

# Statistical Inference

Gene Hunt

NMNH, Smithsonian Institution

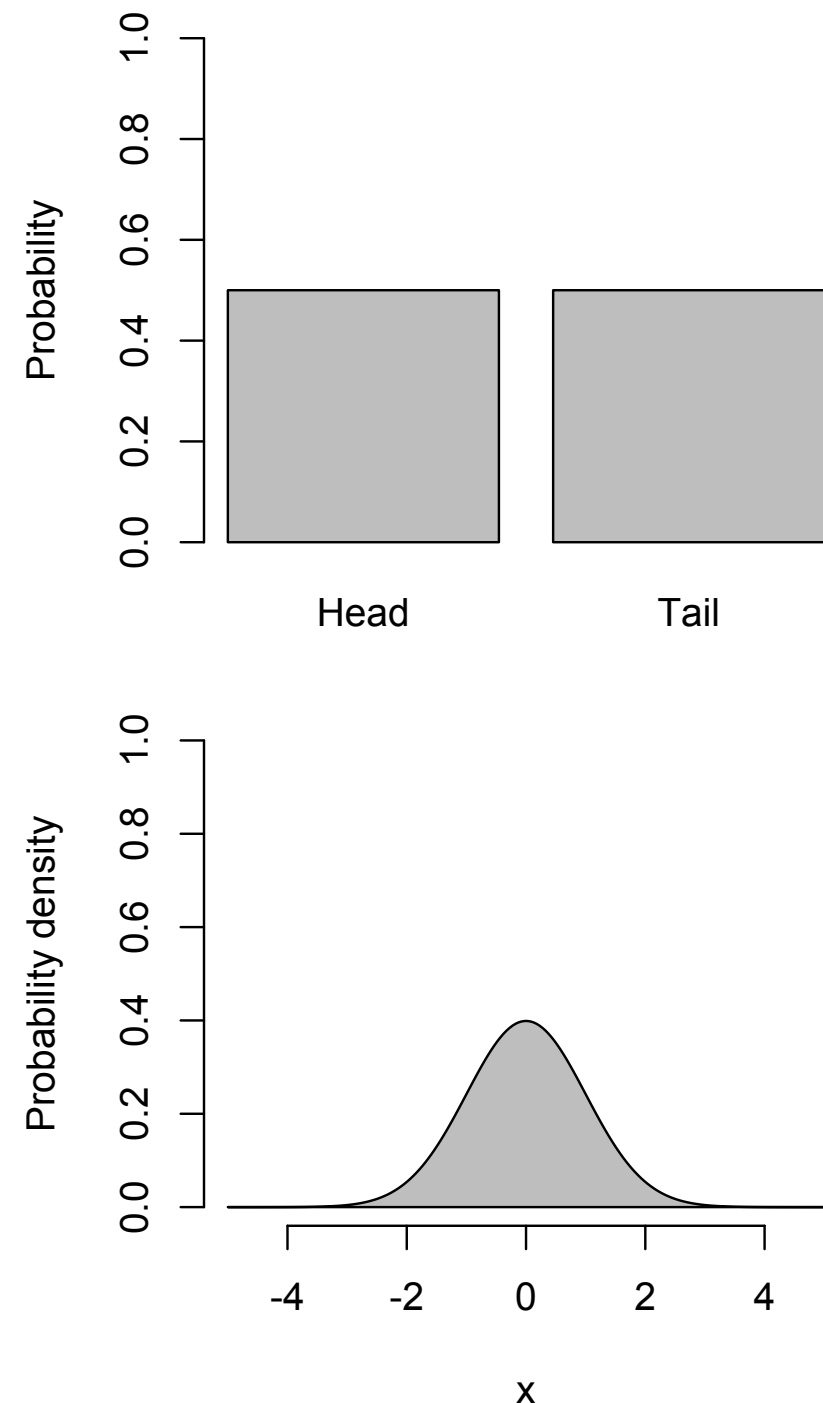
Analytical Paleobiology Workshop, July 2018

# Outline

- Preliminaries: Probability and Samples
- Classical (frequentist) approaches
- Likelihood
- Bayesian Approaches (in part II)
- P-hacking and Crises in Replication (in part II)

