Pengilley-Alana-ada-homework-2

Alana Pengilley

23/03/2021

Challenge 1

IMDB Movies

```r
#load in packages to use
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(readr)
library(dplyr)
library(ggplot2)
library(mosaic)
```

```
## 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
```

```
library(purrr)

#load in dataset
f <- "https://raw.githubusercontent.com/difiore/ADA-datasets/master/IMDB-movies.csv"
d <- read_csv(f, col_names = TRUE)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   tconst = col_character(),
##   titleType = col_character(),
##   primaryTitle = col_character(),
##   startYear = col_double(),
##   runtimeMinutes = col_double(),
##   genres = col_character(),
##   averageRating = col_double(),
##   numVotes = col_double(),
##   nconst = col_character(),
##   director = col_character()
## )
```

```
d <- filter(d, startYear %in% 1920:1979, runtimeMinutes >= 60, runtimeMinutes <= 180)
head(d) #first 6 rows to check dataset
```

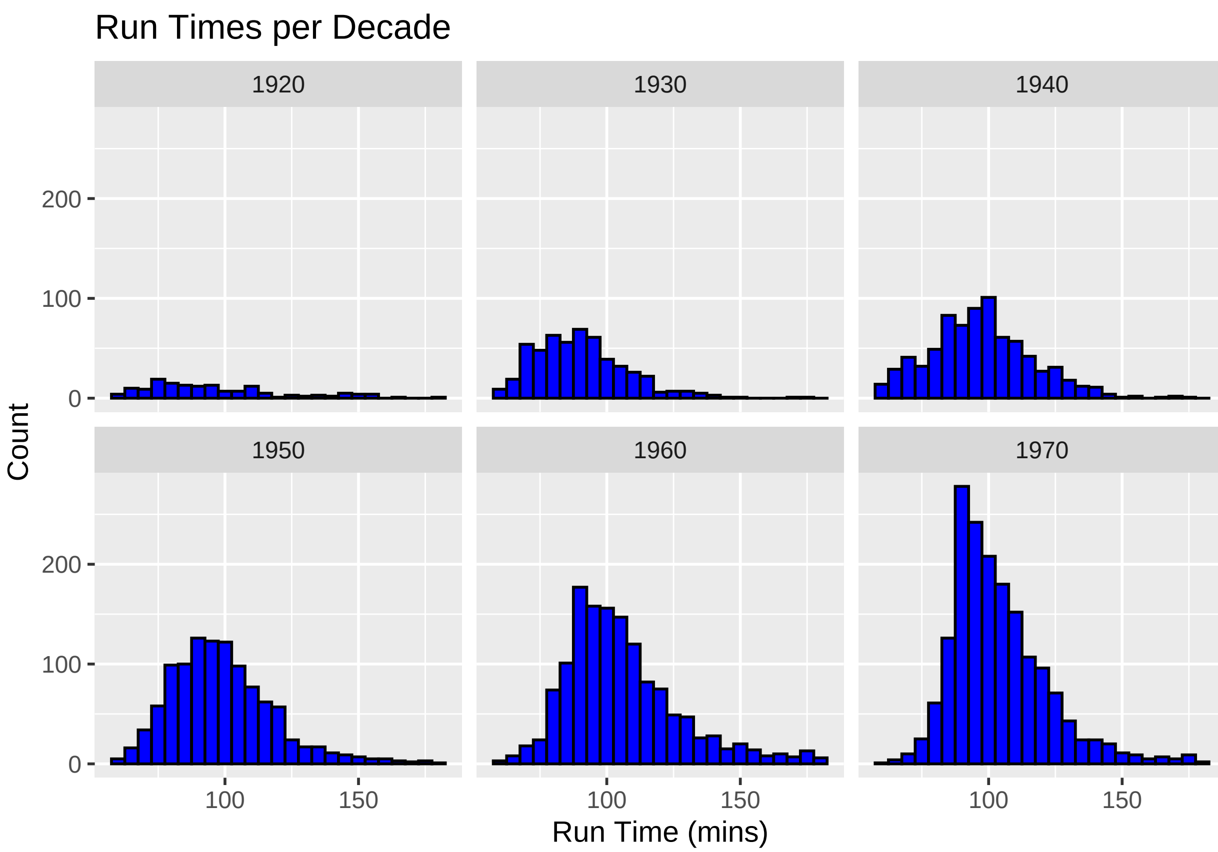
```
## # A tibble: 6 x 10
##   tconst titleType primaryTitle startYear runtimeMinutes genres averageRating
##   <chr>  <chr>     <chr>            <dbl>          <dbl> <chr>          <dbl>
## 1 tt001~ movie     The Cabinet~      1920             76 Fanta~           8.1
## 2 tt001~ movie     Leaves From~      1920            167 Drama            6.7
## 3 tt001~ movie     Dr. Jekyll ~      1920             82 Drama~           7
## 4 tt001~ movie     The Golem         1920             76 Fanta~           7.2
## 5 tt001~ movie     The Last of~      1920             73 Actio~           6.7
## 6 tt001~ movie     The Mark of~      1920            107 Adven~           7.1
## # ... with 3 more variables: numVotes <dbl>, nconst <chr>, director <chr>
```

```
d <- d %>%
  mutate("decade" = floor(startYear/10)*10) #add decade column to dataset
d
```

```
## # A tibble: 5,651 x 11
##    tconst titleType primaryTitle startYear runtimeMinutes genres averageRating
##    <chr>  <chr>     <chr>            <dbl>          <dbl> <chr>          <dbl>
##  1 tt001~ movie     The Cabinet~      1920             76 Fanta~           8.1
##  2 tt001~ movie     Leaves From~      1920            167 Drama            6.7
##  3 tt001~ movie     Dr. Jekyll ~      1920             82 Drama~           7
```

```
##   4 tt001~ movie     The Golem         1920              76 Fanta~           7.2
##   5 tt001~ movie     The Last of~      1920              73 Actio~           6.7
##   6 tt001~ movie     The Mark of~      1920             107 Adven~           7.1
##   7 tt001~ movie     The Penalty       1920              90 Crime~           7.4
##   8 tt001~ movie     The Saphead       1920              77 Comedy           6.2
##   9 tt001~ movie     Way Down Ea~      1920             145 Drama~           7.4
##  10 tt001~ movie     Why Change ~      1920              90 Comed~           6.7
## # ... with 5,641 more rows, and 4 more variables: numVotes <dbl>, nconst <chr>,
## #   director <chr>, decade <dbl>
```

```
#create histograms of run times per decade
histo_runtimes <- ggplot(data = d, aes(x = runtimeMinutes)) + geom_histogram(binwidth = 5, col = "black
  labs(title = "Run Times per Decade", x = "Run Time (mins)", y = "Count") + facet_wrap(~decade)
histo_runtimes
```




```
#population mean and SD for runtimeMins
(results <- group_by(d, decade) %>%
  summarise(avgTime = mean(runtimeMinutes),sdTime = sd(runtimeMinutes)))
```

```
## # A tibble: 6 x 3
##   decade avgTime sdTime
## *  <dbl>   <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
```

```r
#sample of 100 movies per decade and calculate mean, sd, and se for each group
set.seed(1)
n <- 100
s <- group_by(d, decade) %>% sample_n(size = n, replcae = FALSE) %>%
  summarise(single_sample_size = n(), single_sample_mean = mean(runtimeMinutes),
            single_sample_sd = sd(runtimeMinutes), single_sample_se = sd(runtimeMinutes)/sqrt(n))
s
```

```
## # A tibble: 6 x 5
##   decade single_sample_size single_sample_mean single_sample_sd single_sample_se
##    <dbl>              <int>              <dbl>            <dbl>            <dbl>
## 1   1920                100               97.9             27.7             2.77
## 2   1930                100               88.0             15.3             1.53
## 3   1940                100               94.3             18.4             1.84
## 4   1950                100              101.              18.7             1.87
## 5   1960                100              108.              23.0             2.30
## 6   1970                100              103.              15.3             1.53
```

```r
pop_mean <- results$avgTime
pop_sd <- results$sdTime
pop_se <- pop_sd/sqrt(n)

#compare sample and population stats
(c <- cbind(pop_mean, pop_sd, pop_se, s))
```

