Probability Distributions of Y

and Simulated Sampling Distributions of $\bar{Y}$

2023-09-20

In the following examples, consider a discrete random variable Y with the probability distribution $p(y)$. As usual $E(Y) = \mu_Y$ and $VAR(Y) = \sigma^2_Y$. The three examples each display tables and graphs illustrating $p(y)$, and then display simulated sampling distributions of $\bar{Y}$ – the mean of an random sample of n independent observations of the random variable Y – at sample sizes of $n = 5, 25, and 1000$. The simulations were all constructed using the following process:

1. Specify some probability distribution $p(y)$.
2. Draw a sample of size n from the probability distribution.
3. Record the sample mean, $\bar{Y}$.
4. Repeat this process 5,000 times.[1]
5. Display a histogram of the 5,000 $\bar{Y}$'s with 10 bins.

The take-home point here: as N becomes large, the Central Limit Theorem tells us that the distribution of the sampling distribution of $\bar{Y}$ converges to the Normal with an ever-smaller variance. This is true, perhaps unsurprisingly, when the distribution of Y is itself nearly Normal (example 1). But it is also true for any and all possible distributions of Y, including those that are best described as "bimodal" (example 2) or skewed (example 3). Thus when n is large, no assumptions about the distribution of Y are necessary to fully describe the sampling distribution of $\bar{Y}$. Under this circumstance, $\bar{Y}$ is distributed Normal with mean μ_Y and variance $\frac{\sigma^2_Y}{n}$.

Example 1: The Random Variable Y takes on a nearly **Normal Distribution**

```
set.seed(123)
require(tidyverse)
Y <- rnorm(n = 10000,mean = 5,sd = sqrt(3))
```

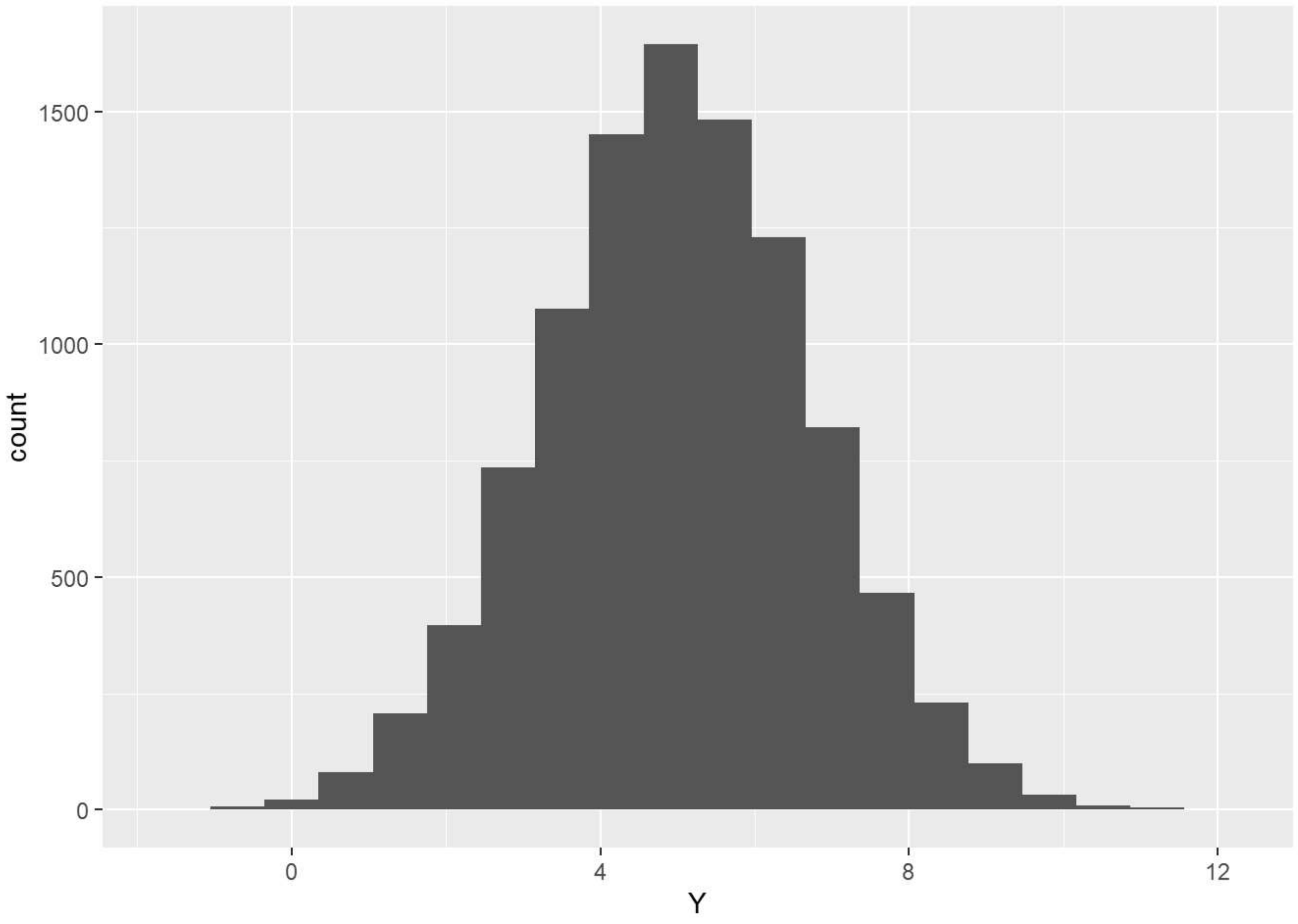
The probability distribution of Y (table)

```
data.frame(Y = round(Y)) %>%
  count(Y) %>%
  mutate(`p(Y=y)` = n / sum(n))
```

```
##     Y    n p(Y=y)
## 1  -2    1 0.0001
## 2  -1    6 0.0006
## 3   0   39 0.0039
## 4   1  169 0.0169
## 5   2  532 0.0532
## 6   3 1194 0.1194
## 7   4 1905 0.1905
## 8   5 2307 0.2307
## 9   6 1915 0.1915
## 10  7 1202 0.1202
## 11  8  521 0.0521
## 12  9  160 0.0160
## 13 10   37 0.0037
## 14 11   11 0.0011
## 15 12    1 0.0001
```