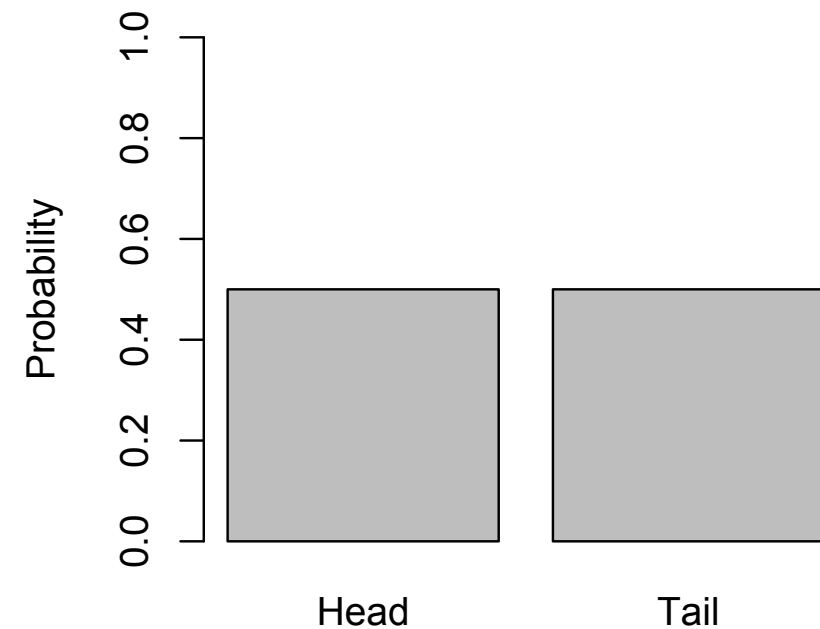
# Probability

- Probability is related to the chances that an event occurs
- Sample space: set of all possible outcomes
- Different outcomes can have different associated probabilities, specified by a model = **probability density**
- Some sample spaces are discrete, others continuous



# Operations on Probabilities

- Probabilities range from 0 to 1
- Probabilities of all possible outcomes sum to 1
- If all outcomes equally probable, can get probabilities by counting outcomes



**Outcomes of tossing two fair dice**  
die 1

die 2

	1	2	3	4	5	6
1	1-1	1-2	1-3	1-4	1-5	1-6
2	2-1	2-2	2-3	2-4	2-5	2-6
3	3-1	3-2	3-3	3-4	3-5	3-6
4	4-1	4-2	4-3	4-4	4-5	4-6
5	5-1	5-2	5-3	5-4	5-5	5-6
6	6-1	6-2	6-3	6-4	6-5	6-6

# Probability Examples

If events are *independent*:

$$\Pr(A \text{ and } B) = \Pr(A) \Pr(B)$$

If events are *disjoint*:

$$\Pr(A \text{ or } B) = \Pr(A) + \Pr(B)$$

Two coin tosses: probability of repeated flip?

Outcomes: HH, HT, TH, TT

$$\Pr(\text{HH or TT}) = \Pr(\text{HH}) + \Pr(\text{TT})$$

$$= 1/4 + 1/4$$

$$= 1/2$$

Two tosses or dice: probability of getting two sixes?

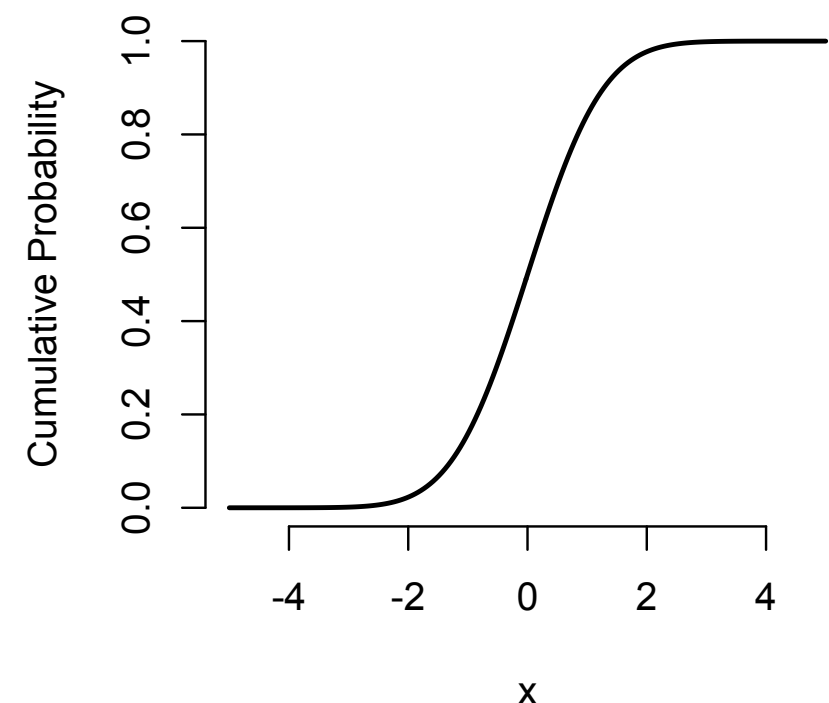
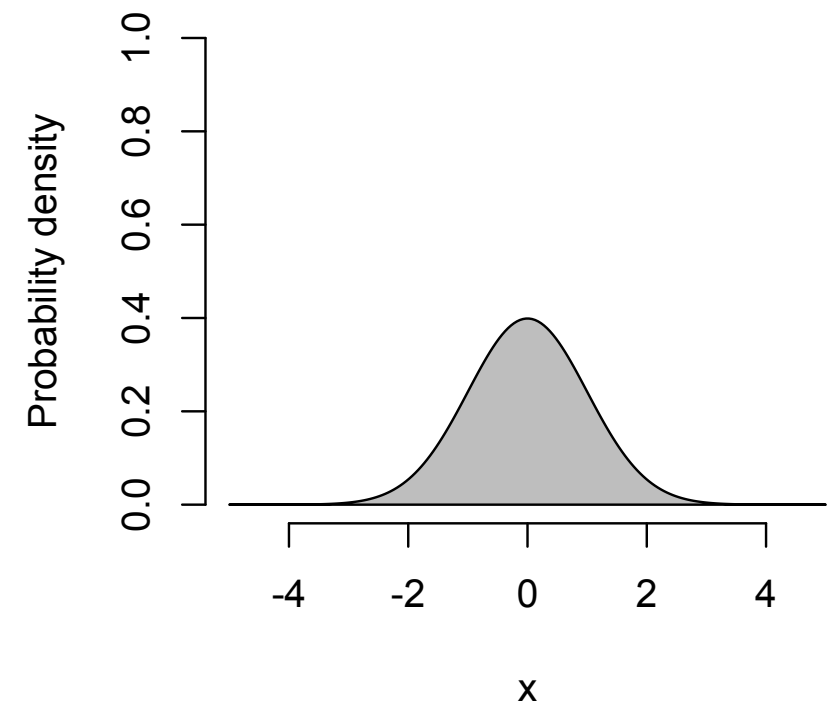
$$\Pr(6 \text{ and } 6) = \Pr(6) \times \Pr(6)$$