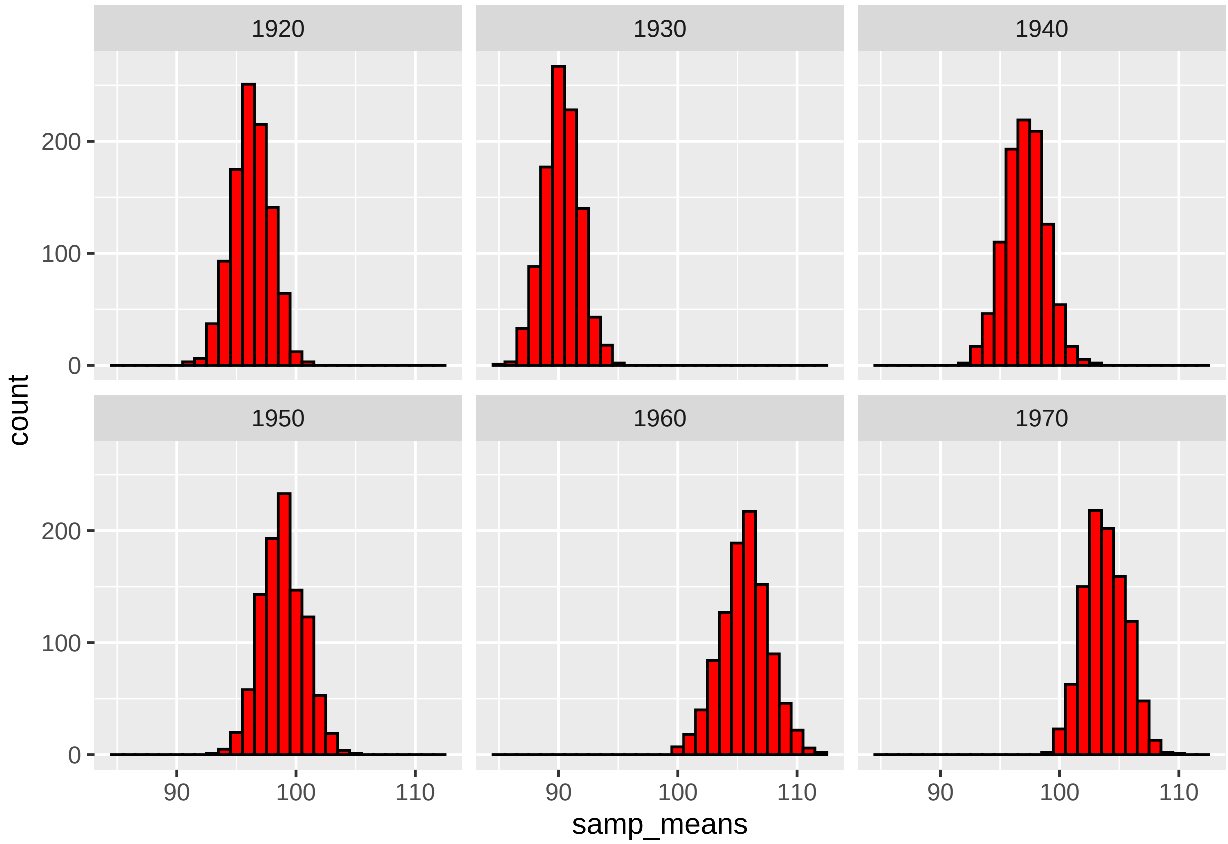
```
##    pop_mean   pop_sd   pop_se decade single_sample_size single_sample_mean
## 1  96.25658 26.20133 2.620133   1920                100              97.87
## 2  90.30000 17.28879 1.728879   1930                100              88.05
## 3  97.20332 19.12372 1.912372   1940                100              94.27
## 4  98.94820 19.20646 1.920646   1950                100             100.60
## 5 105.58586 21.23202 2.123202   1960                100             107.74
## 6 103.75000 17.95934 1.795934   1970                100             102.71
##   single_sample_sd single_sample_se
## 1         27.72836         2.772836
## 2         15.33095         1.533095
## 3         18.40556         1.840556
## 4         18.70343         1.870343
## 5         22.97720         2.297720
## 6         15.25162         1.525162
```

```r
#generate a sampling distribution of mean of runtimeMins
set.seed(1)
n <- 100
num_trials <- 1000
samp_dist_stats<- 1:num_trials %>%
  map_dfr(
    ~ group_by(d, decade) %>%
      slice_sample(n = 100)%>%
      summarize(samp_means = mean(runtimeMinutes), samp_sd = sd(runtimeMinutes))
  ) %>% mutate(n = 100)

samp_dist_stats
```

```
## # A tibble: 6,000 x 4
##    decade samp_means samp_sd     n
```

```
##     <dbl>      <dbl>   <dbl> <dbl>
##  1   1920       97.9    27.7   100
##  2   1930       88.0    15.3   100
##  3   1940       94.3    18.4   100
##  4   1950      101.     18.7   100
##  5   1960      108.     23.0   100
##  6   1970      103.     15.3   100
##  7   1920       97.5    25.3   100
##  8   1930       89.2    18.5   100
##  9   1940       94.6    19.0   100
## 10   1950      100.     19.3   100
## # ... with 5,990 more rows
```

```
#mean and sd for each decade from the sampling distribution
samp_dist_m <- aggregate(samp_dist_stats$samp_means, list(samp_dist_stats$decade), mean)
samp_dist_sd <- aggregate(samp_dist_stats$samp_sd, list(samp_dist_stats$decade), sd) #this will act as

#plot histogram of sample distribution
p <- ggplot(data = samp_dist_stats, aes(x = samp_means)) + geom_histogram(binwidth = 1, col ="black", fi
p #normal distribution
```




```
#compare standard errors
compare <- data.frame("SE estimate from 1 samp" = s$single_sample_se, "SE calculate form pop SD" = pop_
knitr::kable(compare)
```

SE.estimate.from.1.samp	SE.calculate.form.pop.SD	SE.estimate.from.samp.dist
2.772836	2.620133	1.168132
1.533095	1.728879	1.514493

SE.estimate.from.1.samp	SE.calculate.form.pop.SD	SE.estimate.from.samp.dist
1.840556	1.912372	1.412989
1.870343	1.920646	1.656194
2.297720	2.123202	1.837384
1.525162	1.795934	1.821987

Challenge 2

Every morning at the same time, a bee bioloigist goes and sits in a field in the morning and watches for forager bees returning to a hive, counting the number that arrive in a one hour window from 7am to 8am. Based on previous knowledge, she believes that the mean number of foragers that will return in that time is 12, roughly one every 6 minutes.

```
#a random variable x has poisson distribution with a mean of 12.

#P(x<=9)
ppois(q=9, lambda = 12)
```

```
## [1] 0.2423922
```

```
#P(x=0)
dpois(x=0, lambda = 12)
```

```
## [1] 6.144212e-06
```

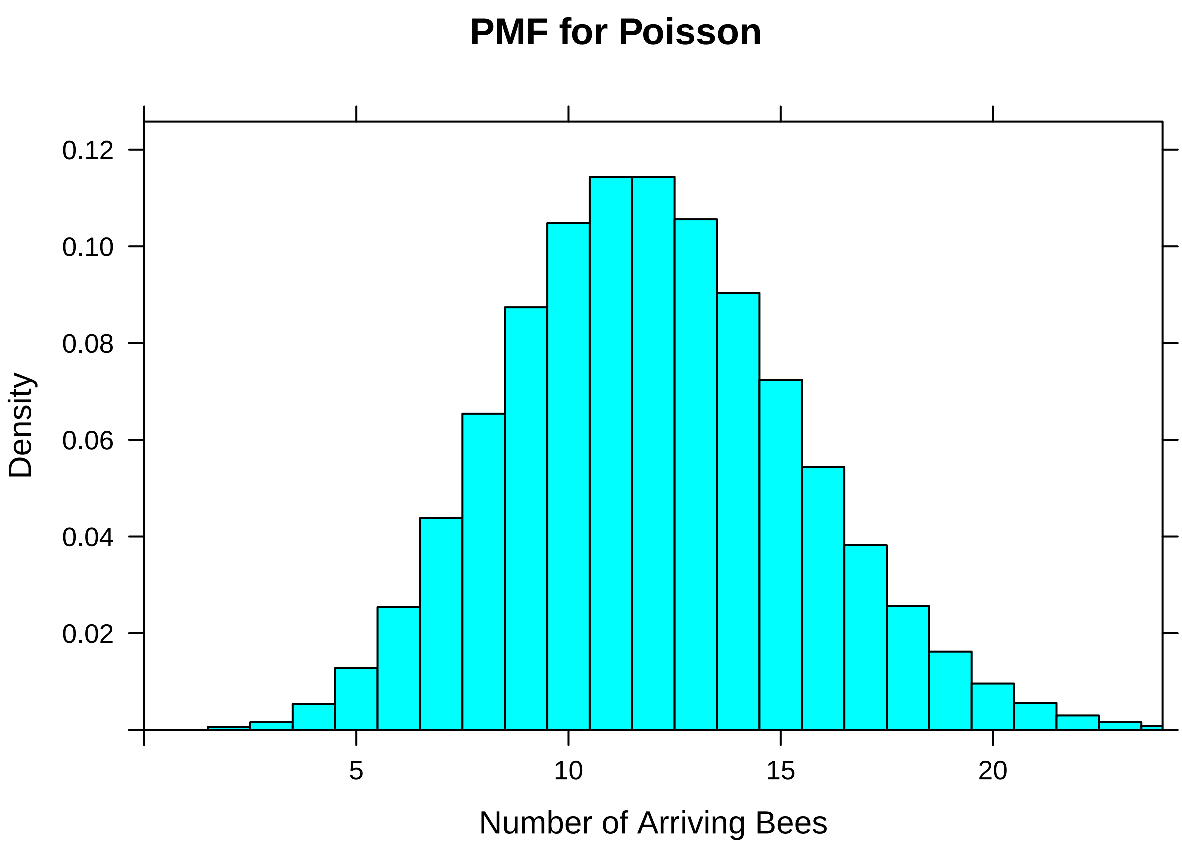
```
#P(x=5)
dpois(x=5, lambda = 12)
```

```
## [1] 0.01274064
```