The probability distribution of Y (histogram)

```
data.frame(Y = Y) %>%
  ggplot(aes(x = Y)) +
  geom_histogram(bins = 20)
```

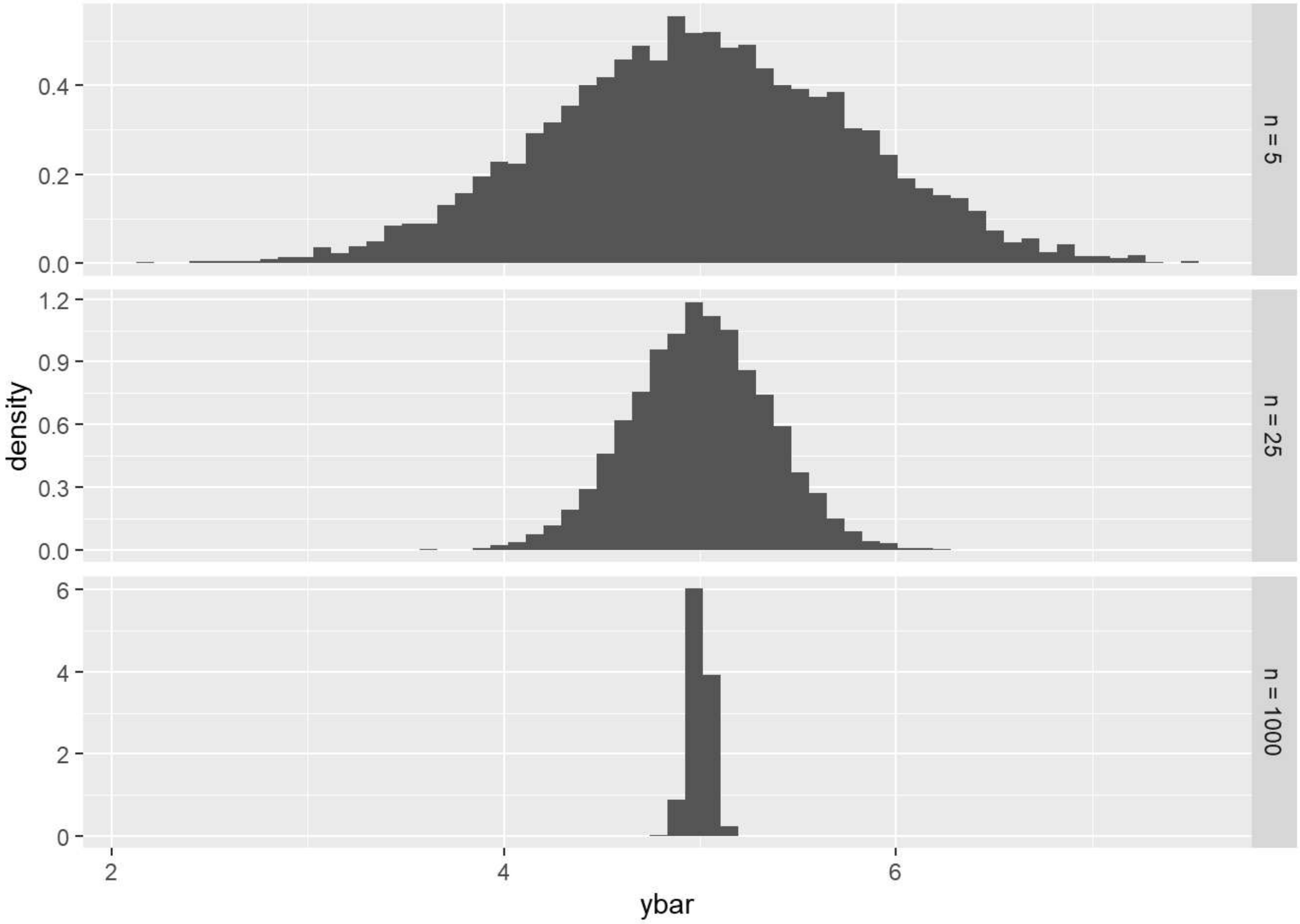


The sampling distribution of $\bar{Y}$ at different sizes

```
toplot <- NULL
for(s in c(5,25,1000)) {
  cat(s,'\n')
  for(i in 1:5000) {
    toplot <- toplot %>%
      bind_rows(data.frame(Y = Y) %>%
      sample_n(size = s) %>%
      summarise(ybar = mean(Y)) %>%
        mutate(size = s))
  }
}
```

```
## 5
## 25
## 1000
```

```
toplot %>%
  mutate(size = factor(paste0('n = ',size),levels = c('n = 5','n = 25','n = 1000'))) %>%
  ggplot(aes(x = ybar)) +
  geom_histogram(bins = 60,aes(y = ..density..)) +
  facet_grid(size~.,scales = 'free_y')
```



Note that we can calculate mean and standard deviation of these results as follows:

```
# Mean: E(Y-bar) = \mu = 5
toplot %>%
  group_by(size) %>%
  summarise(mu = mean(ybar))
```

```
## # A tibble: 3 × 2
##    size    mu
##   <dbl> <dbl>
## 1     5  5.00
## 2    25  5.00
## 3  1000  5.00
```

```
# Std Dev: \sigma_Y-bar = sqrt(var(y-bar)) = sqrt(\sigma^2 / n) approx \sqrt(3) / n
toplot %>%
  group_by(size) %>%
  summarise(sigma = sqrt(3/size)) %>%
  distinct()
```

```
## `summarise()` has grouped output by 'size'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 3 × 2
## # Groups:   size [3]
##    size  sigma
##   <dbl>  <dbl>
## 1     5 0.775
## 2    25 0.346
## 3  1000 0.0548
```

Example 2: The Random Variable Y takes on a nearly **Bimodal Distribution**

```
set.seed(123)
Y <- c(rnorm(n = 5000,mean = 2,sd = sqrt(2)),rnorm(n = 5000,mean = 7,sd = sqrt(2)))
```

