assignment 3

Raul Rodriguez

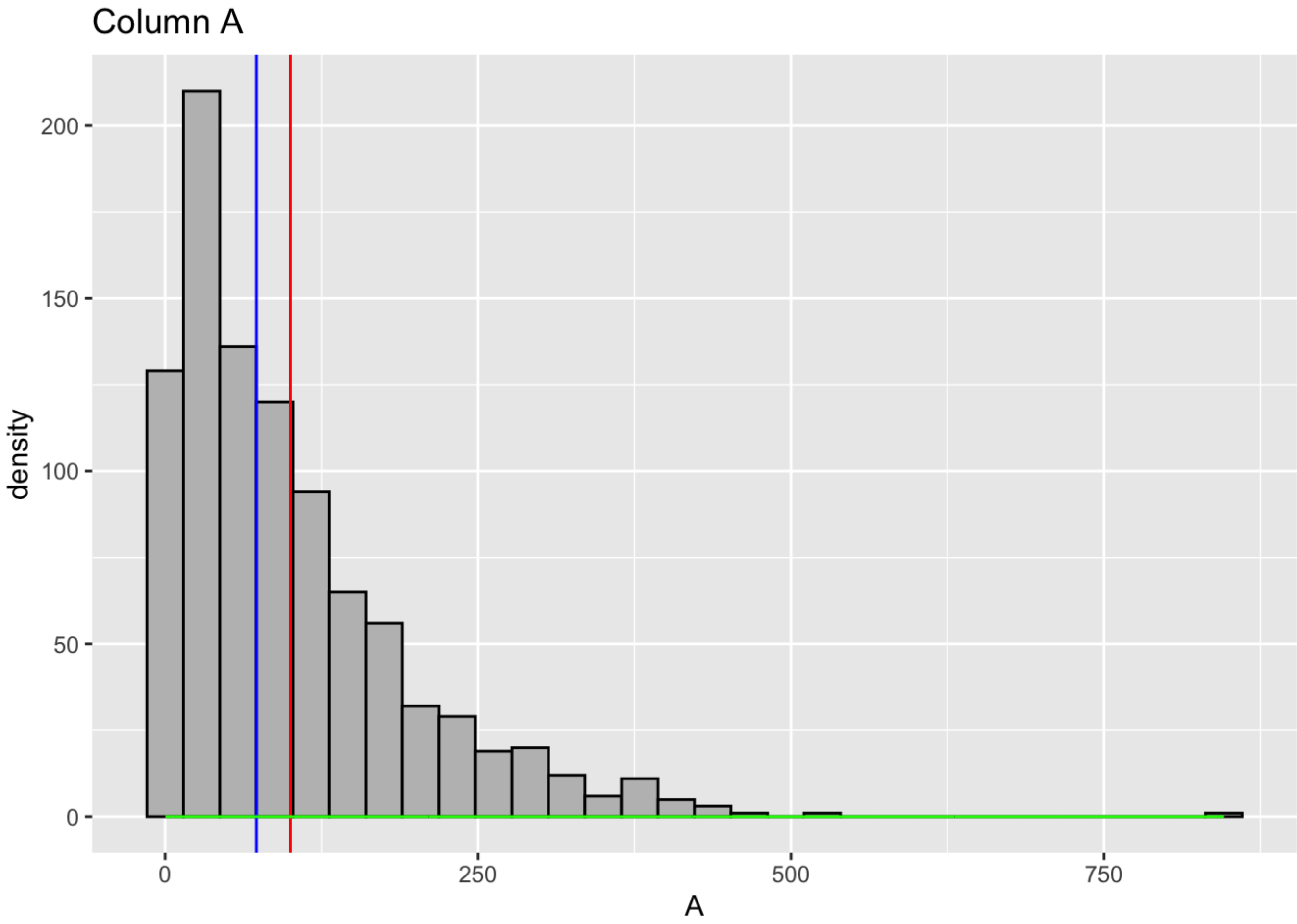
2023-07-05

Question 1

```
df <- read.csv('homework3Data.csv')

# a)
# Load the ggplot2 library
library(ggplot2)



ggplot_factory <- function(df, df_col, title_name) {
  ggplot(data = df, aes(x = .data[[df_col]])) +
    geom_histogram(fill = 'gray', color = 'black') +
    geom_vline(aes(xintercept = mean(.data[[df_col]])), color = 'red') +
    geom_vline(aes(xintercept = median(.data[[df_col]])), color = 'blue') +
    geom_density(color = 'green') +
