Lab 4: Sampling Variation

July 12, 2016

1 Variation of the Sample Mean as Sample Size Changes

Given a random sample drawn from a population, two of the simplest descriptive statistics we can calculate are the sample mean

$$\frac{1}{n}\sum_i X_i,$$

and the sample standard deviation

$$\sqrt{\frac{1}{n-1}\sum_i (X_i - \overline{X}_n)^2}.$$

The sample mean and standard deviation are functions of the random sample so they are also random, i.e. each time we are given a different random sample from the population we get different results, so they themselves have mean and standard deviations (variation).

The sample mean and sample variance (i.e. sample standard deviation squared) are known to be *unbiased* estimators of their population counterparts, meaning that they have mean equal to their population values. In other words, if we repeatedly calculate the sample mean (resp. sample variance) for a random sample and repeat this for infinitely many times, the average of these sample means (and variances) will be equal to the true population mean (and variance).

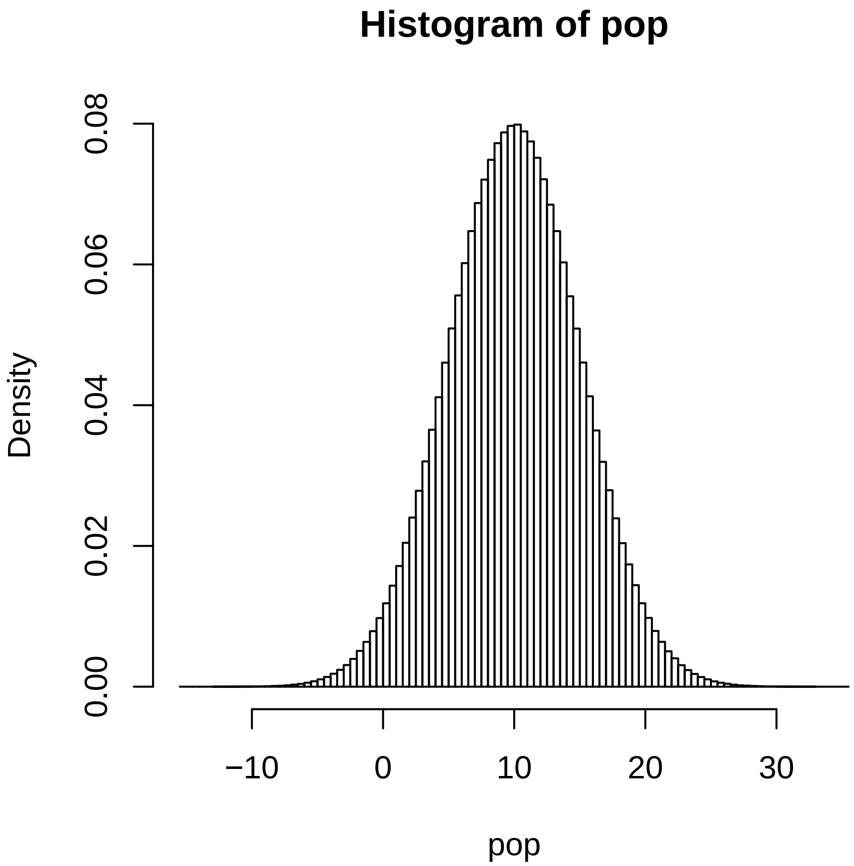
Intuitively when we are given more samples (i.e. n becomes larger) we should estimate these parameters more accurately, so the sample estimates should vary less and be more concentrated at their true population parameters. Today we will examine how this variability changes as we vary our sample size n.