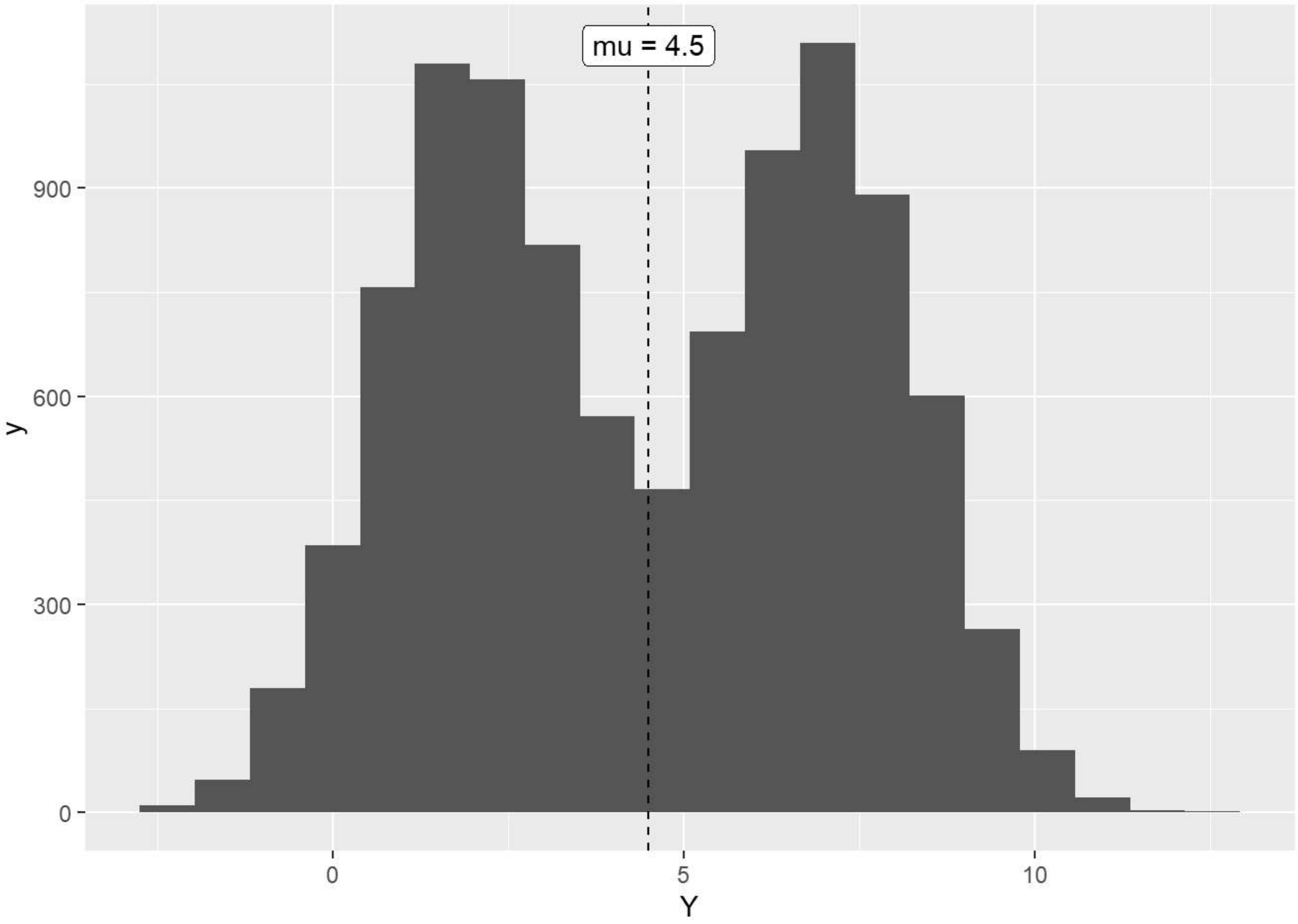
The probability distribution of Y (table)

```
data.frame(Y = round(Y)) %>%
  count(Y) %>%
  mutate(`p(Y=y)` = n / sum(n))
```

```
##     Y    n p(Y=y)
## 1  -2   30 0.0030
## 2  -1  173 0.0173
## 3   0  513 0.0513
## 4   1 1066 0.1066
## 5   2 1414 0.1414
## 6   3 1123 0.1123
## 7   4  686 0.0686
## 8   5  701 0.0701
## 9   6 1079 0.1079
## 10  7 1426 0.1426
## 11  8 1061 0.1061
## 12  9  526 0.0526
## 13 10  175 0.0175
## 14 11   23 0.0023
## 15 12    4 0.0004
```