    labs(title = title_name)
}


a_plot <- ggplot_factory(
  df,
  df_col = 'A',
  title_name = 'Column A')

b_plot <- ggplot_factory(
  df,
  df_col = 'B',
  title_name = 'Column B')

c_plot <- ggplot_factory(
  df,
  df_col = 'C',
  title_name = 'Column C')

c_plot <- ggplot_factory(
  df,
  df_col = 'C',
  title_name = 'Column C')

x_plot <- ggplot_factory(
  df,
  df_col = 'X',
  title_name = 'Column X')

y_plot <- ggplot_factory(
  df,
  df_col = 'Y',
  title_name = 'Column Y')

z_plot <- ggplot_factory(
  df,
  df_col = 'Z',
  title_name = 'Column Z')
```

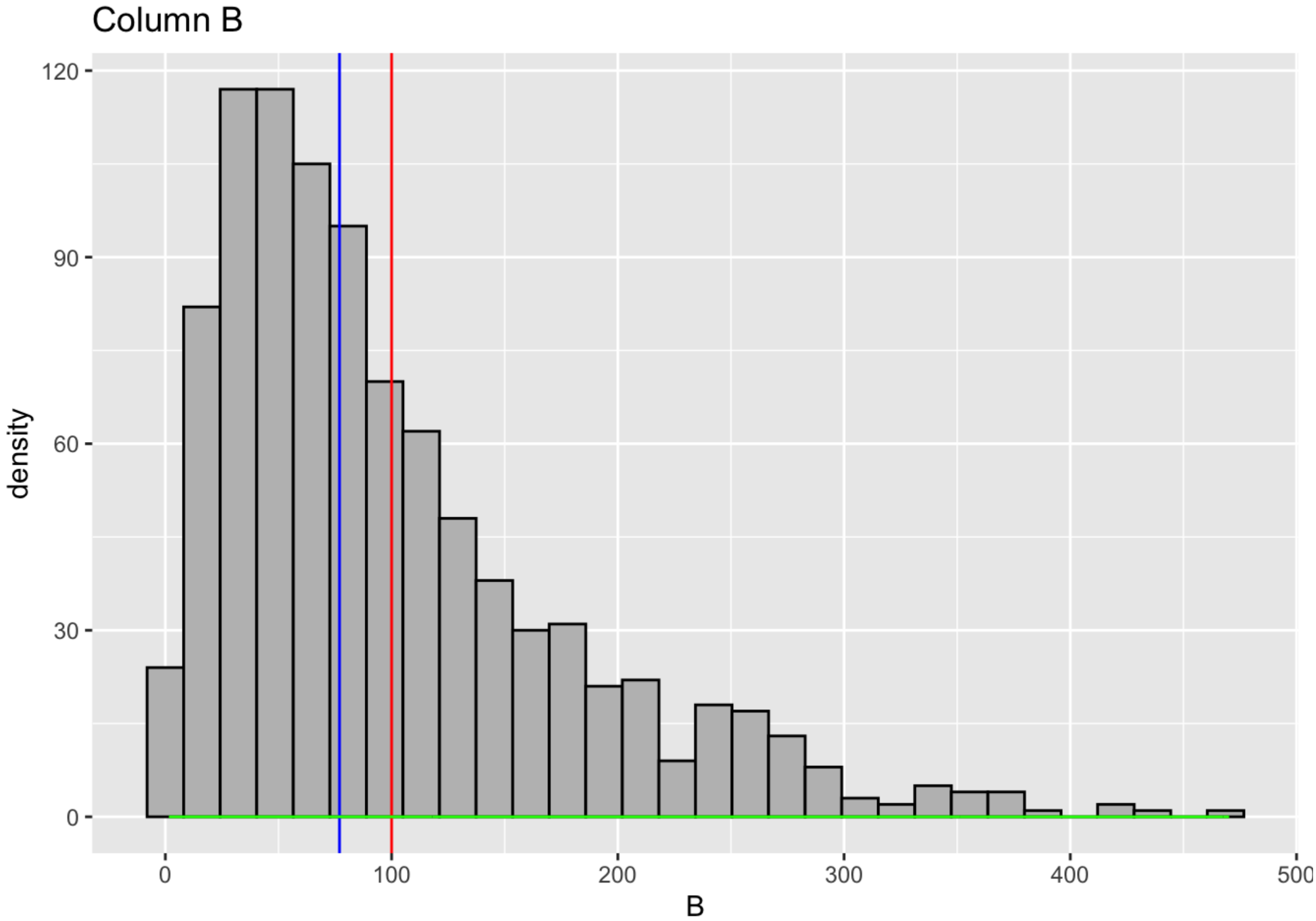
```
a_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



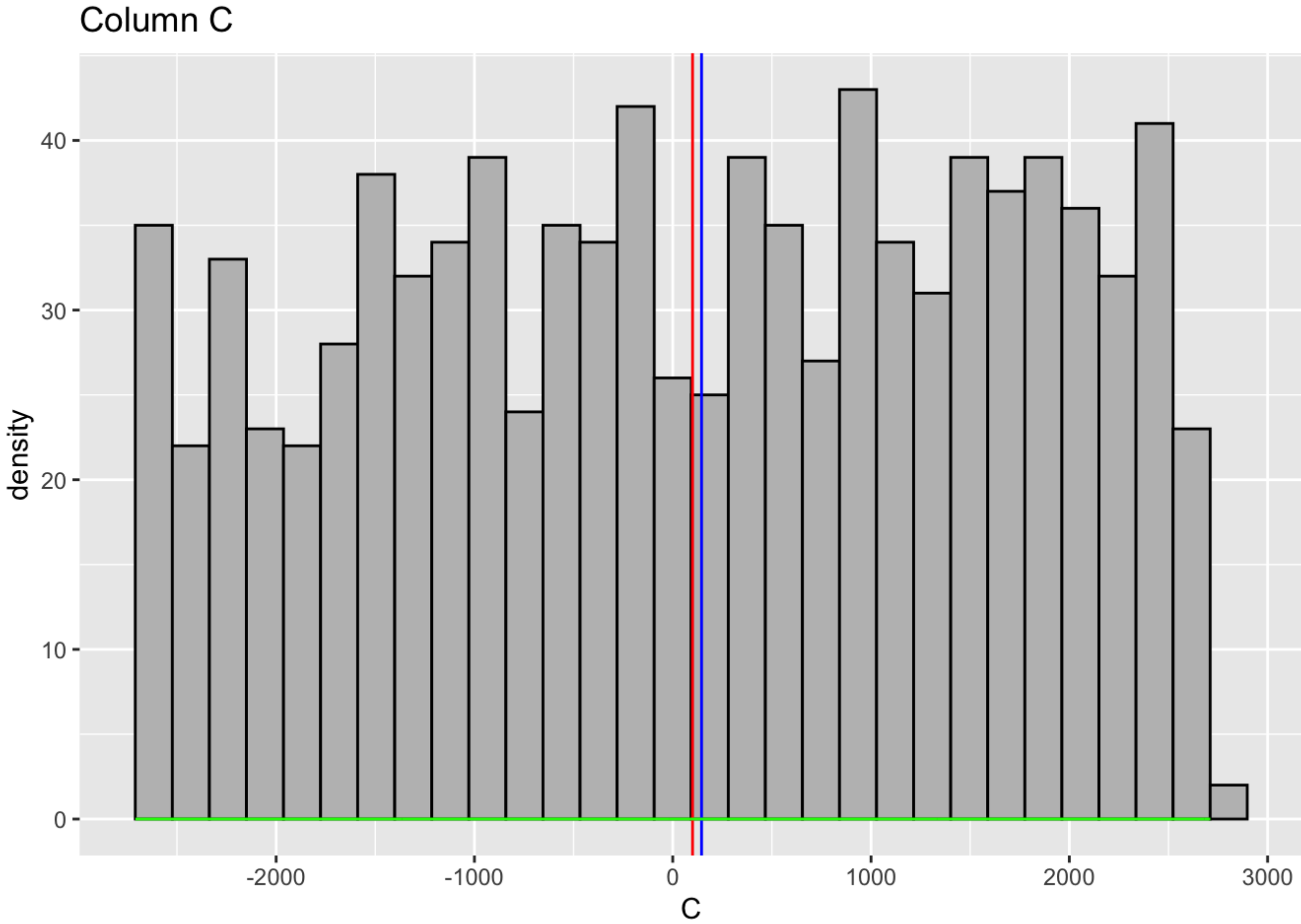

```
b_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



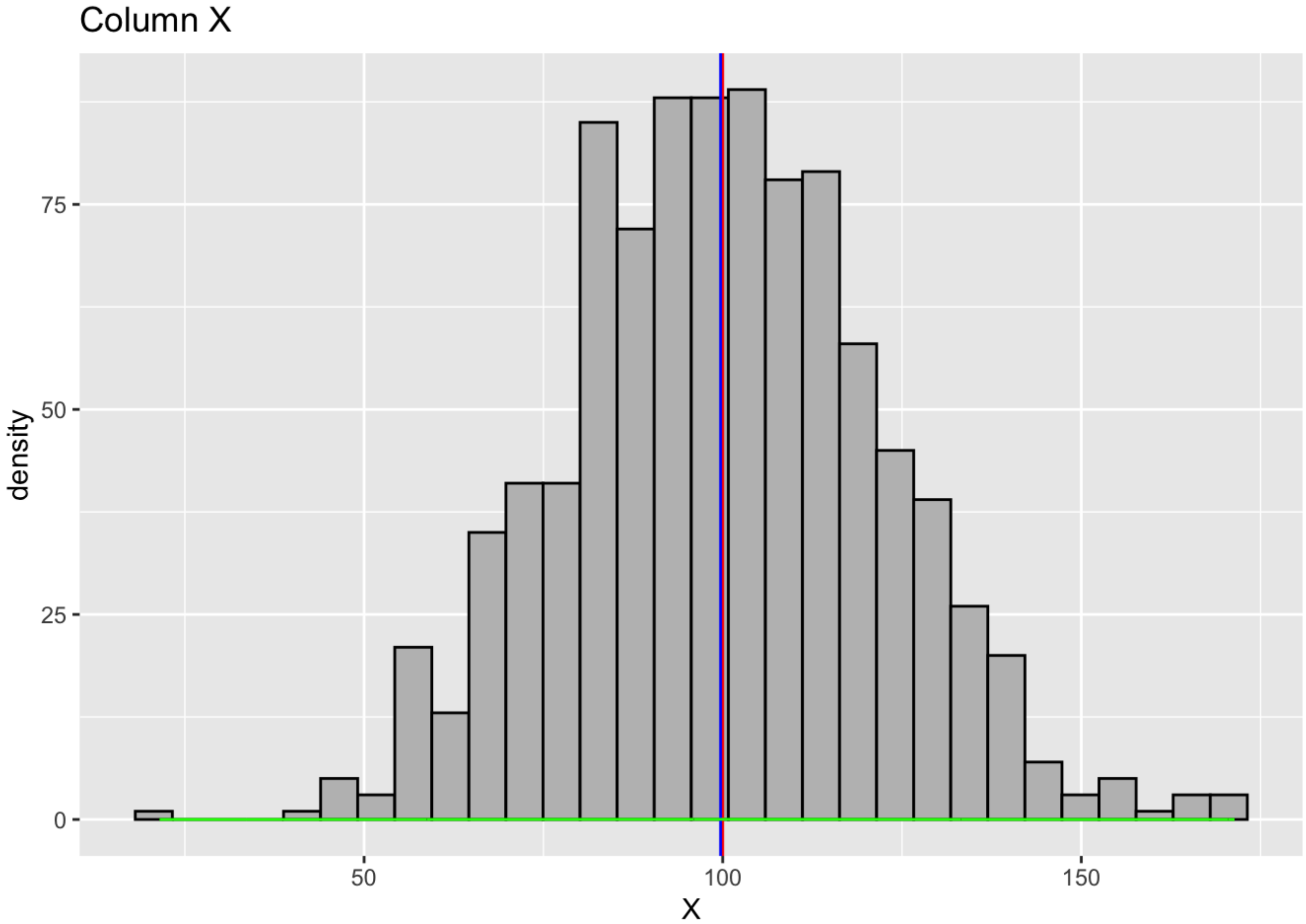

```
c_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



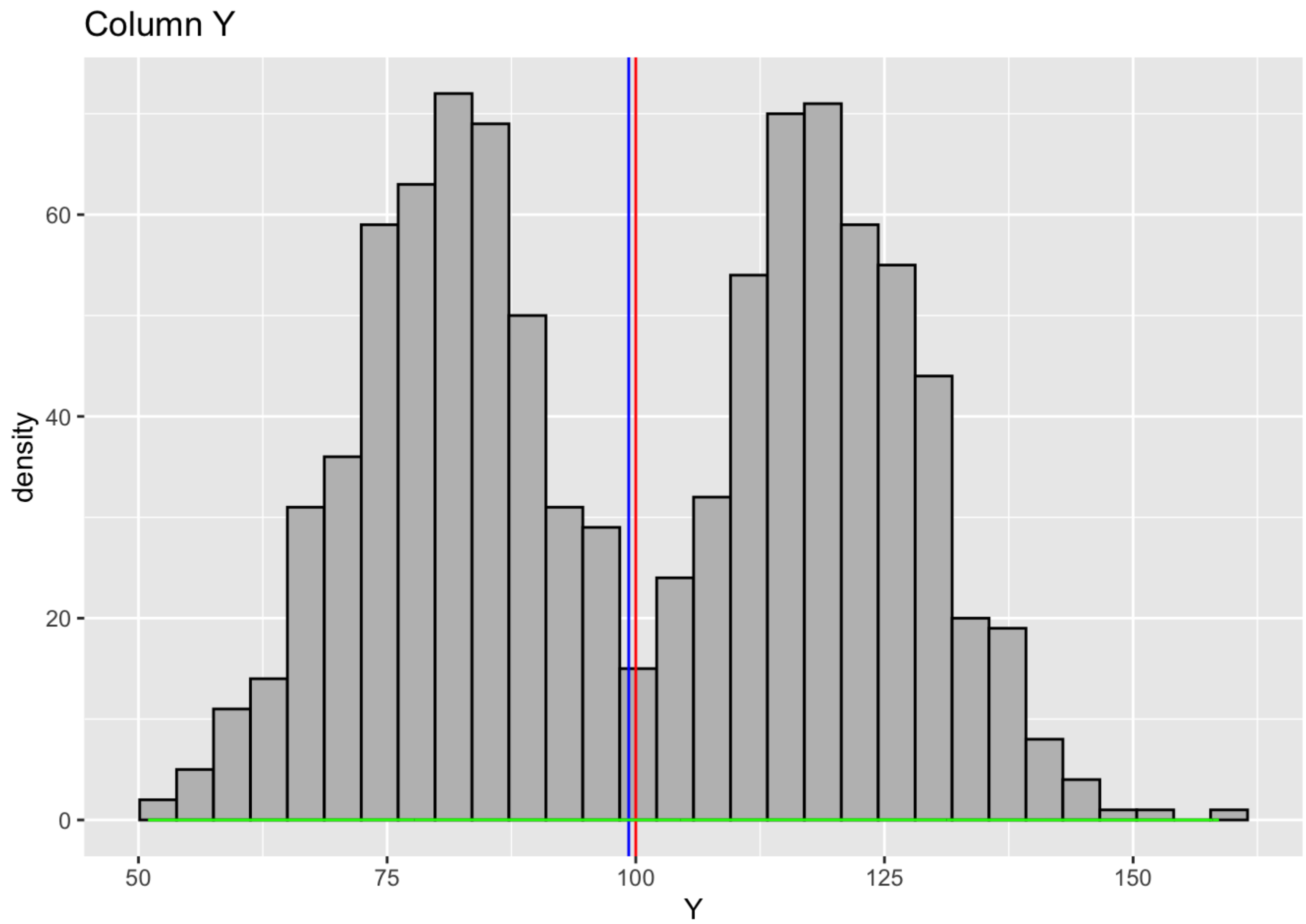

```
x_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