$$= 1/6 \times 1/6$$

$$= 1/36$$

# Probability Distributions in R

- Functions available for common probability distributions such as normal, t, uniform, binomial, etc.
- Example: normal distribution (`norm`)
  - `dnorm()` **density** function
  - `pnorm()` cumulative **distribution**  $\Pr(X < x)$
  - `qnorm()` quantile function (values from prob's)
  - `rnorm()` generate random variates



# Probability Distributions in R

Distribution	R name	additional arguments
beta	beta	shapel, shape2, ncp
binomial	binom	size, prob
Cauchy	cauchy	location, scale
chi-squared	chisq	df, ncp
exponential	exp	rate
F	f	df1, df2, ncp
gamma	gamma	shape, scale
geometric	geom	prob
hypergeometric	hyper	m, n, k
log-normal	lnorm	meanlog, sdlog
logistic	logis	location, scale
negative binomial	nbinom	size, prob
normal	norm	mean, sd
Poisson	pois	lambda
signed rank	signrank	n
Student's t	t	df, ncp
uniform	unif	min, max
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m, n

## Examples

density of gamma: `dgamma()`

quantiles of normal: `qnorm()`

random uniform variable: `runif()`

# Probabilities from Distribution Functions

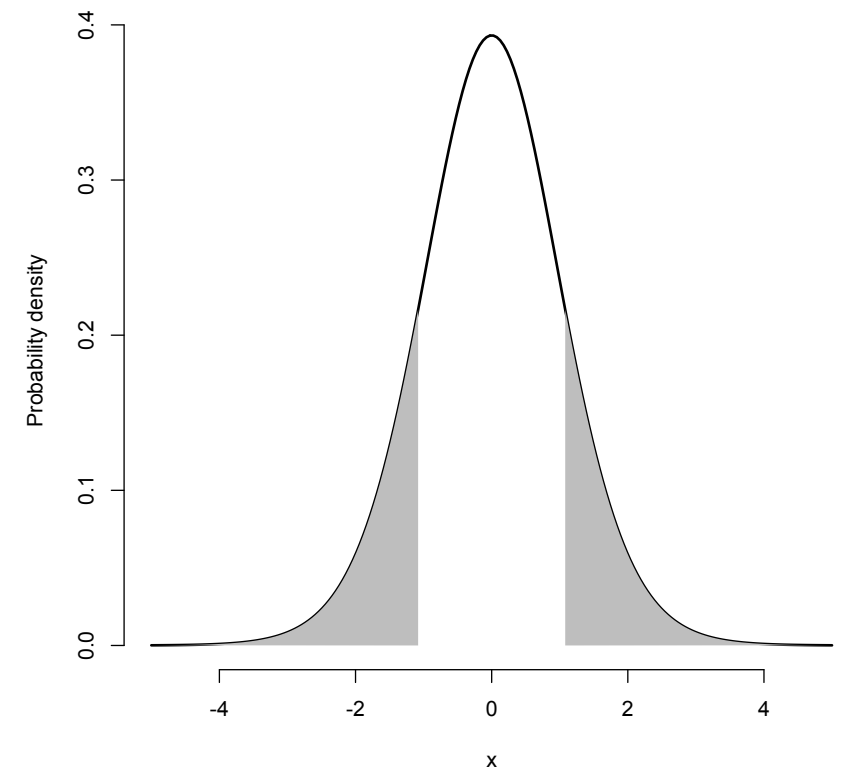
- Common task: get probability of observing value as extreme or more so than one observed ( $P$ -value)
- Area near tails of density function
- Calculation done from **distribution function** - cumulative area of density function