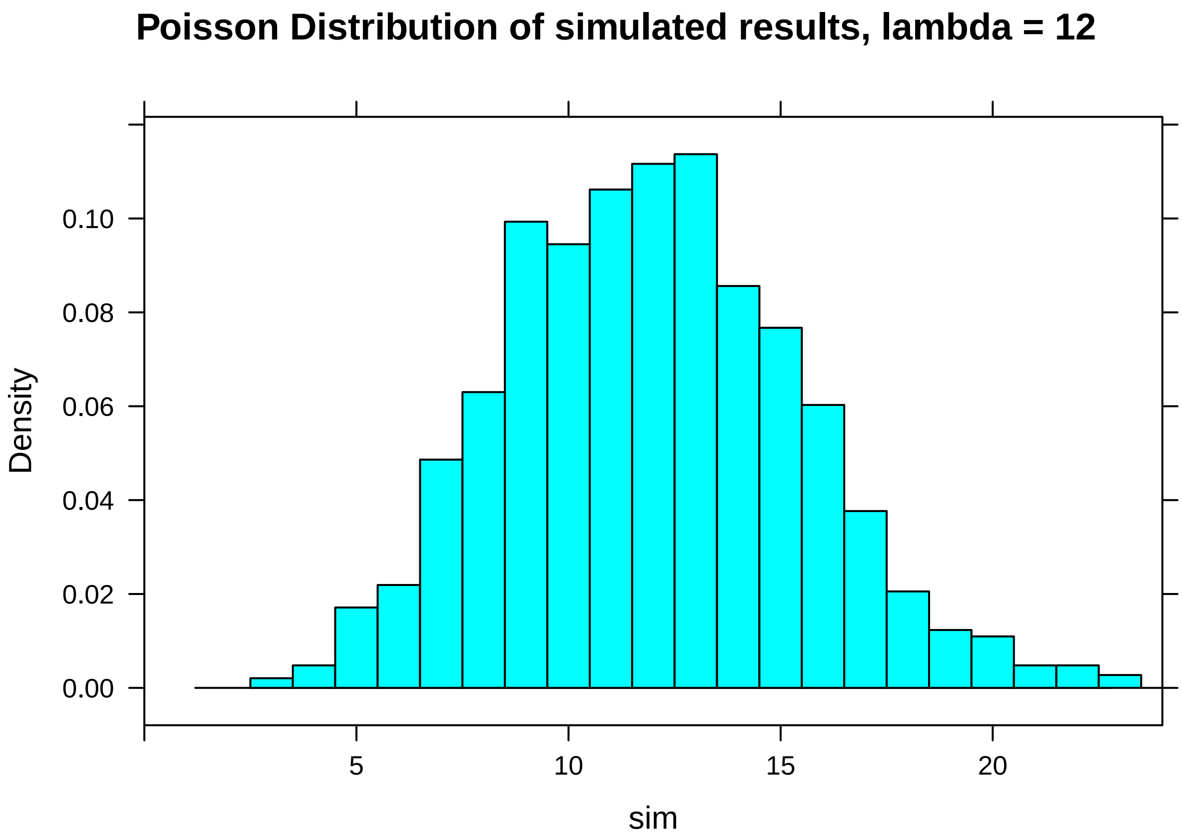
```
#P(x>18)
1 - ppois(q=18, lambda = 12)
```

```
## [1] 0.03741649
```

```
#plot relevant Poisson mass function in range 0:24
plotDist("pois", lambda = 12, xlim = c(0,24), kind = "histogram", main = "PMF for Poisson",xlab = "Numb
```




```
#simulate 1460 results from this distribution
set.seed(100)
sim <- rpois(n=1460, lambda = 12)

#histogram

p2 <- histogram(sim, xlim = c(0, 24),
  main = "Poisson Distribution of simulated results, lambda = 12",
  type = "density", width = 1, center = 12)
p2
```




```
mean(sim)
```

```
## [1] 11.97534
```

```
var(sim)
```

```
## [1] 12.12413
```

```
#the simulated results are very similar to those displayed on the probability mass function plot above.
```

##Challenge 3

Zombies data set. This data includes the first and last name and gender of the entire population of 1000 people who have survived the zombie apocalypse and are now ekeing out an existence somewhere on the East Coast, along with several other variables (height, weight, age, number of years of education, number of zombies they have killed, and college major).

```
library(ggplot2)
library(patchwork)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(radiant)
```

```
## Loading required package: radiant.data

## Loading required package: magrittr

##
## Attaching package: 'magrittr'

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

## The following object is masked from 'package:tidyr':
##
##     extract

## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

##
## Attaching package: 'radiant.data'

## The following objects are masked from 'package:lubridate':
##
##     month, wday

## The following object is masked from 'package:magrittr':
##
##     set_attr

## The following objects are masked from 'package:mosaic':
##
##     n_missing, prop

## The following object is masked from 'package:forcats':
##
##     as_factor

## The following objects are masked from 'package:purrr':
##
##     is_double, is_empty, is_numeric

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

## The following object is masked from 'package:base':
##
##     date

## Loading required package: radiant.design

## Loading required package: mvtnorm

## Loading required package: radiant.basics
```

```
## Loading required package: radiant.model

## Loading required package: radiant.multivariate
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp

## The following object is masked from 'package:patchwork':
##
##     align_plots

## The following object is masked from 'package:mosaic':
##
##     theme_map
```

```
library(purrr)

f <- "https://raw.githubusercontent.com/difiore/ada-2021-datasets/master/zombies.csv"
z <- read_csv(f, col_names = TRUE)
```

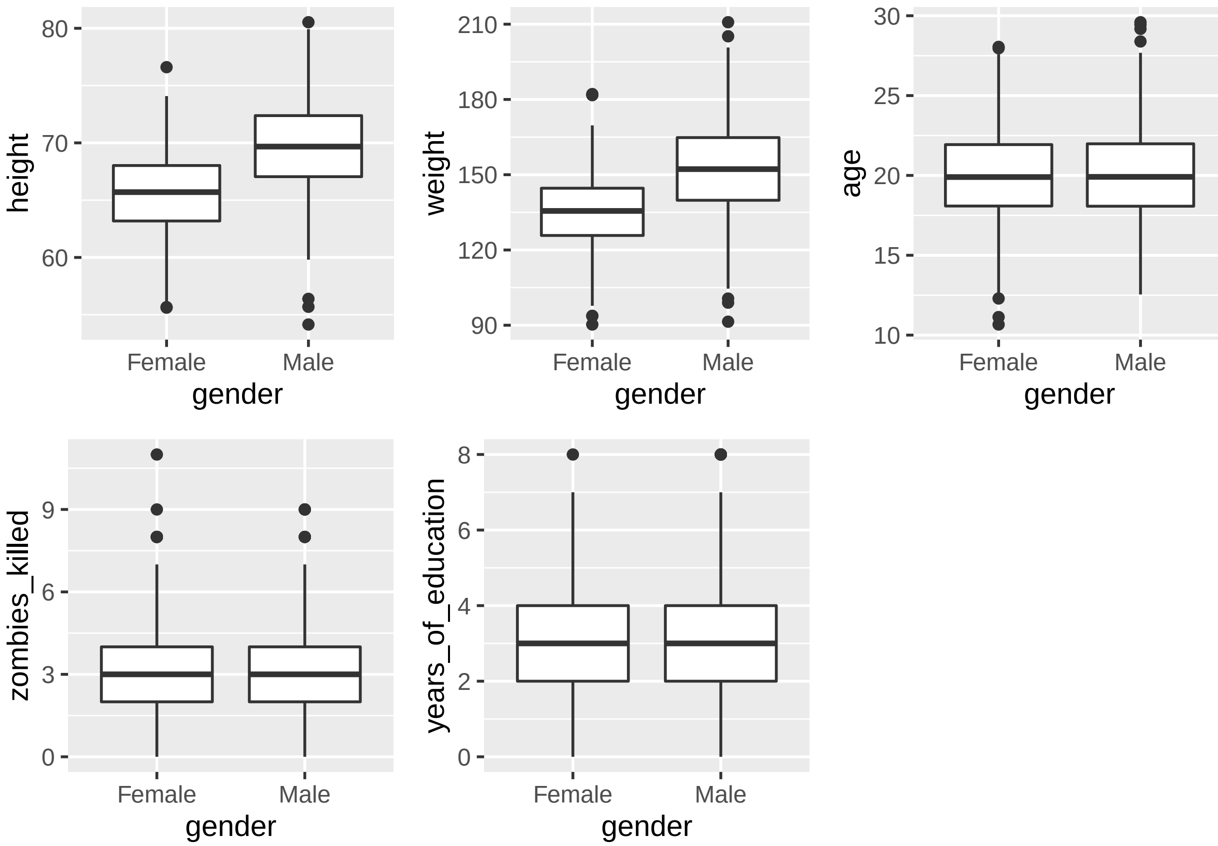
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   id = col_double(),
##   first_name = col_character(),
##   last_name = col_character(),
##   gender = col_character(),
##   height = col_double(),
##   weight = col_double(),
##   zombies_killed = col_double(),
##   years_of_education = col_double(),
##   major = col_character(),
##   age = col_double()
## )
```

```
survivors <- select(z, "gender", "height", "weight", "age", "zombies_killed", "years_of_education")

#population mean and SD for each quantitative random variable

pop_mean <- summarise(survivors, mean(height), mean(weight), mean(age), mean(zombies_killed), mean(years

pop_sd <- summarise(survivors, sdpop(height), sdpop(weight), sdpop(age), sdpop(zombies_killed), sdpop(ye

