QTM 220 HW #2

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

QTM 220 Homework #2

Exercise #2 - Estimated Sampling Distributions

```
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.3.3
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.3     ✓ readr     2.1.4
✓ forcats   1.0.0     ✓ stringr   1.5.0
✓ ggplot2   3.4.3     ✓ tibble    3.2.1
✓ lubridate 1.9.2     ✓ tidyr     1.3.0
✓ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
nba_df <- read.csv("C:/Users/13015/OneDrive - Emory University/Documents/Fall 2024/QTM
         220/nba.data.csv")
knitr::kable(head(nba_df))
```

X	Player	POS	Team	Age	GP	W	L	Min	PTS
1	Jayson Tatum	SF	BOS	25	74	52	22	2732.2	2225
2	Joel Embiid	C	PHI	29	66	43	23	2284.1	2183
3	Luka Doncic	PG	DAL	24	66	33	33	2390.5	2138
4	Shai Gilgeous-Alexander	PG	OKC	24	68	33	35	2416.0	2135
5	Giannis Antetokounmpo	PF	MIL	28	63	47	16	2023.6	1959
6	Anthony Edwards	SG	MIN	21	79	40	39	2841.5	1946

(a)

```
set.seed(42)
# creating a sample
sample <- sample(nba_df, 100, replace = T)

# mean PTS
sample_mean <- mean(sample$PTS)
sample_mean
```

```
[1] 523.4267
```

```
# standard deviation PTS
sample_sd <- sd(sample$PTS)
sample_sd
```

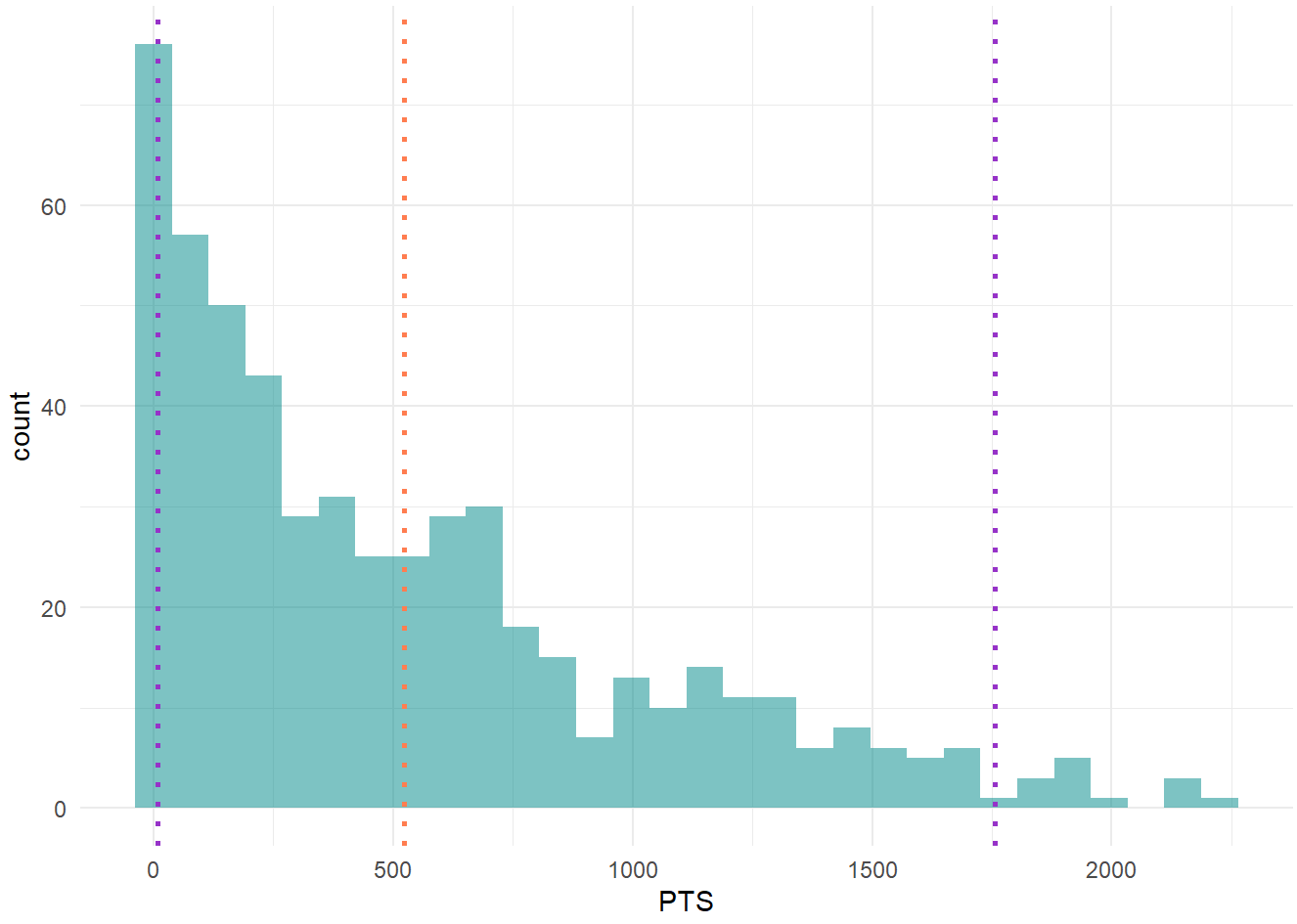
```
[1] 498.0844
```

```
# calculating 95% CI
quantile(sample$PTS, c(0.05, 0.975))
```

```
    5%  97.5%
   9.0 1755.2
```

```
# histogram w/ plus or minus 1.96 standard deviations
ggplot(data = sample, aes(x = PTS)) +
  geom_histogram(fill = "cyan4", alpha = 0.5, position = 'identity') +
   geom_vline(xintercept = sample_mean, linetype="dotted",
              color = "coral", linewidth=1) +
  geom_vline(xintercept = quantile(sample$PTS, 0.05), linetype = 'dotted',
              color = "darkorchid", linewidth = 1) +
    geom_vline(xintercept = quantile(sample$PTS, 0.975), linetype = "dotted",
              color = "darkorchid", linewidth=1) +
  theme_minimal()
```

