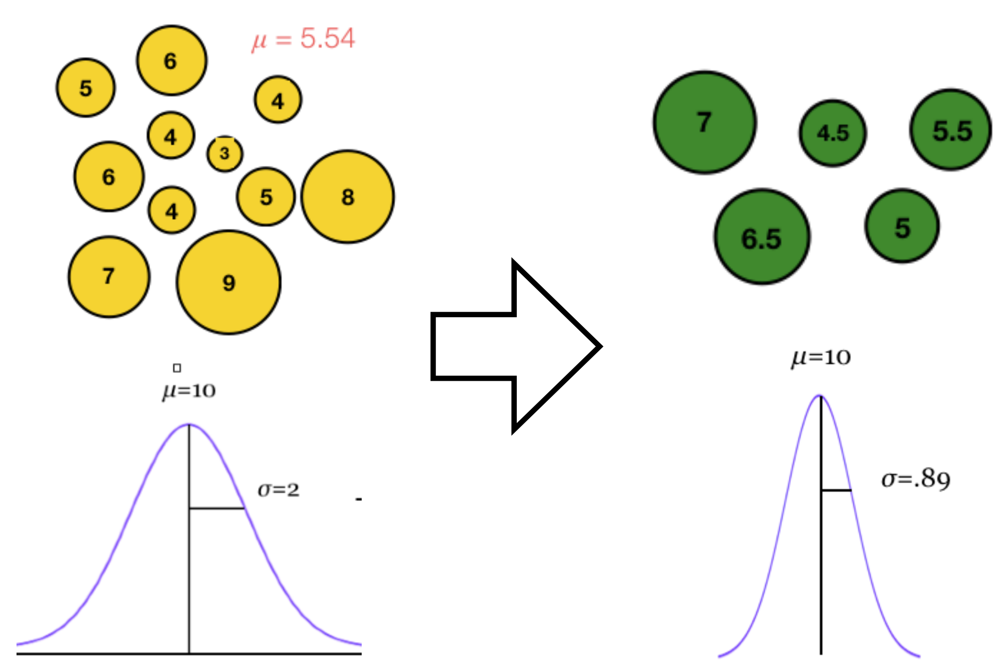
Statistical theory: Central limit theorem

Research Methods for Human Inquiry
Andrew Perfors

Remember last time…

thing	"usual" symbol	thing	"usual" symbol	what is it?	do we know its value?
true population mean	μ	**true population sd**	σ	the truth	no
estimated population mean	$\hat{\mu}$	**estimated population sd**	$\hat{\sigma}$	a statistical inference	yes
sample mean	$\bar{X}$ or M	**sample sd**	s	a description of our dataset	yes

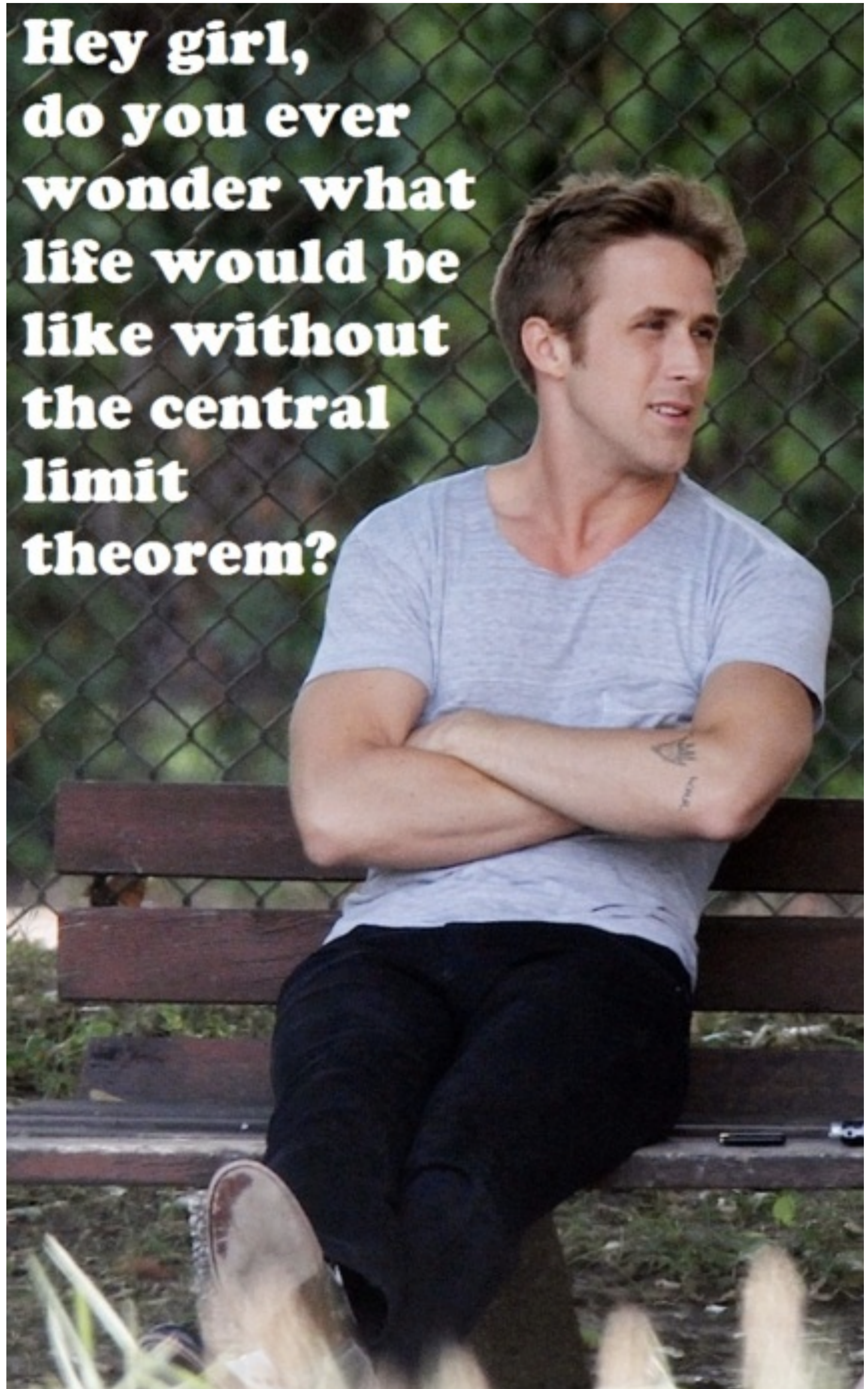
Sampling distribution of the mean is a theoretical idea that captures what you would expect the means of lots of samples from a population to look like