pop_summary<- data.frame(variable=c("Height", "Weight", "Age", "Kills", "Education"), MEAN=t(pop_mean),
knitr::kable(pop_summary)
```

	variable	MEAN	SD
mean(height)	Height	67.63010	4.307970
mean(weight)	Weight	143.90748	18.391857
mean(age)	Age	20.04696	2.963583
mean(zombies_killed)	Kills	2.99200	1.747551

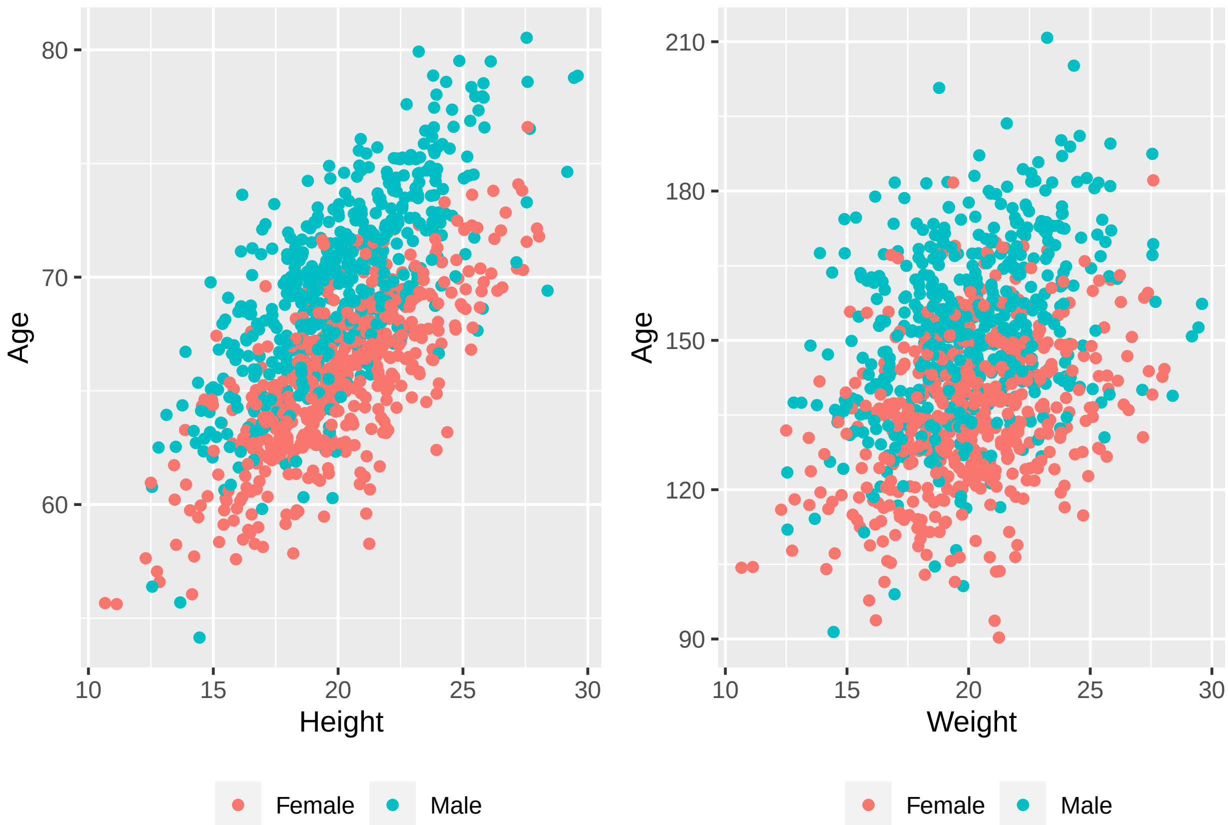
	variable	MEAN	SD
mean(years_of_education)	Education	2.99600	1.675704

```
#boxplots of each variable by gender
b1 <- ggplot(survivors, aes(x = gender, y = height)) + geom_boxplot()
b2 <- ggplot(survivors, aes(x = gender, y = weight)) + geom_boxplot()
b3 <- ggplot(survivors, aes(x = gender, y = age)) + geom_boxplot()
b4 <- ggplot(survivors, aes(x = gender, y = zombies_killed)) + geom_boxplot()
b5 <- ggplot(survivors, aes(x = gender, y = years_of_education)) + geom_boxplot()
plot_grid(b1, b2, b3, b4, b5, ncol = 3)
```




```
#height and weight in relation to age by gender
p1 <- ggplot(data = survivors, aes(
  x = age,
  y = height,
  color = factor(gender)))
p1 <- p1 + xlab("Height") + ylab("Age")
p1 <- p1 + geom_point(na.rm = TRUE)
p1 <- p1 + theme(legend.position = "bottom", legend.title = element_blank())

p2 <- ggplot(data = survivors, aes(
  x = age,
  y = weight,
  color = factor(gender)))
p2 <- p2 + xlab("Weight") + ylab("Age")
p2 <- p2 + geom_point(na.rm = TRUE)
```

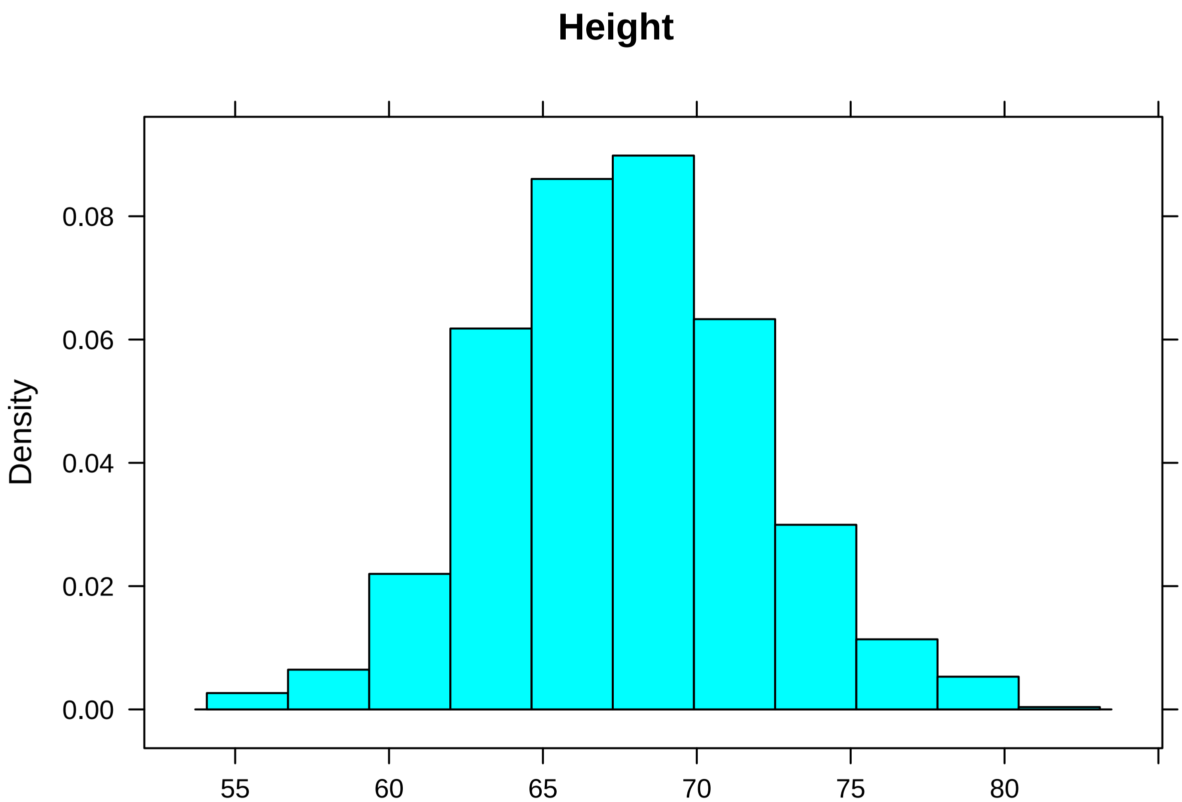
```
p2 <- p2 + theme(legend.position = "bottom", legend.title = element_blank())

plot_grid(p1, p2, nrow = 1)
```



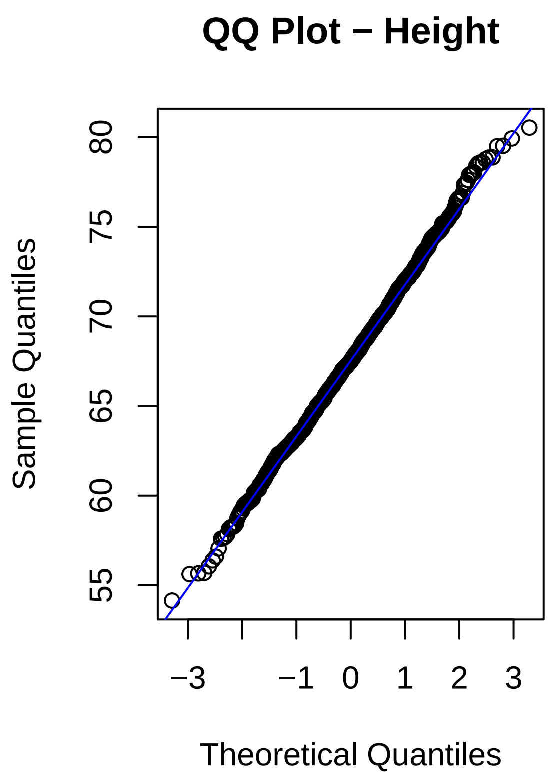
Do these variables (age, weight and height) seem related? Yes, both variables (height and weight) increase with age. —

Using histograms and Q-Q plots, check whether the quantitative variables seem to be drawn from a normal distribution. W
(normal = height, weight, age), (Possion = kills and education)

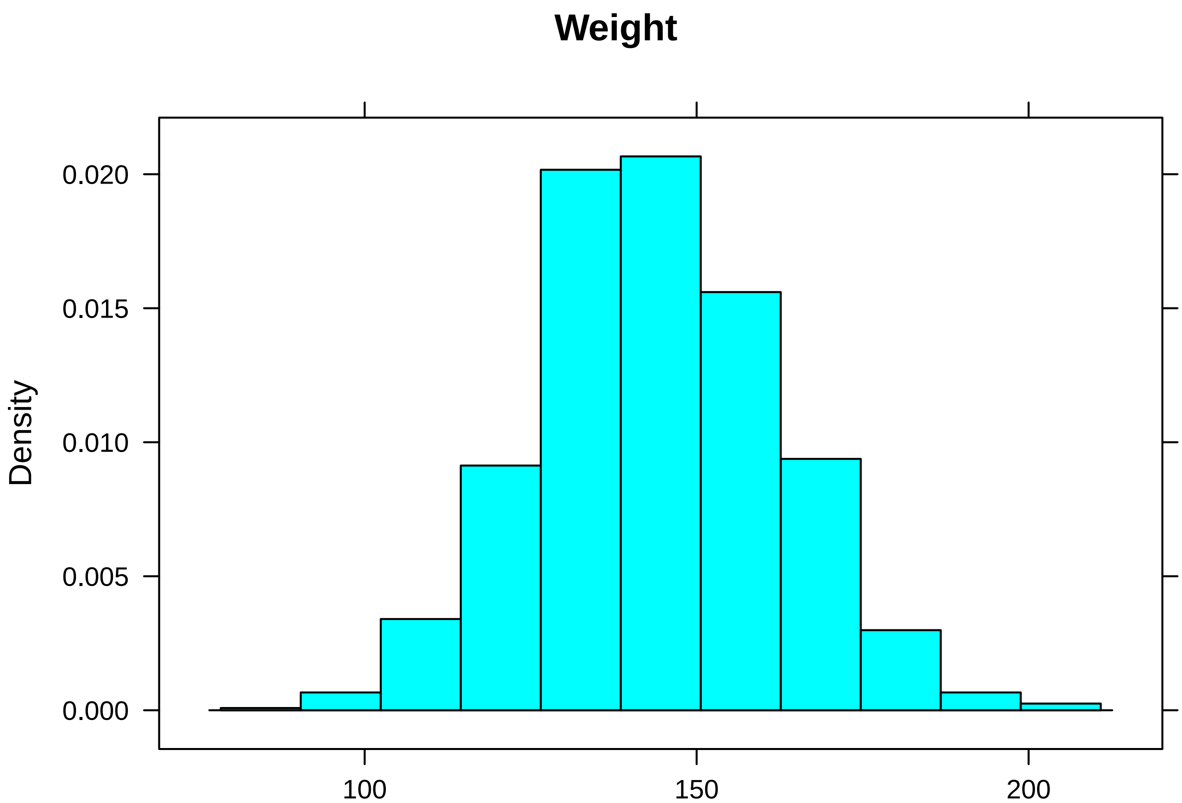
```
#Height
par(mfrow= c(1,2))
histogram(survivors$height, main = "Height", xlab = "")
```




