Large Sample Theory Project

Jieqi Tu

11/5/2020

The algorithm of this project:

- Draw n (n = 10, 20 and 30) samples from $Gamma(k = 1.67, \theta = 49.98)$.
- Compute $\bar{x}$ and s.
- Repeat 1000 times. Then we will get $(\bar{x_1}, s_1), (\bar{x_2}, s_2), \cdots, (\bar{x_{1000}}, s_{1000})$.
- Compute $\bar{\bar{x}} = \frac{\sum \bar{x_i}}{1000}$, $\bar{s} = [\frac{\sum (\bar{x_i} - \bar{\bar{x}})^2}{1000}]^{\frac{1}{2}}$
- Compute $z = \frac{\bar{x_i} - \bar{\bar{x}}}{\bar{s}/\sqrt{n}}$.
- Draw density function based on $z_i, i = 1, 2, \cdots, 1000$.
- Compute $\bar{\bar{s_1^2}} = \frac{\sum (\bar{x_i} - \bar{\bar{x}})^2}{1000} \times n$ and $\bar{\bar{s_2^2}} = \frac{\sum s_i^2}{1000}$

In this project, we want to compare $\bar{\bar{s_1^2}}$, $\bar{\bar{s_2^2}}$ and $k\theta^2$.

Scenario: n = 10

Then let's firstly try n = 10.

```
set.seed(24)
k = 1.67
n = 10
theta = 49.98
x_bar = numeric(1000)
s = numeric(1000)
for (i in 1:1000) {
  sample = rgamma(n, shape = k, scale = theta)
  x_bar[i] = mean(sample)
  s[i] = sd(sample)
}
mean(x_bar)
```

```
## [1] 82.93522
```

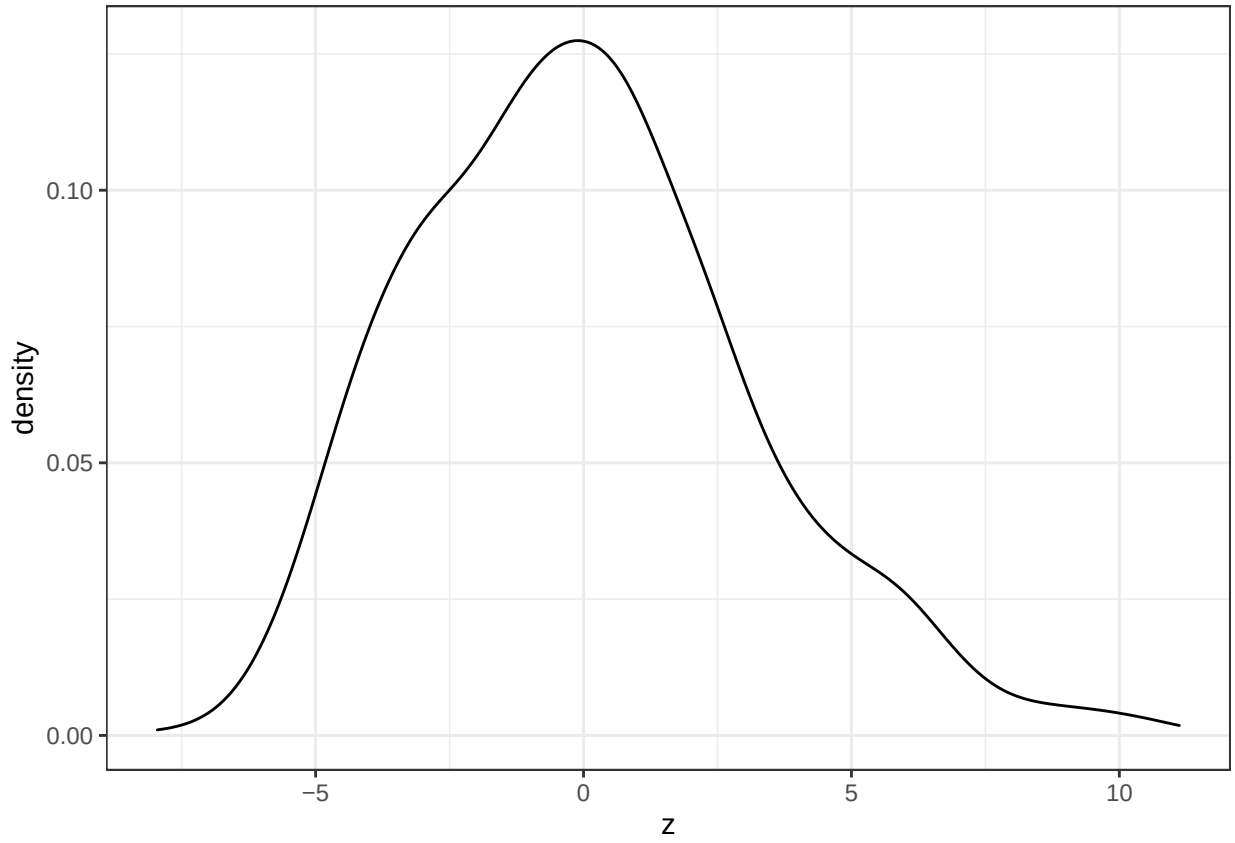
```
sd(x_bar)
```

```
## [1] 19.90318
```

```
z = (x_bar - mean(x_bar))/(sd(x_bar)/sqrt(n))
df = data.frame(z = z)
```