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



(b)

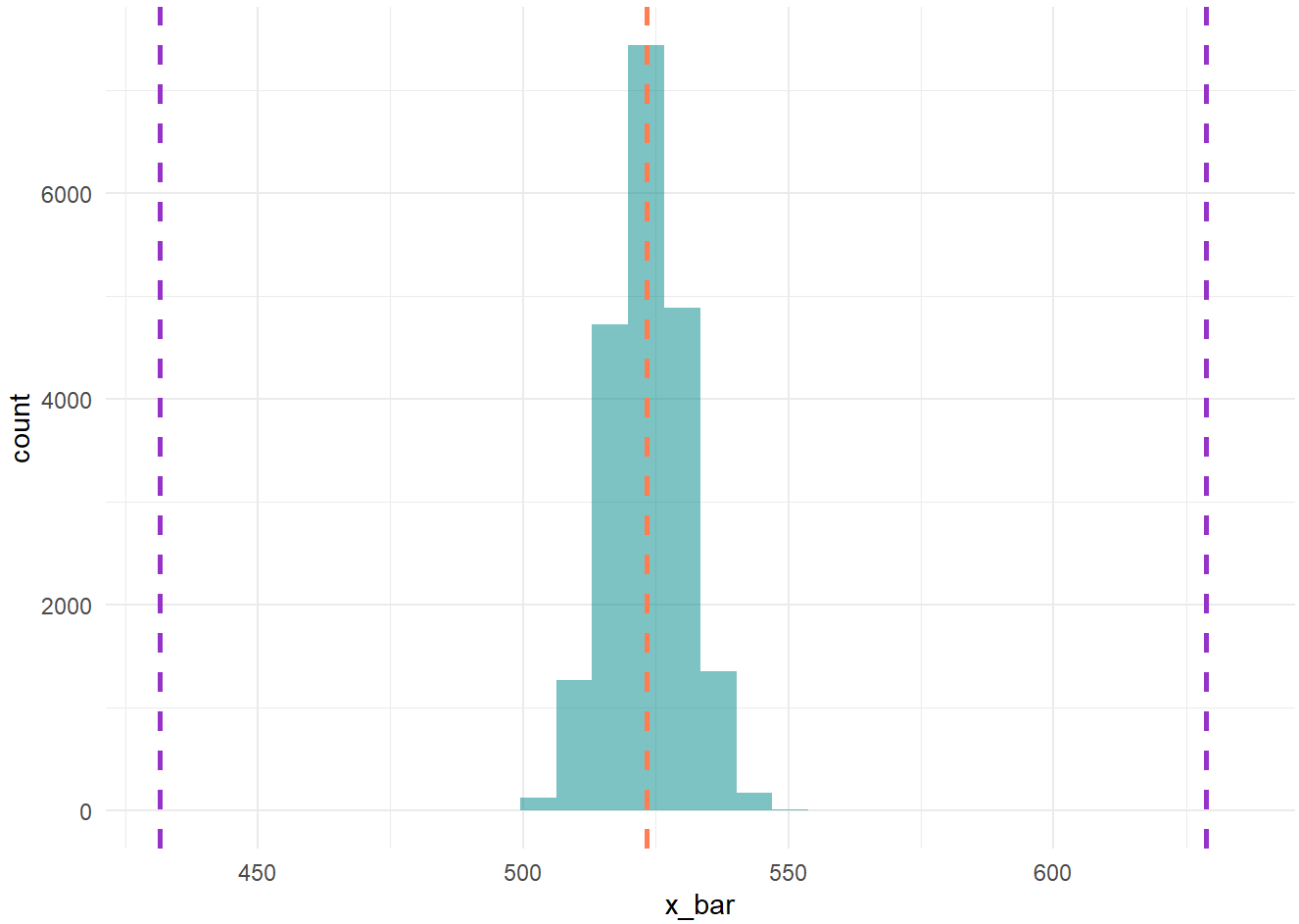
```
mean <- mean(nba_df$PTS)
sd <- sd(nba_df$PTS)
```

```
confidence_95 <- quantile(nba_df$PTS, c(0.025, 0.975))
confidence_95
```

```
  2.5%  97.5% 
   4.0 1755.2 
```

```
set.seed(42)
# creating null vector
n <- 10000
x_bar <-  rep(NA, n)

for(i in 1:n){
  x_bar[i] <- mean(sample(nba_df$PTS, 5000, replace = T))
}

set.seed(42)
# bootstrap mean & standard deviation
N <- 20000
simulation_list <- list(rep(NA, N), rep(NA, N))

for (i in 1:N){
  sample_i <- sample(nba_df$PTS, 5000, replace = T)
  simulation_list[[1]][i] <- mean(sample_i)
  simulation_list[[2]][i] <- sd(sample_i)
}

# creating sample
PTS_sample <- sample(nba_df$PTS, 5000, replace = T)

set.seed(42)
# sample mean
N <- 20000
n <- length(sample)
mean.boot <-  rep(NA, n)
#Loop
for(i in 1:N){
  mean.boot[i] <- mean(sample(PTS_sample, n, replace = T))
}

set.seed(42)
# sample standard deviation
N <- 50000
n <- length(sample)
sd.boot <-  rep(NA, n)

for(i in 1:N){
  sd.boot[i] <- sd(sample(PTS_sample, n, replace = T))
}

# plotting
ggplot(data = data.frame(x_bar = simulation_list[[1]]), aes(x = x_bar)) +
  geom_histogram(fill = "cyan4", alpha = 0.5, position = "identity") +
  geom_vline(xintercept = quantile(mean.boot, 0.025), linetype="dashed", # bootstrap CI
               color = "darkorchid", linewidth=1) +
  geom_vline(xintercept = quantile(mean.boot, 0.975), linetype="dashed",
               color = "darkorchid", linewidth=1) +
  geom_vline(xintercept = mean(x_bar), linetype="dashed",
               color = "coral", linewidth=1) +
  theme_minimal()
```

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




```
# sample mean
mean(mean.boot)
```

```
[1] 527.0932
```

```
# sample standard deviation
sd(mean.boot)
```

```
[1] 50.12356
```

```
# 95% CI
quantile(mean.boot, c(0.05, 0.975))
```

```
      5%    97.5%
446.2280 628.6702
```

