ADA Homework 2

Samantha Rabinowitz, sar4357

3/3/2020

Challenge 1

The R code below will load the 'movies.csv' dataset from GitHub and organize the data into a tibble.

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

```
## Parsed with 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()
## )
```

```
glimpse(d)
```

```
## Observations: 28,938
## Variables: 10
## $ tconst         <chr> "tt0002130", "tt0002844", "tt0003037", "tt0003165", ...
## $ titleType      <chr> "movie", "movie", "movie", "movie", "movie", "movie"...
## $ primaryTitle   <chr> "Dante's Inferno", "Fantômas: In the Shadow of the G...
## $ startYear      <dbl> 1911, 1913, 1913, 1913, 1913, 1914, 1914, 1914, 1914...
## $ runtimeMinutes <dbl> 68, 54, 61, 90, 85, 78, 148, 59, 61, 82, 195, 59, 72...
## $ genres         <chr> "Adventure,Drama,Fantasy", "Crime,Drama", "Crime,Dra...
## $ averageRating  <dbl> 7.0, 7.0, 7.0, 7.0, 6.5, 6.5, 7.1, 6.9, 6.2, 6.3, 6....
## $ numVotes       <dbl> 2082, 1877, 1307, 1010, 1686, 1068, 2907, 1126, 1207...
## $ nconst         <chr> "nm0078205", "nm0275421", "nm0275421", "nm0275421", ...
## $ director       <chr> "Francesco Bertolini", "Louis Feuillade", "Louis Feu...
```

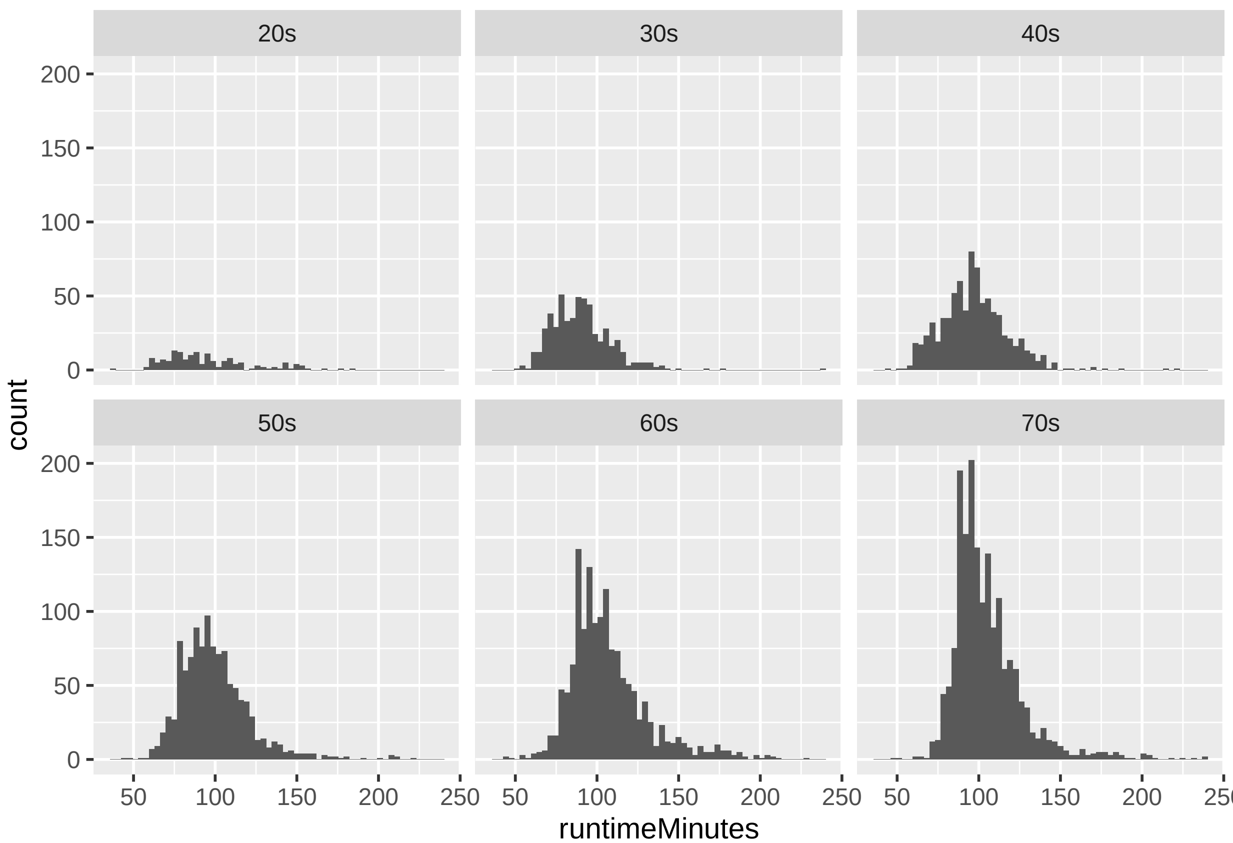
The following code will filter the dataset to just include movies from 1920 to 1979 and movies that are less than 4 hours long. Columns were also added to make **startYear** a new variable called **decade**.

```r
d1 <- d %>%
  filter(startYear >= "1920" & startYear <="1979" & runtimeMinutes < 240) %>%
  mutate(decade = case_when(startYear >=1920 & startYear<=1929 ~ "20s",
                            startYear >=1930 & startYear<=1939 ~ "30s",
                            startYear >=1940 & startYear<=1949 ~ "40s",
                            startYear >=1950 & startYear<=1959 ~ "50s",
                            startYear >=1960 & startYear<=1969 ~ "60s",
                            startYear >=1970 & startYear<=1979 ~ "70s"))
d1 %>% glimpse()
```

```
## Observations: 5,741
## Variables: 11
## $ tconst         <chr> "tt0010323", "tt0011000", "tt0011130", "tt0011237", ...
## $ titleType      <chr> "movie", "movie", "movie", "movie", "movie", "movie"...
## $ primaryTitle   <chr> "The Cabinet of Dr. Caligari", "Leaves From Satan's ...
## $ startYear      <dbl> 1920, 1920, 1920, 1920, 1920, 1920, 1920, 1920, 1920...
## $ runtimeMinutes <dbl> 76, 167, 82, 76, 73, 107, 90, 77, 145, 90, 79, 75, 1...
## $ genres         <chr> "Fantasy,Horror,Mystery", "Drama", "Drama,Horror,Sci...
## $ averageRating  <dbl> 8.1, 6.7, 7.0, 7.2, 6.7, 7.1, 7.4, 6.2, 7.4, 6.7, 6....
## $ numVotes       <dbl> 52649, 1047, 4561, 6128, 1063, 2053, 1927, 1356, 479...
## $ nconst         <chr> "nm0927468", "nm0003433", "nm0731910", "nm0091380", ...
## $ director       <chr> "Robert Wiene", "Carl Theodor Dreyer", "John S. Robe...
## $ decade         <chr> "20s", "20s", "20s", "20s", "20s", "20s", "20s", "20...
```

The code below utilizes *ggplot* to plot histograms of the distribution of **runtimeMinutes** for each decade.

```r
d1 %>%
  ggplot(aes(x=runtimeMinutes)) + geom_histogram(bins=60) + facet_wrap(~ decade)
```



The R code below will compute the population mean and population standard deviation in **runtimeMinutes** for each decade and store the values in a new dataframe, *results*.

```
results <- d1 %>% group_by(decade) %>% summarize(mean=mean(runtimeMinutes),
                                                 pop_sd=sdpop(runtimeMinutes))
results %>% glimpse()
```

```
## Observations: 6
## Variables: 3
## $ decade <chr> "20s", "30s", "40s", "50s", "60s", "70s"
## $ mean   <dbl> 95.95513, 90.24254, 97.30341, 99.64168, 106.82781, 105.06640
## $ pop_sd <dbl> 27.43508, 18.64468, 20.55356, 21.54956, 24.30888, 21.35976
```

The following code will generate a function to calculate the standard error of the mean for each decade as well as a single sample of 100 movies from each decade and calculate the sample mean and standard deviation for each decade. Additionally, the SE around each population mean for each decade will be estimated using the standard deviation and sample size of these samples.

```
std_error <- function(x) {
  sd(x) / sqrt(length(x))
}
d1 %>% group_by(decade) %>% sample_n(100, replace=FALSE) %>%
  summarize(mean(runtimeMinutes), sd(runtimeMinutes),std_error(runtimeMinutes))
```

```
## # A tibble: 6 x 4
```

```
##   decade `mean(runtimeMinutes)` `sd(runtimeMinutes)` `std_error(runtimeMinutes)`
##   <chr>                  <dbl>                <dbl>                       <dbl>
## 1 20s                     95.8                 29.2                        2.92
## 2 30s                     88.8                 19.1                        1.91
## 3 40s                     96.4                 23.4                        2.34
## 4 50s                    104.                  26.7                        2.67
## 5 60s                    109.                  28.3                        2.83
## 6 70s                    104.                  20.7                        2.07
```

The code below will write a function to calculate the standard error of the mean for each deacde using the population standard deviation for purposes of comparison to the values obtained from the sample created above.

```
pop_std_error <- function(x) {
  sdpop(x) / (sqrt(100))
}
d1 %>% group_by(decade) %>%
  summarize(mean = mean(runtimeMinutes),
            sdpop = sdpop(runtimeMinutes),
            pop_se = pop_std_error(runtimeMinutes))
```

```
## # A tibble: 6 x 4
##   decade  mean sdpop pop_se
##   <chr>  <dbl> <dbl>  <dbl>
## 1 20s     96.0  27.4   2.74
## 2 30s     90.2  18.6   1.86
## 3 40s     97.3  20.6   2.06
## 4 50s     99.6  21.5   2.15
## 5 60s    107.   24.3   2.43
## 6 70s    105.   21.4   2.14
```

The mean **runtimeMinutes** for the sample and population are very close to each other. The sample standard deviations and standard errors are both consistently somewhat greater than those for the population.