The probability distribution of Y (histogram)

```
data.frame(Y = Y) %>%
  ggplot(aes(x = Y)) +
  geom_histogram(bins = 20) +
  geom_vline(xintercept = mean(Y),linetype = 'dashed') +
  annotate(geom = 'label',x = mean(Y),y = Inf,label = paste0('mu = ',round(mean(Y),2)),vjust =
1.5)
```

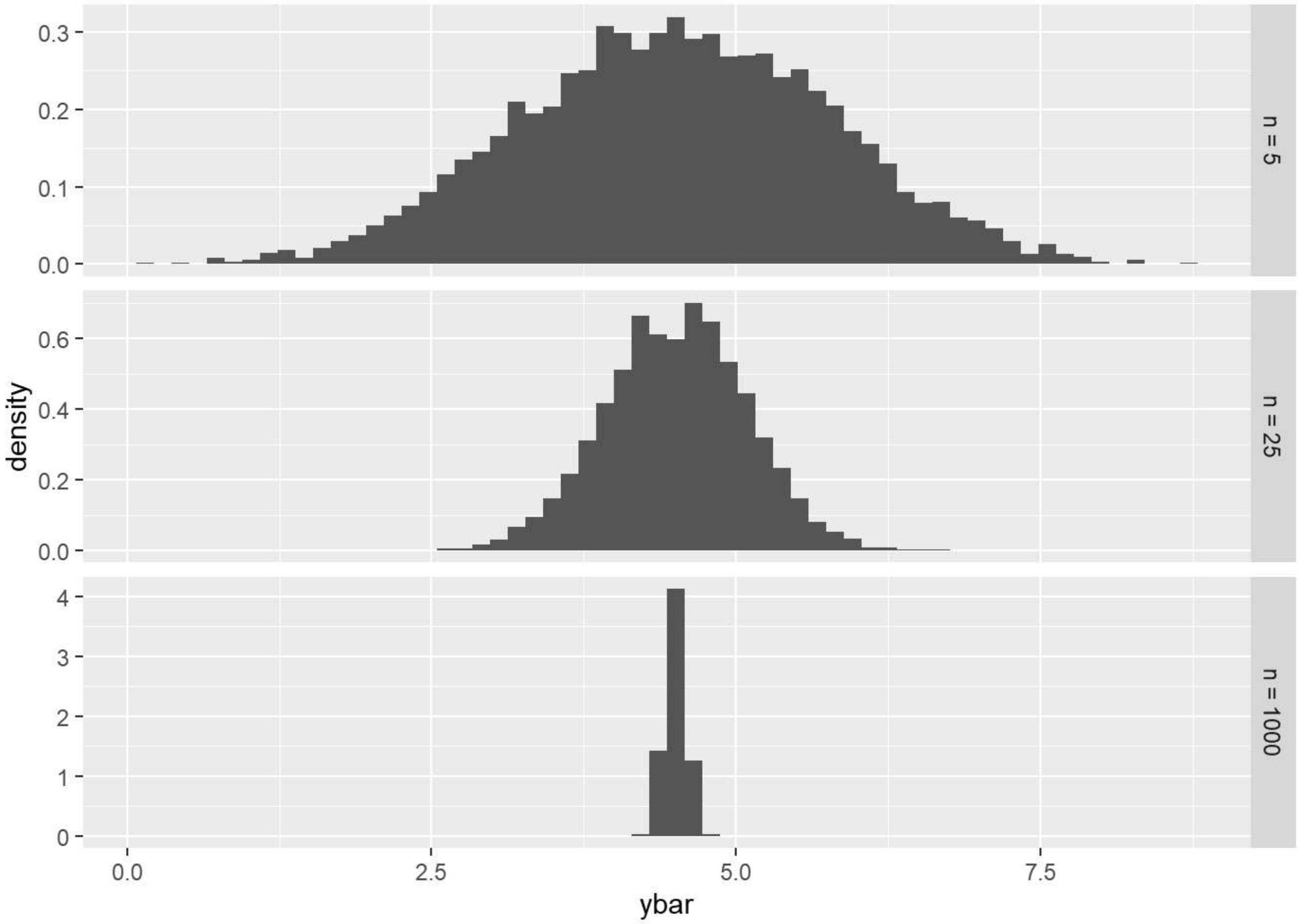


The sampling distribution of $\bar{Y}$ at different sizes

```
toplot <- NULL
for(s in c(5,25,1000)) {
  cat(s,'\n')
  for(i in 1:5000) {
    toplot <- toplot %>%
      bind_rows(data.frame(Y = Y) %>%
      sample_n(size = s) %>%
      summarise(ybar = mean(Y)) %>%
        mutate(size = s))
  }
}
```

```
## 5
## 25
## 1000
```

```
toplot %>%