**Example: value of -1.06 from Normal (0,1)**

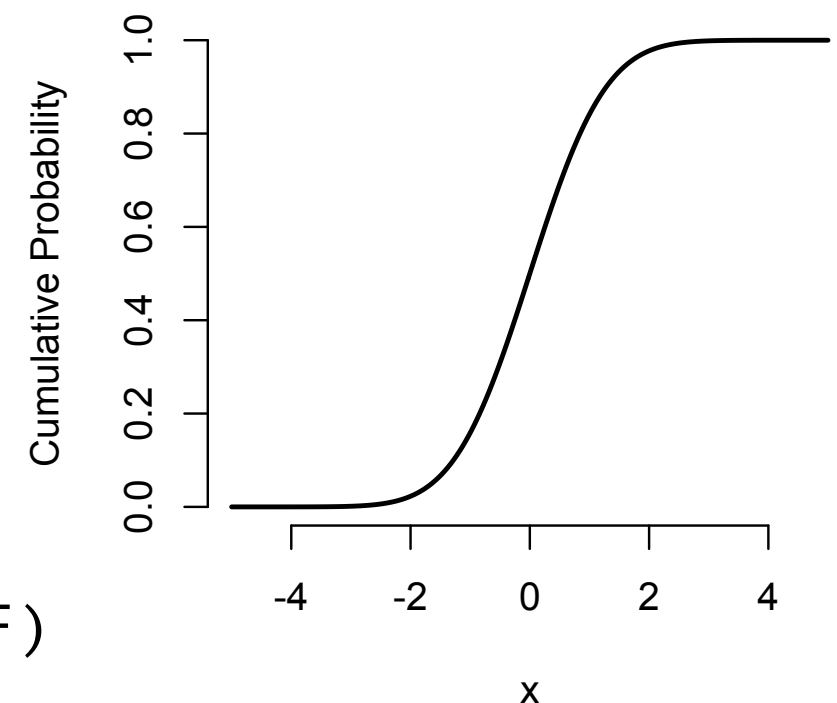
Left tail:  $\Pr(x < -1.06)$  `pnorm(-1.06)`

Right tail:  $\Pr(x > 1.06)$  `pnorm(1.06, lower.tail=F)`

`dnorm (x, mean, sd)`



`pnorm (x, mean, sd)`





### *Exercise 1. Probabilities and probability distributions*

1. When tossing two fair dice, what is the probability of observing exactly one six?
2. Generate a vector of 50 variates uniformly distributed between 10 and 20.
3. Suppose a set of femora lengths are drawn from a normal distribution with a mean of 10 and standard deviation of 1. Use the cumulative distribution function of the normal distribution to compute the probability of observing a value of less than 12. How about greater than 12? What should these sum to?
4. Same distribution as question #2. How can you get R to tell you what the femoral lengths define the 25th and 75th percentiles of this distribution?

# Populations and Samples

- A **population** is the complete set of objects or events of interest for some question
- A **sample** is a subset of objects chosen to represent a population

## Examples

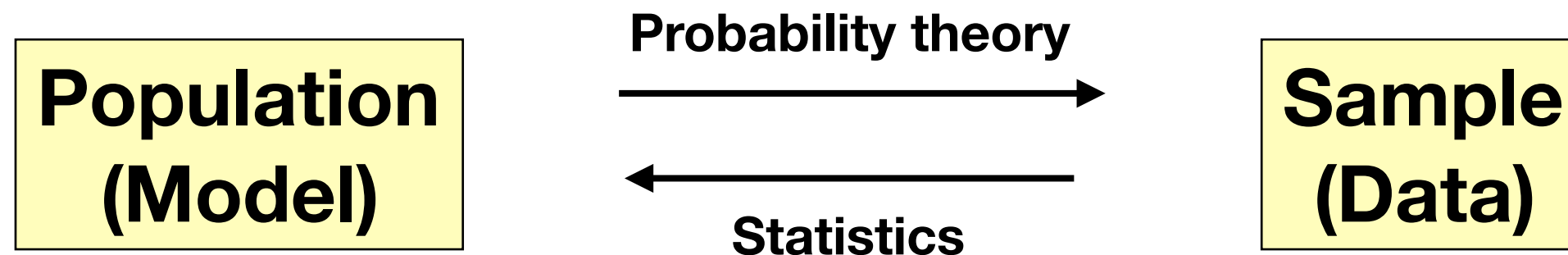
To a pollster, the population may be the full set of U.S. voters, but the sample may be a set of 1,000 respondents.

For a paleontologist, the population may be all individuals that ever existed of *Phacops rana*, with the sample being 25 specimens collected from an outcrop in New York.

**Ideally, samples are representative of populations with respect to the attributes under study**

# Populations and Samples

Samples, because they are finite subsets, will not match the population exactly. These differences are ascribed to **sampling error**.