```
qqnorm(survivors$height, main = "QQ Plot - Height")
qqline(survivors$height, col = "blue")
#normal

#Weight
par(mfrow= c(1,2))
```

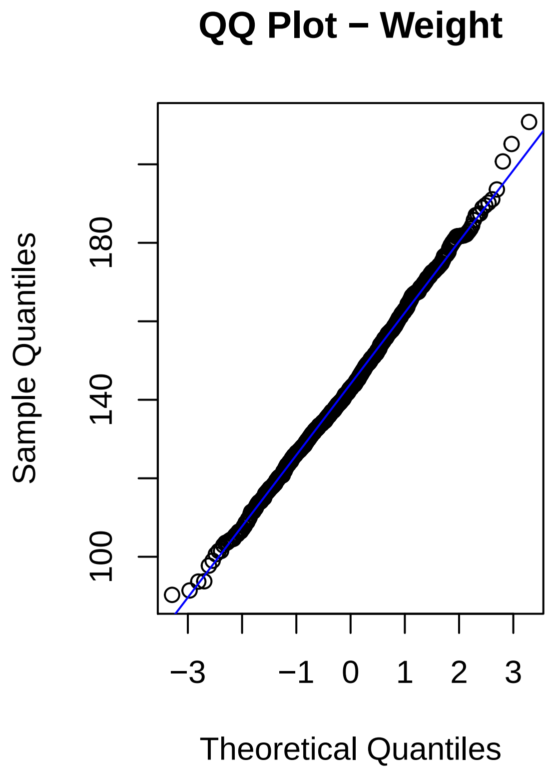



```
histogram(survivors$weight, main = "Weight", xlab = "")
```

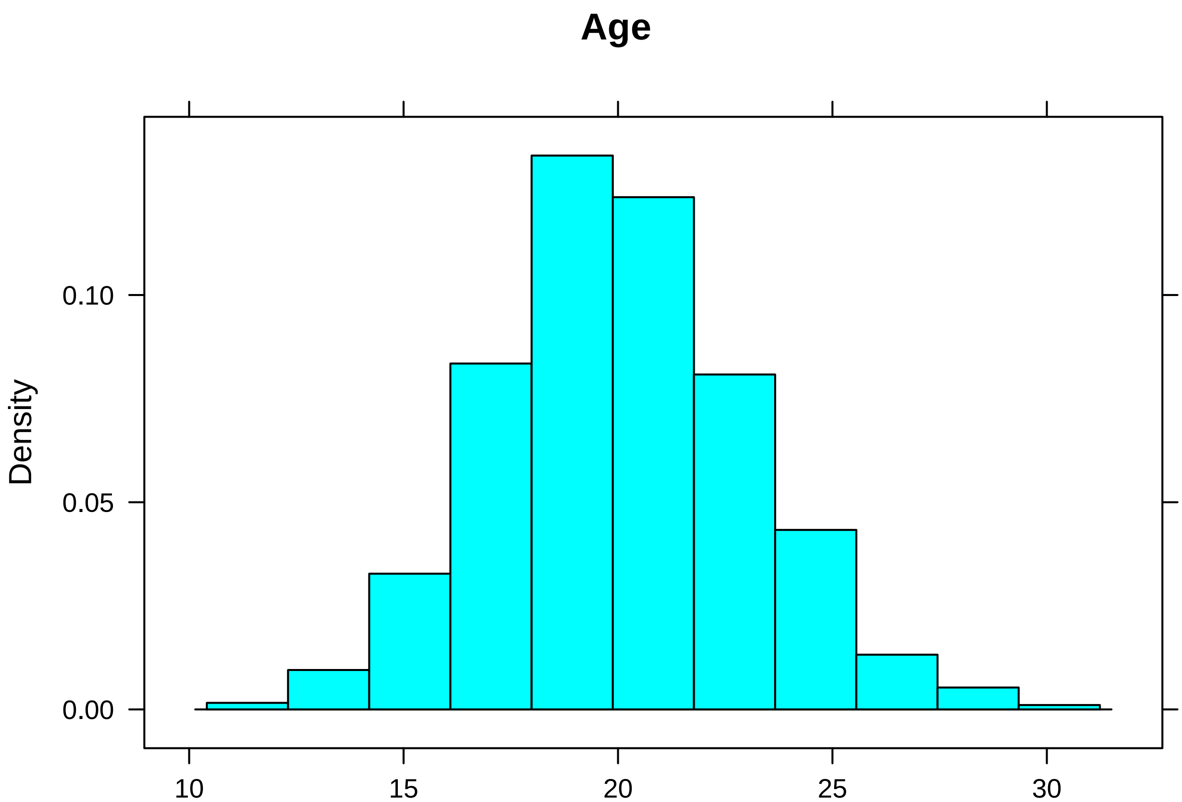



```
qqnorm(survivors$weight, main = "QQ Plot - Weight")
qqline(survivors$weight, col = "blue")
```

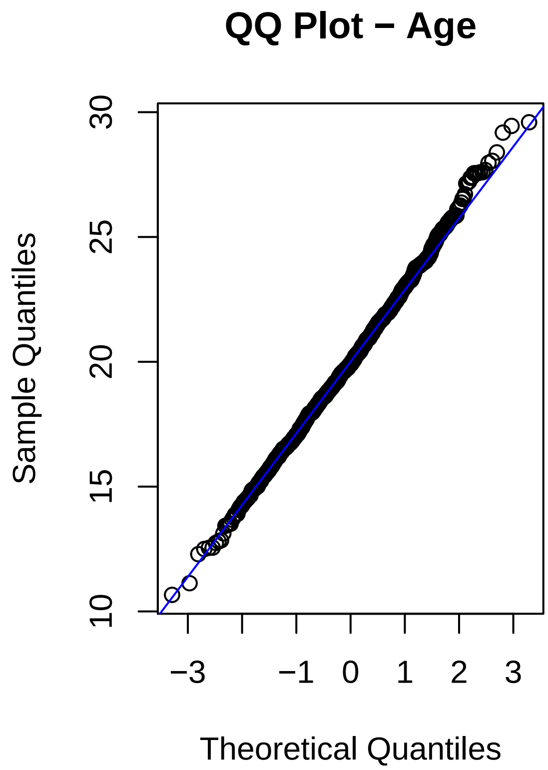
```r
#normal

#Age
par(mfrow= c(1,2))
```



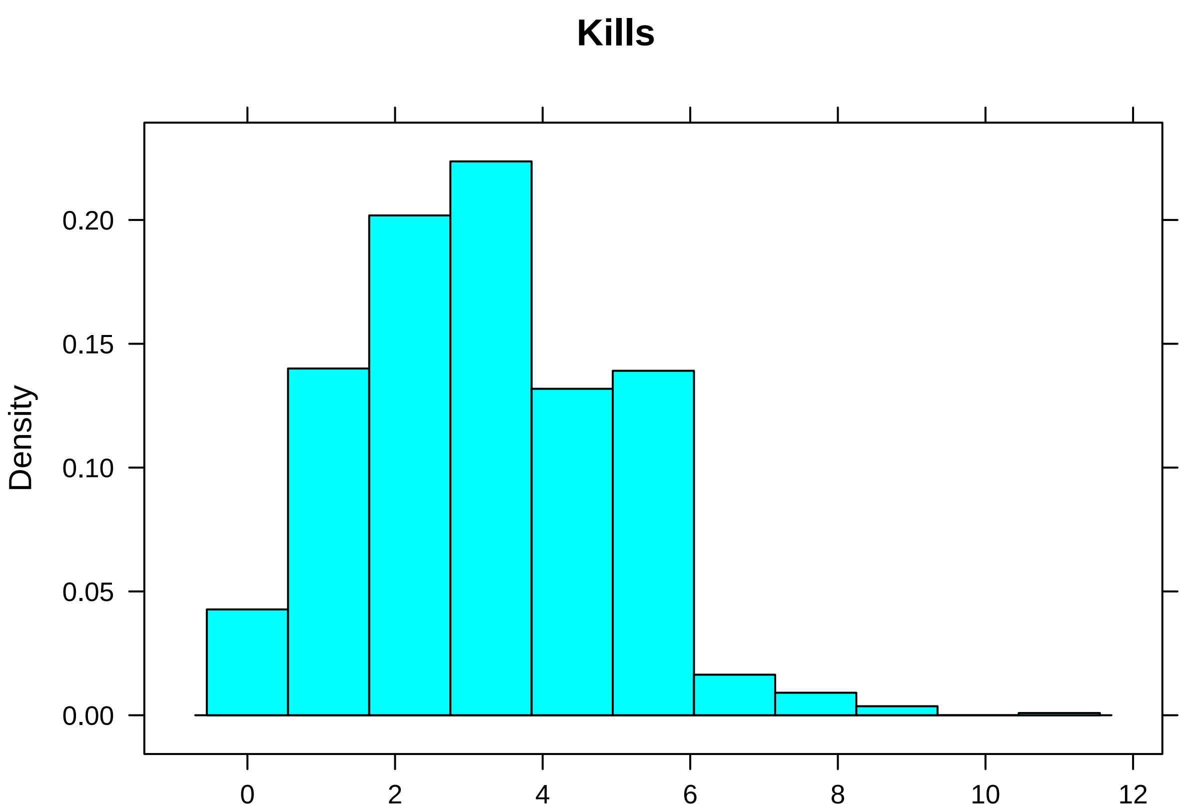

```r
histogram(survivors$age, main = "Age", xlab = "")
```




```
qqnorm(survivors$age, main = "QQ Plot - Age")
qqline(survivors$age, col = "blue")
#normal

