J.Kimbrough_DATA-413_HW2

2024-10-02

Problem 1:

```
library(tidyverse)
tribble( ~x,     ~y,   ~w,    ~z,
         210,    300,  220,   180,
         102,    100,  119,   187,
         176,    175,  188,   173,
         87,     95,   91,    94,
         202,    210,  234,   218,
         110,    122,  131,   128,
) -> dt
dt
```

```
## # A tibble: 6 x 4
##       x     y     w     z
##   <dbl> <dbl> <dbl> <dbl>
## 1   210   300   220   180
## 2   102   100   119   187
## 3   176   175   188   173
## 4    87    95    91    94
## 5   202   210   234   218
## 6   110   122   131   128
```

1a.

Use and show a map function to find the "mean" of each column of the dt data table.

```
column_means <- dt %>%
  map_dbl(mean)
column_means
```

```
##        x        y        w        z
## 147.8333 167.0000 163.8333 163.3333
```

1b.

Use and show a map function that will calculate the "standard deviation" of each value of each column of the data table dt.

```
column_sds <- dt %>%
  map_dbl(sd)
column_sds
```

```
##        x        y        w        z
## 54.45151 79.12016 58.40348 44.66617
```

1c.

Use and show a map function that will calculate the "square root" of each value of each column of the data table dt.

```
column_sqrts <- dt %>%
  map_df(sqrt)
column_sqrts
```

```
## # A tibble: 6 x 4
##       x     y     w     z
##   <dbl> <dbl> <dbl> <dbl>
## 1 14.5  17.3  14.8  13.4
## 2 10.1  10    10.9  13.7
## 3 13.3  13.2  13.7  13.2
## 4  9.33  9.75  9.54  9.70
## 5 14.2  14.5  15.3  14.8
## 6 10.5  11.0  11.4  11.3
```

1d.

Use R code to find the "mean", "max", "1st Quartile", "3rd Quartile", "Median", and "Min" for each column of the dt data table. (Hint: You do not have to use a map function).

```
column_summaries <- dt %>%
  summary()
column_summaries
```

```
##        x                y                w                z
##  Min.   : 87.0   Min.   : 95.0   Min.   : 91.0   Min.   : 94.0
##  1st Qu.:104.0   1st Qu.:105.5   1st Qu.:122.0   1st Qu.:139.2
##  Median :143.0   Median :148.5   Median :159.5   Median :176.5
##  Mean   :147.8   Mean   :167.0   Mean   :163.8   Mean   :163.3
##  3rd Qu.:195.5   3rd Qu.:201.2   3rd Qu.:212.0   3rd Qu.:185.2
##  Max.   :210.0   Max.   :300.0   Max.   :234.0   Max.   :218.0
```

