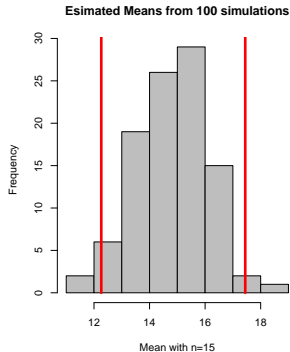
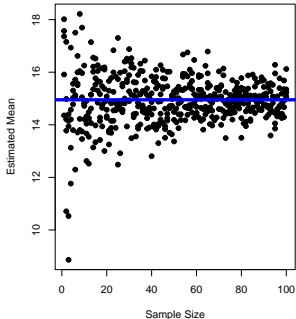


# Sample Properties & Simulation



## But first, a gratuitous advertisement



<http://scifundchallenge.org>

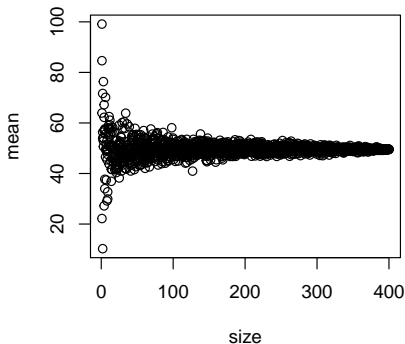
What is #SciFund?

- ▶ Crowdfunding your research (avg project \$1500)
- ▶ An opportunity to try your hand at *outreach*
- ▶ Training in video and communication
- ▶ Signup by Oct. 8th

## Loops: Simulation to Estimate Precision

Last time...

How does sample size influence precision of our estimate of the mean?



# The anatomy of a simulation

1) Create a vector of sample sizes you want to iterate over

```
n <- rep(1:400, times = 4)
```

# The anatomy of a simulation

2) Create a blank vector of means

```
m <- rep(NA, times = length(n))
```

*length* gets length of a vector

# The anatomy of a simulation

## 3) The For Loop

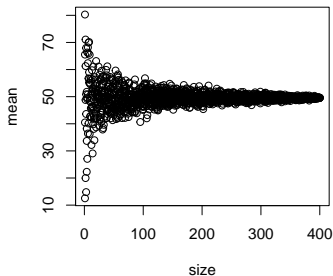
```
for (i in 1:length(n)) {  
  m[i] <- mean(sample(population, size = n[i]))  
}
```

- ▶  $i$  is an index to iterate over
- ▶ the values of  $i$  are from the vector  $1:length(n)$

# The anatomy of a simulation

## 4) Plot it

```
plot(n, m, xlab = "size", ylab = "mean")
```



Precision plateaus around 50.

## Exercise

- ▶ Write a for loop that calculates the first 15 numbers of the fibonacci sequence

*1, 1, 2, 3, 5, 7, 9...*

(Challenge: do it with a starting vector of only NA's )

(hint - create a blank vector, but with the first two entries as 1)

(hint - `aVec[i+1]` is `aVec[2]` if `i=1`)



## Exercise

```
# start with a blank vector with some 1's
fibVec <- c(1, 1, rep(NA, 13))

# now loop
for (i in 3:15) {
  fibVec[i] <- fibVec[i - 1] + fibVec[i - 2]
}

fibVec

## [1] 1 1 2 3 5 8 13 21 34 55 89 144 233 377
## [15] 610
```

## Sample Properties: Variance

How variable was that population?

$$s^2 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n-1}$$

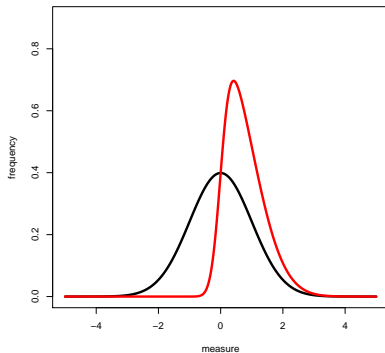
- ▶ **Sums of Squares** over  $n-1$
- ▶  $n-1$  corrects for both sample size and sample bias
- ▶  $\sigma^2$  if describing the population
- ▶ Units in square of measurement...

