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 $\bar{\bar{s}}_1^2 = \frac{\sum (\bar{x_i} - \bar{\bar{x}})^2}{1000}$ 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 $k\theta^2$

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)
```

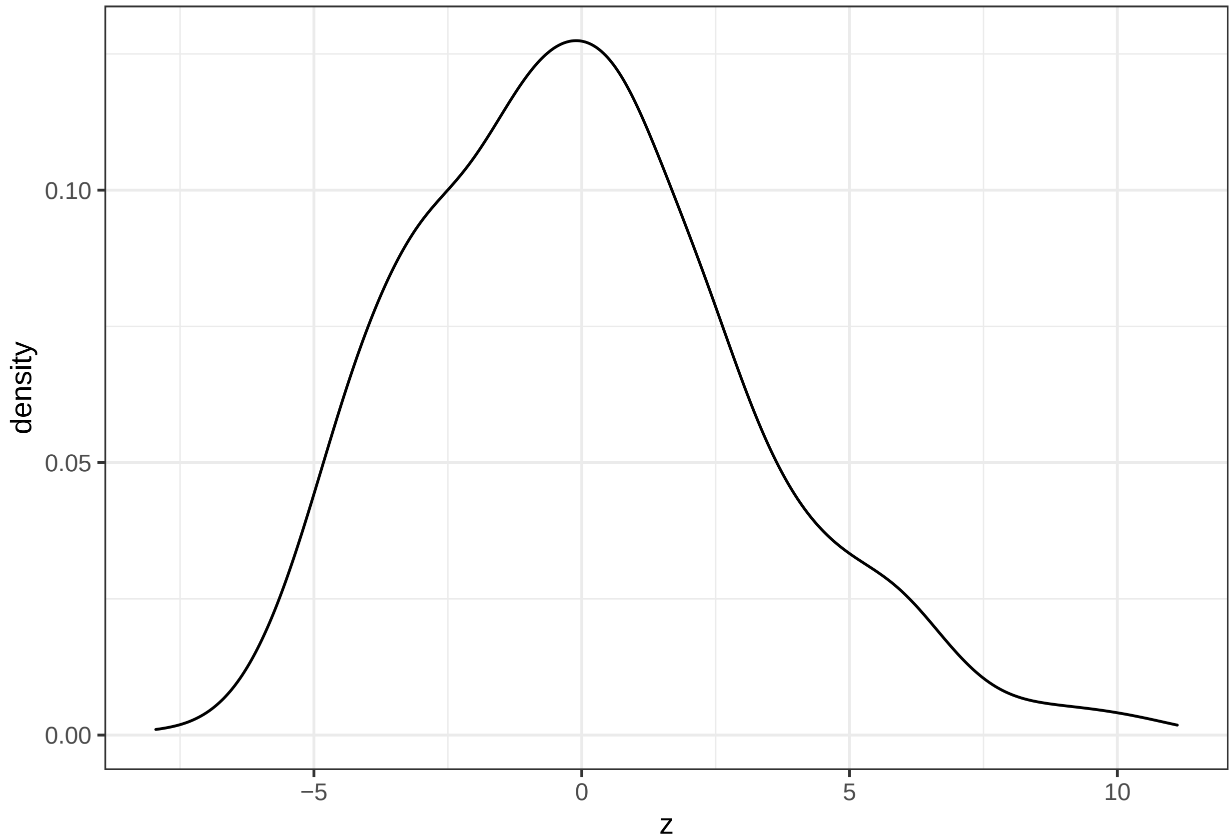
```
## [1] 82.93522
```

```
sd(x_bar)
```

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

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

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




```r
s1_1 = (sd(x_bar))^2 * n; s1_1
```

```
## [1] 3961.364
```

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

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

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