When the sample size n = 10, of 1000 samples, the mean of sample mean is 82.9352159, and the standard deviation of the sample mean is 19.9031755.

```
density1 = ggplot(df, aes(x = z)) + geom_density() + theme_bw()
density1
```



Here we plotted the density of z. We can see that the distribution is a little bit right skewed and goes down more rapidly after the peak. The peak is approximately a slightly left to the origin.

```
# Calculate s1 and s2 when n = 10
s1_1 = sum((x_bar-mean(x_bar))^2)/1000 * n; s1_1
```

```
## [1] 3957.403
```

```
s2_1 = sum(s^2)/1000; s2_1
```

```
## [1] 4138.226
```

The values for $\bar{s_1^2}$ and $\bar{s_2^2}$ are calculated when n = 10. Here, the $\bar{s_2^2}$ is slightly larger than $\bar{s_1^2}$, and is more close to the theoretical value 4171.660668.

Scenario: n = 20

Then let us increase the sample size. This time, n = 20.

```
set.seed(24)
n = 20
x_bar2 = numeric(1000)
s2 = numeric(1000)
for (i in 1:1000) {
  sample = rgamma(n, shape = k, scale = theta)
  x_bar2[i] = mean(sample)
  s2[i] = sd(sample)
}
mean(x_bar2)
```

```
## [1] 83.25347
```