1.1 Population

Suppose we have a population of size 10 million that comes from the famous bell-curved normal distribution. A univariate normal distribution takes one parameter that specifies the mean, and another for standard deviation.

```r
# Setting the seed to the same number will ensure you obtain the same results each time
# you run the code. You will not run into trouble without this line -- you will simply get
# different results every time.
set.seed(311)
# Population size
N <- 10000000
# Draw a population from the normal distribution
pop <- rnorm(N, 10, 5)
# True population mean
pop.mean <- mean(pop)
pop.mean
```

```
## [1] 10.00087

# True population standard deviation
pop.sd <- sd(pop)
pop.sd

## [1] 4.99784

# Histogram of the population.
hist(pop, breaks = 100, freq = FALSE)
```



1.2 One Sample Mean and Standard Deviation

Recall how we used the `sample` command to draw a sample from the population. Suppose we are given a sample of size 100 and we calculate its sample mean and sample standard deviation.

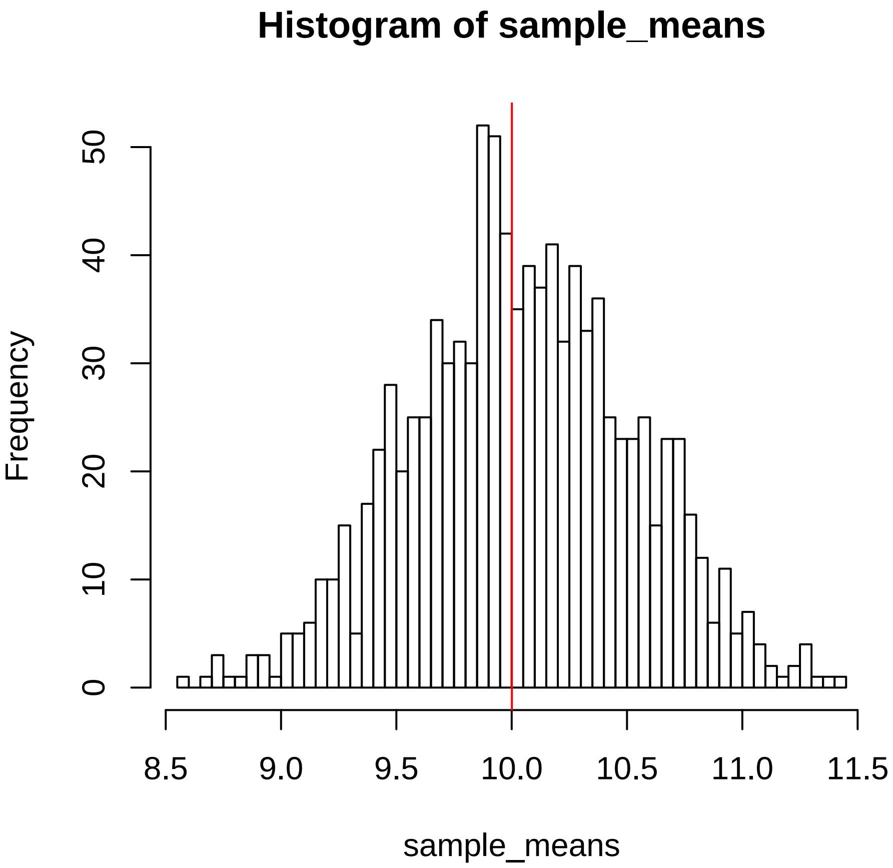
```
# Sample size n = 100
sample_size <- 100
# Draw one sample from the population
one_sample <- sample(pop, sample_size)
mean(one_sample)

## [1] 10.52653

sd(one_sample)
```

```
## [1] 4.843895
```

1.3 Histogram of Sample Means

From now on we focus on the sample mean only. Suppose we fix our sample size $n = 100$ and repeat this for 1000 times. Let us see how these 1000 sample means vary.

```
# Try for 1000 times with sample size 100
trials <- 1000
n <- 100
# c(0,0,...,0) with 1000 slots
sample_means <- rep(0, trials)
# Repeat for 1000 times
for (i in 1:trials){
  # Draw one sample from the population
  one_sample <- sample(pop, n)
  # Store its sample mean
  sample_means[i] <- mean(one_sample)
}
# Draw histogram
hist(sample_means, breaks = 50)
# True population mean, which should be at about the center of the histogram
abline(v = pop.mean, col = "red")
```