#Number of zombies killed
par(mfrow= c(1,2))
```

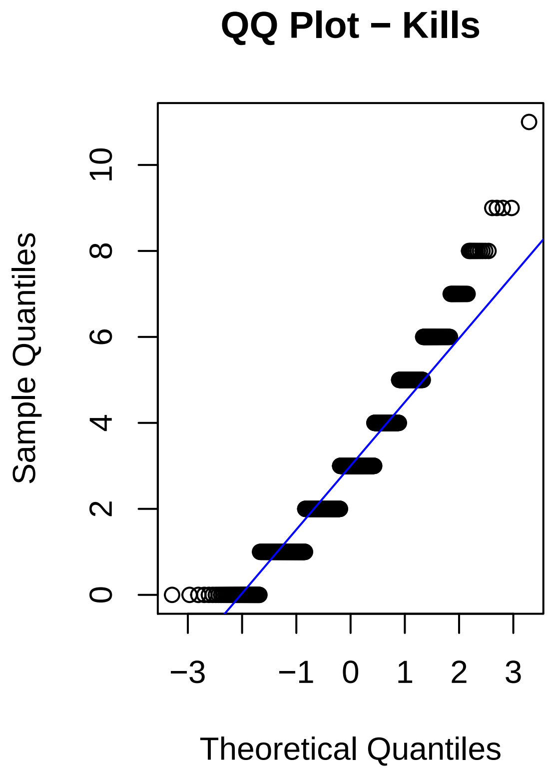



```
histogram(survivors$zombies_killed, main = "Kills", xlab = "")
```

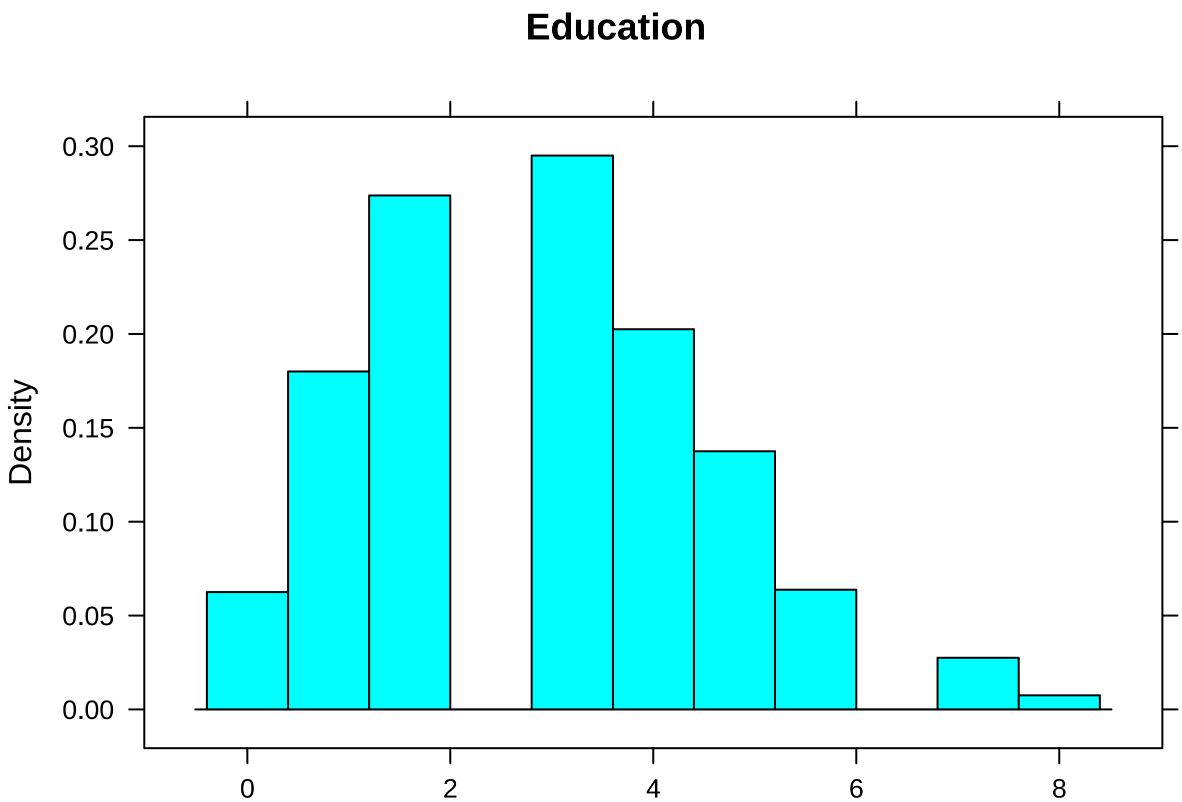



```
qqnorm(survivors$zombies_killed, main = "QQ Plot - Kills")
qqline(survivors$zombies_killed, col = "blue")
```

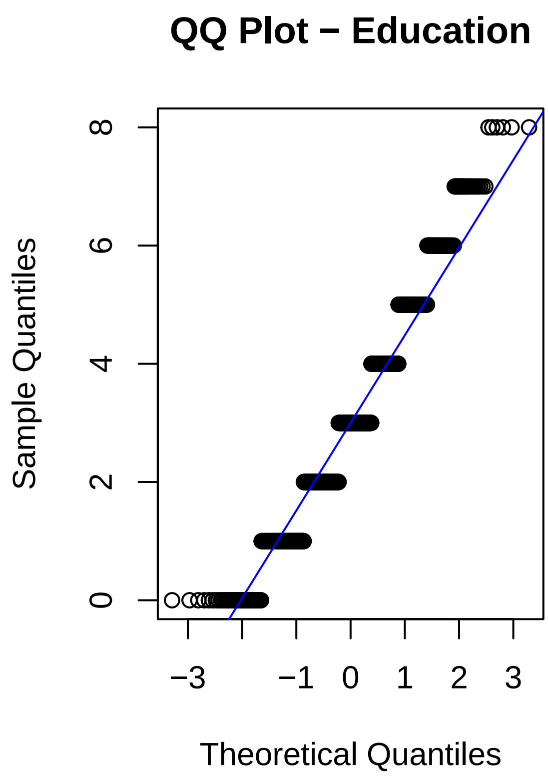
```r
#Poisson

#Years of education
par(mfrow= c(1,2))
```




```r
histogram(survivors$years_of_education, main = "Education", xlab = "")
```




```
qqnorm(survivors$years_of_education, main = "QQ Plot - Education")
qqline(survivors$years_of_education, col = "blue")
#Poisson
```



Sample ONE subset of 50 zombie apocalypse survivors (without replacement) from this population and calculate the mean and sample standard deviation for each variable. Also estimate the standard error for each variable based on this sample and use that to construct a 95% confidence interval for each mean.

```
#sample of 50 survivors
set.seed(1)
n <- 50
s <- sample_n(survivors, size = n, replace = FALSE)

#means
s_means <- summarise(s, across(c(height, weight, age, zombies_killed, years_of_education), list(mean=mea
#sds
s_sd <- summarise(s, height_sd = sd(height),weight_sd = sd(weight),age_sd = sd(age),kills_sd = sd(zombi
#(note: tried to use across function for sd but got an error message i couldn't solve)

#function for SE
SE <- function(x, type="normal") {
  if (type == "normal") {
    SE <- sd(x)/sqrt(length(x))
  }
  if (type == "poisson"){
    SE<- sqrt(mean(x)/length(x))
  }
  return(SE)
} #need to distinguish between distribution type in this function

s_se <- tibble(SE(s$height), SE(s$weight),SE(s$age), SE(s$zombies_killed, type = "poisson"), SE(s$years_
```

```r
#Confidence interval functions
alpha = 0.05
CI.norm <- function(x){
  CI <- mean(x) + c(-1, 1) * qnorm(1-alpha/2) * SE(x) #se function - check there is one
  names(CI) <- c("Lower", "Upper")
  return(CI)
} #normal

CI.sim <- function(x){
  sim <- NULL
  for (i in 1:1000){
    sim[i] <- mean(sample(x), length(x), replace = T)
  }
  CI <- quantile(sim, c(alpha/2, 1-alpha/2))
names(CI) <- c("Lower", "Upper")
return(CI)
} #simulation

#CI based on normal distribution
s_CI.norm <- rbind(CI.norm(s$height), CI.norm(s$weight), CI.norm(s$age), CI.norm(s$zombies_killed), CI.n

#CI based on simulation
s_CI.sim <- rbind(CI.sim(s$height), CI.sim(s$weight), CI.sim(s$age), CI.sim(s$zombies_killed), CI.sim(s$

single_samp_summary <- data.frame(Variable=c("Height", "Weight", "Age", "Kills", "Education"), MEAN = t
```

Then draw another 99 random samples of 50 zombie apocalypse survivors out and calculate the mean for each of the these s

---