  mutate(size = factor(paste0('n = ',size),levels = c('n = 5','n = 25','n = 1000'))) %>%
  ggplot(aes(x = ybar)) +
  geom_histogram(bins = 60,aes(y = ..density..)) +
  facet_grid(size~.,scales = 'free_y')
```



```
# Mean: E(Y-bar) = \mu = 5
toplot %>%
  group_by(size) %>%
  summarise(mu = mean(ybar))
```

```
## # A tibble: 3 × 2
##    size    mu
##   <dbl> <dbl>
## 1     5  4.50
## 2    25  4.50
## 3  1000  4.50
```

```
# Std Dev: \sigma_Y-bar = sqrt(var(y-bar)) = sqrt(\sigma^2 / n) approx \sqrt(3) / n
toplot %>%
  group_by(size) %>%
  summarise(sigma = sqrt(3/size)) %>%
  distinct()
```

```
## `summarise()` has grouped output by 'size'. You can override using the
## `.groups` argument.
## # A tibble: 3 × 2
## # Groups:   size [3]
##    size  sigma
##   <dbl>  <dbl>
## 1     5 0.775
## 2    25 0.346
## 3  1000 0.0548
```

Example 3: The Random Variable Y takes on a **Skewed Distribution**

```
set.seed(123)
Y <- rgamma(n = 10000,shape = 1,rate = 1)
```

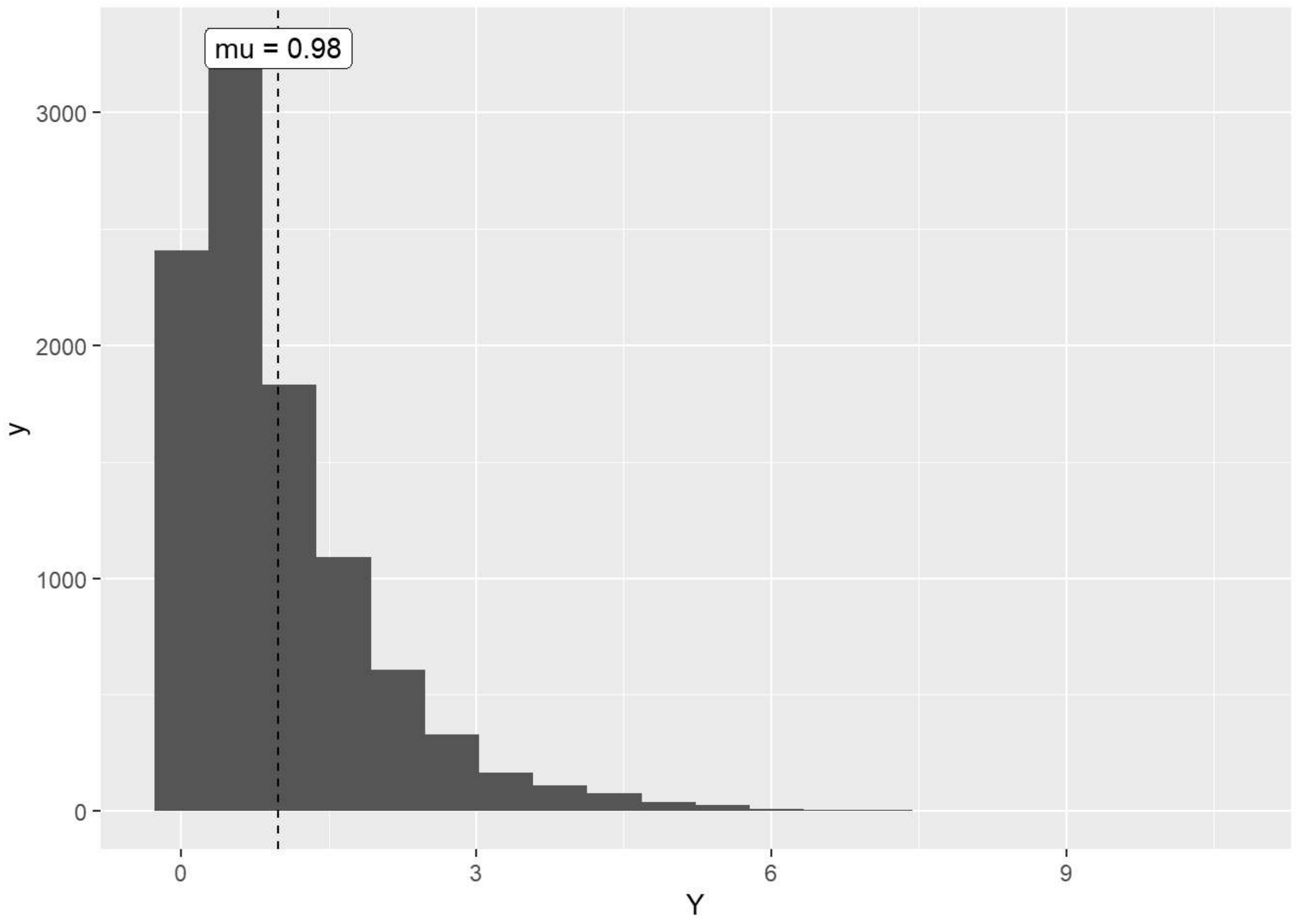
The probability distribution of Y (table)

```
data.frame(Y = round(Y)) %>%
  count(Y) %>%
  mutate(`p(Y=y)` = n / sum(n))
```

```
##     Y    n p(Y=y)
## 1   0 4000 0.4000
## 2   1 3860 0.3860
## 3   2 1381 0.1381
## 4   3  457 0.0457
## 5   4  191 0.0191
## 6   5   68 0.0068
## 7   6   27 0.0027
## 8   7    9 0.0009
## 9   8    4 0.0004
## 10  9    2 0.0002
## 11 10    1 0.0001
```