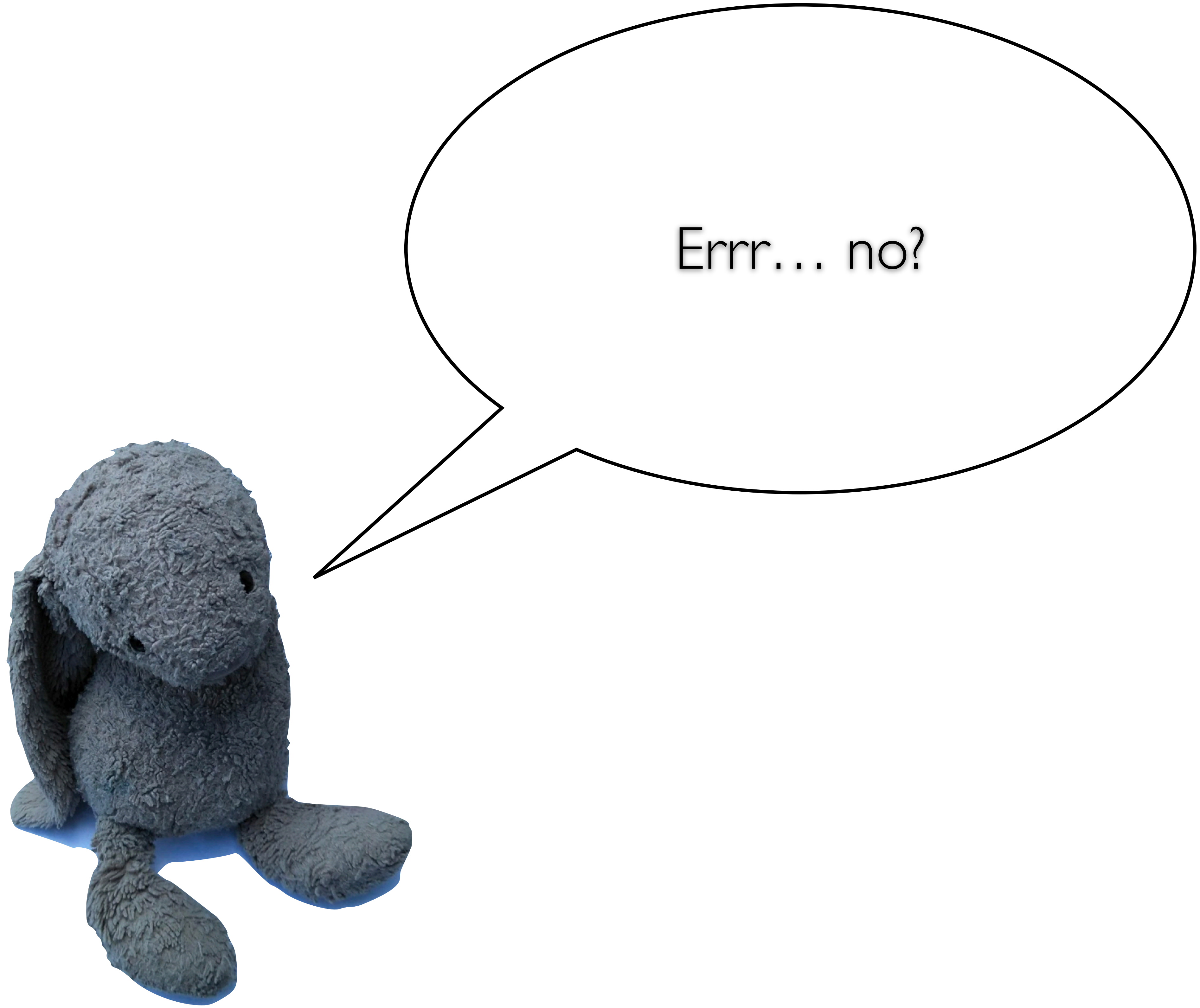


It is less variable than the original distribution

The sampling distribution of the mean is super cool.