# Outline

- Preliminaries: Probability and Samples
- Classical (frequentist) approaches
- Likelihood
- Bayesian Approaches (in part II)
- P-hacking and Crises in Replication (in part II)

# Classical Hypothesis Testing

- From the work of Pearson, Neyman, and Fisher early 20th century
- Dominates undergraduate statistics teaching
- Based on frequentist probability



**Karl Pearson**  
wikipedia



**R. A. Fisher**  
wikipedia

# Classical Hypothesis Tests

## Steps

- Specify the **null** and **alternative** hypotheses
- Determine a **test statistic** that reflects the hypotheses; compute it for observed data
- Using known distribution of test statistic, compute the **P-value** of the test statistic: probability of observing that result, or one more extreme, if the null hypothesis is true
- Compare the P-value to the **significance level**,  $\alpha$ . If  $P < \alpha$ , reject the null. If  $P > \alpha$ , do not reject the null.

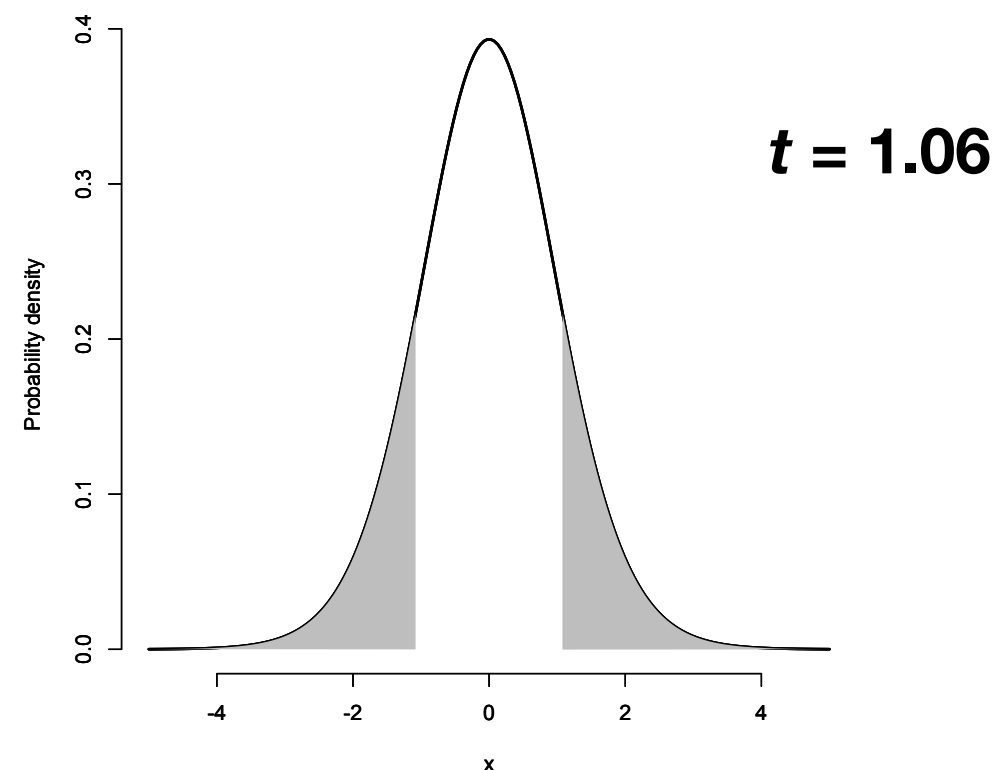
# Example: do two populations differ in means?

- **Null and alternative hypotheses**
- **Test statistic**
- **P-value** from distribution of test statistic
- Compare to the **significance level,  $\alpha$ .**