## Sample Properties: Standard Deviation

$$s = \sqrt{s^2}$$

- ▶ Units the same as the measurement
- ▶ If distribution is normal, 67% of data within 1 SD
- ▶ 95% within 2 SD
- ▶  $\sigma$  if describing the population

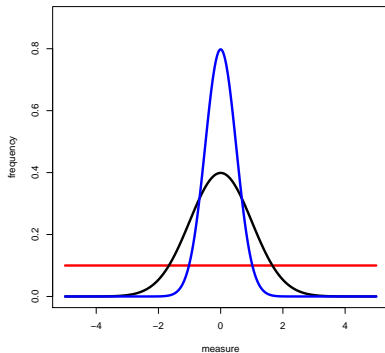
## Sample Properties: Skew



### Right-Skewed

Skew calculated using additional moments (think sums of squares, but cubed)

# Sample Properties: Kurtosis

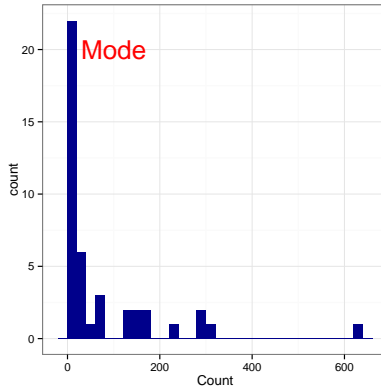


Platykurtic

Leptokurtic

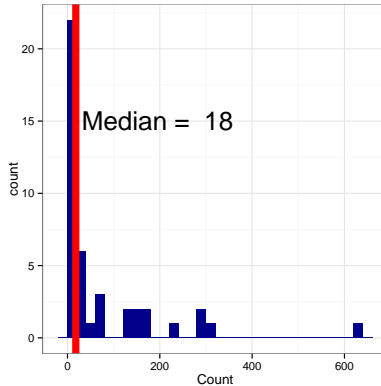
Normal

## Sample Properties: Mode



This highest point on a frequency plot.

## Sample Properties: Median



This middle value of a dataset.

## Sample Properties: Median

We obtain the median by sorting and picking the middle value.

```
sort(bird$Count)
```

```
## [1] 1 1 1 1 1 2 2 2 2 3 3 4 5 7  
## [15] 7 10 12 13 14 15 16 18 23 23 25 28 33 33  
## [29] 59 64 67 77 128 135 148 152 173 173 230 282 297 300  
## [43] 625
```

```
nrow(bird) #this is the # of rows in the data frame
```

```
## [1] 43
```

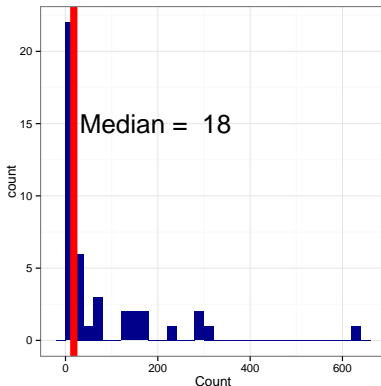
```
sort(bird$Count)[22]
```

```
## [1] 18
```



## Sample Properties: Median

The midpoint of the data-set is the 50th percentile!



