- vector(length = k)   
for (i in 1:k) {  
 m\_age[[i]] <- mean(s[[i]])  
}  
  
# each sample mean  
  
m\_age

## [1] 19.61305 20.38201 20.05912 20.23918 20.20136 19.59087 19.25051 20.73431  
## [9] 20.28163 19.51297 20.80614 20.46777 20.34437 19.72870 19.78363 19.56823  
## [17] 20.36601 19.30568 20.52085 20.35515 20.58105 20.18040 19.18545 19.78234  
## [25] 20.10000 19.71775 20.30066 20.26921 19.28969 19.56301 19.57142 19.59154  
## [33] 19.77574 19.96034 20.27520 20.63067 19.94297 19.90969 20.39181 19.98777  
## [41] 19.76677 20.44676 19.71118 19.79179 20.02226 20.75107 20.33486 20.21891  
## [49] 19.97715 19.94380 20.24809 20.00341 20.15117 19.68451 19.88577 20.32618  
## [57] 21.31412 20.27643 19.68231 19.32079 20.57795 20.24031 19.86152 19.83360  
## [65] 20.35147 20.44117 20.22580 20.18132 20.02932 20.93338 20.29783 19.90037  
## [73] 19.96693 20.33917 20.07296 20.49484 20.27893 20.03327 20.30574 20.40798  
## [81] 19.90621 20.48567 19.77616 19.78761 19.95700 20.42224 20.18344 19.72342  
## [89] 20.60866 19.98571 20.20640 19.88438 21.00092 19.72931 20.36100 20.09671  
## [97] 19.93623 19.92100 19.94073

# mean of the sampling distribution  
  
mean(m\_age)

## [1] 20.0875

# standard deviation of the sampling distribution  
  
sd(m\_age)

## [1] 0.3935884

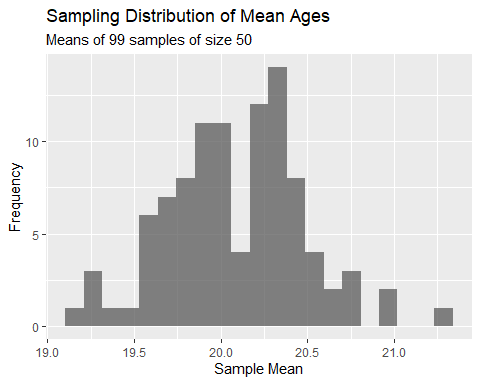
# The standard deviation of the sampling distribution and the standard error of the initial sample are very similar, with only about a 0.2 difference (SE: 0.427252 and SD: 0.3991022).  
  
# confidence interval for the sampling distribution  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- mean(m\_age) + qnorm(alpha / 2) \* sd(m\_age)  
upper <- mean(m\_age) + qnorm(1 - alpha / 2) \* sd(m\_age)  
(ci <- c(lower, upper))

## [1] 19.31608 20.85892

ci

## [1] 19.31608 20.85892

# The confidence interval for age calculated from the initial sample is very similar to the confidence interval for age calculated from the sampling distribution.  
  
(  
 p <- ggplot(data = as.data.frame(m\_age), aes(x = m\_age)) +  
 geom\_histogram(binwidth = function(x) (max(m\_age) - min(m\_age)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Mean Ages"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency")  
)



# The sampling distribution for age is similar to the sampling distribution for height. It is roughly normally distributed, with some notable gaps missing that prevent it from being a complete bell curve.  
  
# 99 samples for number of zombies killed  
  
k <- 99  
n <- 50  
s <- list()  
for (i in 1:k) {  
 s[[i]] <- sample(d$zombies\_killed, size = n, replace = FALSE)  
}  
  
  
  
m\_zombieskilled <- vector(length = k)   
for (i in 1:k) {  
 m\_zombieskilled[[i]] <- mean(s[[i]])  
}  
  
# each sample mean  
  
m\_zombieskilled

## [1] 3.44 3.04 3.32 3.52 3.30 2.80 3.06 2.84 2.64 3.04 2.92 3.04 2.96 3.00 2.92  
## [16] 3.54 2.70 3.56 3.04 3.04 3.12 2.94 3.00 2.72 3.02 2.98 2.84 3.12 2.54 3.24  
## [31] 3.12 2.92 3.14 2.72 2.62 2.62 2.76 2.88 2.98 3.06 2.66 3.30 2.72 2.86 2.82  
## [46] 2.96 3.12 3.08 2.92 3.02 2.82 3.02 2.88 2.86 2.98 2.74 2.54 3.34 3.14 2.84  
## [61] 3.00 3.04 3.14 3.12 3.28 2.82 3.34 2.66 3.12 2.86 2.68 2.58 3.22 3.02 3.10  
## [76] 3.30 2.86 3.06 2.70 2.94 2.98 2.82 3.24 2.82 2.86 2.88 3.36 3.38 2.86 3.12  
## [91] 2.86 3.24 2.76 2.68 3.10 2.80 2.94 3.28 2.80

# mean of the sampling distribution  
  
mean(m\_zombieskilled)

## [1] 2.982828

# standard deviation of the sampling distribution  
  
sd(m\_zombieskilled)

## [1] 0.2289285

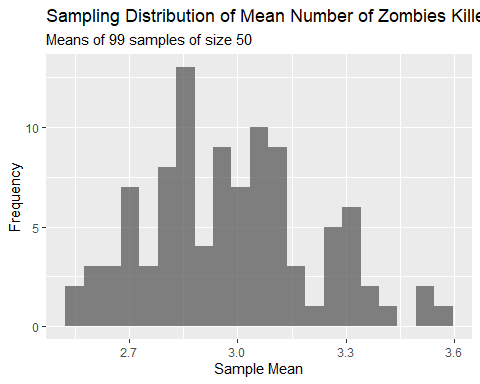
# The standard deviation of the sampling distribution is fairly similar to the standard error initially calculated from the sample (SE: 0.2587411, SD: 0.213464).  
  