not significant,  $P = 0.308$

$$H_0 : \mu_1 = \mu_2 \quad H_a : \mu_1 \neq \mu_2$$

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$



`t.test(x1, x2)`

`Pval= 2 * pt(t, df = n1 + n2 - 2, lower = F)`

# Example: even sex ratio?

**Data: 20 male, 30 female**

- **Null and alternative hypotheses**

$$H_0 : p = 0.5 \quad H_a : p \neq 0.5$$

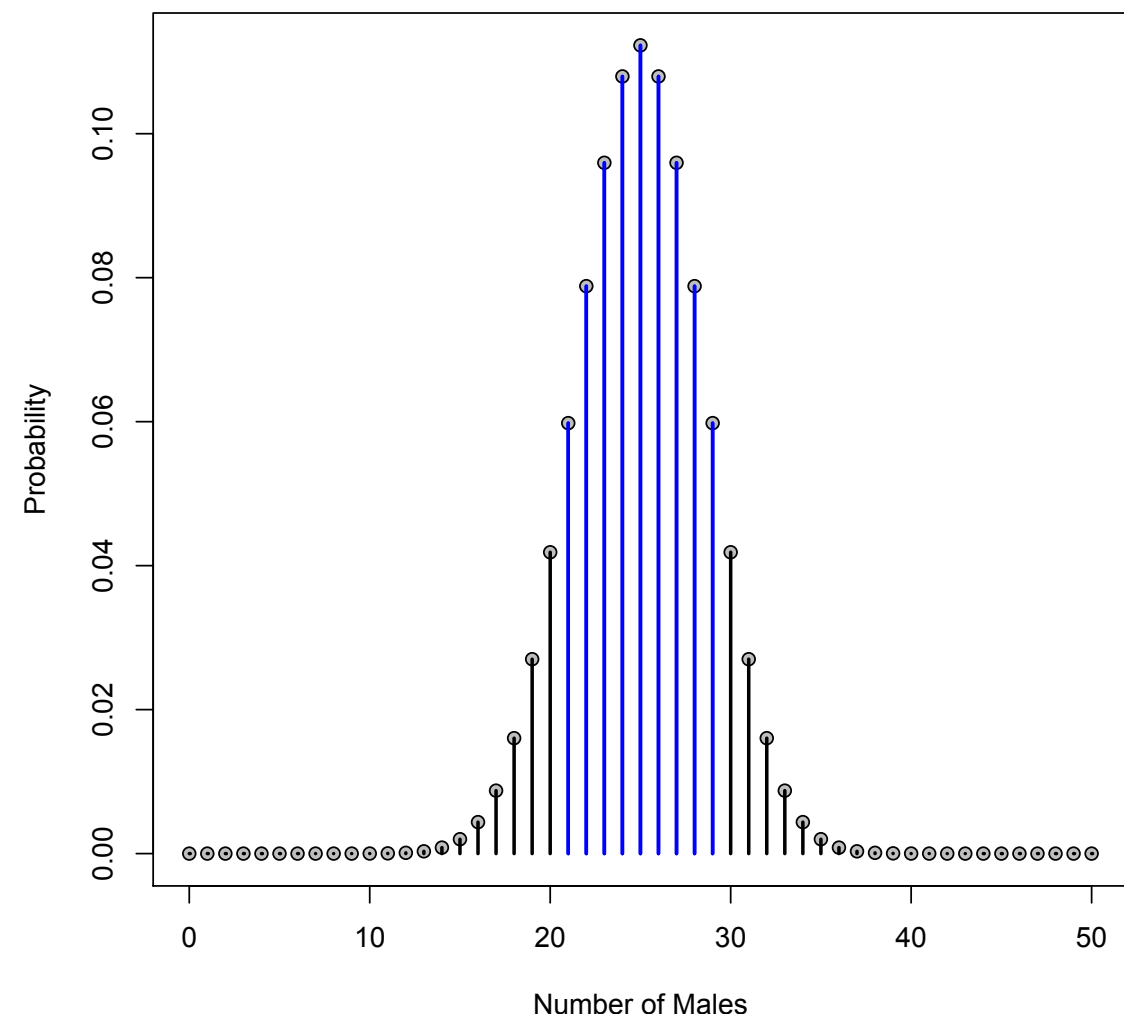
- **Test statistic**

- **P-value** from distribution of test statistic

- Compare to the **significance level,  $\alpha$ .**

**not significant,  $P = 0.203$**

$x$  = number of males (20)