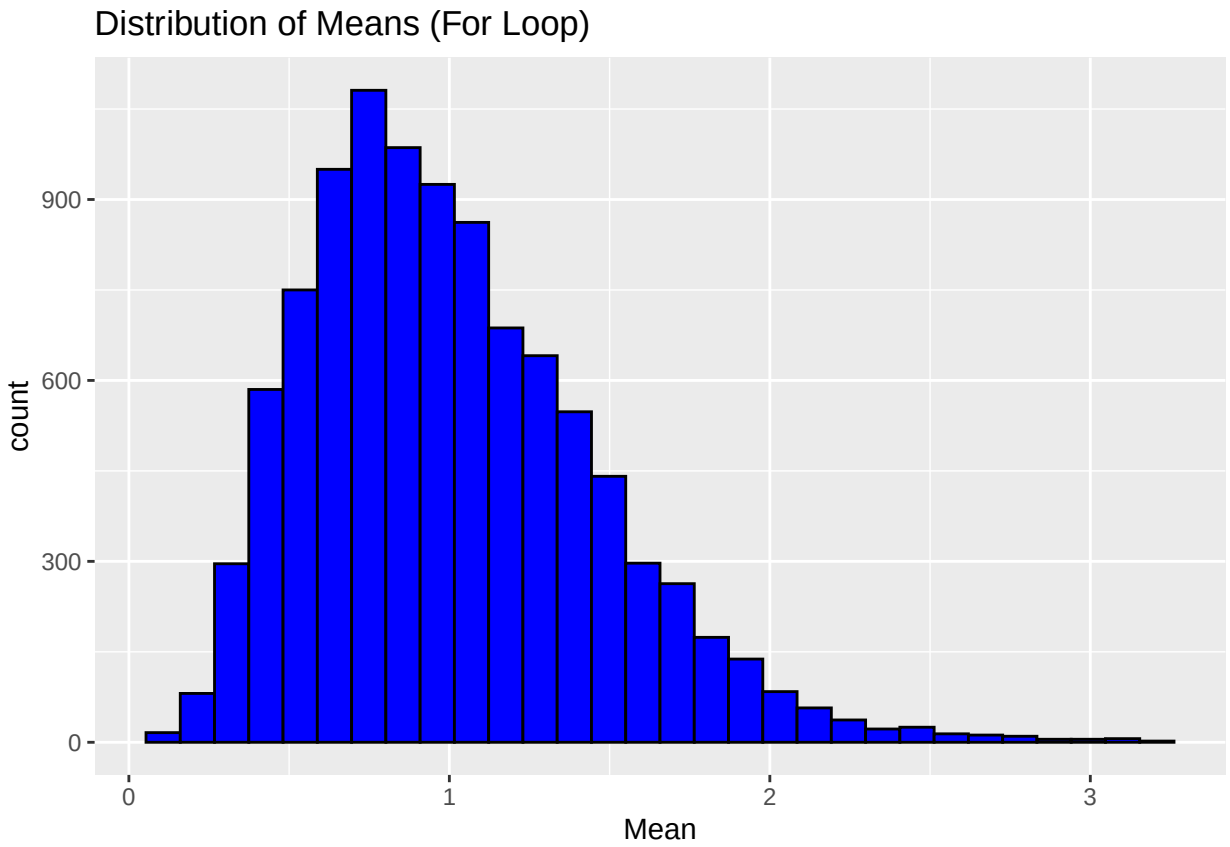
Problem 2:

Write a function that uses a for loop for each iteration, randomly draws 5 observations from an exponential distribution with "rate" parameter 1 (use rexp()) and calculates its "mean". It should do this 10,000 times. Choose an appropriate plot to plot the distribution of "means".

```
set.seed(123)

draw_means_for <- function(iterations = 10000, sample_size = 5, rate = 1){
  means <- numeric(iterations)

  for(i in 1:iterations){
  sample <- rexp(sample_size, rate)
  means[i] <- mean(sample)
  }
  return (means)
}

means_for <- draw_means_for()
```

```
means_df <- data.frame(means = means_for)

ggplot(means_df, aes(x = means)) +
  geom_histogram(fill = "blue", color = "black") +
  xlab("Mean") +
  ggtitle("Distribution of Means (For Loop)")
```