The following will generate a sampling distribution of mean **runtimeMinutes** for each decade by (a) drawing 10,000 samples of 100 movies from each decade and, for each sample, (b) calculating the mean **runtimeMinutes** and the standard deviation in **runtimeMinutes**.

```
k <- 10000
n <- 100
s <- list()
t <- list()
  for (i in unique(d1$decade)) {
    d2 <- filter(d1, decade==i)
    s[[i]] <- do(k) * mean(~runtimeMinutes, data = sample_n(d2, size = n, replace = FALSE))
    t[[i]] <- do(k) * sd(~runtimeMinutes, data = sample_n(d2, size = n, replace = FALSE))
  }
head(s[[1]])
```

```
##    mean
## 1 96.92
## 2 94.12
## 3 94.39
```

```
## 4 96.62
## 5 93.31
## 6 98.03
```

```
head(t[[1]])
```

```
##         sd
## 1 26.48750
## 2 29.35157
## 3 26.84867
## 4 27.61732
## 5 29.28538
## 6 26.45631
```

Challenge 2

```
ppois(13, lambda = 18)
```

```
## [1] 0.1425978
```

```
dpois(0, lambda = 18)
```

```
## [1] 1.522998e-08
```

```
dpois(7, lambda = 18)
```

```
## [1] 0.00185002
```

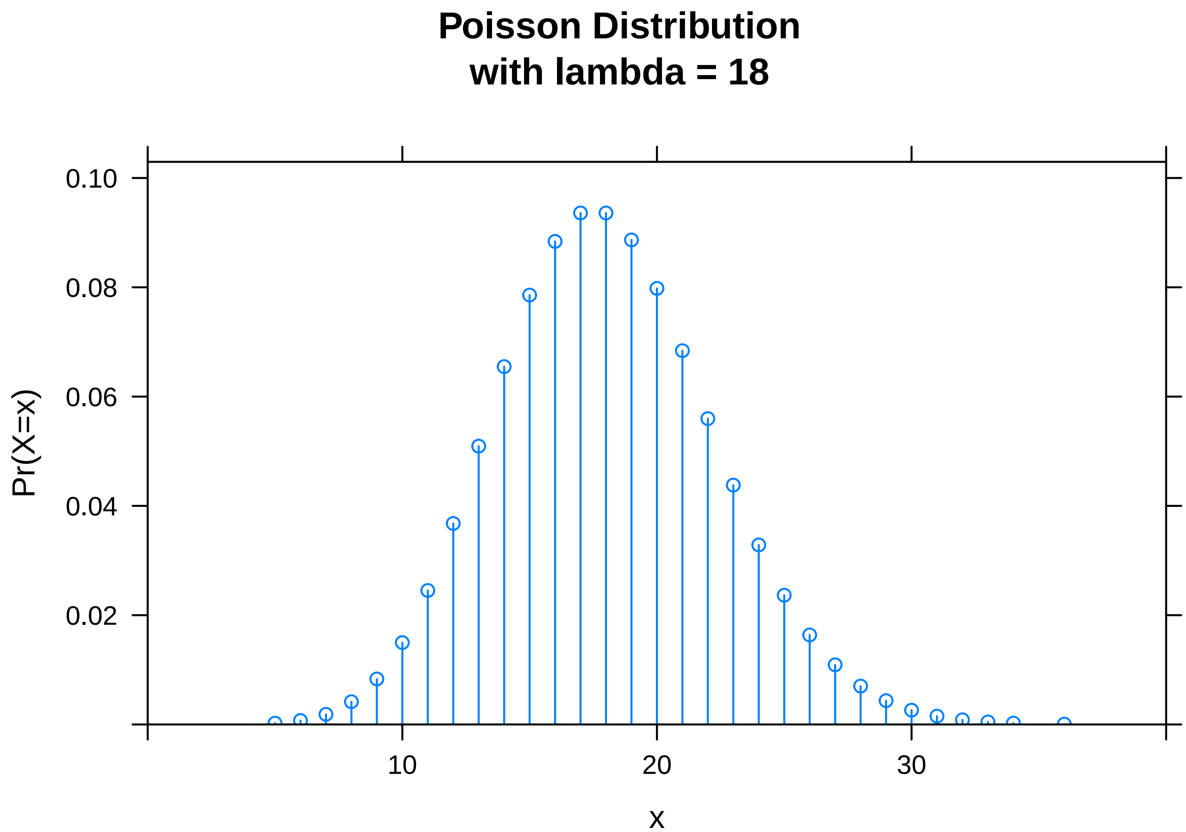
```
1-ppois(20, lambda = 18)
```

```
## [1] 0.2692798
```

The probability that she will hear 13 or fewer calls is 0.1425978. The probability that she will hear no calls is 1.522998e-08. The probability that she will hear exactly 7 calls is 0.00185002. The probability tha she will hear 20 calls or more 0.2692798.

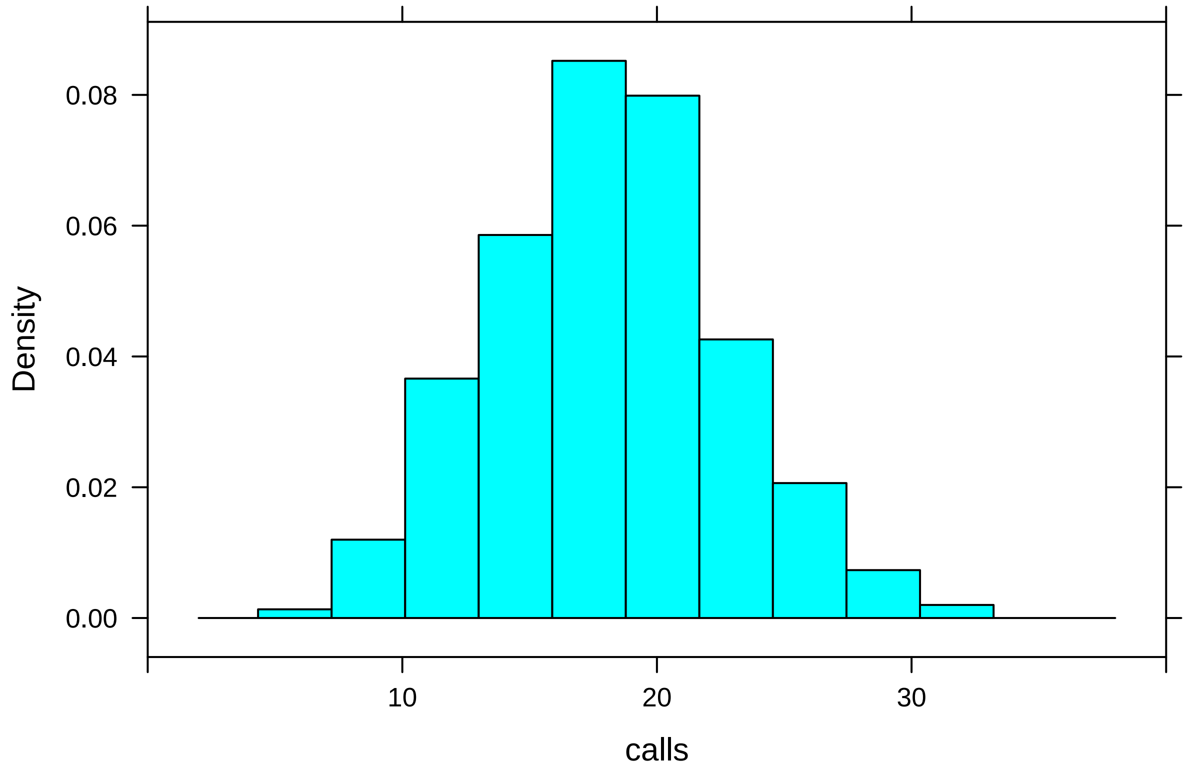
The below code will plot a Poisson mass function with a lambda value of 18 over an x range of 0 to 40.

```
plotDist("pois", lambda = 18, main = "Poisson Distribution\nwith lambda = 18", xlab = "x", xlim = c(0:4
```



The following R code will simulate 520 (10 years of Saturday call omnitoring sessions) results from the above distribution. Following creation of these results, a histogram will be plotted to compare to the shape of the above probability mass function.

```
calls <- rpois(520, lambda = 18)
histogram(calls, xlim=c(0:40))
```



The general shape of the histogram of simulated results above reflects what is seen in the Poisson probability mass function. If the simulation included a greater number of results, this histogram may more closely match what is shown in the probability ass function plot.

Challenge 3

The R code below will load the 'zombies.csv' dataset from GitHub and organize the data into a tibble.

```
f <- "https://raw.githubusercontent.com/difiore/ADA-datasets/master/zombies.csv"
d <- read_csv(f, col_names = T)
```

```
## Parsed with 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()
## )
```

```r
glimpse(d)
```

```
## Observations: 1,000
## Variables: 10
## $ id                 <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
## $ first_name         <chr> "Sarah", "Mark", "Brandon", "Roger", "Tammy", "A...
## $ last_name          <chr> "Little", "Duncan", "Perez", "Coleman", "Powell"...
## $ gender             <chr> "Female", "Male", "Male", "Male", "Female", "Mal...
## $ height             <dbl> 62.88951, 67.80277, 72.12908, 66.78484, 64.71832...
## $ weight             <dbl> 132.0872, 146.3753, 152.9370, 129.7418, 132.4265...
## $ zombies_killed     <dbl> 2, 5, 1, 5, 4, 1, 0, 4, 9, 2, 4, 4, 2, 5, 4, 2, ...
## $ years_of_education <dbl> 1, 3, 1, 6, 3, 4, 4, 0, 3, 3, 4, 3, 1, 5, 5, 2, ...
## $ major              <chr> "medicine/nursing", "criminal justice administra...
## $ age                <dbl> 17.64275, 22.58951, 21.91276, 18.19058, 21.10399...
```

The code below will calculate the population mean and standard deviation for each quantitative random variable.

```r
d %>% summarize(height_mean=mean(height),
                weight_mean=mean(weight),
                age_mean=mean(age),
                n_zombies_mean=mean(zombies_killed),
                ed_mean=mean(years_of_education))
```