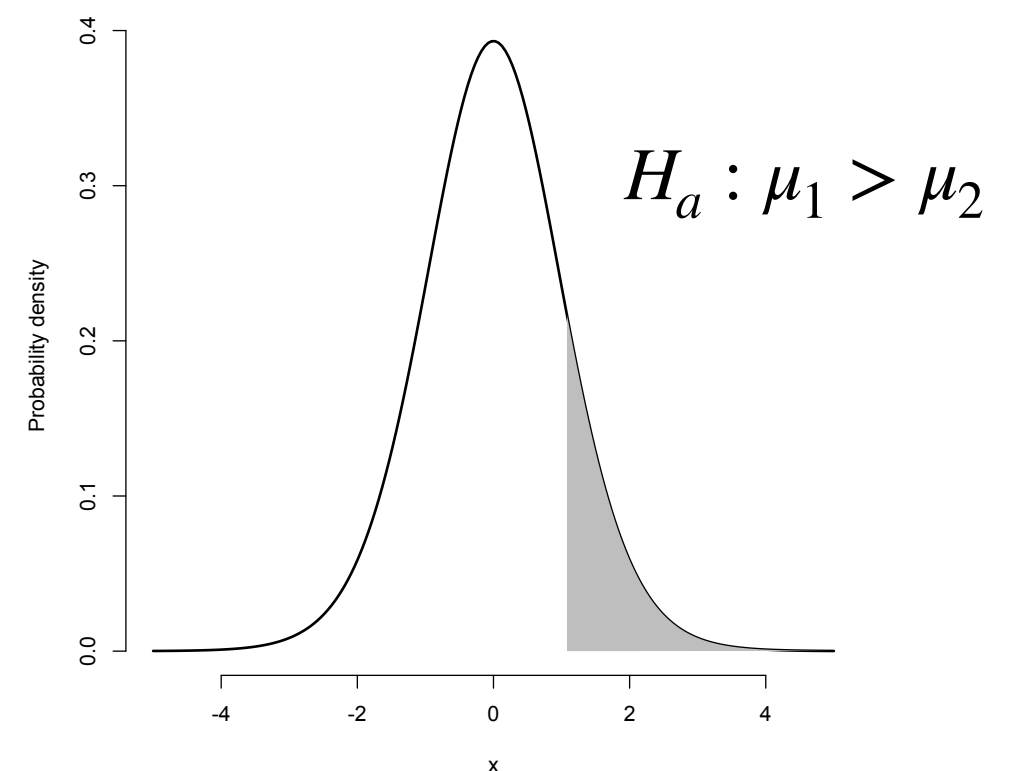
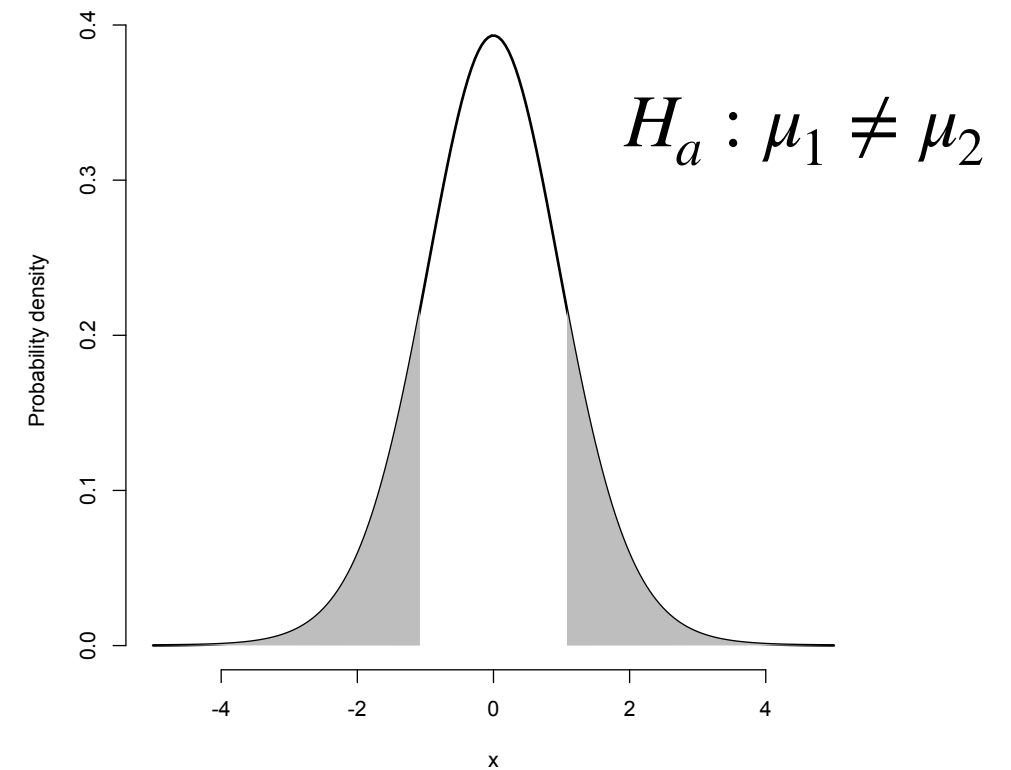
```
binom.test(x = 20, n = 50)
```

```
Pval = 2 * pbinom(x = 20, size = 50)
```



# 1- vs 2-tailed tests

- Thus far, tests have been **two-tailed**;  $H_a$  includes both tails of the distribution of the test statistic
- Some tests are **one-tailed** because interest is only in departures in one direction (e.g., Chi-square)
- Tests should usually be two-tailed.
- Ask: would I also be interested in a significant finding in the other direction?



# Types of Error

		Reality	
		$H_0$ is true	$H_0$ is false
Test Result	Fail to Reject $H_0$	OK	Type II error False negative
	Reject $H_0$	Type I error False positive	OK

$$\Pr(\text{Type I}) = \alpha$$

$$\Pr(\text{Type II}) = \beta$$

$$\text{Power} = 1 - \beta$$

There is a tradeoff between  
Type I and Type II error

## *Exercise 2. Null hypothesis testing*

1. Simulate two samples drawn from a normal distribution with equal means and standard deviations. Perform a t-test using the R function `t.test()`. What is the p-value? What did you expect?
2. One iteration is not enough to get a sense of the behavior of this test. Write a script that replicates the previous exercise 1,000 times, recording the p-value of each replication into a vector. What does this distribution look like? What proportion of tests show a significant p-value assuming  $\alpha = 0.05$ ?
3. Now, investigate Type II error by simulation, again for the t-test. Recall that Type II means that  $H_0$  is false but not rejected, therefore we need to specify how  $H_0$  is false. Let's say that sample 1 was drawn from a population with a mean of 10 and sd of 1, whereas sample 2 the mean is 11 while sd is still 1. Assume a sample size of  $n = 20$  for each sample. Run simulations to estimate the probability of Type II error in this situation.
4. Repeat question 3, but now for  $n = 10$  per sample. How much does the probability of Type II error change? How about when  $n = 50$  per sample?