```r
#99 random samples of 50 zombies
set.seed(1)
k <- 99 # number of samples
n <- 50 # size of each sample
sample <- 1:k %>%
  map_dfr(
    ~ survivors %>%
      slice_sample(n = n)%>%
      summarize(height_mean = mean(height), weight_mean =mean(weight), age_mean = mean(age), zombies_kil
  ) %>% mutate(n = 50)
sample
```

```
## # A tibble: 99 x 6
##    height_mean weight_mean age_mean zombies_killed_me~ years_of_education~     n
##          <dbl>       <dbl>    <dbl>              <dbl>               <dbl> <dbl>
##  1        67.3        143.     20.1               3.08                3.04    50
##  2        68.0        145.     20.3               3.04                3.18    50
##  3        66.8        142.     19.1               3.08                2.72    50
##  4        68.5        147.     20.6               2.96                2.72    50
##  5        68.4        147.     20.5               2.82                3.04    50
##  6        67.8        147.     19.8               2.98                2.64    50
##  7        67.8        147.     20.0               3.08                2.54    50
##  8        67.2        142.     19.9               3.88                3.1     50
##  9        66.8        142.     20.0               2.96                2.36    50
## 10        67.0        141.     19.7               2.5                 3.3     50
```

```
## # ... with 89 more rows
```

```
(samp_dist <- bind_rows(sample, s_means)) #combine single sample and sample of 99 into one df
```

```
## # A tibble: 100 x 6
##    height_mean weight_mean age_mean zombies_killed_me~ years_of_education~     n
##          <dbl>       <dbl>    <dbl>              <dbl>               <dbl> <dbl>
##  1        67.3        143.     20.1               3.08                3.04    50
##  2        68.0        145.     20.3               3.04                3.18    50
##  3        66.8        142.     19.1               3.08                2.72    50
##  4        68.5        147.     20.6               2.96                2.72    50
##  5        68.4        147.     20.5               2.82                3.04    50
##  6        67.8        147.     19.8               2.98                2.64    50
##  7        67.8        147.     20.0               3.08                2.54    50
##  8        67.2        142.     19.9               3.88                3.1     50
##  9        66.8        142.     20.0               2.96                2.36    50
## 10        67.0        141.     19.7               2.5                 3.3     50
## # ... with 90 more rows
```

```
samp_dist_mean <- samp_dist %>% summarise(height = mean(height_mean), weight = mean(weight_mean), age =

#calculate mean of SE for variables from each sample
se_fun <- function(x){
  sd(x)/sqrt(100)
} #create a function for standard error
samp_dist_se <- samp_dist %>% summarise(height = se_fun(height_mean), weight = se_fun(weight_mean), age

comparison <- data.frame(Variables =c("Height", "Weight", "Age", "Kills", "Education"), "Mean_Samp_Dist
```

Finally, construct an 95% confidence interval for each mean directly from the sampling distribution of sample means using t

---

```
#95% CI for each mean from the sampling dist.
percent_ci <- 95
alpha <- 1 - percent_ci / 100
lower <- samp_dist_mean + qnorm(alpha / 2) * samp_dist_se
upper <- samp_dist_mean + qnorm(1 - alpha / 2) * samp_dist_se
ci <- c(lower, upper)
```

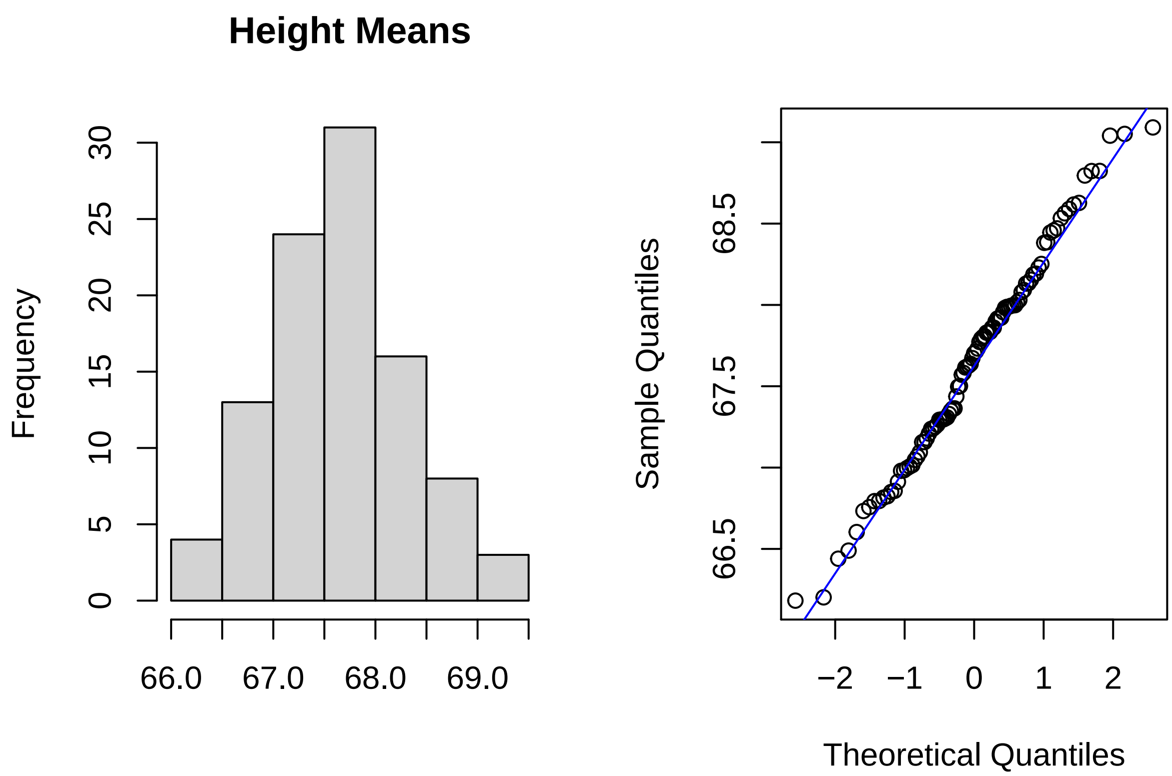
How do the standard deviations of the sampling distribution for each variable compare to the standard errors estimated fro

---

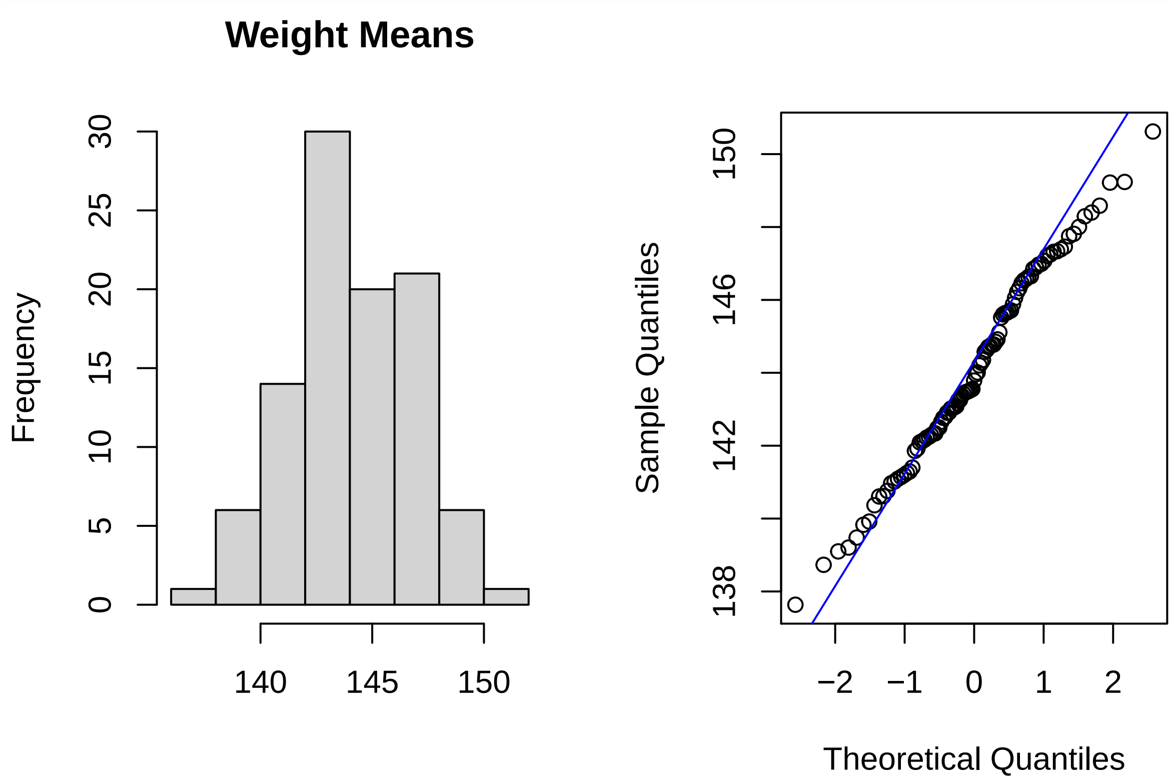
```
compare <- data.frame(Variables=c("Height", "Weight", "Age", "Kills", "Education"), "Mean of Samp Dist"