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



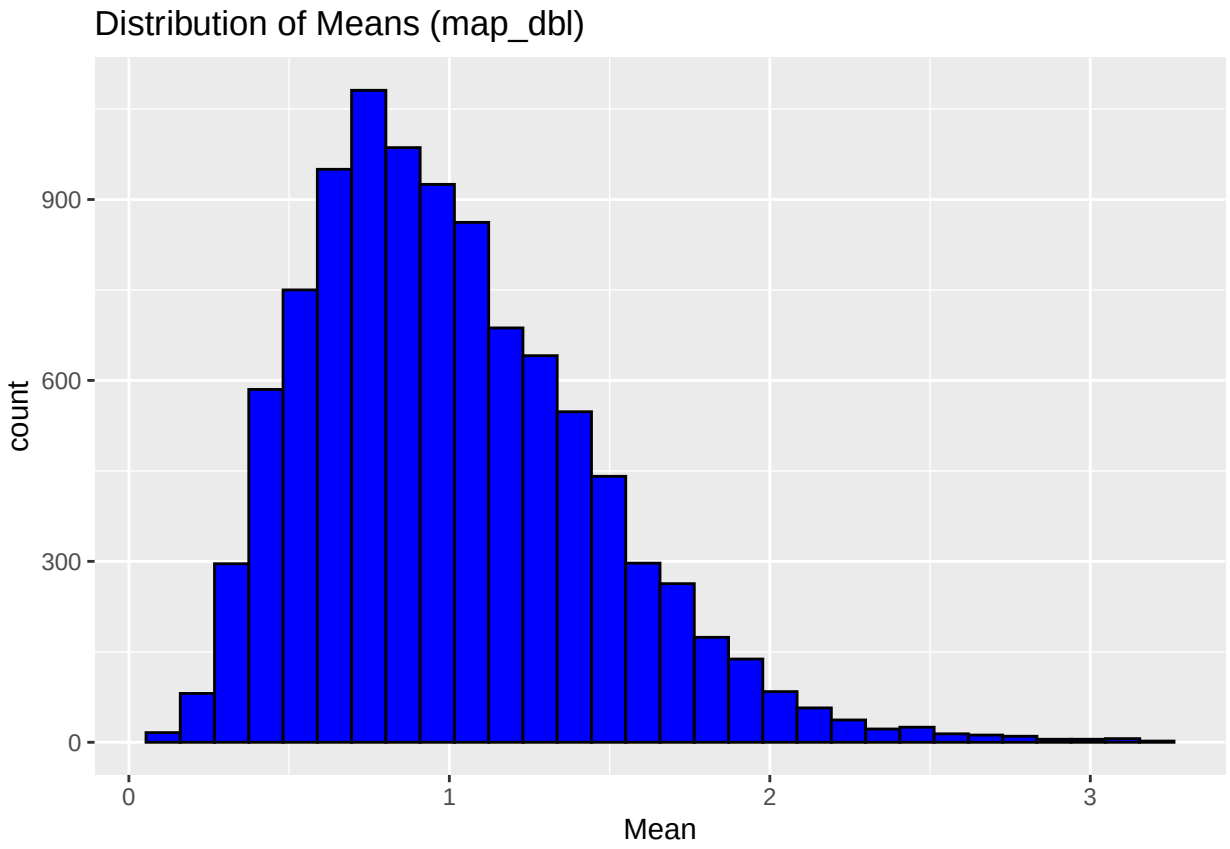
2a.

Repeat part 1 by using a map_* () function.

```
set.seed(123)

means_map <- map_dbl(1:10000, ~ mean(rexp(5, rate = 1)))
```

```
means_map_df <- data.frame(means = means_map)

ggplot(means_map_df, aes(x = means)) +
  geom_histogram(fill = "blue", color = "black") +
  xlab("Mean") +
  ggtitle("Distribution of Means (map_dbl)")
```