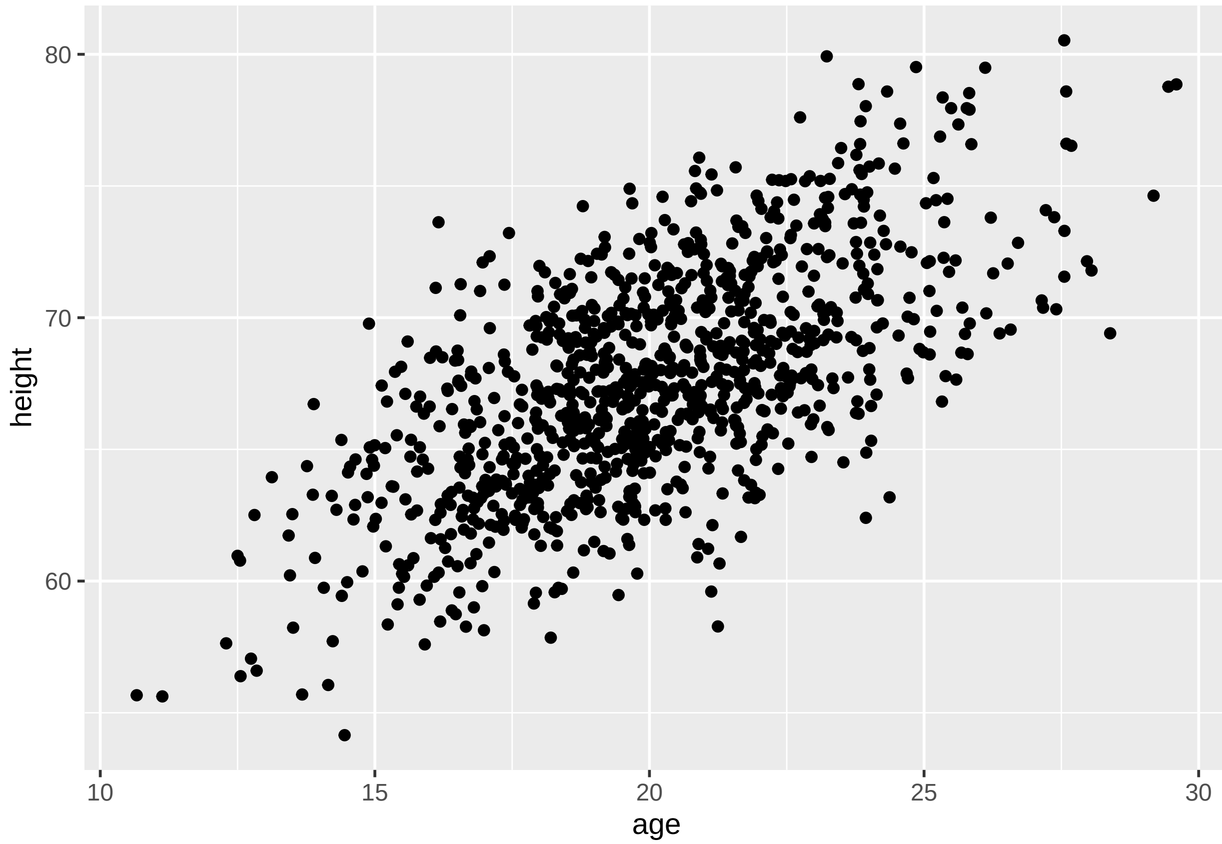
```
## # A tibble: 1 x 5
##   height_mean weight_mean age_mean n_zombies_mean ed_mean
##         <dbl>       <dbl>    <dbl>          <dbl>   <dbl>
## 1        67.6        144.     20.0           2.99    3.00
```

```r
d %>% summarize(height_sd=sdpop(height),
                weight_sd=sdpop(weight),
                age_sd=sdpop(age),
                n_zombies_sd=sdpop(zombies_killed),
                ed_sd=sdpop(years_of_education))
```

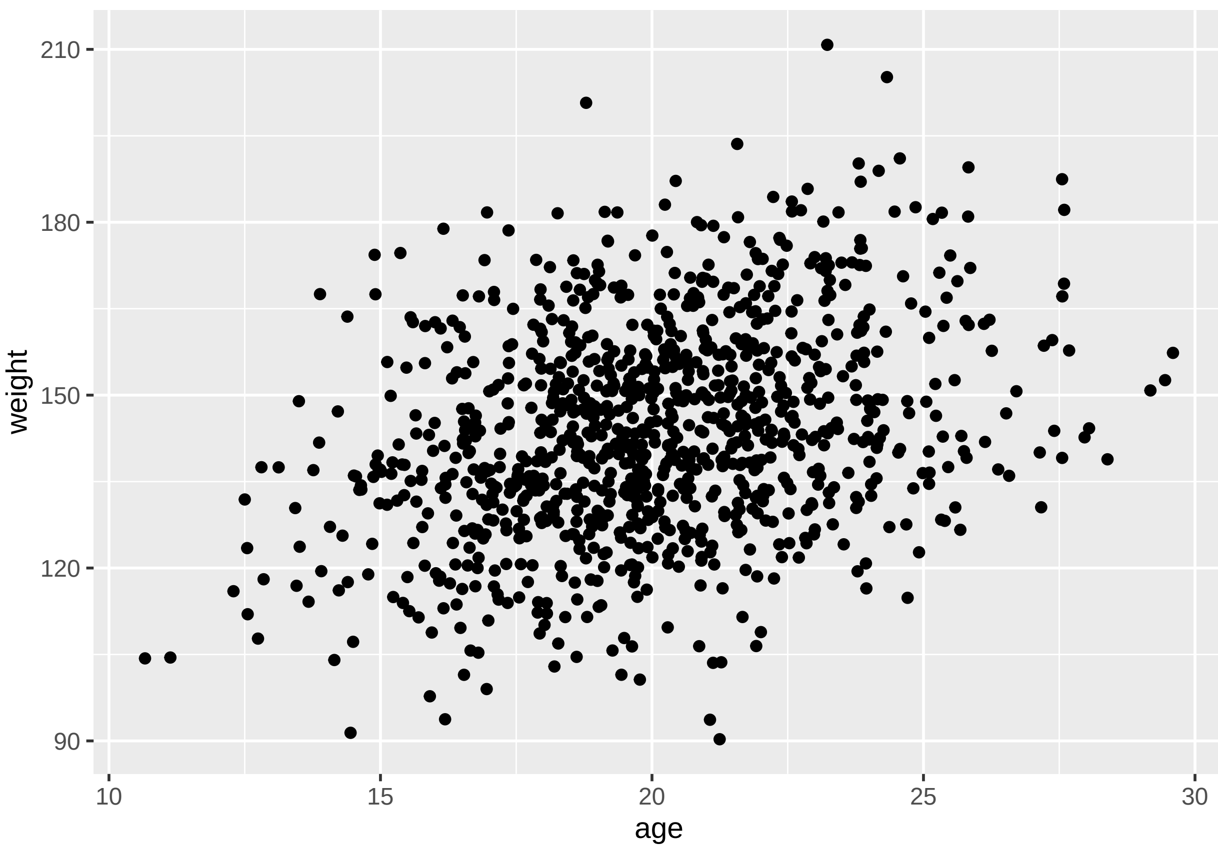
```
## # A tibble: 1 x 5
##   height_sd weight_sd age_sd n_zombies_sd ed_sd
##       <dbl>     <dbl>  <dbl>        <dbl> <dbl>
## 1      4.31      18.4   2.96         1.75  1.68
```

The following will utilize *ggplot2* to make scatterplots of height and weight in relation to age.

```r
d %>% ggplot(aes(x=age,y=height)) + geom_point()
```



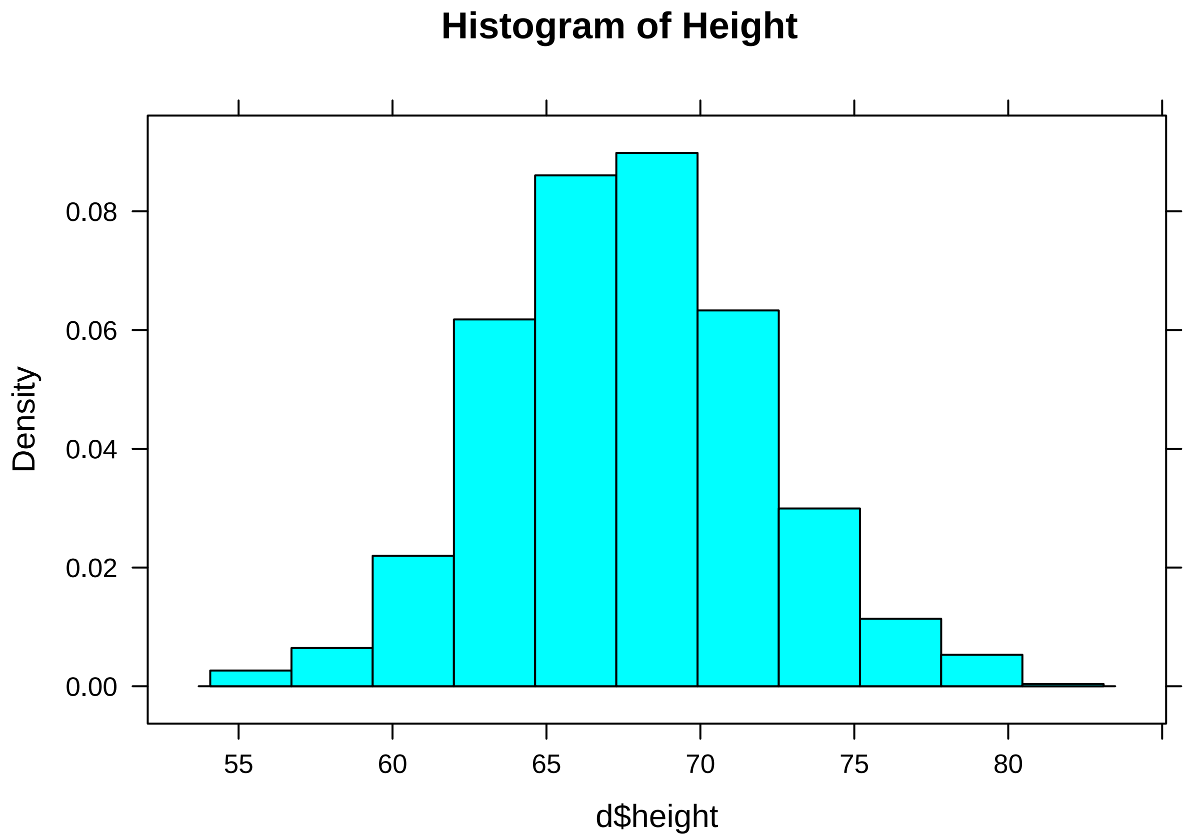

```
d %>% ggplot(aes(x=age,y=weight)) + geom_point()
```