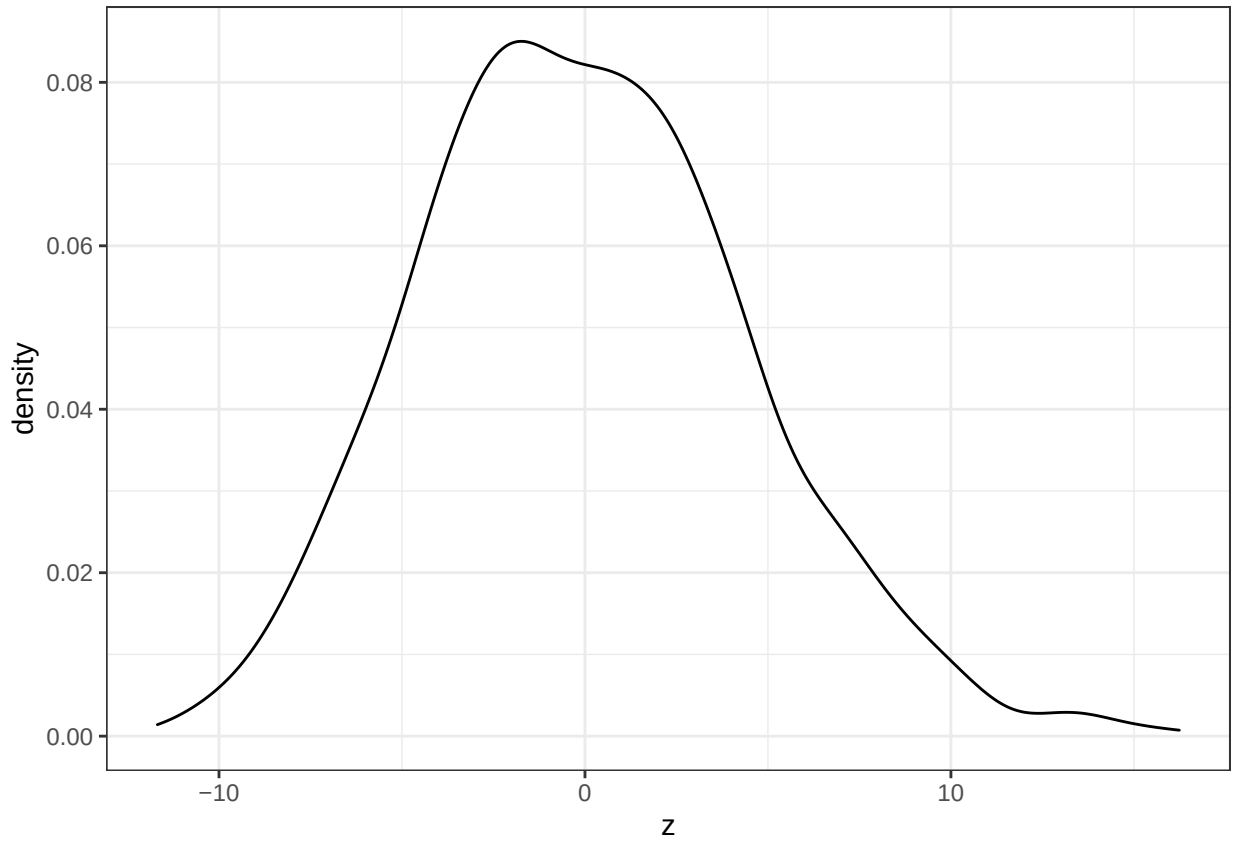
```
sd(x_bar2)
```

```
## [1] 14.74353
```

```
z2 = (x_bar2 - mean(x_bar2))/(sd(x_bar2)/sqrt(n))
df2 = data.frame(z = z2)
```

When the sample size n = 20, of 1000 samples, the mean of sample mean is 83.2534668, and the standard deviation of the sample mean is 14.7435267. We could notice that, although the mean remains the same as it of n = 10, the standard deviation becomes lower than before.

```
density2 = ggplot(df2, aes(x = z)) + geom_density() + theme_bw()
density2
```



This is the distribution of z when sample size n = 20. Here we could see that the distribution becomes more symmetric than that of n = 10. This time, the peak also arrives left to the origin. The distribution is still slightly right skewed.

```
s1_2 = sum((x_bar2-mean(x_bar2))^2)/1000 * n; s1_2
```

```
## [1] 4343.084
```

```
s2_2 = sum(s2^2)/1000; s2_2
```

```
## [1] 4232.641
```

Here we could see that, $\bar{s_2^2}$ is still more close to the theoretical value, compared to $\bar{s_1^2}$.

Scenario: n = 30

Then we want to increase the sample size from 20 to 30, and see what will happen.

```
set.seed(24)
n = 30
x_bar3 = numeric(1000)
s3 = numeric(1000)
for (i in 1:1000) {
  sample = rgamma(n, shape = k, scale = theta)
```

```
  x_bar3[i] = mean(sample)
  s3[i] = sd(sample)
}
mean(x_bar3);
```

```
## [1] 83.49558
```