# confidence interval for the sampling distribution  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- mean(m\_zombieskilled) + qnorm(alpha / 2) \* sd(m\_zombieskilled)  
upper <- mean(m\_zombieskilled) + qnorm(1 - alpha / 2) \* sd(m\_zombieskilled)  
(ci <- c(lower, upper))

## [1] 2.534137 3.431520

ci

## [1] 2.534137 3.431520

# The confidence interval for zombies killed calculated from the sampling distribution is smaller than the confidence interval estimated from the initial sample. It is also higher by about 0.3 on the lower bound and 0.15 on the upper bound.  
  
(  
 p <- ggplot(data = as.data.frame(m\_zombieskilled), aes(x = m\_zombieskilled)) +  
 geom\_histogram(binwidth = function(x) (max(m\_zombieskilled) - min(m\_zombieskilled)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Mean Number of Zombies Killed"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency")  
)



# The sampling distribution for number of zombies killed is somewhat normally distributed, but instead of peaking at closer to the median value, the shape of the distribution is somewhat bimodal. This distribution does not show the heavy right skew indicated in the distribution calculated from the sample.  
  
# 99 samples for years of education  
  
k <- 99  
n <- 50  
s <- list()  
for (i in 1:k) {  
 s[[i]] <- sample(d$years\_of\_education, size = n, replace = FALSE)  
}  
  
  
  
m\_yearsofeducation <- vector(length = k)   
for (i in 1:k) {  
 m\_yearsofeducation[[i]] <- mean(s[[i]])  
}  
  
# each sample mean  
  
m\_yearsofeducation

## [1] 2.88 2.72 2.80 3.28 3.00 3.14 3.24 3.22 2.80 3.02 2.74 2.90 2.90 2.98 3.10  
## [16] 2.82 3.02 2.54 2.80 3.10 3.44 2.90 2.78 3.04 3.22 3.16 2.58 3.06 3.00 3.10  
## [31] 2.94 2.76 3.00 2.64 3.24 2.82 3.24 2.96 2.72 3.24 3.14 3.04 2.90 3.22 2.80  
## [46] 3.32 3.26 3.34 3.22 3.26 2.92 3.14 2.86 2.72 2.60 3.10 3.10 3.06 3.02 3.22  
## [61] 3.10 3.28 2.94 2.70 2.88 3.06 2.96 3.26 3.20 3.04 3.04 3.14 2.58 3.14 3.16  
## [76] 2.90 3.18 2.74 3.16 2.98 3.16 3.00 3.26 3.08 2.64 3.24 2.78 3.04 2.96 3.12  
## [91] 3.00 3.28 3.02 3.52 2.98 2.86 2.80 2.80 2.48

# mean of the sampling distribution  
  
mean(m\_yearsofeducation)

## [1] 3.005455

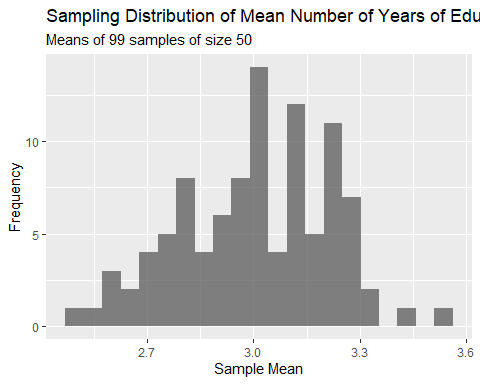
# standard deviation of the sampling distribution  
  
sd(m\_yearsofeducation)

## [1] 0.2118879

# The SD of the sampling distribution of years of education (0.20978833) is very close to the SE calculated initially (0.2157852) and is only 0.1 off.  
  
# confidence interval for the sampling distribution  
  
percent\_ci <- 95  
alpha <- 1 - percent\_ci / 100  
lower <- mean(m\_yearsofeducation) + qnorm(alpha / 2) \* sd(m\_yearsofeducation)  
upper <- mean(m\_yearsofeducation) + qnorm(1 - alpha / 2) \* sd(m\_yearsofeducation)  
(ci <- c(lower, upper))

## [1] 2.590162 3.420747

# The confidence interval estimates are about 0.5 off from each other, with the sampling distribution estimate being higher.  
  
(  
 p <- ggplot(data = as.data.frame(m\_yearsofeducation), aes(x = m\_yearsofeducation)) +  
 geom\_histogram(binwidth = function(x) (max(m\_yearsofeducation) - min(m\_yearsofeducation)) / 20, alpha = 0.75) +  
 labs(  
 title = paste0("Sampling Distribution of Mean Number of Years of Education"),  
 subtitle = paste0("Means of ", k, " samples of size ", n)  
 ) +  
 xlab("Sample Mean") +  
 ylab("Frequency")  
)



# The sampling distribution is roughly normally distributed and perhaps slightly right skewed. This is contrast to the histogram generated from the initial sample, which was not normally distributed and instead moderately right skewed.