Why? Because of the central limit theorem.

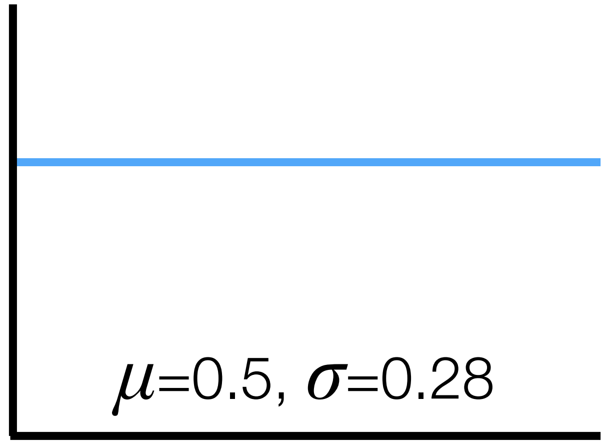

Hey girl,
do you ever
wonder what
life would be
like without
the central
limit
theorem?


Errr… no?

The central limit theorem

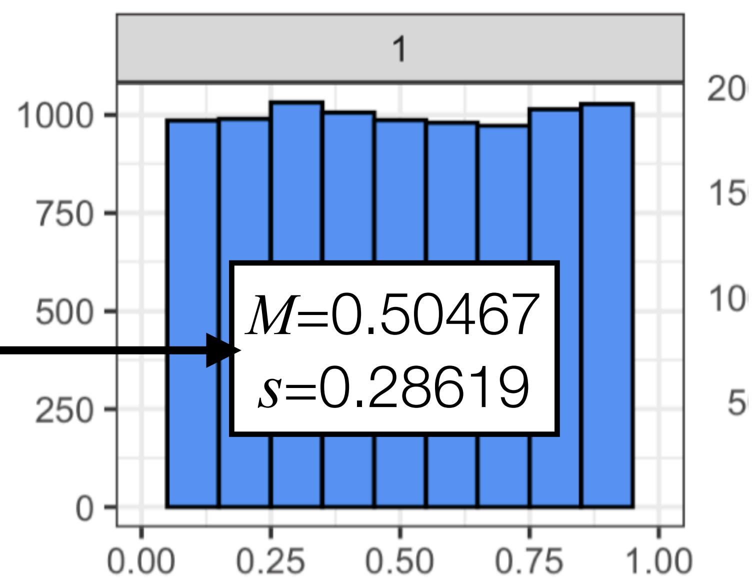
- One of the most important results in statistics
- What does it say?
  - The sampling distribution of the mean becomes normal
  - As long as you're averaging lots of independent things
  - Weirdly, it doesn't matter what the distribution looks like
  - Don't believe me? Here, I'll show you…

Original distribution: uniform (flat)



sampling distributions of the mean for samples of different sizes

0.28/
sqrt(1)



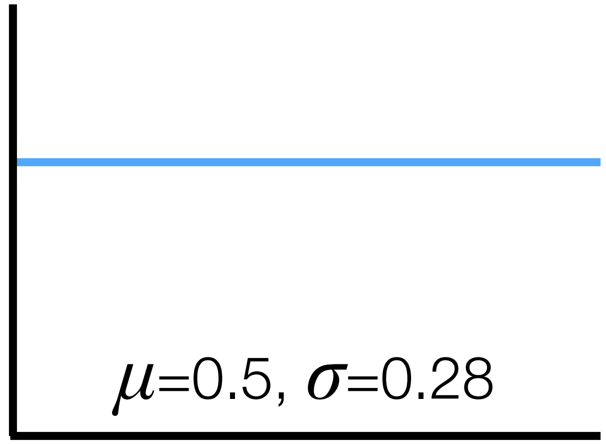
Let's demonstrate it with an underlying uniform distribution.

```
```{r centrallimitunif, echo=FALSE, warning=FALSE}
N <- 10000
one <- 1:N
two <- 1:N
five <- 1:N
ten <- 1:N
fifty <- 1:N
hundred <- 1:N

for (i in 1:N) {
  one[i] <- mean(runif(n=1,min=0,max=1))
  two[i] <- mean(runif(n=2,min=0,max=1))
  five[i] <- mean(runif(n=5,min=0,max=1))
  ten[i] <- mean(runif(n=10,min=0,max=1))
  fifty[i] <- mean(runif(n=50,min=0,max=1))
  hundred[i] <- mean(runif(n=100,min=0,max=1))
}
size <- c(rep(1,N),rep(2,N),rep(5,N),rep(10,N),rep(50,N),rep(100,N))
sample <- c(one,two,five,ten,fifty,hundred)
```

NOTE: I am providing this to you just in case you want to try playing with it on your own. It has several elements I have not introduced and which are not going to be assessed in any way.
```