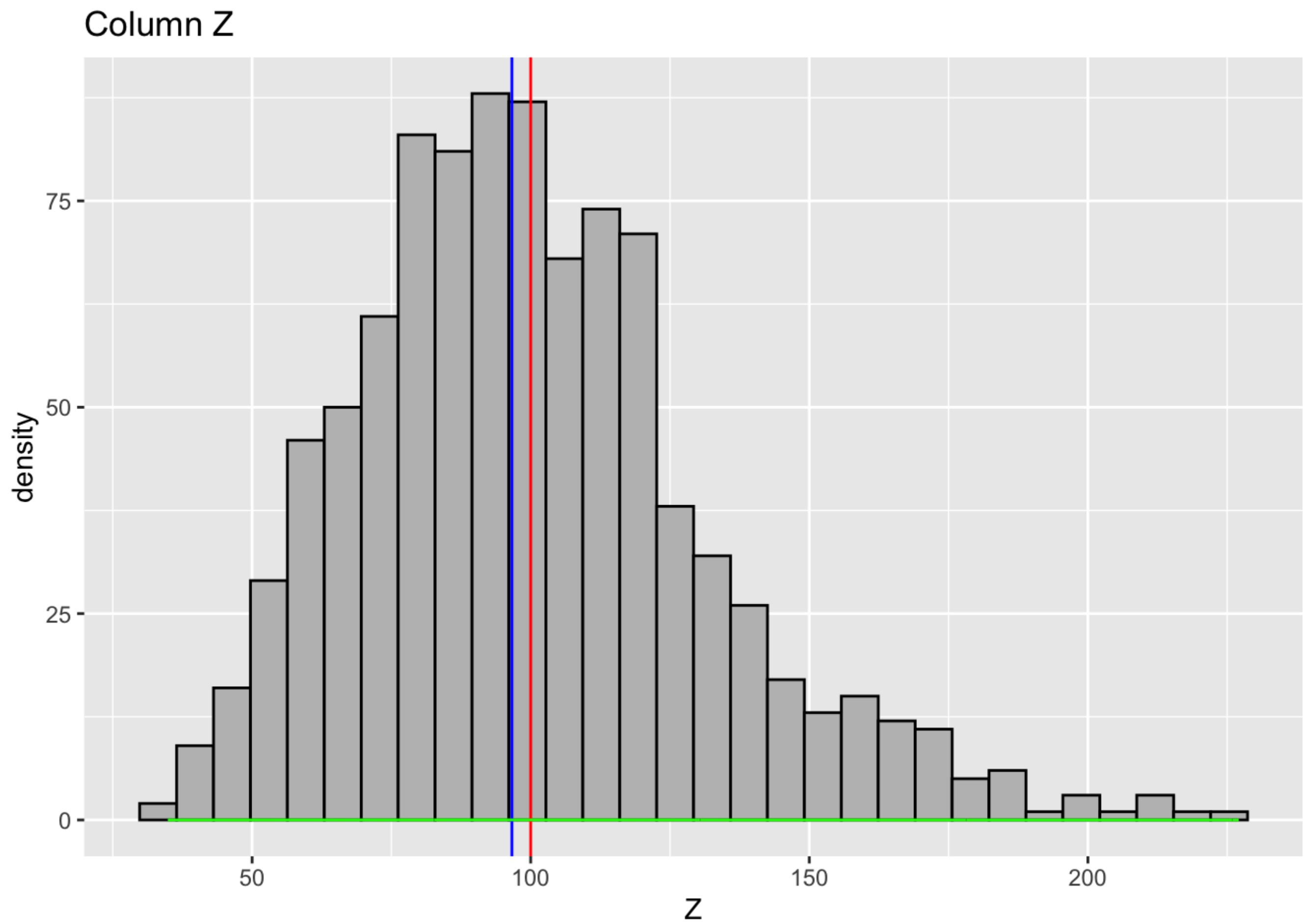

```
y_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
z_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#b)
# answered in text

#c)

mean_X <- mean(df$X)
sd_X <- sd(df$X)

interval_68 <- c(mean_X - sd_X, mean_X + sd_X)
interval_95 <- c(mean_X - 2 * sd_X, mean_X + 2 * sd_X)
interval_997 <- c(mean_X - 3 * sd_X, mean_X + 3 * sd_X)

interval_68
```

```
## [1]  78.28366 121.71634
```

```
interval_95
```

```
## [1]  56.56733 143.43267
```

```
interval_997
```

```