# Percentiles, Quantiles, Quartiles, and all that

1. Sort a data set
2. The index of the *i*th value minus 0.5 divided by *n* is its quantile
3. Quantile \* 100 is the percentile
4. Quartiles are those points that divide data into 4 equal chunks (25th, 50th, and 75th percentile)

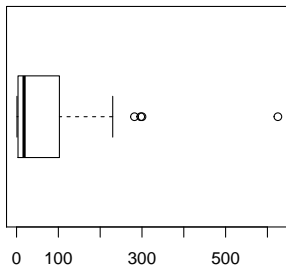
# Percentiles, Quantiles, Quartiles, and all that

```
sort(bird$Count)
```

```
##  [1]  1  1  1  1  1  2  2  2  2  3  3  4  5  7
## [15]  7 10 12 13 14 15 16 18 23 23 25 28 33 33
## [29] 59 64 67 77 128 135 148 152 173 173 230 282 297 300
## [43] 625
```

## Boxplots to Represent Quartile Information

```
boxplot(bird$Count, horizontal = T)
```

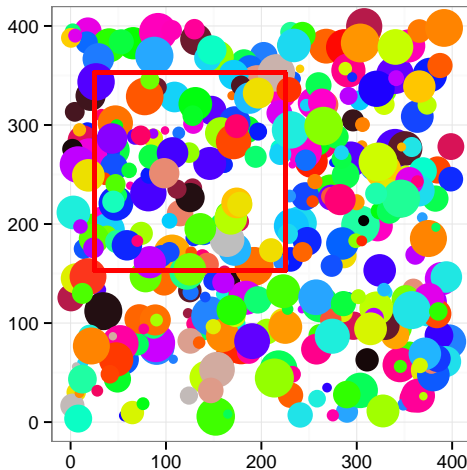


Whiskers show  $1.5 \times$  interquartile range, Points show outliers

# Variation in Sample Estimates

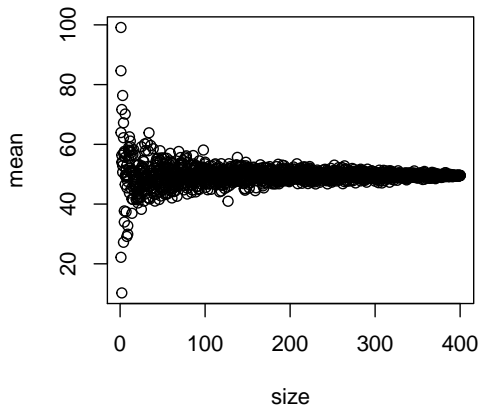
## Remember Samples and Populations?

How representative of our population are the estimates from our sample?



## Remember Samples and Populations?

We've seen that we get variation in point estimates at any sample size



## Exercise: Variation in Estimation

- ▶ Consider a population with some distribution (`rnorm`, `runif`, `rgamma`)
- ▶ Think of the mean of one sample as an individual replicate
- ▶ Take many (50) 'replicates' from this population of means
- ▶ What does the distribution of means look like? Use *hist*
- ▶ How does it depend on sample size (within replicates) or distribution type?

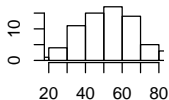
Extra: Show the change in distributions with sample size in one figure.



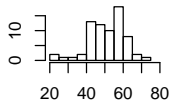
# Central Limit Theorem

The distribution of means converges on normality

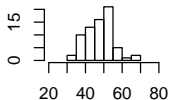
**n = 3**



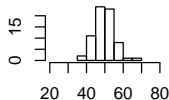
**n = 9**



**n = 15**



**n = 27**



# Central Limit Theorem Simulation

```
set.seed(697)
n <- 3
mvec <- rep(NA, times = 100)
# simulate sampling events!
for (i in 1:length(mvec)) {
  mvec[i] <- mean(runif(n, 0, 100))
}
hist(mvec, main = "n=3")
```

## Estimating Variation Around a Mean

Great, so, if we can draw many replicated means from a larger population, we can the standard deviation of an estimate!

This standard deviation of the estimate of the mean is the **Standard Error**.

**But for a single study, we only have one sample...**

## A Bootstrap Simulation Approach to Standard Error

- ▶ Our sample is representative of the entire population
- ▶ Therefore, we can resample it *with replacement* for 1 simulated sample
- ▶ We use our sample size as the new sample size as well

We set the replace argument in `sample` = TRUE

Try sampling from the bird data with replacement.