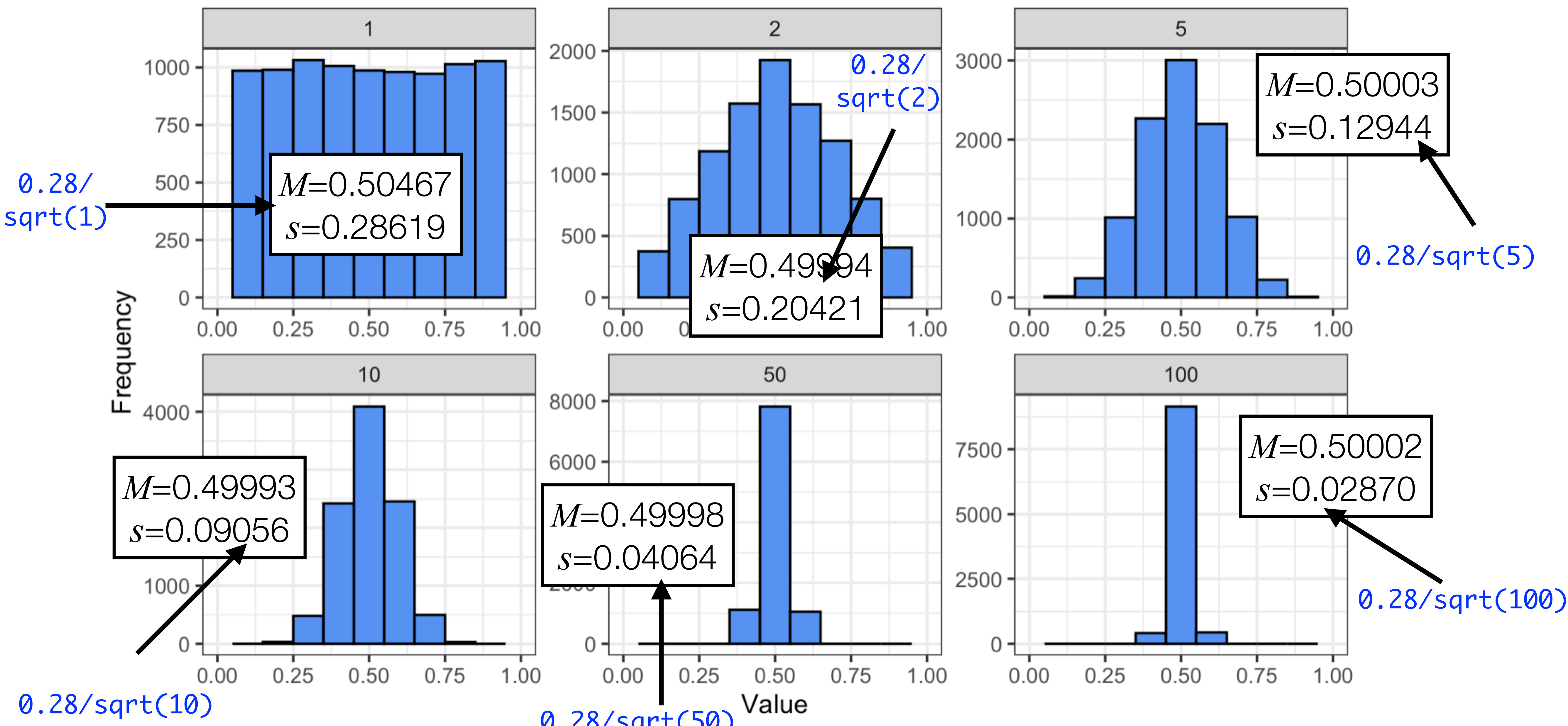
Original distribution: uniform (flat)



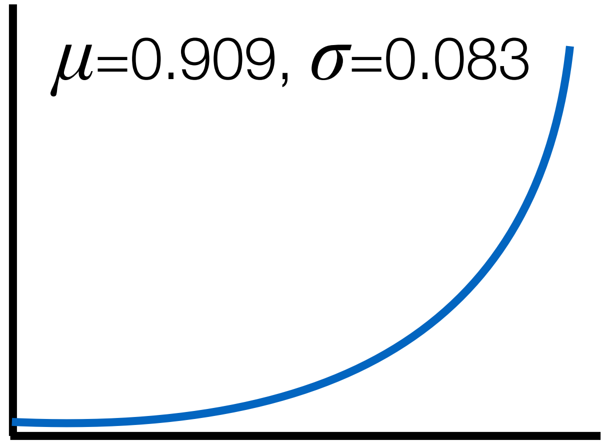
note that the means of the sampling distributions converge on the true population mean μ