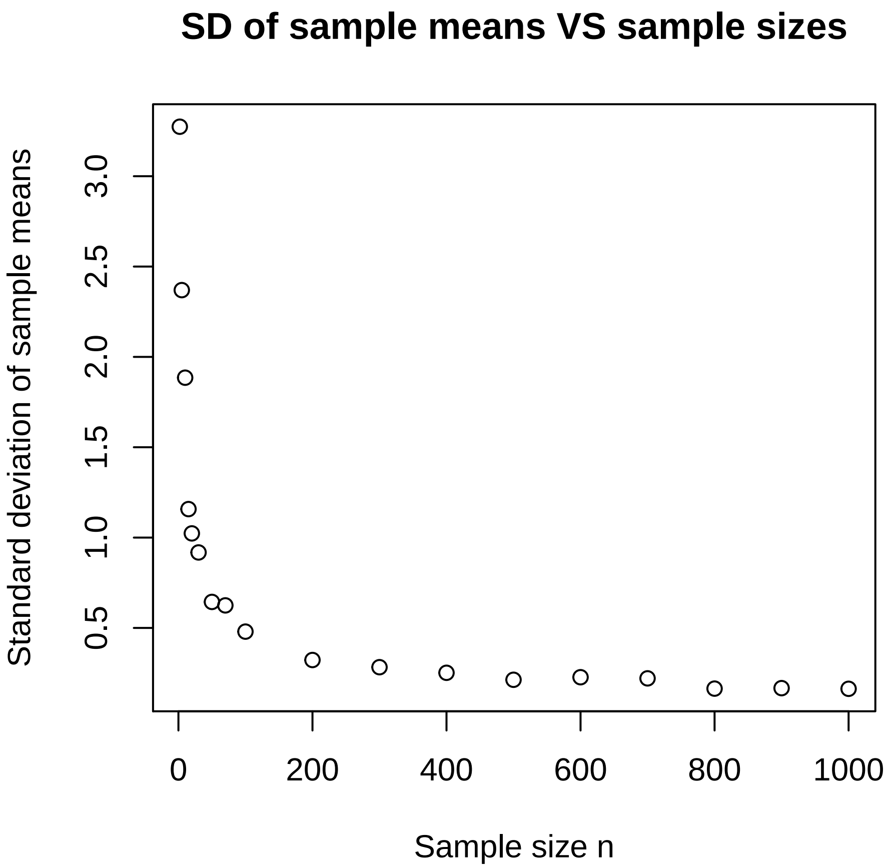


```
# The sample mean of these 1000 sample means, which should be close to pop.mean
mean(sample_means)

## [1] 10.04377

pop.mean

## [1] 10.00087

# The sample standard deviation of these 1000 sample means
sd(sample_means)

## [1] 0.4835545
```