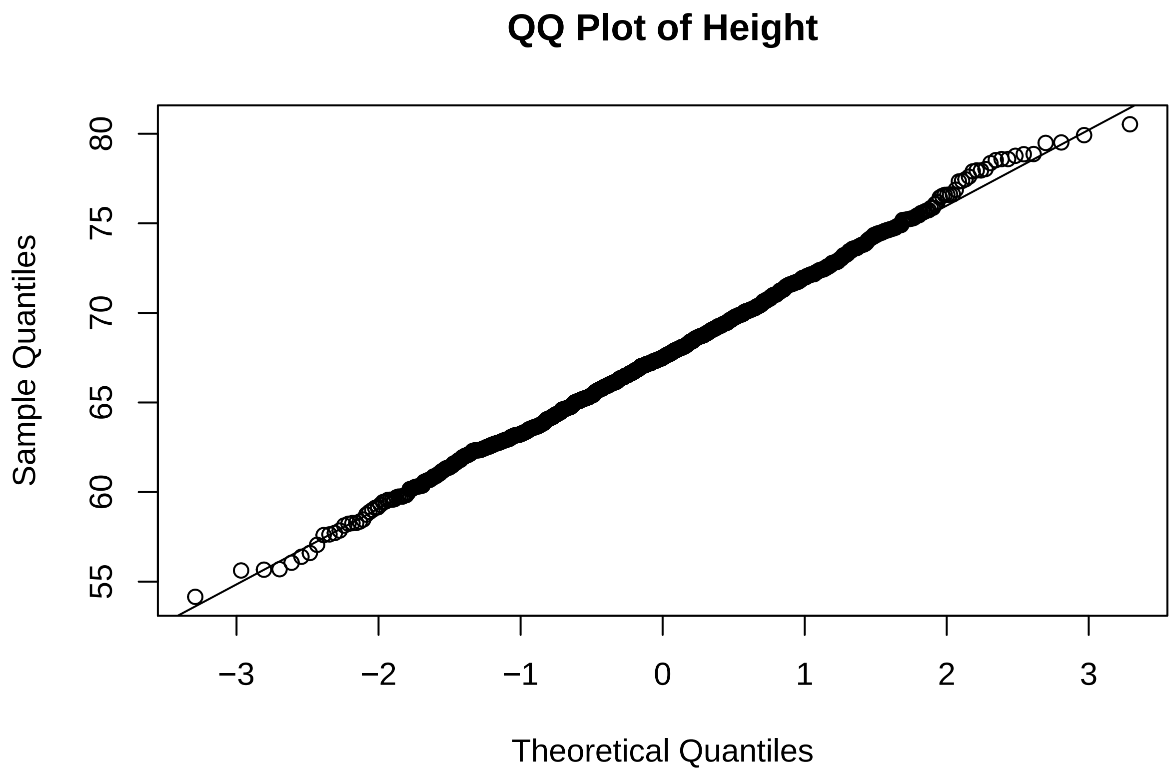
The scatterplot generated for height in relation to age shows a distinct positive relationship. The scatterplot generated for weight in relation to age does not show as strong of a relationship however there is still a positive tendency to the relationship between the two variables.

The following code will generate histograms and Q-Q plots for the numeric variables in the dataset to test for normality of the data.

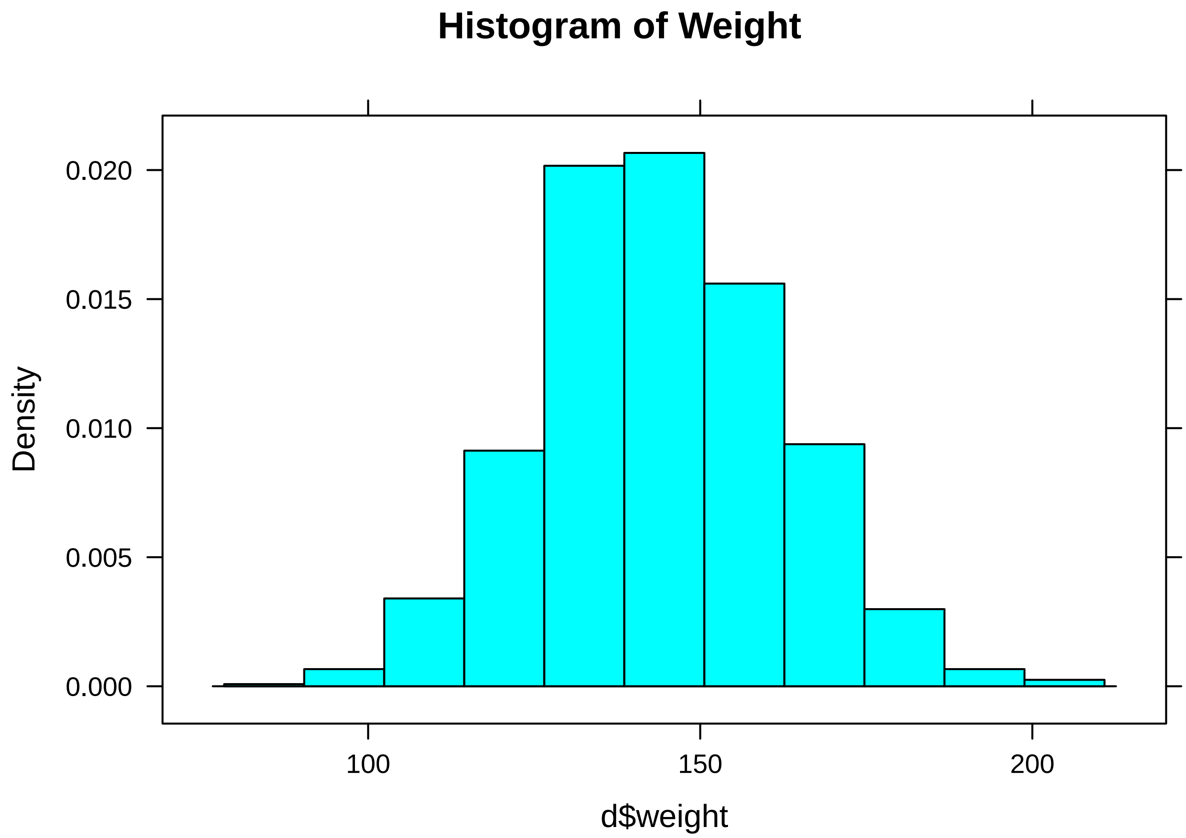
```
histogram(d$height, main= "Histogram of Height")
```



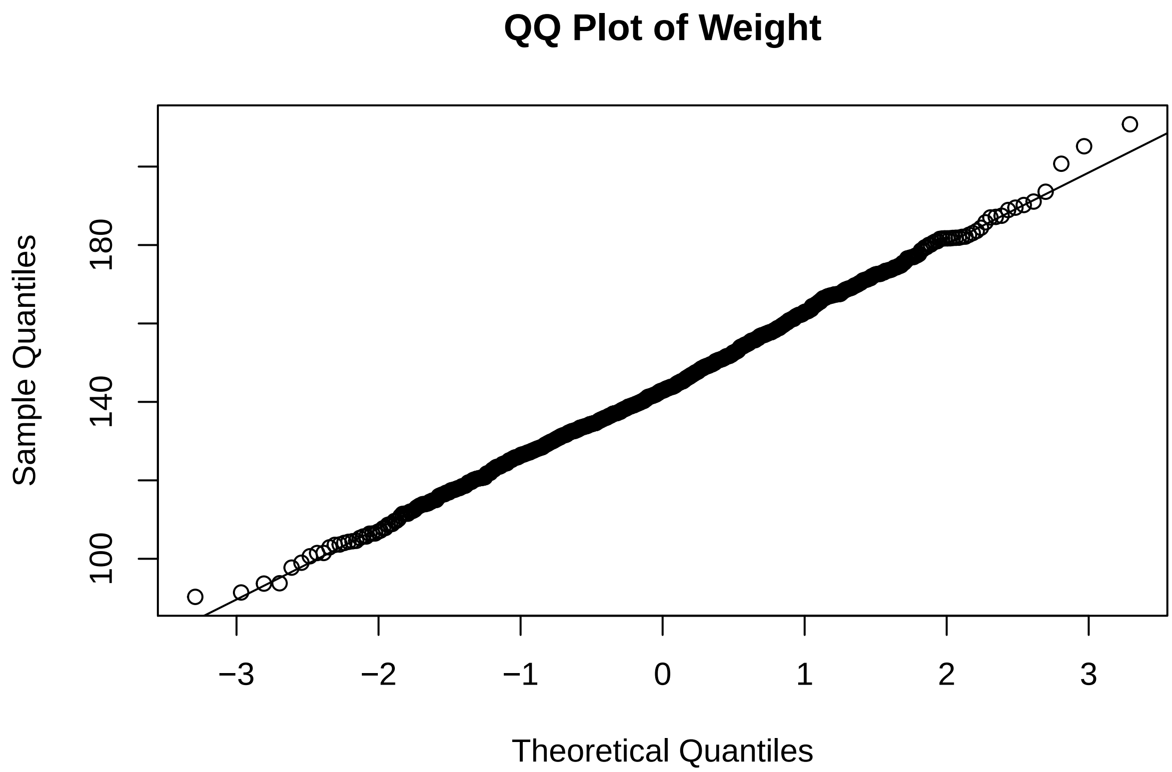

```r
qqnorm(d$height, main="QQ Plot of Height")
qqline(d$height)
```