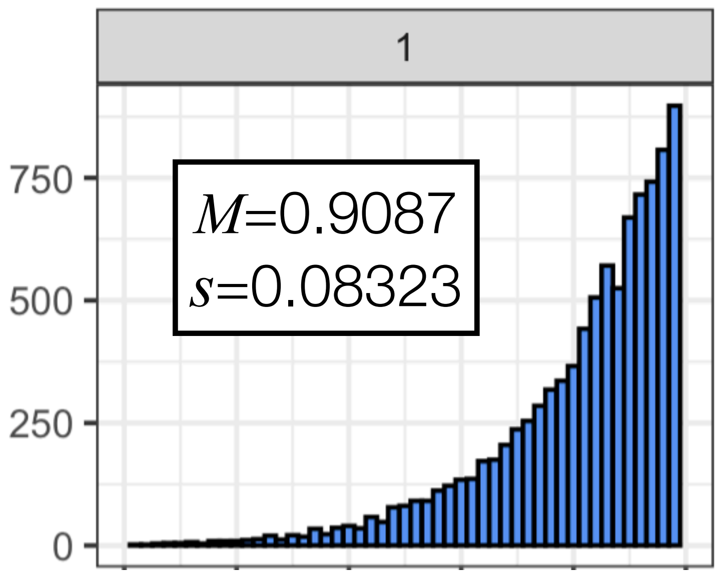
but the standard deviations get smaller: lower variance with larger samples!
these s=SEM of the original ($\frac{\sigma}{\sqrt{N}}$)

sampling distributions of the mean for samples of different sizes



Original distribution: skewed






```r
```{r centrallimitskewed, echo=FALSE, warning=FALSE}
N <- 10000
one <- 1:N
two <- 1:N
five <- 1:N
ten <- 1:N
fifty <- 1:N
hundred <- 1:N

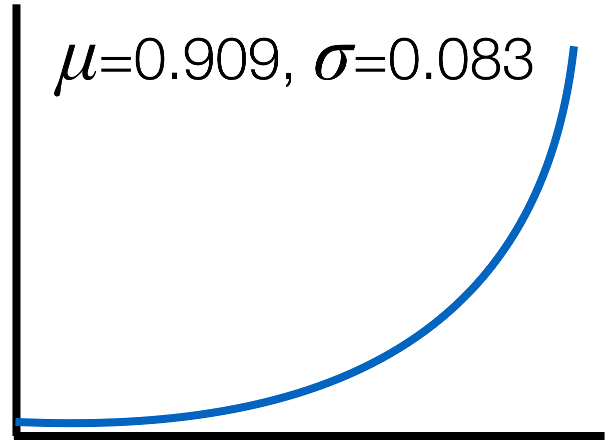
for (i in 1:N) {
  one[i] <- mean(rbeta(n=1,shape1=10,shape2=1))
  two[i] <- mean(rbeta(n=2,shape1=10,shape2=1))
  five[i] <- mean(rbeta(n=5,shape1=10,shape2=1))
  ten[i] <- mean(rbeta(n=10,shape1=10,shape2=1))
  fifty[i] <- mean(rbeta(n=50,shape1=10,shape2=1))
  hundred[i] <- mean(rbeta(n=100,shape1=10,shape2=1))
}
size <- c(rep(1,N),rep(2,N),rep(5,N),rep(10,N),rep(50,N),rep(100,N))
sample <- c(one,two,five,ten,fifty,hundred)
d <- tibble(size,sample)