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



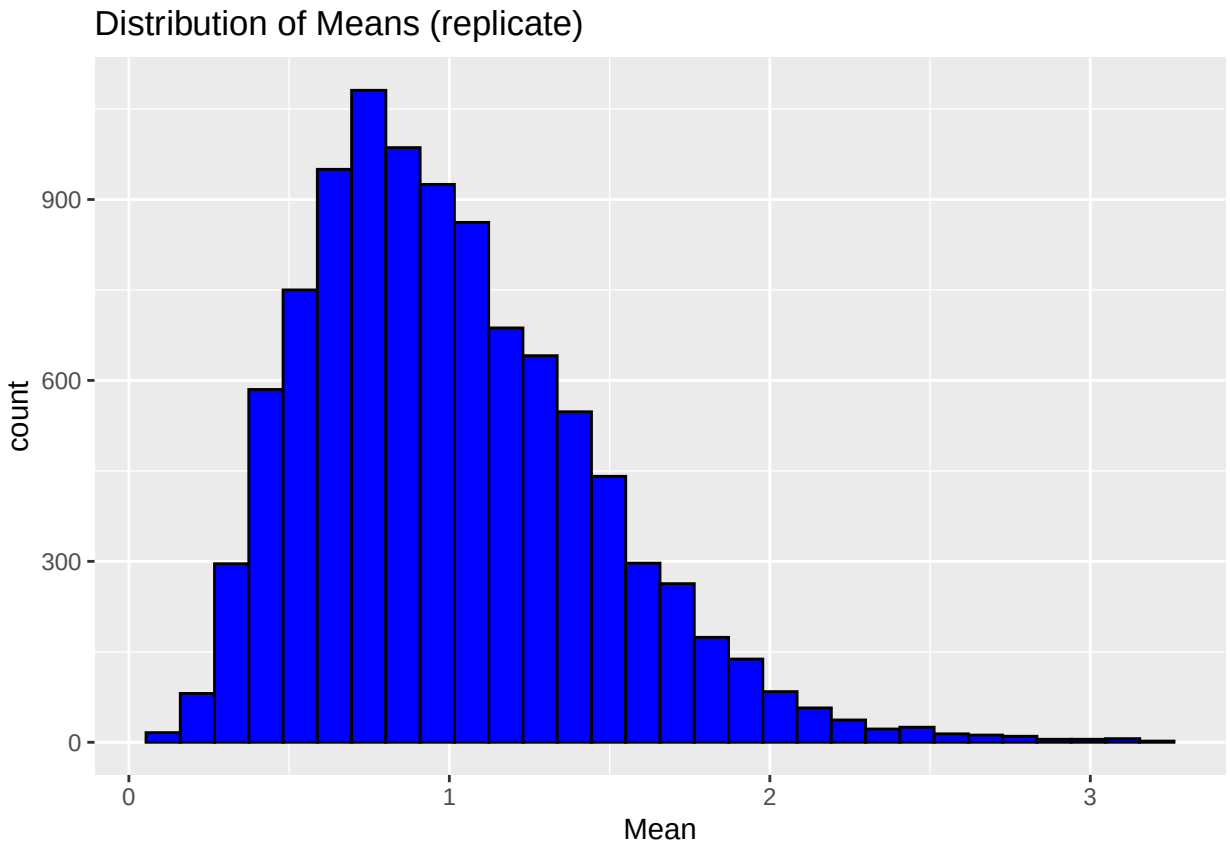
2b.

Repeat part 1 by using the replicate () function.

```
set.seed(123)

means_replicate <- replicate(10000, mean(rexp(5, rate = 1)))
```

```
means_replicate_df <- data.frame(means = means_replicate)

ggplot(means_replicate_df, aes(x = means)) +
  geom_histogram(fill = "blue", color = "black") +
  xlab("Mean") +
  ggtitle("Distribution of Means (replicate)")
```

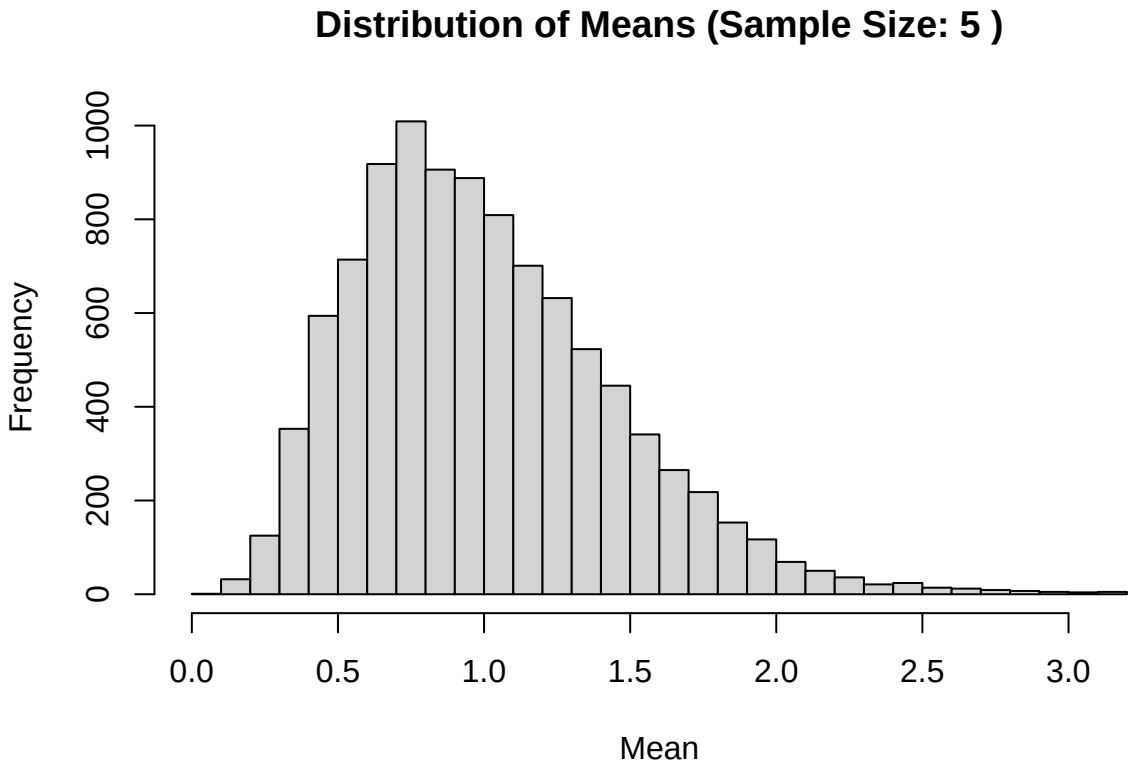
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