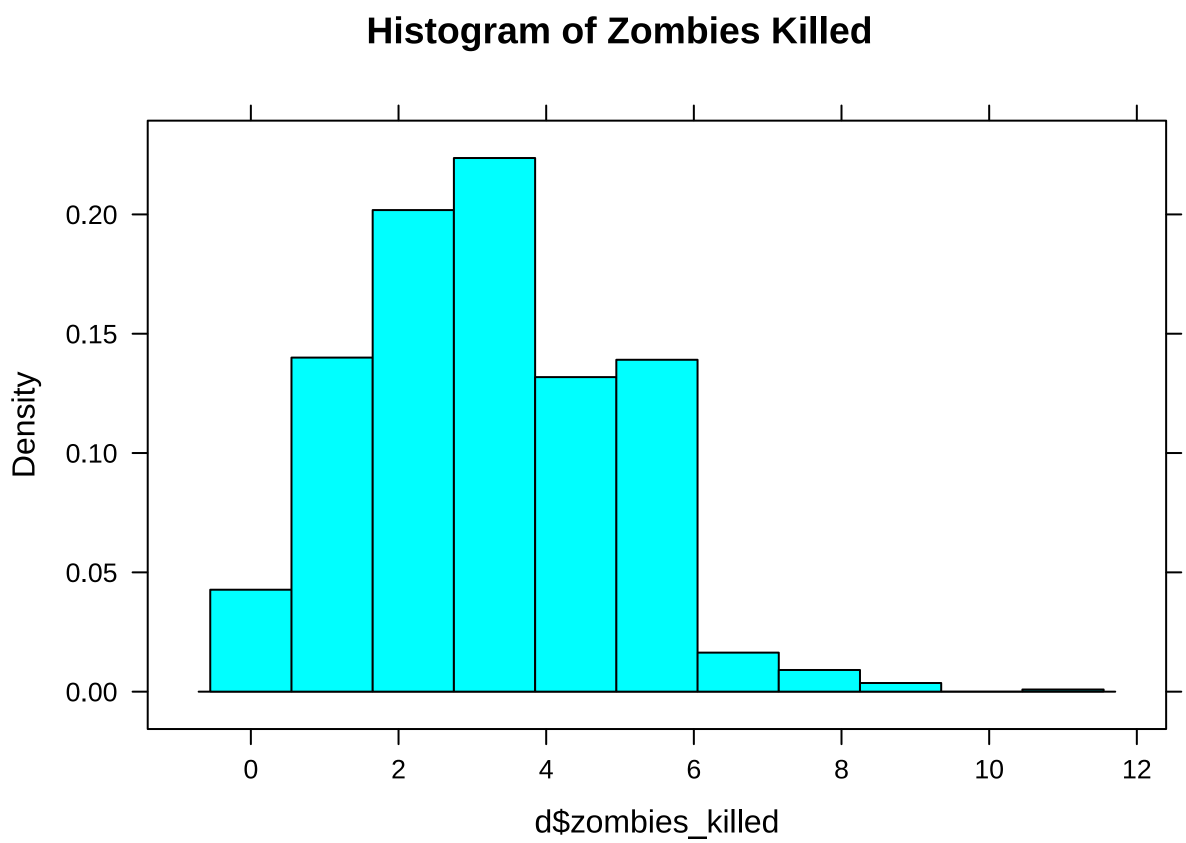

```
histogram(d$weight, main= "Histogram of Weight")
```



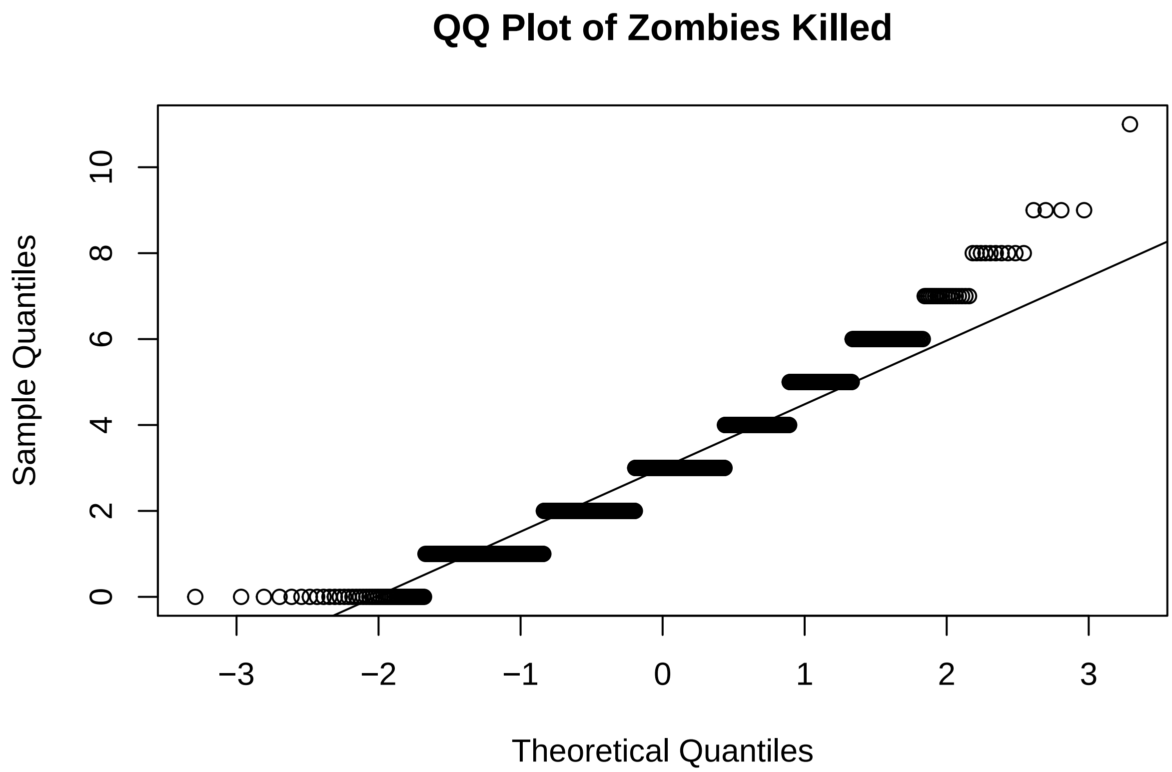

```
qqnorm(d$weight, main="QQ Plot of Weight")
qqline(d$weight)
```



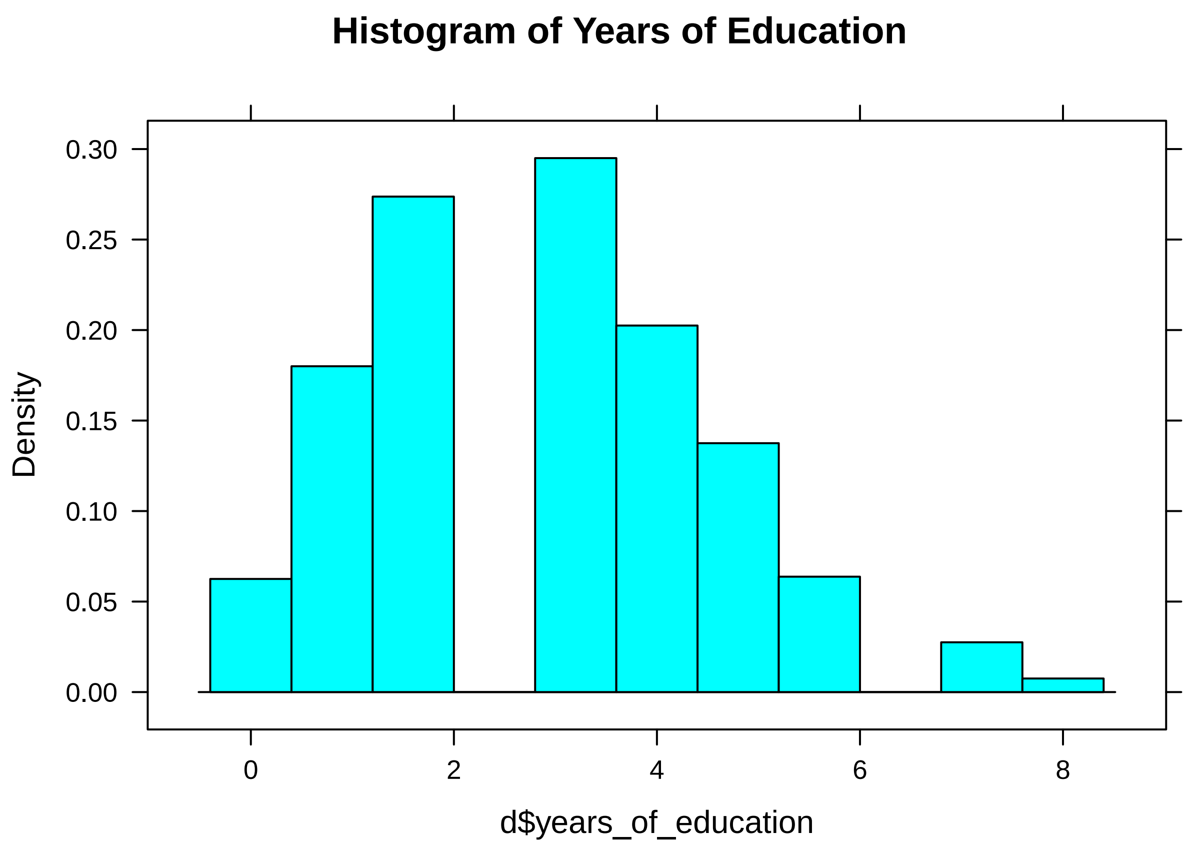

```
histogram(d$zombies_killed, main= "Histogram of Zombies Killed")
```