d %>%
  ggplot(mapping = aes(x=sample)) +
  geom_histogram(binwidth=0.01,fill="cornflowerblue",colour="black") +
```

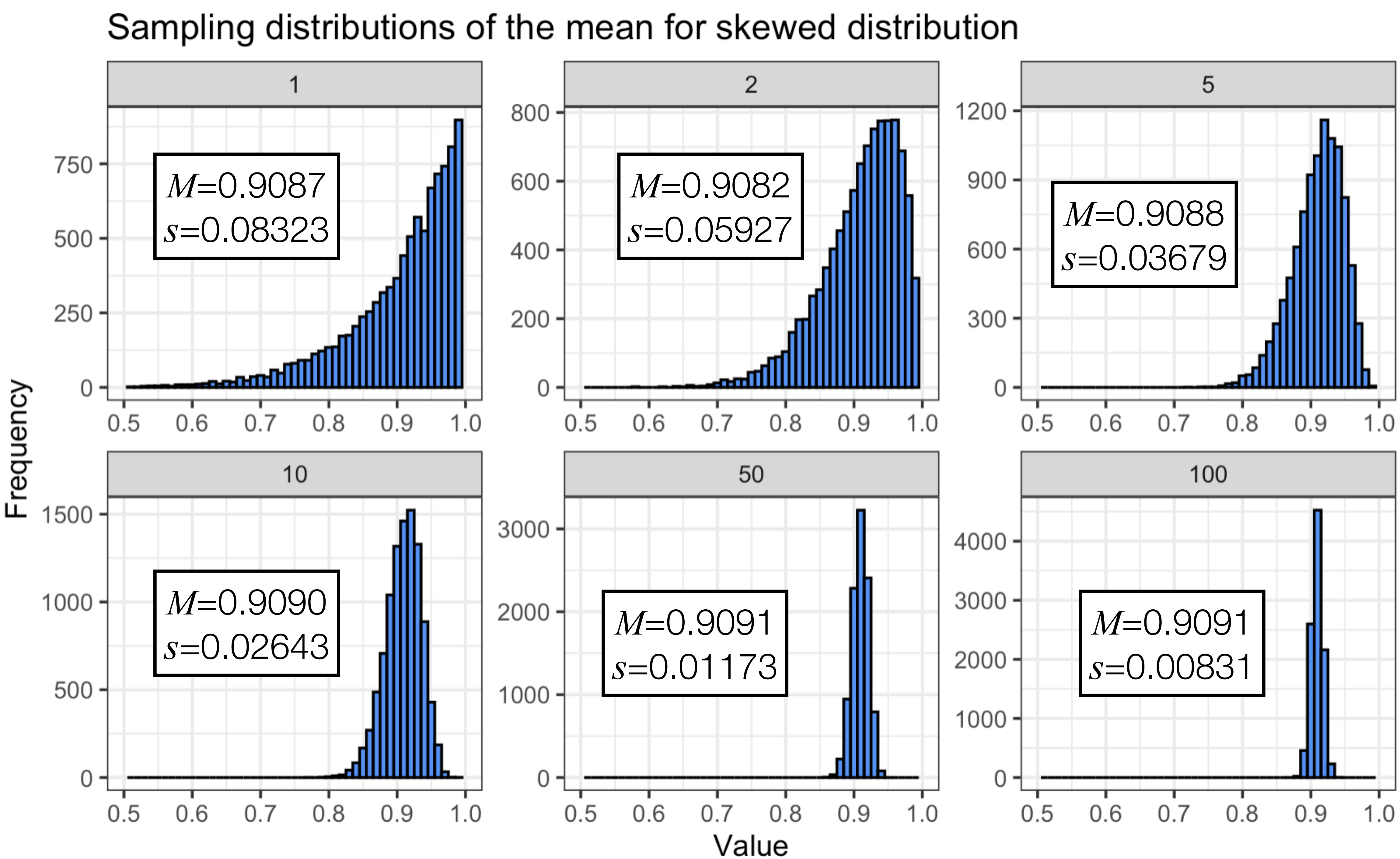
Chunk 8: centrallimitskewed

R Mark
```