(c)

```
bootstrap_ci90 <- quantile(mean.boot, c(0.05,0.95))
bootstrap_ci90
```

```
     5%     95%
446.228 613.001
```

```
bootstrap_ci95 <- quantile(mean.boot, c(0.05, 0.975))
bootstrap_ci95
```

```
      5%    97.5% 
446.2280 628.6702 
```

```
bootstrap_ci99 <- quantile(mean.boot, c(0.005, 0.995))
bootstrap_ci99
```

```
    0.5%    99.5% 
402.4770 657.7106 
```

```
# as the confidence interval gets larger, so does the difference between the quantiles!

# for instance a 90% CI is smaller than a 95% CI
# additionally, a 99% CI is larger than a 95% CI
```

(d)

```
# calculating 95%
est.mean.se <- sd/sqrt(length(PTS_sample))
q <- qnorm(1 - 0.05/2)
lower.bound <- mean(PTS_sample) - q*est.mean.se
upper.bound <- mean(PTS_sample) + q*est.mean.se

# text answer
print(paste0("The Plug-in 95% CI is {", lower.bound,", ",upper.bound,"}"))
```

```
[1] "The Plug-in 95% CI is {513.523259618271, 541.135140381729}"
```

(e)

```
set.seed(42)
# creating a sample
new_sample <- sample(nba_df, 5000, replace = T)

# mean PTS
sample_mean <- mean(new_sample$PTS)
sample_mean
```

```
[1] 523.4267
```

```
# standard deviation PTS
sample_sd <- sd(new_sample$PTS)
sample_sd
```

```
[1] 498.0844
```

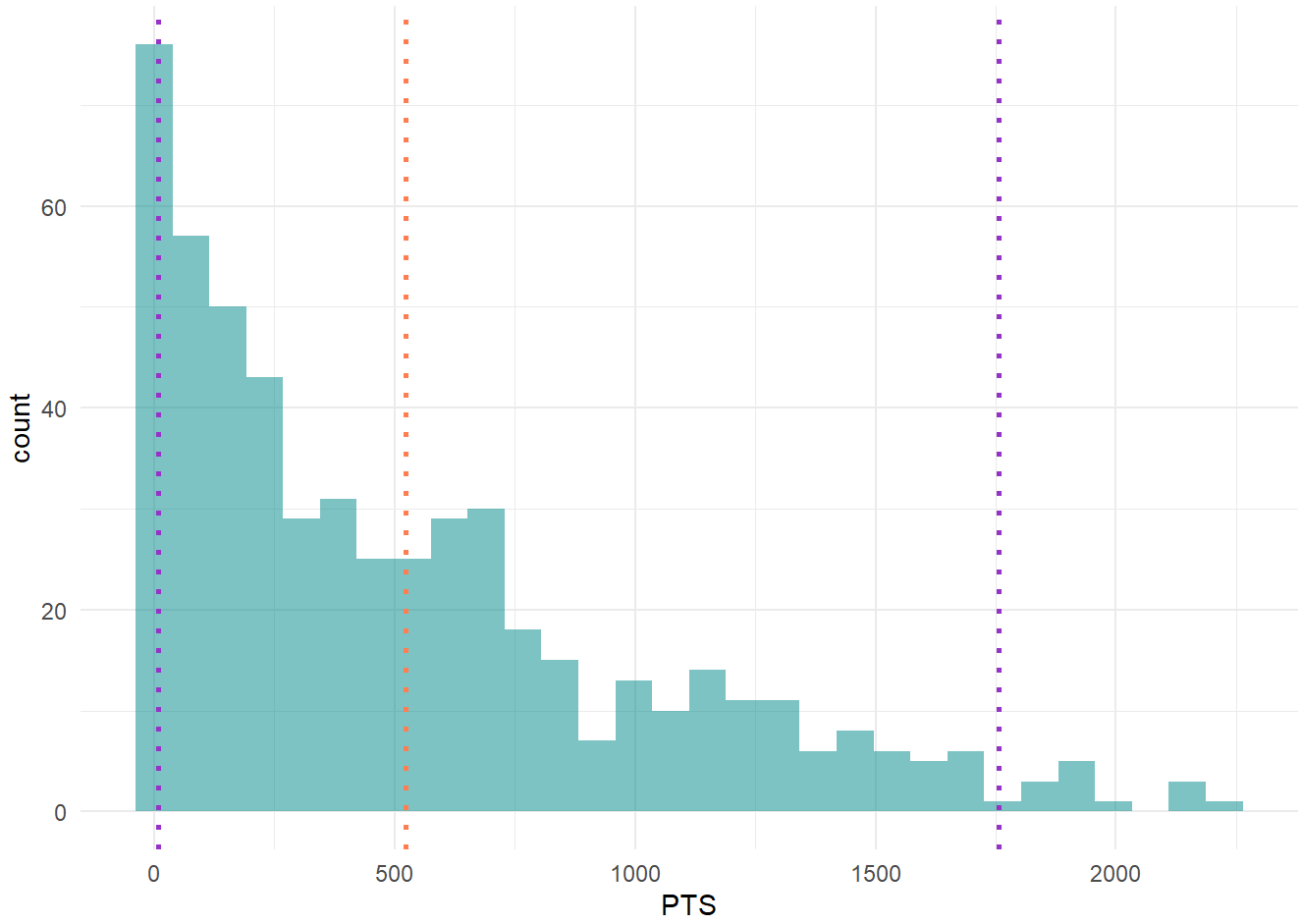
```
# calculating 95% CI
quantile(sample$PTS, c(0.05, 0.975))
```

```
    5%  97.5% 
   9.0 1755.2 
```

```
# histogram w/ plus or minus 1.96 standard deviations
ggplot(data = new_sample, aes(x = PTS)) +
  geom_histogram(fill = "cyan4", alpha = 0.5, position = 'identity') +
   geom_vline(xintercept = sample_mean, linetype="dotted",
              color = "coral", linewidth=1) +
  geom_vline(xintercept = quantile(sample$PTS, 0.05), linetype = 'dotted',
              color = "darkorchid", linewidth = 1) +
    geom_vline(xintercept = quantile(sample$PTS, 0.975), linetype = "dotted",
```

```
                color = "darkorchid", linewidth=1) +
  theme_minimal()
```