## [1]  34.85099 165.14901
```

```
# d)
quantiles <- quantile(df$X, c(0.003, 0.025, 0.1587, 0.5, 0.8413, 0.975, 0.997))
quantiles
```

```
##      0.3%      2.5%    15.87%       50%    84.13%     97.5%     99.7%
##  47.68037  57.93873  79.17083  99.70620 121.58789 141.58849 166.42299
```

```
# e)
intervals_qnorm <- qnorm(c(0.0015, 0.025, 0.1587, 0.5, 0.8413, 0.975, 0.9985), mean =
mean_X, sd = sd_X)

intervals_qnorm
```

```
## [1]  35.55160  57.43676  78.28768 100.00000 121.71232 142.56324 164.44840
```

```
# f)

#pop mean estimate
sample_mean <- mean(df$X)
sample_mean
```

```
## [1] 100
```

```
#sd
sample_sd <- sd(df$X)
sample_size <- length(df$X)
#estimated error
estimated_se <- sample_sd / sqrt(sample_size)
estimated_se
```

```
## [1] 0.704571
```

```
#critical value
diff <- sample_size - 1
critical_value_t <- qt(0.86, diff)
critical_value_t
```

```
## [1] 1.080936
```

```
confidence_interval_lower <- sample_mean - (critical_value_t * estimated_se)
confidence_interval_upper <- sample_mean + (critical_value_t * estimated_se)

confidence_interval_lower
```

```
## [1] 99.2384
```

```
confidence_interval_upper
```

```
## [1] 100.7616
```

b. Without any formal tests and purely based on visual aid, column x appears to be normally distributed because the graph closely conforms with the normal distribution bell curve.

c. comparing these intervals, we can see that the intervals obtained from part (e) using qnorm() are closest to the intervals from part (c) calculated using the mean and standard deviation.