Original
distribution:
skewed

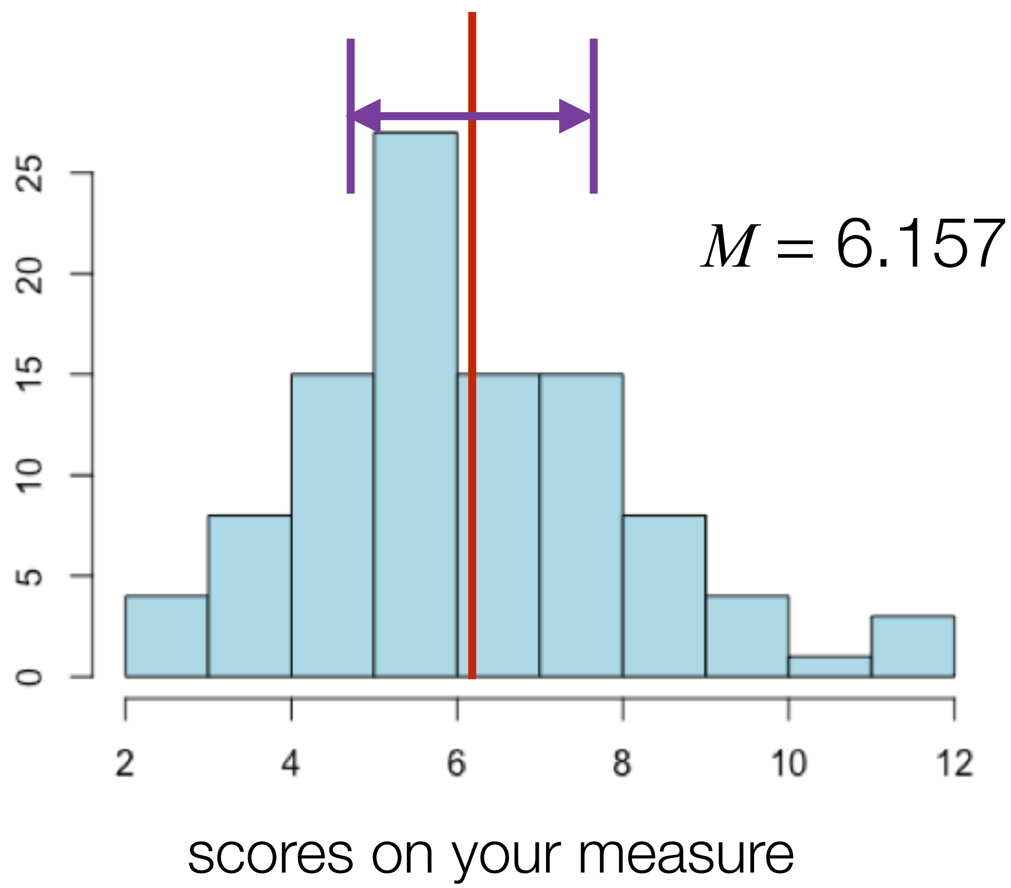


As before, the means converge and the standard deviations get smaller



What does all of this buy us?

Suppose instead of running 10,000 experiments with 100 people each, you run only one (much more likely!)



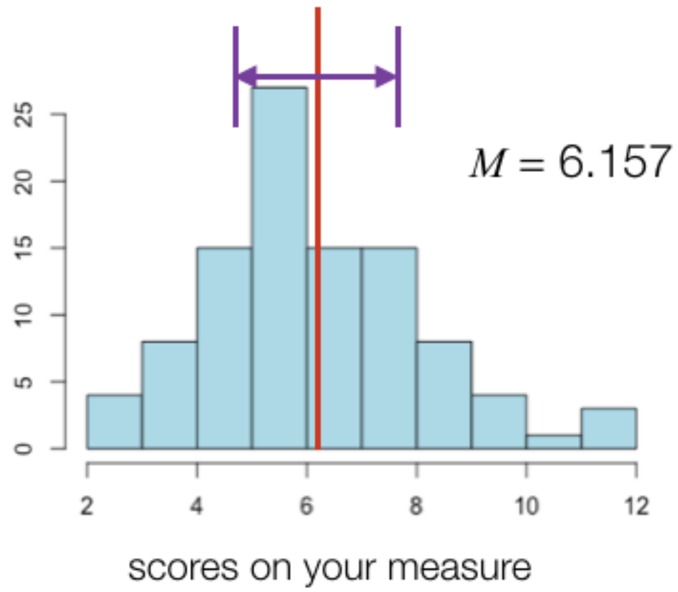
What you really want to know is, how much can you trust this mean?

If you ran this experiment again, would you get a similar mean?

If you ran this experiment 100 times, what range of means would you get?

Ideally we want a **confidence interval** around the mean: a range that we're confident covers the mean