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



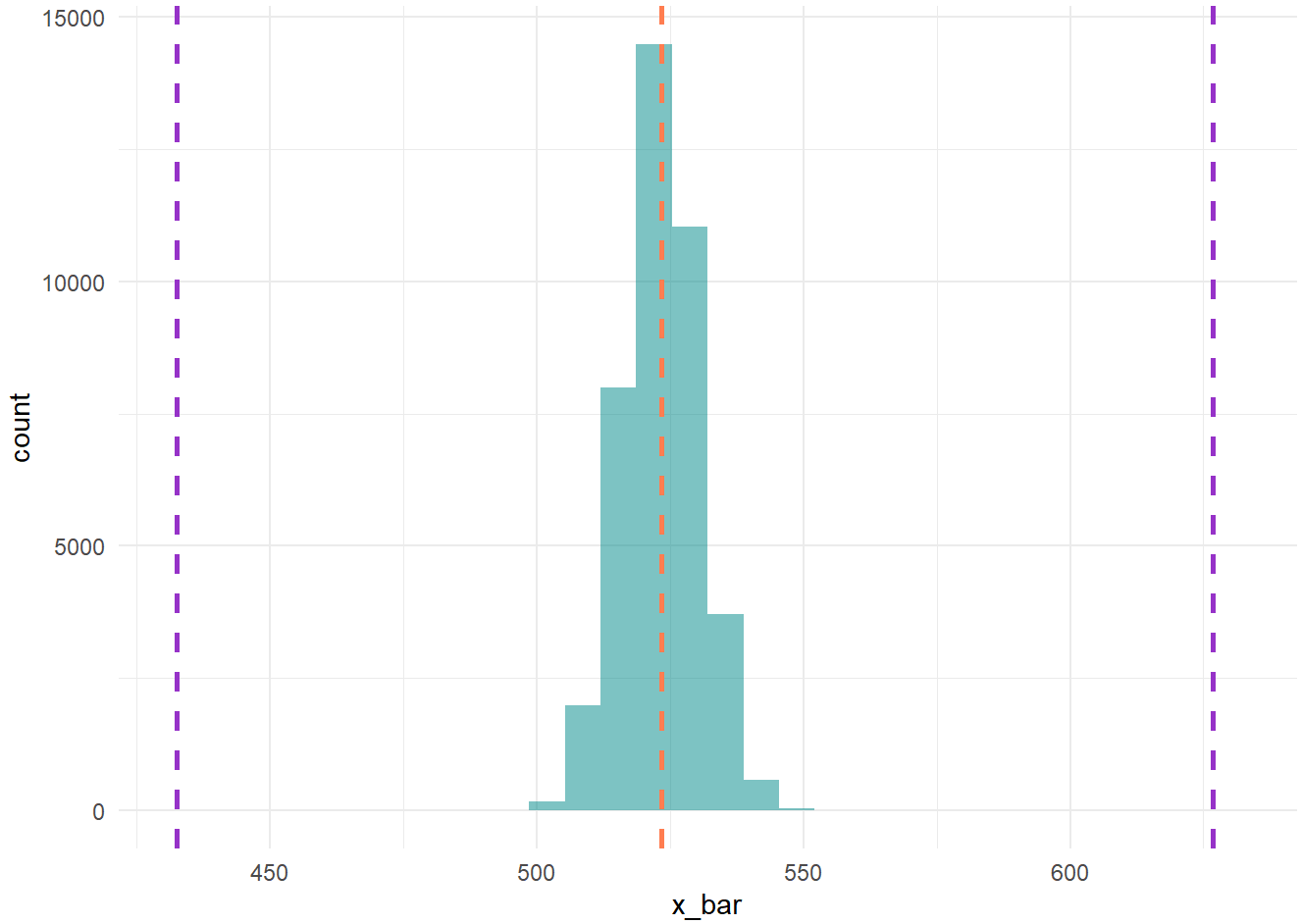

```
mean <- mean(nba_df$PTS)
sd <- sd(nba_df$PTS)

confidence_95 <- quantile(nba_df$PTS, c(0.025, 0.975))
confidence_95
```

```
  2.5%  97.5%
   4.0 1755.2
```

```
set.seed(42)
# creating null vector
n <- 50000
x_bar <-  rep(NA, n)

for(i in 1:n){
  x_bar[i] <- mean(sample(nba_df$PTS, 5000, replace = T))
}

set.seed(42)
# bootstrap mean & standard deviation
N <- 40000
simulation_list <- list(rep(NA, N), rep(NA, N))

for (i in 1:N){
  sample_i <- sample(nba_df$PTS, 5000, replace = T)
```

```
    simulation_list[[1]][i] <- mean(sample_i)
    simulation_list[[2]][i] <- sd(sample_i)
  }

  # creating sample
  new_PTS_sample <- sample(nba_df$PTS, 10000, replace = T)

  set.seed(42)
  # sample mean
  N <- 40000
  n <- length(sample)
  mean.boot <-  rep(NA, n)
  #Loop
  for(i in 1:N){
    mean.boot[i] <- mean(sample(new_PTS_sample, n, replace = T))
  }

  set.seed(42)
  # sample standard deviation
  N <- 50000
  n <- length(sample)
  sd.boot <-  rep(NA, n)

  for(i in 1:N){
    sd.boot[i] <- sd(sample(new_PTS_sample, n, replace = T))
  }