Question 2

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

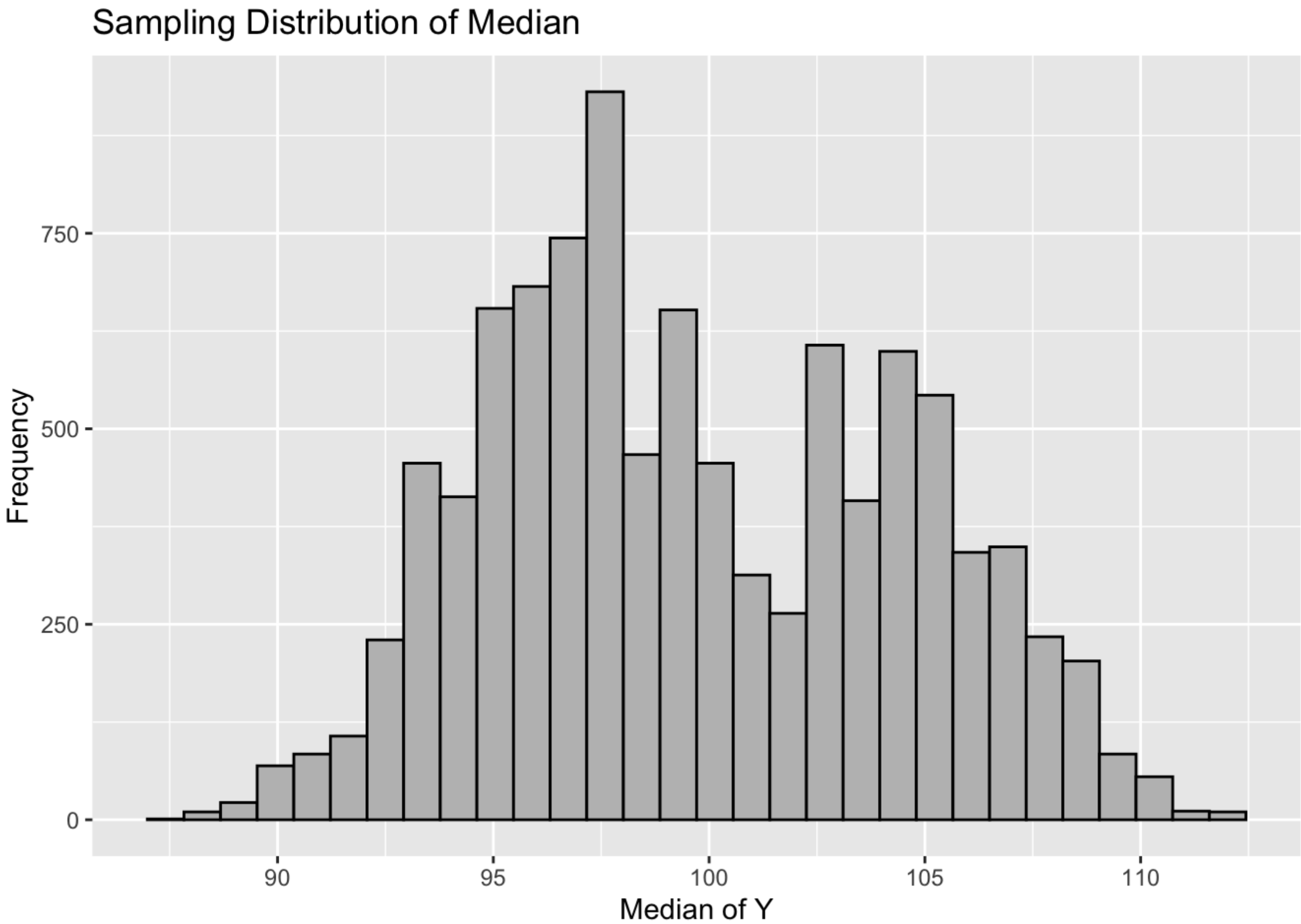
```
# a)
names <- starwars$name


# b)
nchar(names[c(7, 19, 31)])
```

```
## [1] 18  4 12
```

```
# c)
num_char <- numeric(length(names))

for (i in 1:length(names)) {
  num_char[i] <- nchar(names[i])
}


# d)
num_char <- sapply(names[c(7, 19, 31)], nchar)
```

Question 3

```
library(ggplot2)
data <- read.csv("homework3Data.csv")
set.seed(123)

# a)
n_bootstrap <- 10000
sample_size <- 250
n_replicate <- 10000

bootstrap_medians <- numeric(n_bootstrap)

for (i in 1:n_bootstrap) {
  bootstrap_sample <- sample(data$Y, size = sample_size, replace = TRUE)
  bootstrap_medians[i] <- median(bootstrap_sample)
}

# b)
ggplot(data.frame(bootstrap_medians), aes(x = bootstrap_medians)) +
  geom_histogram(fill = "gray", color = "black", bins = 30) +
  labs(x = "Median of Y", y = "Frequency", title = "Sampling Distribution of Median")
```



```
# c)
estimated_median <- median(data$Y)

# d)
lower_ci <- quantile(bootstrap_medians, 0.025)
upper_ci <- quantile(bootstrap_medians, 0.975)
```