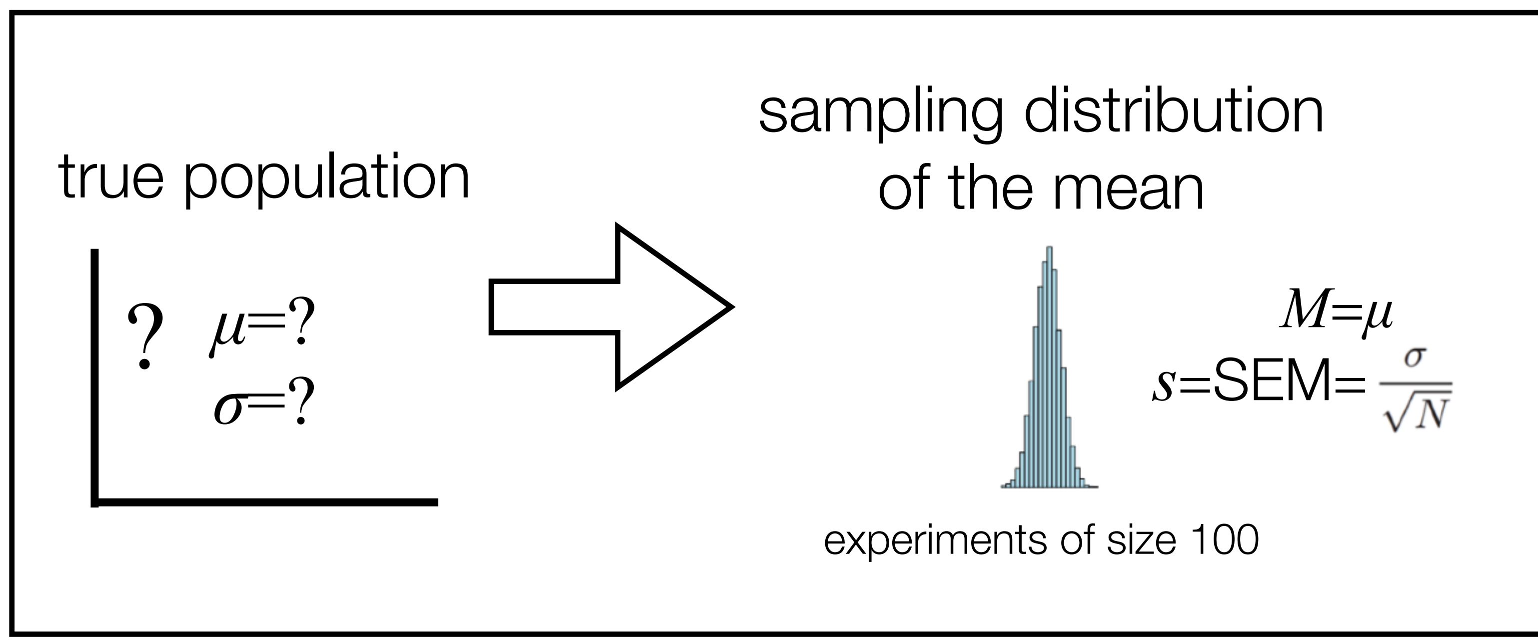
Deriving a confidence interval



confidence interval around the mean: a range that we're confident covers the mean

of course, we have to say with what probability it covers the mean. **95%** is traditional.

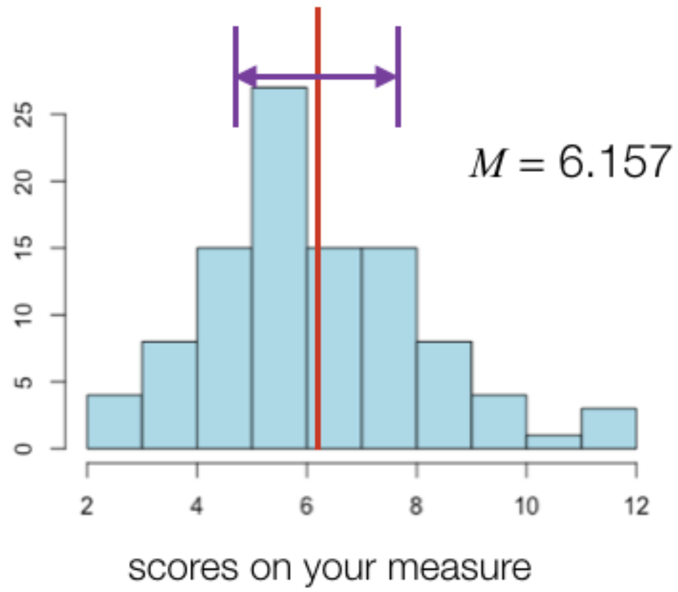
we already know that the sampling distribution of the mean converges on the mean, and the standard deviation of that sample is just the SEM



remember that 2 standard deviations = 95% of the probability
(technically it's 1.96)

thus we know that the range bounded by ± 1.96 SEMs gives the **95% confidence interval**

Confidence intervals in a nutshell



the 95% confidence interval (CI) is the range that covers the mean 95% of the time

(if you did 100 experiments, the mean would be within that range 95 times)

95% CI = mean ± 1.96*SEM

$$\mathrm{CI}_{95} = \bar{X} \pm 1.96 \frac{\hat{\sigma}}{\sqrt{N}}$$

Confidence intervals in R

load the dataset

```
> loc <- here("stolendata.csv")
> stolendata <- read_csv(file=loc)
> head(stolendata)
# A tibble: 6 × 14
   year location population water chickens  eggs  cows  pigs wheat  corn carrots lettuce
  <dbl> <chr>         <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>
1  2022 OZMXM             6   541        3    12     1     2     2    18      53       8
2  2022 MGSVB             4   450        3     8     1     2     3    50      49       4
3  2022 ORLBA             5   495        3    17     2     2     0    24      34      19
4  2022 5LO29             5   325        1     7     1     1     1    15      17      10
5  2022 KJYDE             7   567        4    17     2     3     2    47      59      33
6  2022 QF90Y             7   866        3    16     1     2     2    29      33      28
# … with 2 more variables: potatoes <dbl>, strawberries <dbl>
# ℹ Use `colnames()` to see all variable names
```

Confidence intervals in R

Use `ciMean()` in the `lsr` package for all variables in the data frame

```
> library(lsr)
> ciMean(stolendata)
                    2.5%       97.5%
year         2016.366085 2017.633915
location*             NA          NA
population      5.493277    6.183491
water         533.126832  625.943875
chickens        5.142833    6.049086
eggs           29.884897   36.862578
cows            2.669785    3.229205
pigs            3.593200    4.305790
wheat           3.956883    5.012814
corn           50.826709   62.930867
carrots        58.295331   71.482446
lettuce        27.983438   36.562016
potatoes       43.195241   52.097688
strawberries   67.022809   82.997393
```

Can do for single variables

```
> ciMean(stolendata$water)
         2.5%    97.5%
[1,] 533.1268 625.9439
```

it calculates other CIs (here is 80%)

```
> ciMean(stolendata$water,conf=0.8)
          10%     90%
[1,] 549.3617 609.709
```

See the `w5day2exercises.Rmd` file for the exercises!