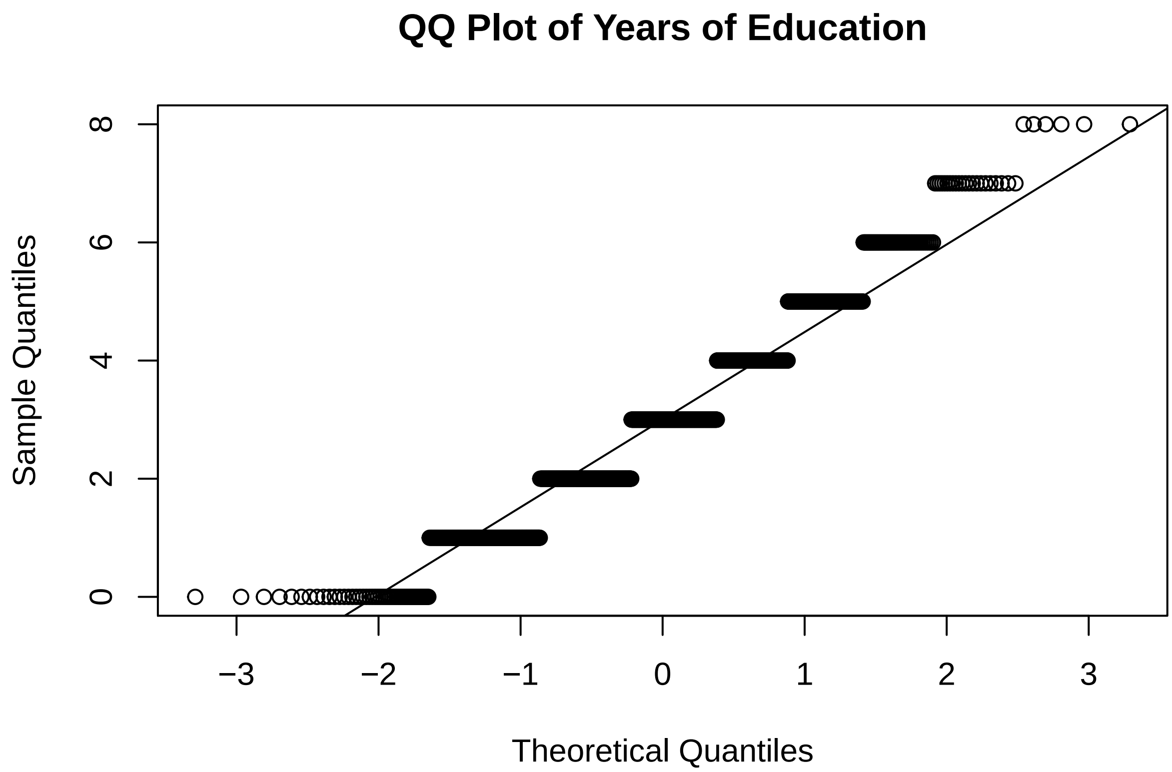

```r
qqnorm(d$zombies_killed, main="QQ Plot of Zombies Killed")
qqline(d$zombies_killed)
```



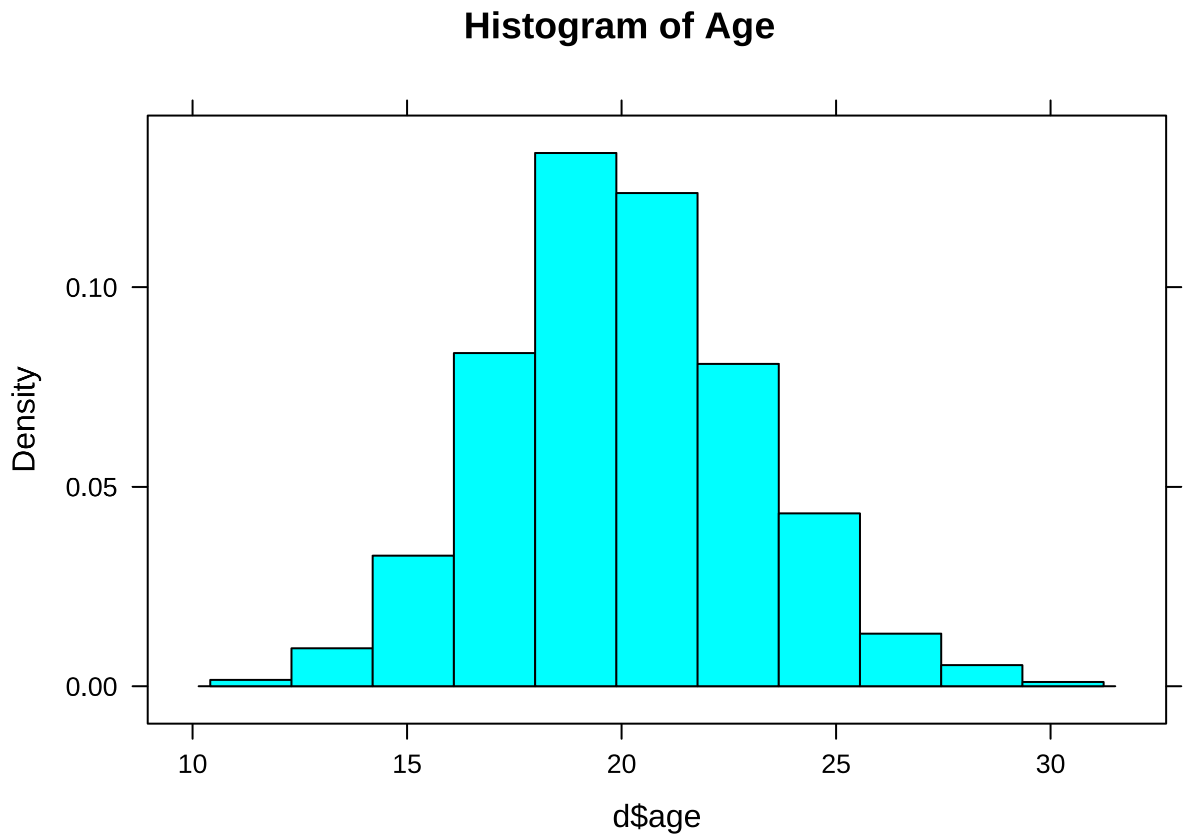

```
histogram(d$years_of_education, main= "Histogram of Years of Education")
```



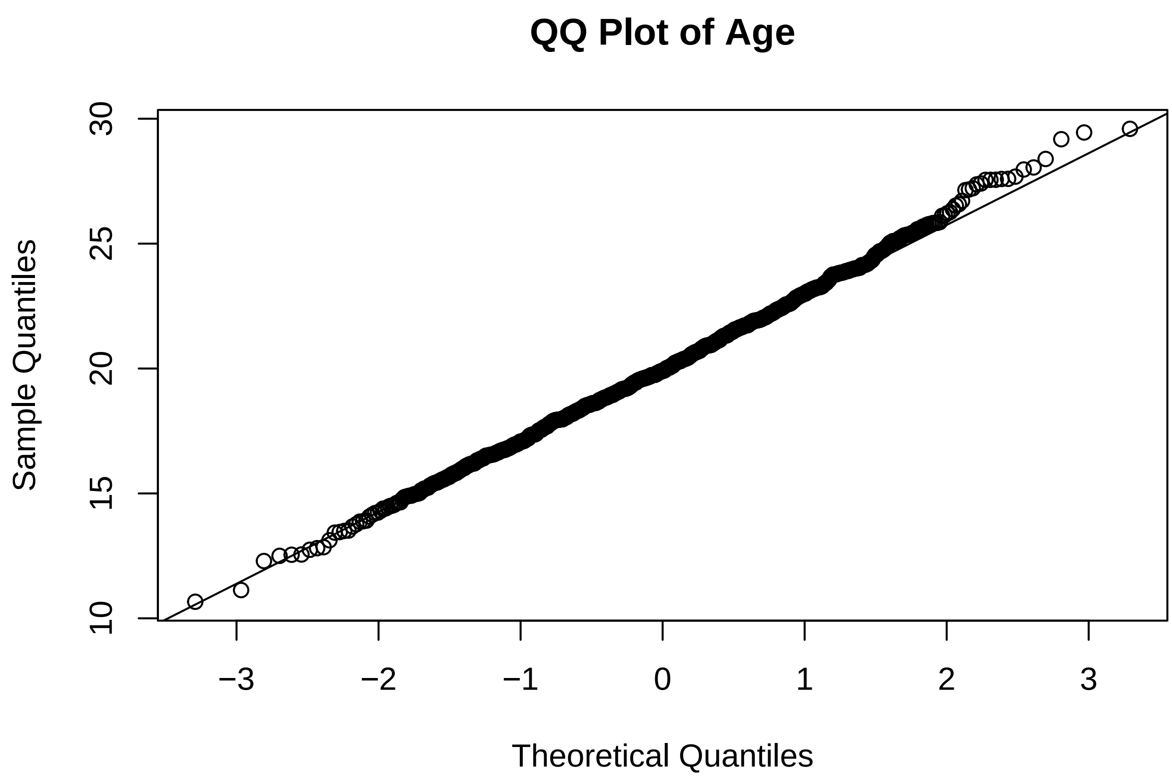

```r
qqnorm(d$years_of_education, main="QQ Plot of Years of Education")
qqline(d$years_of_education)
```




```
histogram(d$age, main= "Histogram of Age")
```




```r
qqnorm(d$age, main="QQ Plot of Age")
qqline(d$age)
```



As demonstrated by the histograms and Q-Q plots above, the height, weight, and age variables seem to be drawn from a normal distribution. The plots generated for the number of zombies killed and years of education indicate that these variables are not drawn from a normal distribution.