The probability distribution of Y (histogram)

```
data.frame(Y = Y) %>%
  ggplot(aes(x = Y)) +
  geom_histogram(bins = 20) +
  geom_vline(xintercept = mean(Y),linetype = 'dashed') +
  annotate(geom = 'label',x = mean(Y),y = Inf,label = paste0('mu = ',round(mean(Y),2)),vjust =
1.5)
```

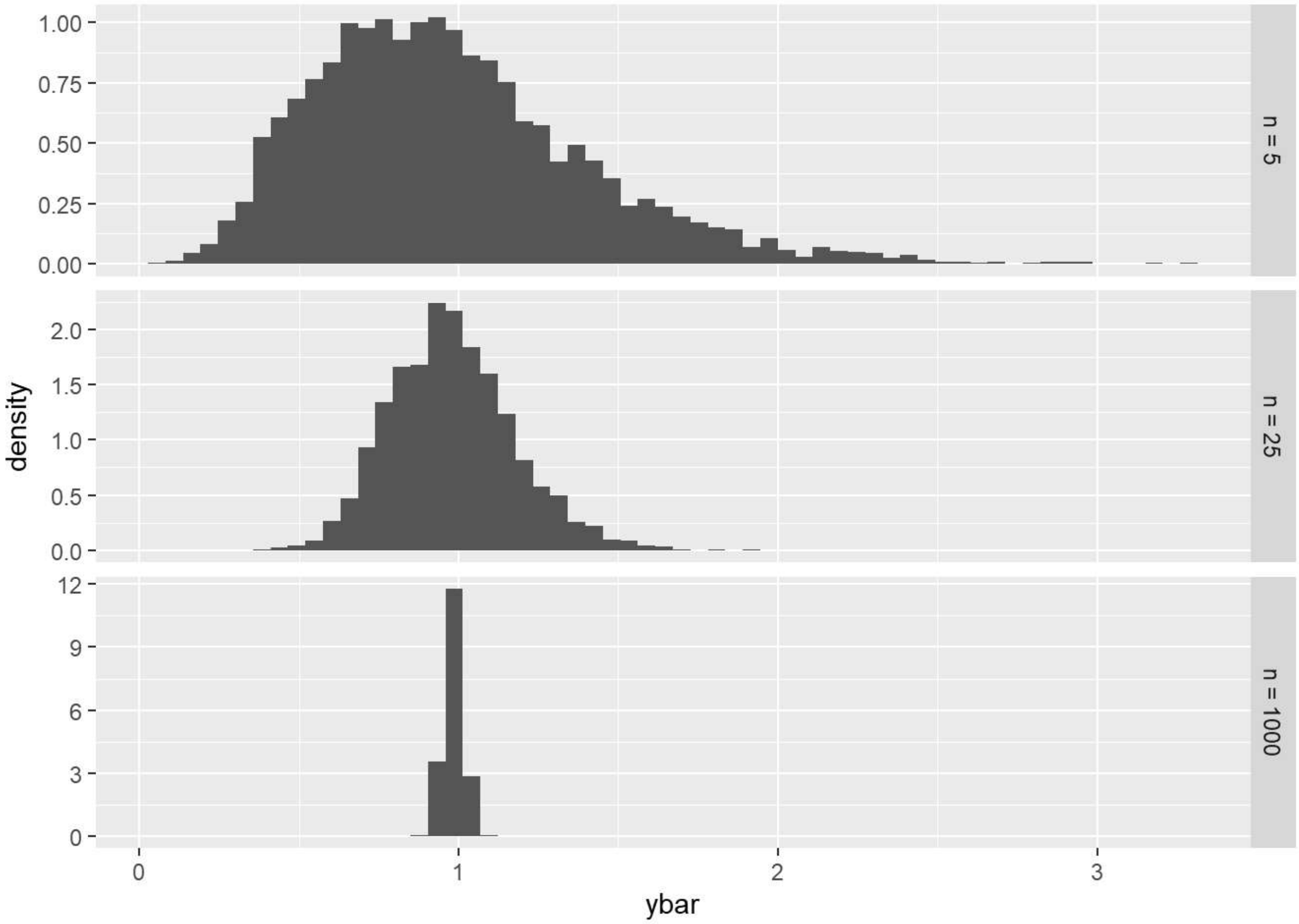


The sampling distribution of $\bar{Y}$ at different sizes

```
toplot <- NULL
for(s in c(5,25,1000)) {
  cat(s,'\n')
  for(i in 1:5000) {
    toplot <- toplot %>%
      bind_rows(data.frame(Y = Y) %>%
      sample_n(size = s) %>%
      summarise(ybar = mean(Y)) %>%
        mutate(size = s))
  }
}
```

```
## 5
## 25
## 1000
```

```
toplot %>%
  mutate(size = factor(paste0('n = ',size),levels = c('n = 5','n = 25','n = 1000'))) %>%
  ggplot(aes(x = ybar)) +
  geom_histogram(bins = 60,aes(y = ..density..)) +
  facet_grid(size~.,scales = 'free_y')
```



```
# Mean: E(Y-bar) = \mu = 5
toplot %>%
  group_by(size) %>%
  summarise(mu = mean(ybar))
```

```
## # A tibble: 3 × 2
##    size    mu
##   <dbl> <dbl>
## 1     5 0.974
## 2    25 0.980
## 3  1000 0.984
```

```
# Std Dev: \sigma_Y-bar = sqrt(var(y-bar)) = sqrt(\sigma^2 / n) approx \sqrt(3) / n
toplot %>%
  group_by(size) %>%
  summarise(sigma = sqrt(3/size)) %>%
  distinct()
```

```
## `summarise()` has grouped output by 'size'. You can override using the
## `.groups` argument.
## # A tibble: 3 × 2
## # Groups:   size [3]
##    size  sigma
##   <dbl>  <dbl>
## 1     5 0.775
## 2    25 0.346
## 3  1000 0.0548
```

1. Note that I chose 5,000 as a large number that could nevertheless be done in a short amount of time on a standard computer. But this number doesn’t matter. I could have picked 10,000 or 10 million such iterations: at higher numbers of iterations, the histograms would be smoother but would otherwise remain similar.↩