## A Bootstrap Simulation Approach to Standard Error

```
sample(bird$Count, replace = T, size = nrow(bird))
```

```
## [1] 23 135 1 23 59 4 67 15 3 1 135 13 152 128  
## [15] 67 148 7 1 3 2 67 1 23 3 300 64 2 282  
## [29] 297 33 297 2 25 128 128 173 14 64 1 33 2 297  
## [43] 282
```

```
sample(bird$Count, replace = T, size = nrow(bird))
```

```
## [1] 297 2 625 230 13 33 25 12 4 28 297 2 12 7  
## [15] 3 1 18 28 297 1 282 15 300 148 23 2 33 1  
## [29] 625 282 77 23 12 25 297 2 2 33 230 135 67 18  
## [43] 77
```

## Standard Error

$$SE_{\bar{Y}} = \frac{s}{\sqrt{n}}$$

$\bar{Y}$  - sample mean

s - sample standard deviation

n - sample size

## 95% Confidence Interval and SE

- ▶ Recall that 95% of the data in a sample is within 2SD of its mean
- ▶ So, 95% of the times we sample a population, the *true* mean will lie within 2SE of our estimated mean
- ▶ This is the 95% **Confidence Interval**

$$\bar{Y} - 2SE \leq \mu \leq \bar{Y} + 2SE$$

## Exercise: 95% Confidence Interval

$$\bar{Y} - 2SE \leq \mu \leq \bar{Y} + 2SE$$

- ▶ Draw 20 simulated samples with  $n=10$  from a normal distribution of mean 0
- ▶ Calculate the upper and lower confidence interval for each
- ▶ Compare the 95% CIs to the true value of the mean
- ▶ Extra: graph it with segments

Tip: To bind two vectors together as columns, use *cbind*



## Exercise: 95% Confidence Interval

```
set.seed(697)
n <- 20
upperCIvec <- rep(NA, n)
lowerCIvec <- rep(NA, n)

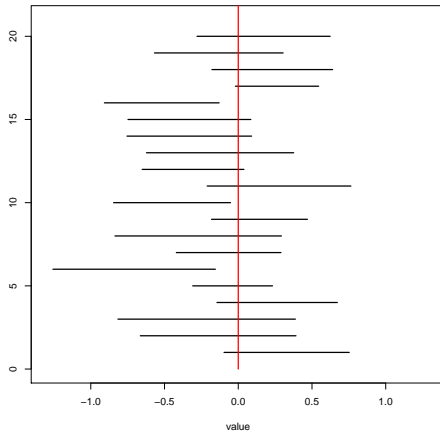
# loop and calculate the 95% CI
for (i in 1:n) {
  samp <- rnorm(10)
  upperCIvec[i] <- mean(samp) + 2 * sd(samp)/sqrt(n)
  lowerCIvec[i] <- mean(samp) - 2 * sd(samp)/sqrt(n)
}
```

## Exercise: 95% Confidence Interval

```
# examine the numbers
cbind(upperCIvec, lowerCIvec)[1:10, ]
```

##		upperCIvec	lowerCIvec
##	[1,]	0.75237	-0.09638
##	[2,]	0.39117	-0.66417
##	[3,]	0.38746	-0.81584
##	[4,]	0.67183	-0.14438
##	[5,]	0.23227	-0.30878
##	[6,]	-0.15508	-1.25684
##	[7,]	0.28960	-0.41992
##	[8,]	0.29285	-0.83584
##	[9,]	0.46890	-0.18128
##	[10,]	-0.05229	-0.84528

## Exercise: 95% Confidence Interval



## Variation in Other Estimates

- ▶ Many SEs and CIs of estimates have formulae and well understood properties
- ▶ For those that do not, we can bootstrap the SE of any estimate - e.g., the median
- ▶ Bootstrapped estimates (mean of simulated replicates) can be used to assess bias
- ▶ Bootstrapping is not a panacea - requires a good sample size to start