```r
  x_bar2[i] = mean(sample)
  s2[i] = sd(sample)
}
mean(x_bar2)
```

n = 20

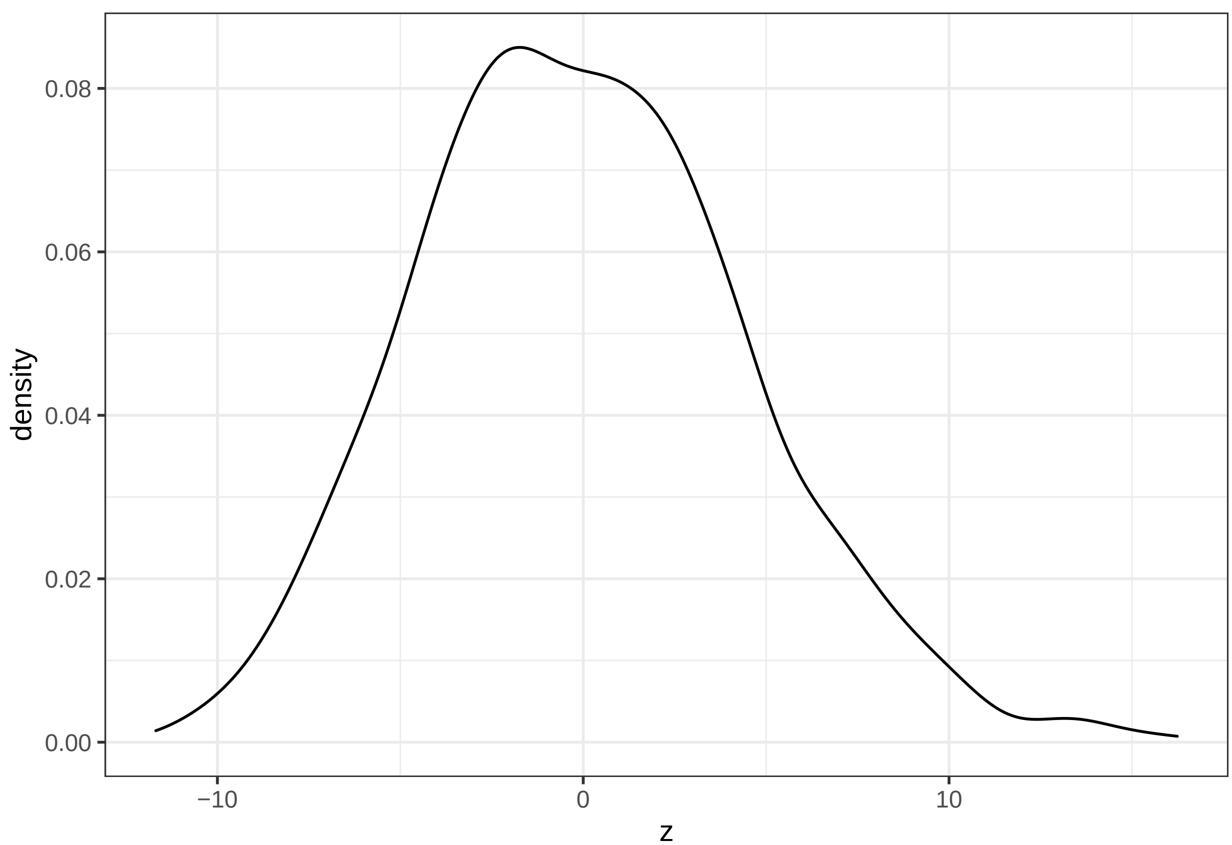
```
## [1] 83.25347
```

```r
sd(x_bar2)
```

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

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

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




```r
s1_2 = (sd(x_bar2))^2 * n; s1_2
```

```
## [1] 4347.432
```

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

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

n = 30

```r
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)
```

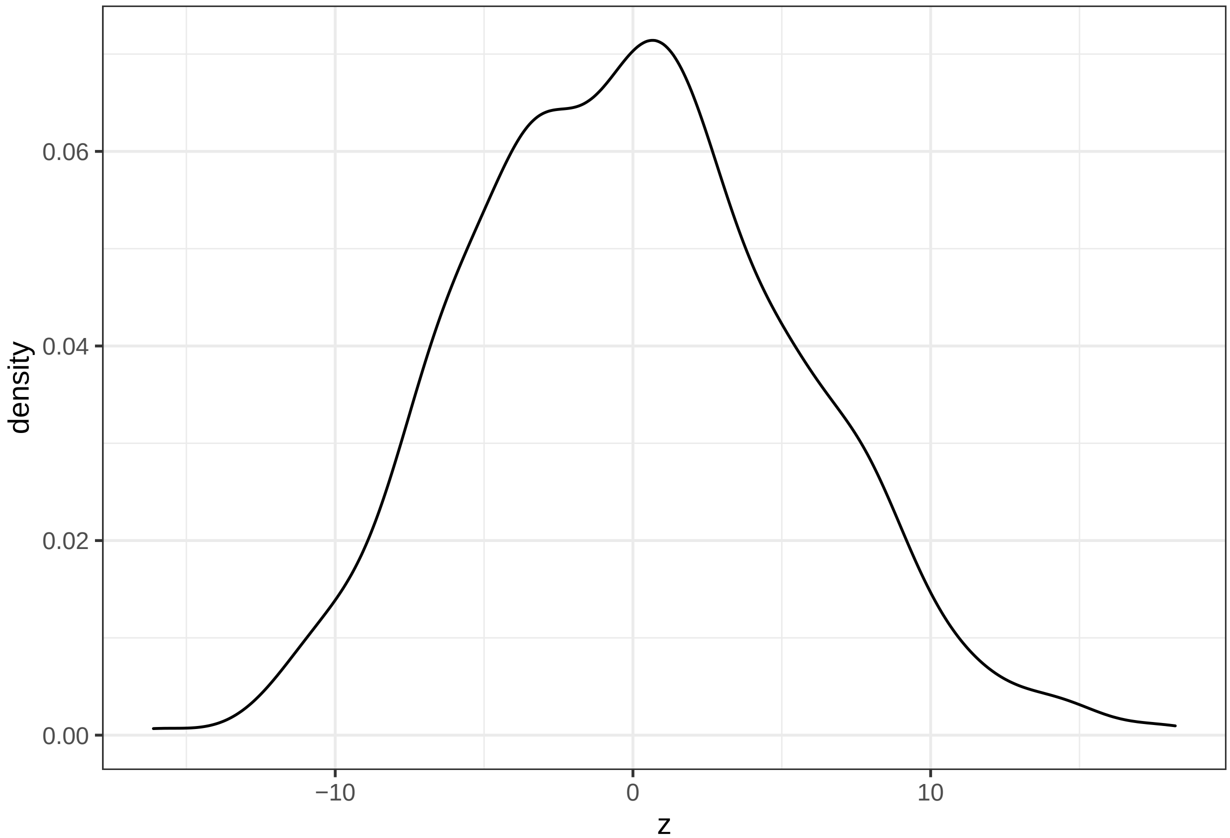
```
## [1] 83.49558
```