  # plotting
  ggplot(data = data.frame(x_bar = simulation_list[[1]]), aes(x = x_bar)) +
    geom_histogram(fill = "cyan4", alpha = 0.5, position = "identity") +
    geom_vline(xintercept = quantile(mean.boot, 0.025), linetype="dashed", # bootstrap CI
                  color = "darkorchid", linewidth=1) +
    geom_vline(xintercept = quantile(mean.boot, 0.975), linetype="dashed",
                  color = "darkorchid", linewidth=1) +
    geom_vline(xintercept = mean(x_bar), linetype="dashed",
                  color = "coral", linewidth=1) +
    theme_minimal()
```

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




```
# sample mean
mean(mean.boot)
```

```
[1] 526.9506
```

```
# sample standard deviation
sd(mean.boot)
```

```
[1] 49.60123
```

```
# 95% CI
quantile(mean.boot, c(0.05, 0.975))
```

```
      5%    97.5%
447.1190 626.7705
```

```
est.mean.se <- sd/sqrt(length(new_PTS_sample))
q <- qnorm(1 - 0.05/2)
lower.bound <- mean(new_PTS_sample) - q*est.mean.se
upper.bound <- mean(new_PTS_sample) + q*est.mean.se

# text answer
print(paste0("The Plug-in 95% CI is {", lower.bound,", ",upper.bound,"}"))
```

```
[1] "The Plug-in 95% CI is {516.914725935423, 536.439274064577}"
```