1.4 How would `sd(sample_means)` change in n?

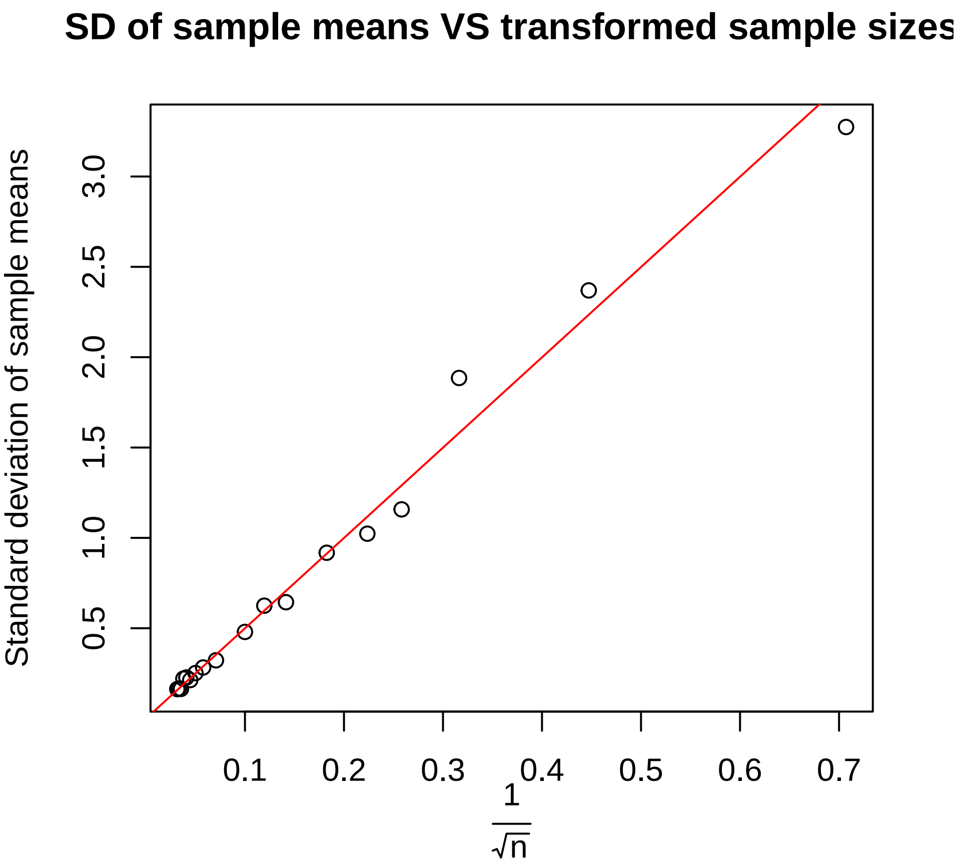
Let us now examine how `sd(sample_means)` (the standard deviation of the sample means) change as we increase the sample size n. To speed up calculations we decrease `trials` to 100.

```
# Different sample sizes
ns <- c(2, 5, 10, 15, 20, 30, 50, 70, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000)
trials <- 100
# sd(sample_means) for each n
sds_of_sample_means <- rep(0, length(ns))
for (n_index in 1:length(ns)){
  current_n <- ns[n_index]
  # 100 sample means for this n
  sample_means <- rep(0, trials)
  for (i in 1:trials){
    # Draw one sample from population
    one_sample <- sample(pop, current_n)
    sample_means[i] <- mean(one_sample)
  }
  # sd(sample_means) for this n
  sds_of_sample_means[n_index] <- sd(sample_means)
}

# Scatter plot of sd(sample_means) versus n
plot(ns, sds_of_sample_means, xlab = "Sample size n",
     ylab = "Standard deviation of sample means",
     main = "SD of sample means VS sample sizes")
```



```
# Scatter plot of sd(sample_means) versus 1/sqrt(n)
plot(1/sqrt(ns), sds_of_sample_means, xlab = expression(frac(1, sqrt(n))),
     ylab = "Standard deviation of sample means",
     main = "SD of sample means VS transformed sample sizes")

# Add a straight line with slope pop.sd to the previous plot
```

```
abline(a = 0, b = pop.sd, col = "red")
```



The previous plot suggests that the standard deviation of $\overline{X}_n$ is possibly $\frac{\sigma}{\sqrt{n}}$, where σ is the (true) population standard deviation. This, as we will see, is in fact theoretically justified.

2 Variation of the Sample Standard Deviation as Sample Size Changes

Repeat the analysis above for sample standard deviation. Can you guess the approximate formula for the standard deviation of the sample standard deviations in terms of n?

It turns out that for data that comes from the *normal* distribution, the theoretical standard deviation of sample variances is *exactly* $\frac{\sqrt{2}\sigma^2}{\sqrt{n-1}}$, and the standard deviation of sample standard deviations is *approximately* $\frac{\sigma}{\sqrt{2n}}$, which is a good approximation when n is large.