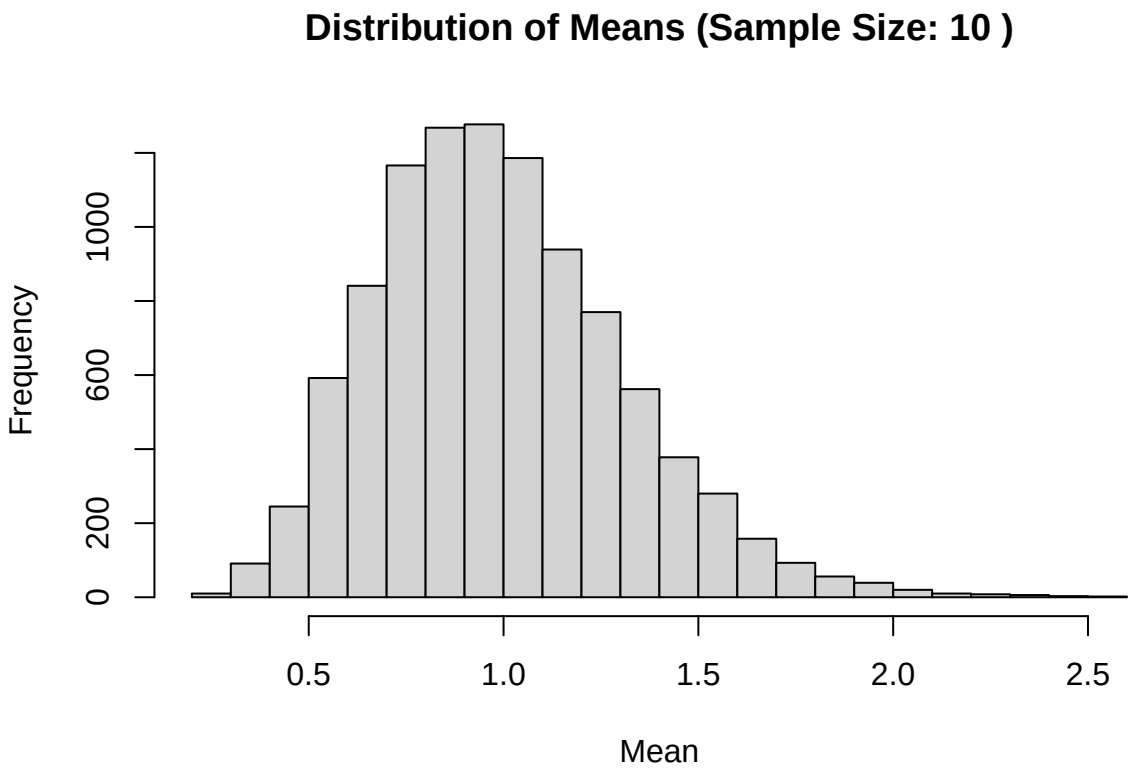


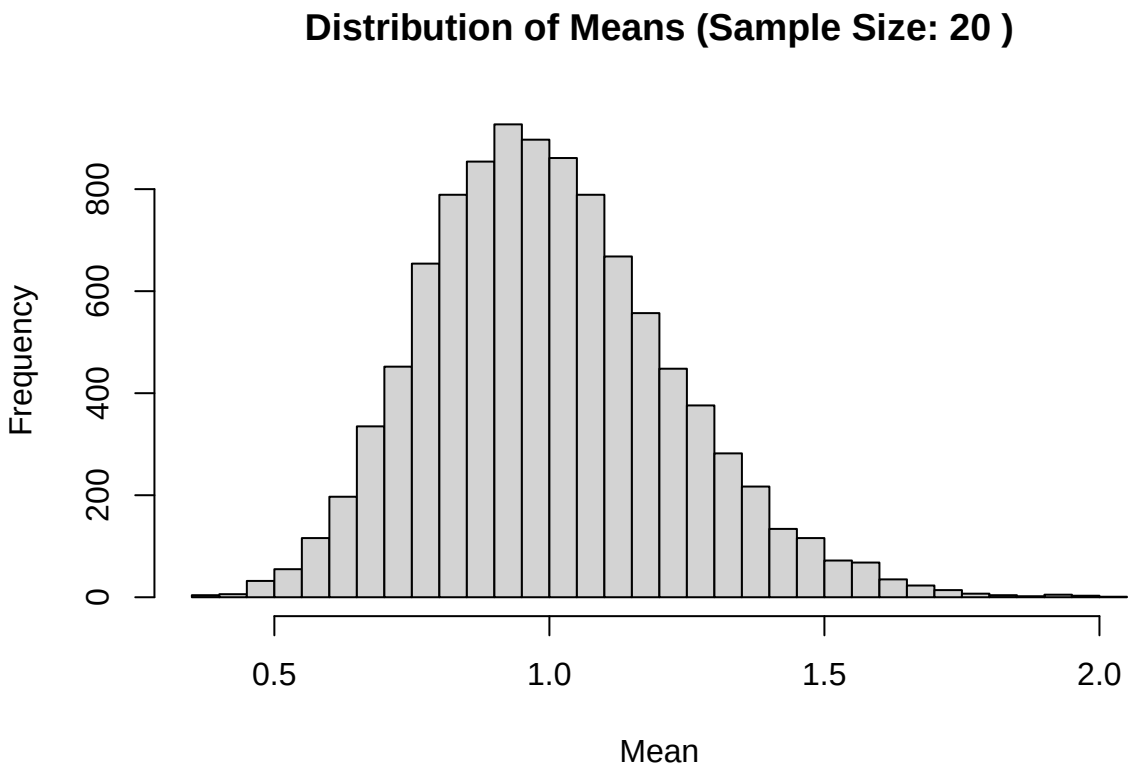
2c.

Use another for loop that will print out plots for sample size of 5, 10, and 20 observations (instead of just 5).

```
sample_sizes <- c(5, 10, 20)

set.seed(123)

for(n in sample_sizes){
  means <- numeric(10000)

  for(i in 1:10000){
    sample <- rexp(n, rate = 1)
    means[i] <- mean(sample)
  }
  hist(means, main = paste("Distribution of Means (Sample Size:", n, ")"),
       xlab = "Mean", breaks = 30)
}
```


Distribution of Means (Sample Size: 5)
Frequency
0
200
400
600
800
1000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Mean


Distribution of Means (Sample Size: 10)
Frequency
0
200
600
1000
0.5
1.0
1.5
2.0
2.5
Mean




```
#helpful link: https://www.geeksforgeeks.org/histograms-in-r-language/
```

Problem 3:

Use and show R coding to calculate the "standard deviation" for each variable of the data table mtcars using the "Special For Loop Method".

```
data("mtcars")
mtcars
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
```

```
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
output <- vector("double", ncol(mtcars))
for(i in seq_along(mtcars)){
  output[[i]] <- sd(mtcars[[i]])
}
names(output) <- colnames(mtcars)

output
```

```
##         mpg         cyl        disp          hp        drat          wt
##   6.0269481   1.7859216 123.9386938  68.5628685   0.5346787   0.9784574
##        qsec          vs          am        gear        carb
##   1.7869432   0.5040161   0.4989909   0.7378041   1.6152000
```