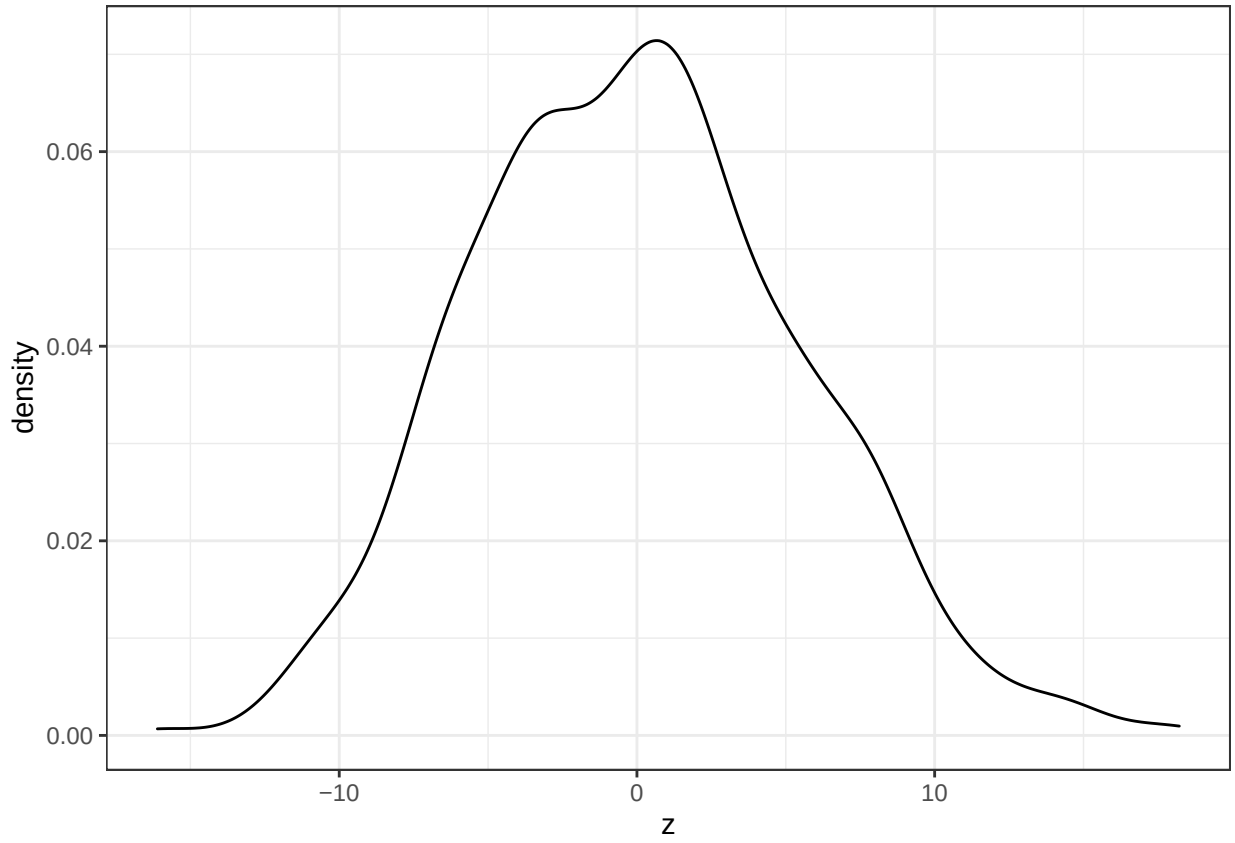
```
sd(x_bar3)
```

```
## [1] 11.89414
```