```r
sd(x_bar3)
```

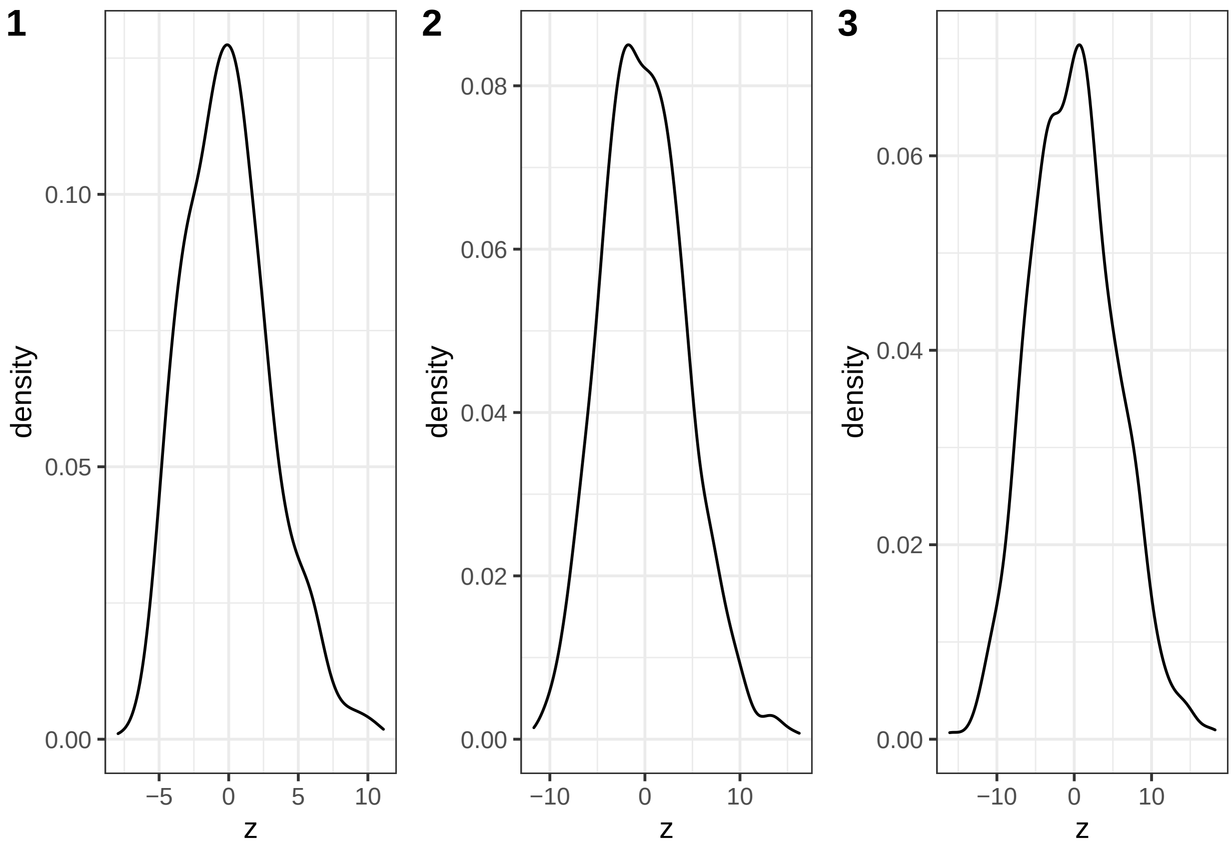
```
## [1] 11.89414
```

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

```r
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 = (sd(x_bar3))^2 * n; s1_3
```

```
## [1] 4244.119
```

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

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

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

Compare 3 scenarios

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

```r
compare$'s1-true' = s1 - true
compare$'s2-true' = s2 - true
compare
```

```
##         s1       s2    s1-true   s2-true
## 1 3961.364 4138.226 -210.29672 -33.43443
## 2 4347.432 4232.641  175.77092  60.98032
## 3 4244.119 4190.516   72.45813  18.85529
```