The R code below will sample one subset of 30 zombie apocalypse survivors from this population and calculate the mean and standard deviation for each variable. Additionally, the standard error for each variable and a 95% confidence interval will be calculated.

```
sample1 <- d %>% sample_n(30, replace = FALSE)
sample1_stats <- sample1 %>% summarize_if(is.numeric, list(mean=mean,sd=sd, std_error=std_error))
sample1_stats %>% glimpse()
```

```
## Observations: 1
## Variables: 18
## $ id_mean                  <dbl> 394.6667
## $ height_mean              <dbl> 67.48079
## $ weight_mean              <dbl> 142.2574
## $ zombies_killed_mean      <dbl> 2.8
## $ years_of_education_mean  <dbl> 2.833333
## $ age_mean                 <dbl> 20.43207
## $ id_sd                    <dbl> 268.1212
## $ height_sd                <dbl> 4.062971
## $ weight_sd                <dbl> 18.33848
## $ zombies_killed_sd        <dbl> 1.562491
## $ years_of_education_sd    <dbl> 1.440386
## $ age_sd                   <dbl> 3.179835
```

```
## $ id_std_error                 <dbl> 48.95201
## $ height_std_error             <dbl> 0.7417937
## $ weight_std_error             <dbl> 3.348132
## $ zombies_killed_std_error     <dbl> 0.2852706
## $ years_of_education_std_error <dbl> 0.2629774
## $ age_std_error                <dbl> 0.5805558
```

```
ci_height <- sample1_stats$height_mean + c(-1, 1) * qt(1 - 0.05 / 2, df = 30 - 1) * sample1_stats$heigh
ci_height
```

```
## [1] 65.96365 68.99793
```

```
ci_weight <- sample1_stats$weight_mean + c(-1, 1) * qt(1 - 0.05 / 2, df = 30 - 1) * sample1_stats$weigh
ci_weight
```

```
## [1] 135.4097 149.1051
```

```
ci_zombies_killed <- sample1_stats$zombies_killed_mean + c(-1, 1) * qt(1 - 0.05 / 2, df = 30 - 1) * sam
ci_zombies_killed
```

```
## [1] 2.216556 3.383444
```

```
ci_years_of_education <- sample1_stats$years_of_education_mean + c(-1, 1) * qt(1 - 0.05 / 2, df = 30 -
ci_years_of_education
```

```
## [1] 2.295484 3.371182
```

```
ci_age <- sample1_stats$age_mean + c(-1, 1) * qt(1 - 0.05 / 2, df = 30 - 1) * sample1_stats$age_std_err
ci_age
```

```
## [1] 19.24470 21.61944
```

The code below will generate an additional 99 samples and calculate the means for each of these samples. These means will make up a sampling distribution for each variable. Following generation of these samples, the means and standard deviations of each sampling distribution will be calculated. Q-Q Plots for the sampling distributions were also created to test for normality in the distributions, including those for variables determined to not be normally distributed.

```
x <- list()
  for (i in 1:99) {
    x[[i]] <- sample_n(d, 30, replace = FALSE) %>% select(-id) %>% summarize_if(is.numeric, list(mean=m
  }
samp_dist <- bind_rows(x)
sample1_stats <- sample1_stats %>% select(height_mean, weight_mean, zombies_killed_mean, years_of_educa
sample1_stats %>% full_join(samp_dist) %>% glimpse()
```

```
## Joining, by = c("height_mean", "weight_mean", "zombies_killed_mean",
## "years_of_education_mean", "age_mean")
```

```
## Observations: 100
## Variables: 5
## $ height_mean            <dbl> 67.48079, 67.27276, 67.42873, 67.46039, 67....
## $ weight_mean            <dbl> 142.2574, 140.2751, 145.3028, 139.3371, 143...
## $ zombies_killed_mean    <dbl> 2.800000, 2.566667, 3.133333, 2.966667, 3.2...
## $ years_of_education_mean <dbl> 2.833333, 3.266667, 2.800000, 3.033333, 2.6...
## $ age_mean               <dbl> 20.43207, 20.53862, 19.98023, 20.60415, 19....
```

```
samp_dist %>% summarize_all(mean)
```

```
## # A tibble: 1 x 5
##   height_mean weight_mean zombies_killed_mean years_of_education_mean age_mean
##         <dbl>       <dbl>               <dbl>                   <dbl>    <dbl>
## 1        67.5        144.                2.97                    2.98     20.0
```

```
samp_dist %>% summarize_all(sd)
```

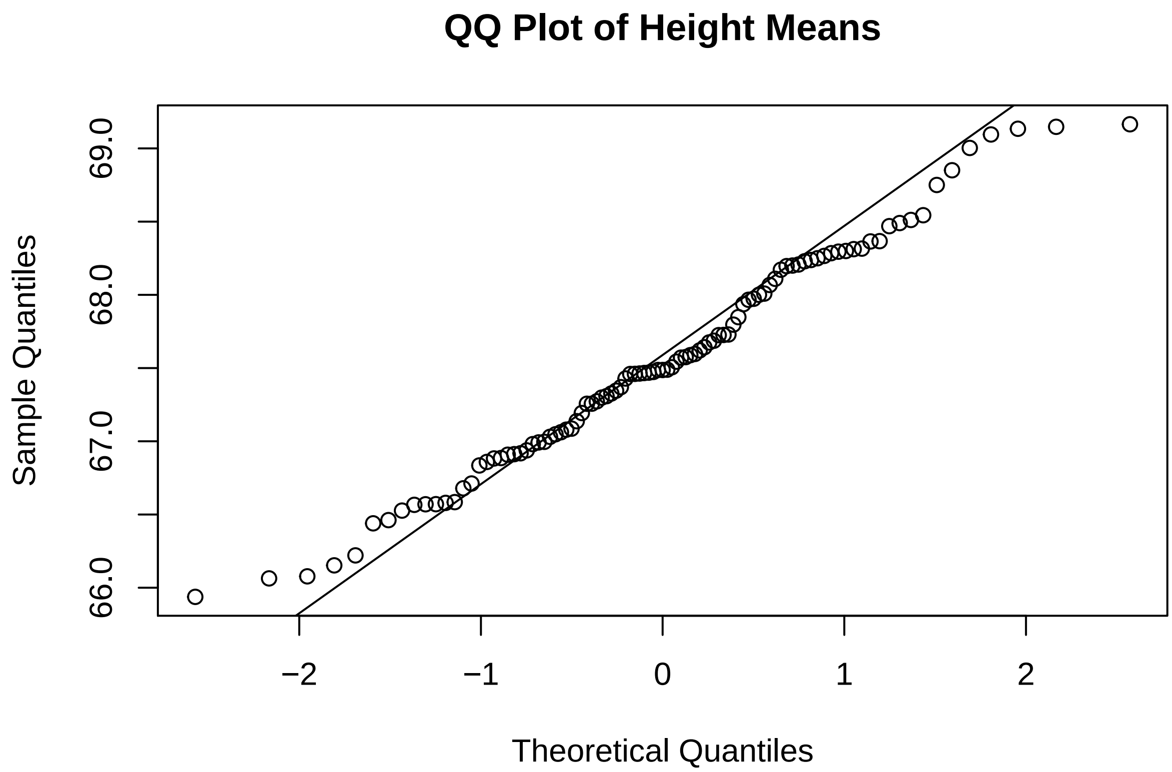
```
## # A tibble: 1 x 5
##   height_mean weight_mean zombies_killed_mean years_of_education_mean age_mean
##         <dbl>       <dbl>               <dbl>                   <dbl>    <dbl>
## 1       0.751        3.65               0.287                   0.310    0.440
```

```
pop_se <- function(x) {
 sdpop(x) / (sqrt(30))
}
sample1 %>% select(-id) %>% summarize_if(is.numeric, pop_se)
```

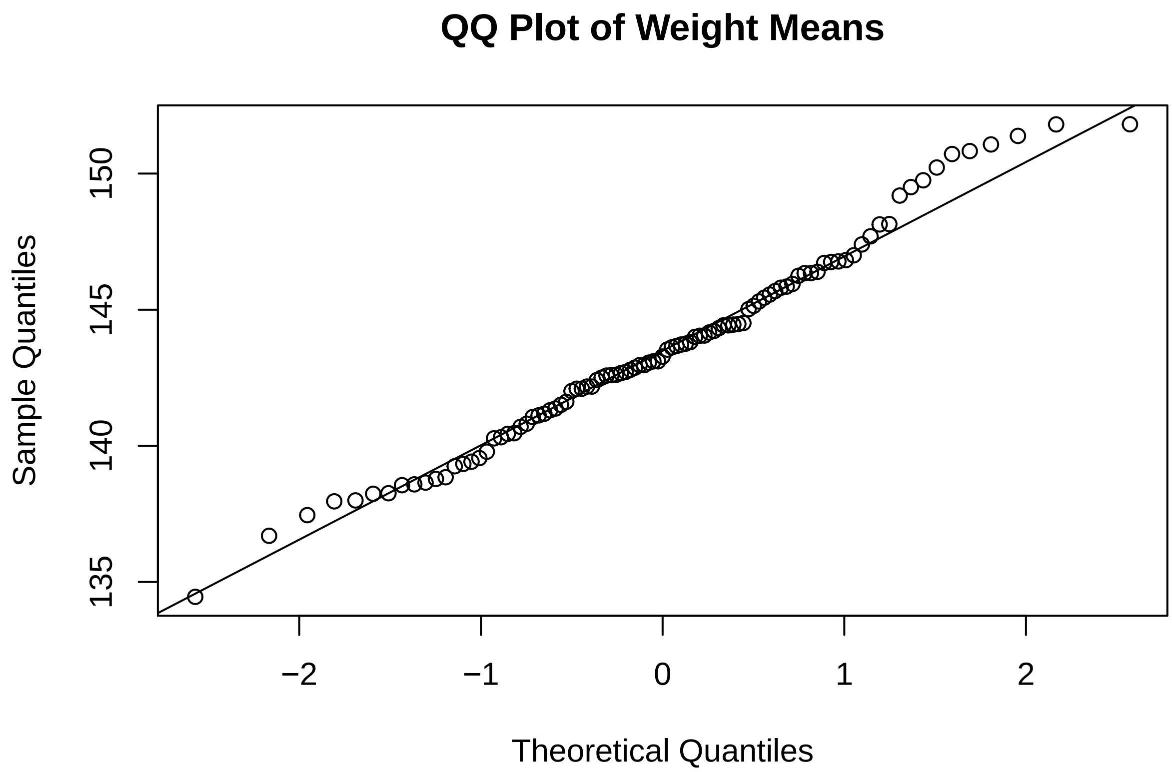
```
## # A tibble: 1 x 5
##   height weight zombies_killed years_of_education   age
##    <dbl>  <dbl>          <dbl>              <dbl> <dbl>
## 1  0.729   3.29          0.280              0.259 0.571
```

The standard deviations above are very close to the values for the standard error of each variable calculated in the above question. Given that the standard deviation across a sampling distribution is a good estimate for the standard error for a variable, this makes sense. Additionally, the standard deviations of the sampling distributions are closer to the standard errors using the sample above as compared to the standard errors calculated using the population standard deviation.