#All are about the same. The SE for each variable is similar to the population estimate
```

What do sampling distributions for each variable mean look like? Are they normally distributed?

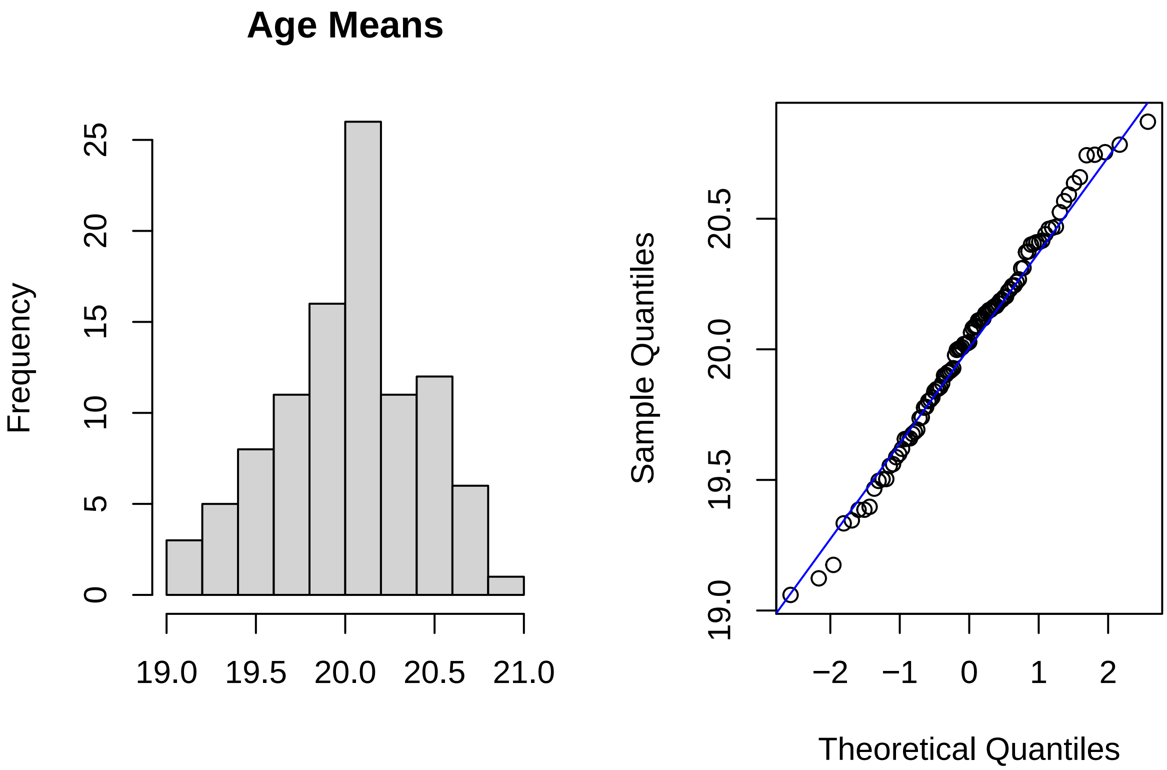
```
par(mfrow = c(1,2))
hist(sample$height_mean, main = "Height Means", xlab = "") #height
qqnorm(sample$height_mean, main = "")
qqline(sample$height_mean, col = "blue")
```



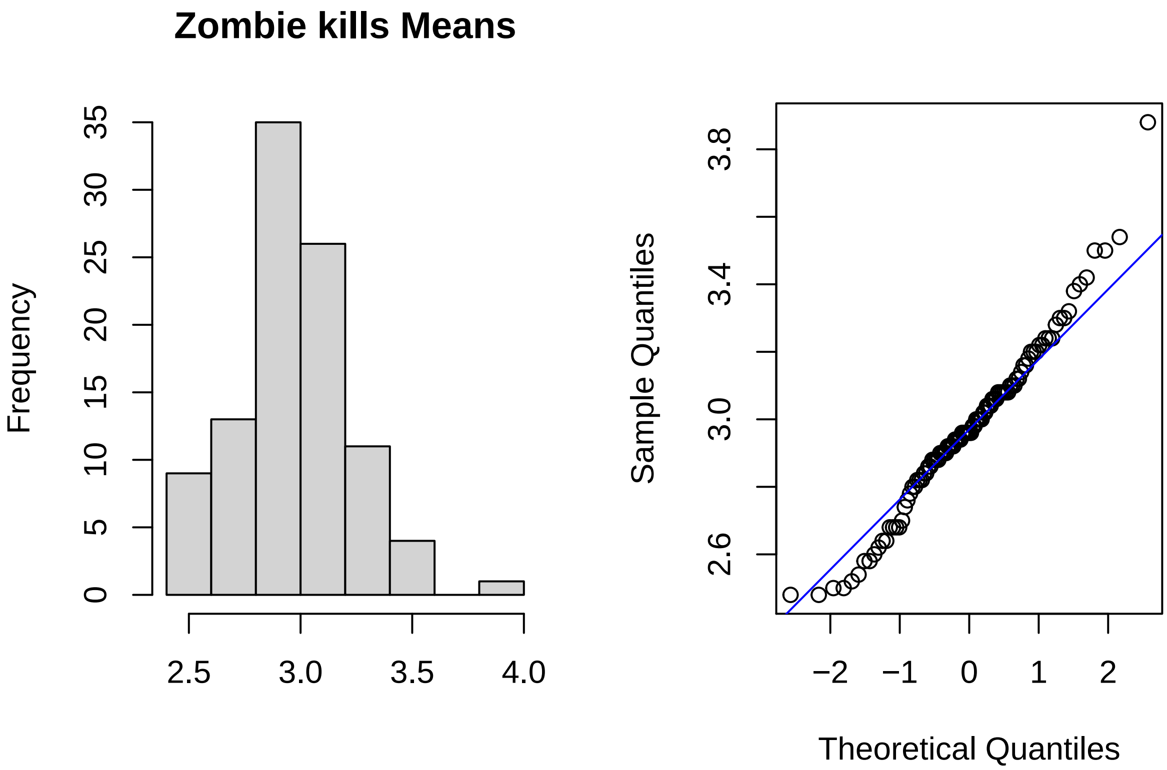

```
hist(sample$weight_mean, main = "Weight Means", xlab = "") #weight
qqnorm(sample$weight_mean, main = "")
qqline(sample$weight_mean, col = "blue")
```



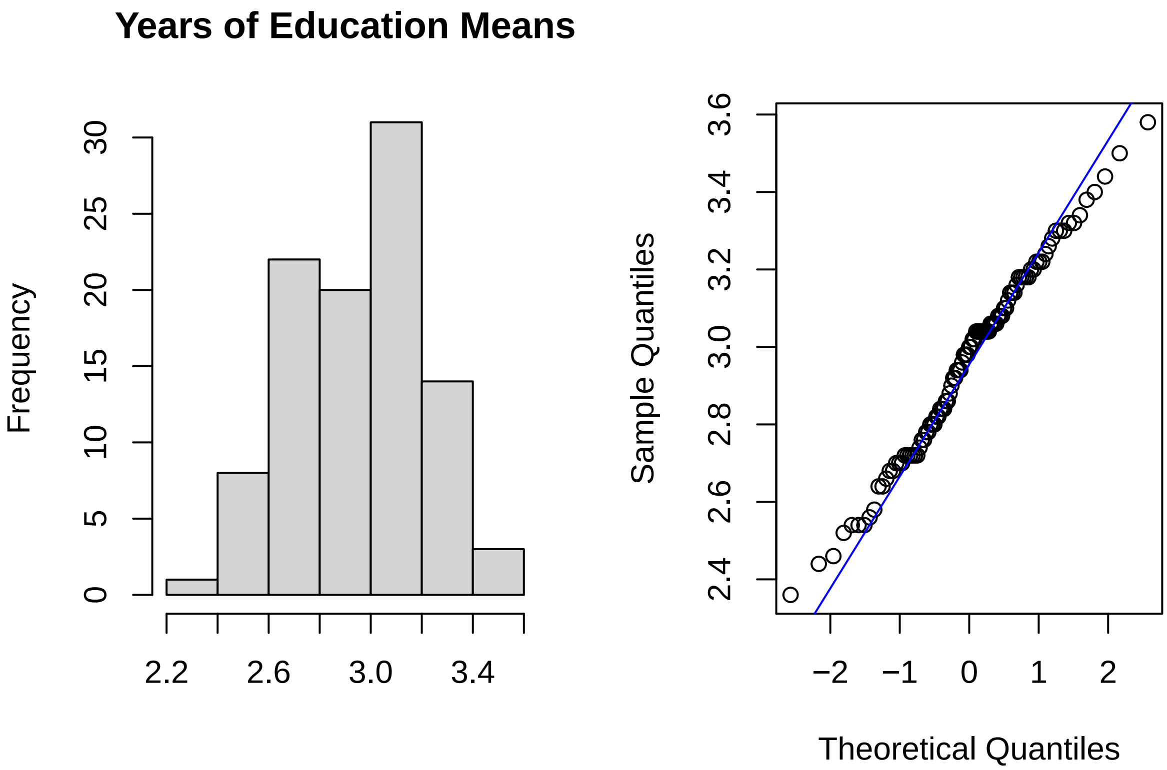
```r
hist(sample$age_mean, main = "Age Means", xlab = "") #age
qqnorm(sample$age_mean, main = "")
qqline(sample$age_mean, col = "blue")
```




```r
hist(sample$zombies_killed_mean, main = "Zombie kills Means", xlab = "") #kills
qqnorm(sample$zombies_killed_mean, main = "")
qqline(sample$zombies_killed_mean, col = "blue")
```




```
hist(sample$years_of_education_mean, main = "Years of Education Means", xlab = "") #edu
qqnorm(sample$years_of_education_mean, main = "")
qqline(sample$years_of_education_mean, col = "blue")
```



What about for those variables that you concluded were not originally drawn from a normal distribution? All these distirbutions are normal, the two varibales that were previously poisson are now normal too. — — How do the two 95% CIs you estimated compare to one another (i.e., the CI based on one sample and its estimated standard deviation versus the CI based on simulation)? —

```
cbind(s_CI.norm, s_CI.sim) #based on normal distribution and based on simulation
```

```
##            Lower      Upper     Lower     Upper
## [1,]  66.088109  68.515543  67.28762  67.28762
## [2,] 137.698908 149.233583 142.96071 142.96071
## [3,]  19.226751  20.948423  19.84796  19.84796
## [4,]   2.567250   3.592750   3.00000   3.00000
## [5,]   2.606378   3.473622   3.00000   3.00000
```