Exercise #3

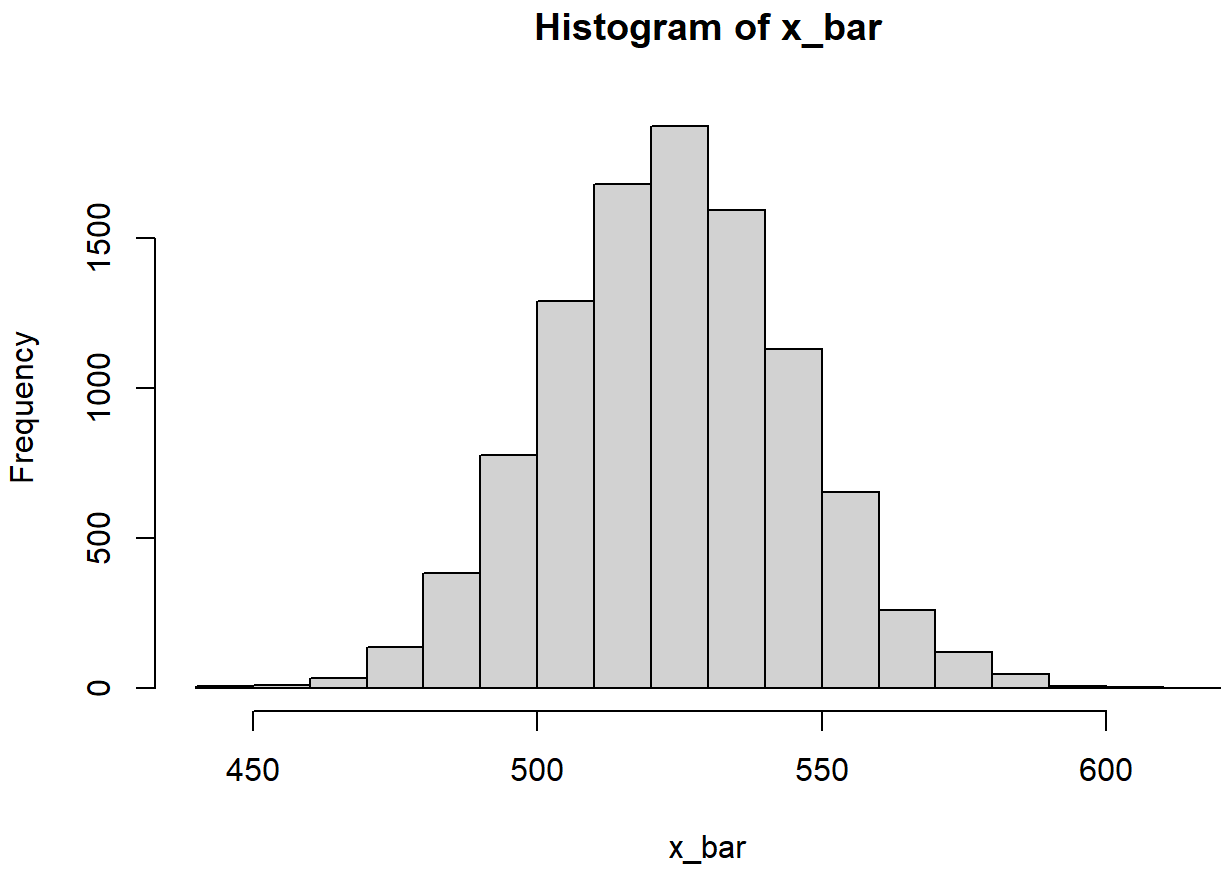
(a)

```
mean(nba_df$PTS)
```

```
[1] 523.4267
```

(b)

```
set.seed(42)
#Save the sample mean
n <- 10000
x_bar <-  rep(NA, n)
#Loop
for(i in 1:n){
  x_bar[i] <- mean(sample(nba_df$PTS, 539, replace = T))
}

hist(x_bar)
```




```
mean(x_bar)
```

```
[1] 523.7285
```

```
sd(x_bar)
```

```
[1] 21.4766
```

```
quantile(x_bar, c(0.05, 0.975))
```

```
       5%    97.5%
 488.5382 566.3769

 width <- quantile(x_bar, 0.975) - quantile(x_bar, 0.05)
 width
```

```
   97.5%
77.83868
```