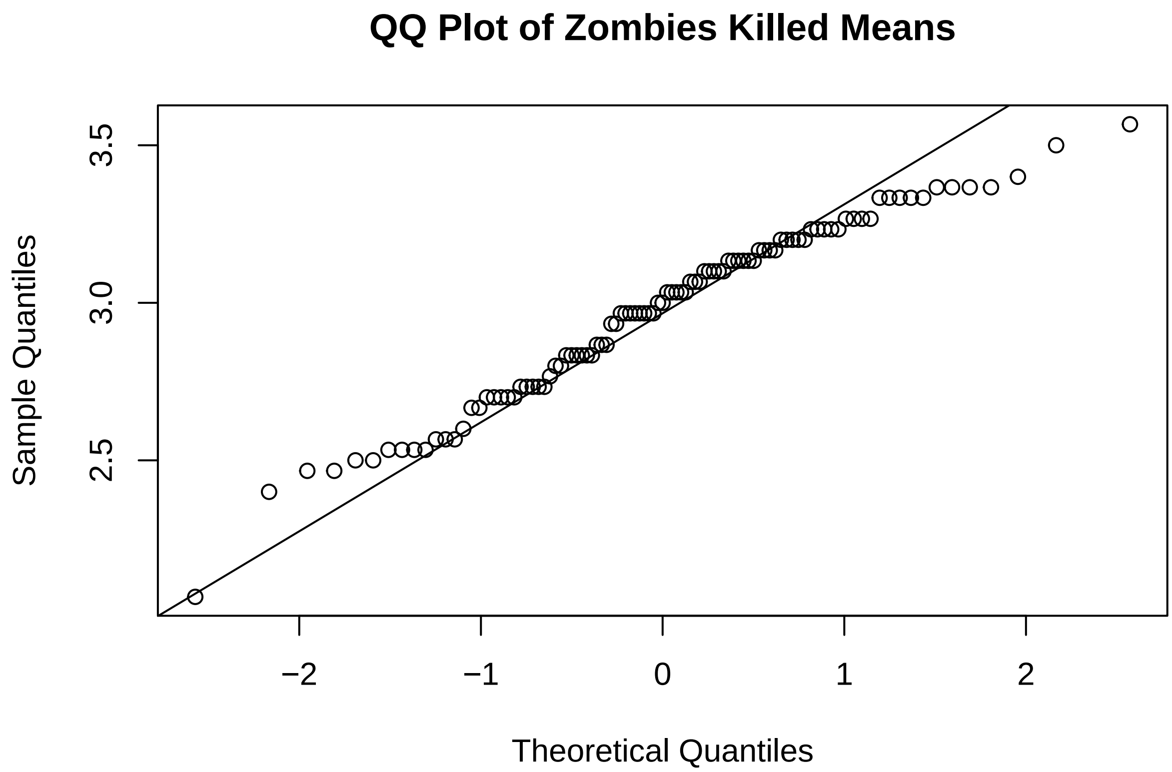
```
qqnorm(samp_dist$height_mean, main="QQ Plot of Height Means")
qqline(samp_dist$height_mean)
```



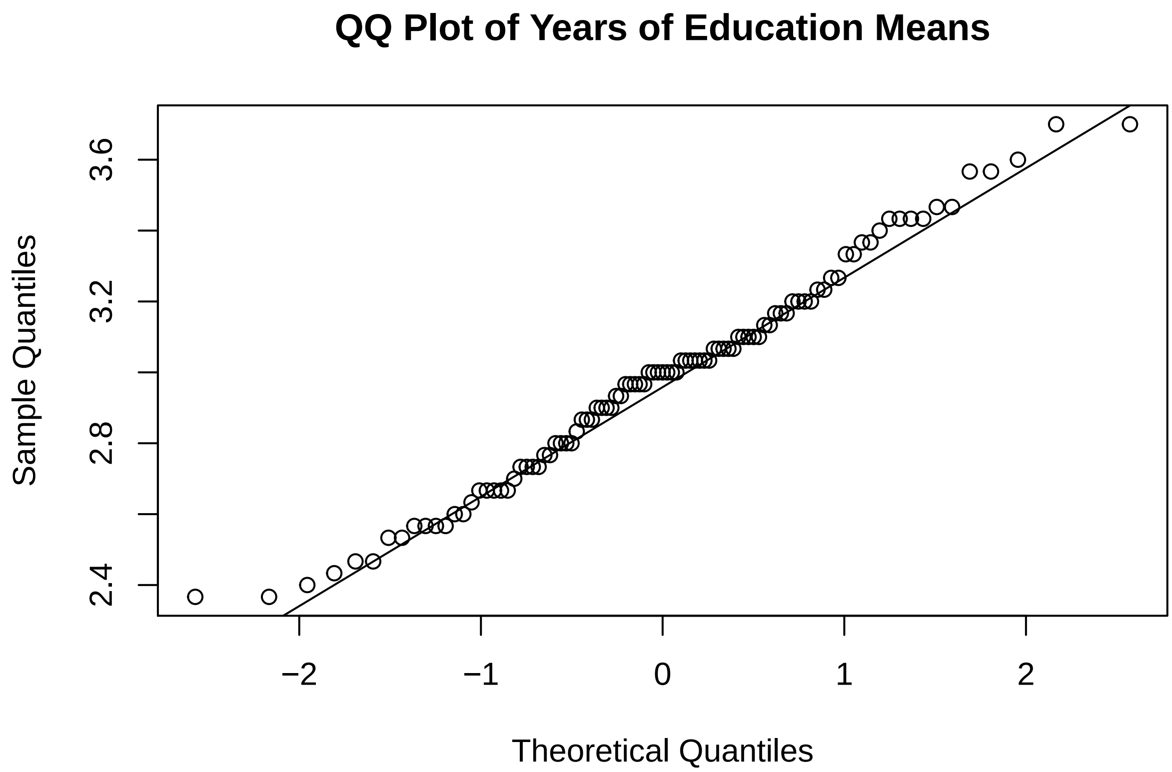

```r
qqnorm(samp_dist$weight_mean, main="QQ Plot of Weight Means")
qqline(samp_dist$weight_mean)
```



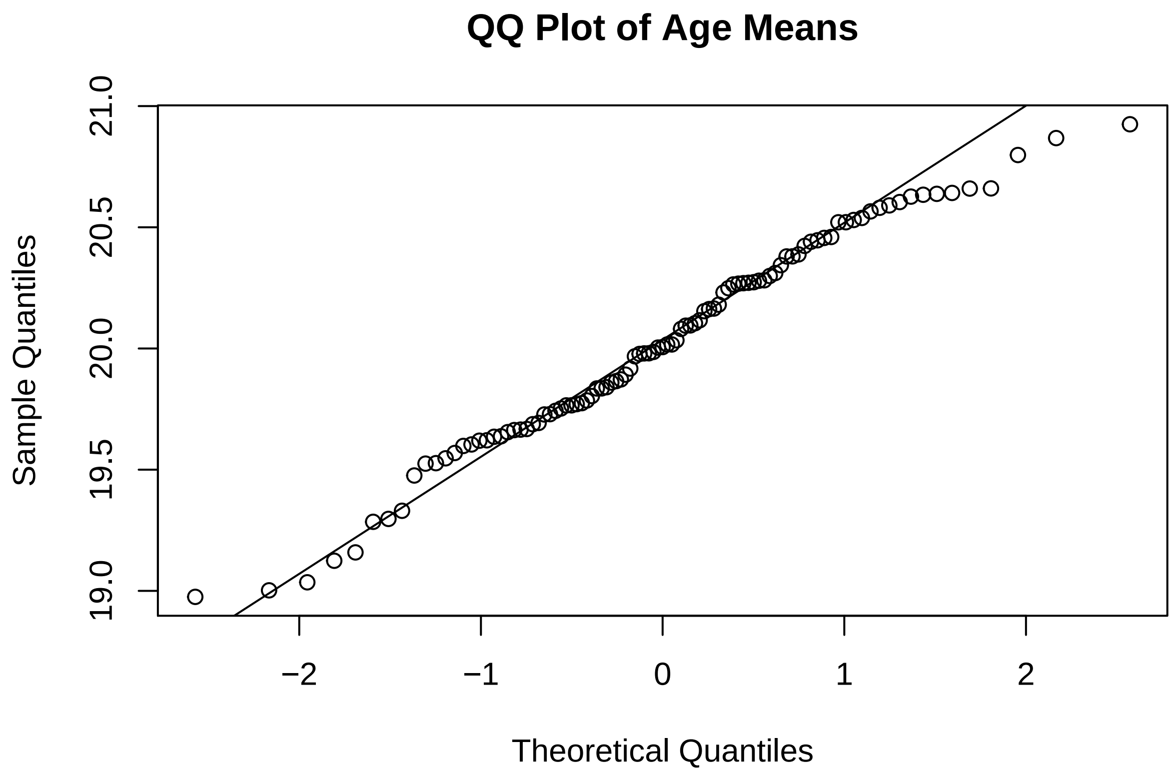

```r
qqnorm(samp_dist$zombies_killed_mean, main="QQ Plot of Zombies Killed Means")
qqline(samp_dist$zombies_killed_mean)
```




```r
qqnorm(samp_dist$years_of_education_mean, main="QQ Plot of Years of Education Means")
qqline(samp_dist$years_of_education_mean)
```




```r
qqnorm(samp_dist$age_mean, main="QQ Plot of Age Means")
qqline(samp_dist$age_mean)
```



The above Q-Q plots demonstrate that the sampling distributions for each variable follow a relatively normal distrubution. This includes the **zombies_killed** and **years_of_education** variables that were found previously to not follow a normal distrubution in the original population.