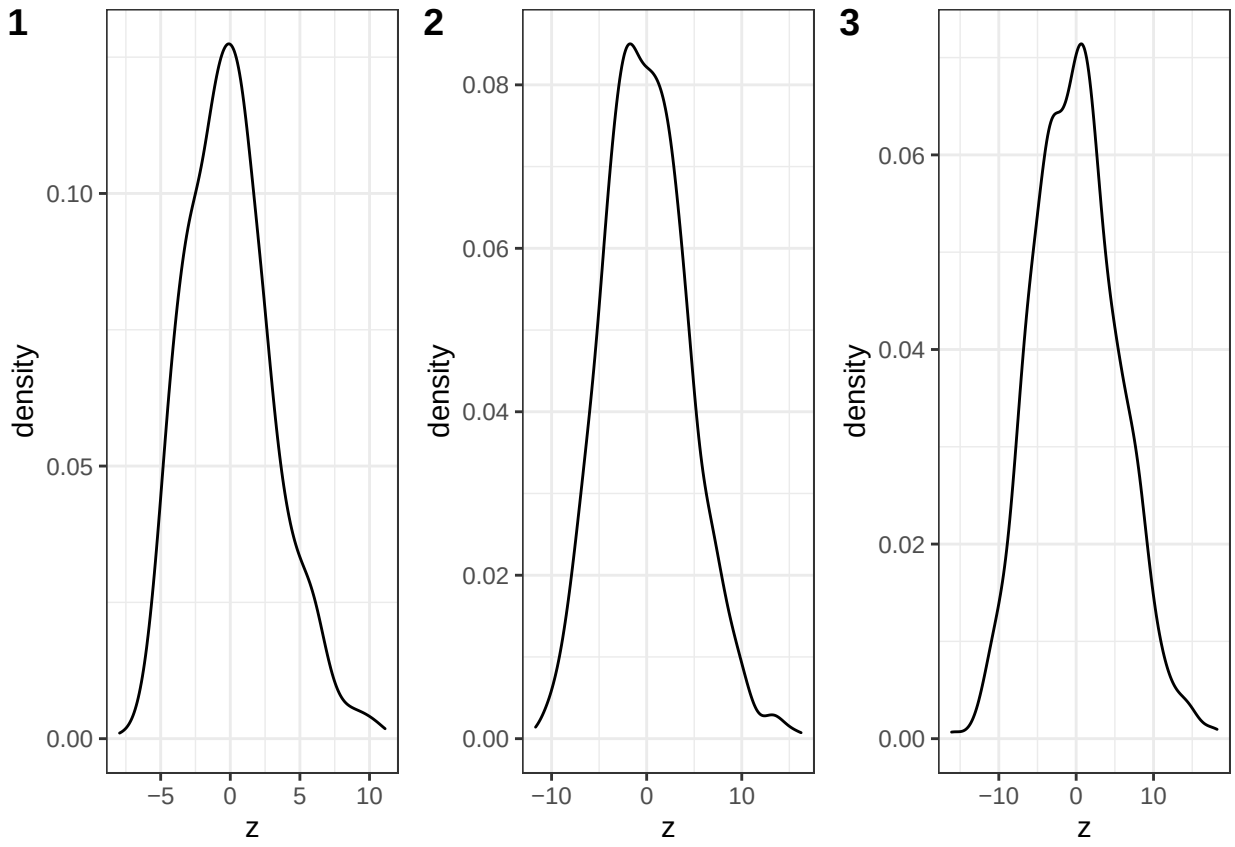
```
z3 = (x_bar3 - mean(x_bar3))/(sd(x_bar3)/sqrt(n))
df3 = data.frame(z = z3)
```

When the sample size n = 30, of 1000 samples, the mean of sample mean is 83.4955773, and the standard deviation of the sample mean is 11.8941425. We could see that, although the mean remains close to that of n = 10 and 20, the standard deviation becomes more lower than n = 10 and 20.

```
density3 = ggplot(df3, aes(x = z)) + geom_density() + theme_bw()
density3
```



```
ggarrange(density1, density2, density3,
          labels = c(1, 2, 3),
          ncol = 3, nrow = 1)
```



From the plot panel, we could see that, although they are all bell-shaped and symmetric distributed, the peak density became lower and lower as the sample size increases.

```
s1_3 = sum((x_bar3-mean(x_bar3))^2)/1000 * n; s1_3
```

```
## [1] 4239.875
```

```
s2_3 = sum(s3^2)/1000; s2_3
```

```
## [1] 4190.516
```

Here we could see that, $\bar{s_2^2}$ is still more close to the theoretical value, compared to $\bar{s_1^2}$.

Compare 3 scenarios

```
s1 = c(s1_1, s1_2, s1_3)
s2 = c(s2_1, s2_2, s2_3)
compare = data.frame(s1, s2)
theoretical = k*theta^2
theoretical
```

```
## [1] 4171.661
```

```
compare$'s1-true' = s1 - theoretical
compare$'s2-true' = s2 - theoretical
compare %>% knitr::kable()
```

s1	s2	s1-true	s2-true
3957.403	4138.226	-214.25808	-33.43443
4343.084	4232.641	171.42349	60.98032
4239.875	4190.516	68.21401	18.85529

We want to compare the $\bar{\bar{s}}_1^2$, $\bar{\bar{s}}_2^2$ and the theoretical value (true) $k\theta^2 = 4171.661$. From the result, we have several findings:

- As the sample size increases, $\bar{\bar{s}}_1^2$ and $\bar{\bar{s}}_2^2$ are getting more closer to the theoretical value.
- No matter what sample size we choose, $\bar{\bar{s}}_2^2$ is always closer to the theoretical value than $\bar{\bar{s}}_1^2$.

Therefore, we want to show that $\bar{\bar{s}}_1^2$ is a biased estimator of the population variance while $\bar{\bar{s}}_2^2$ is an unbiased estimator of the population variance.

$\bar{\bar{s}}_2^2$ is an unbiased estimator of the population variance