# Some Issues with P-values

- For large enough datasets, nearly all tests are significant. Statistical significance is not the same as scientific importance.
- Threshold problems:  $P = 0.049$  and  $P = 0.051$  get different interpretations, but  $P = 0.051$  and  $P = 0.99$  are interpreted the same.
- They can be used to reject hypotheses but not support them.
- They often receive too much focus, at the expense of other important issues (e.g., parameter estimates and confidence intervals).

# ASA's Statement on P-values (2016)

1. P-values can indicate how incompatible the data are with a specified statistical model.
2. P-values do not measure the probability that the studied hypothesis is true, or the probability that the data were produced by random chance alone.
3. Scientific conclusions and business or policy decisions should not be based only on whether a P-value passes a specific threshold.
4. Proper inference requires full reporting and transparency
5. A P-value, or statistical significance, does not measure the size of an effect or the importance of a result.
6. By itself, a p-value does not provide a good measure of evidence regarding a model or hypothesis.

# Point Estimation

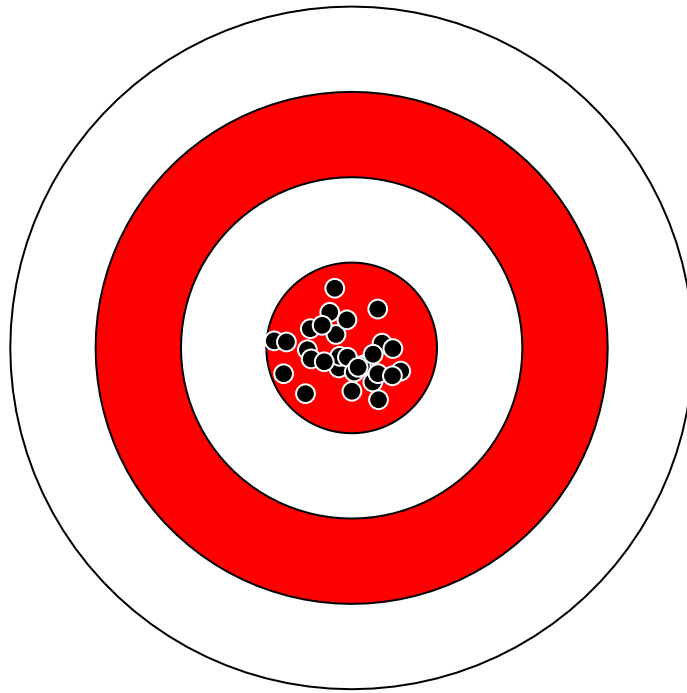
- A **parameter** is a quantity describing an aspect of a population, estimable only indirectly from samples.
- A **point estimate** is a number that can be regarded as the most plausible value of a parameter
- Example: the sample mean is a point estimate of the population mean ( $\hat{\mu}$ )

$$\bar{x} = \frac{\sum x_i}{n}$$

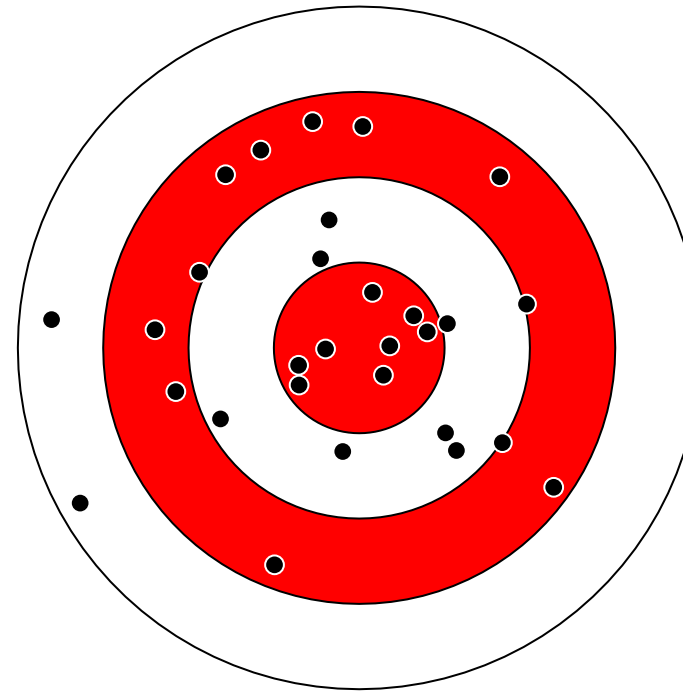
# Point Estimation

Estimators should be unbiased (accurate) and have low variance (precise)

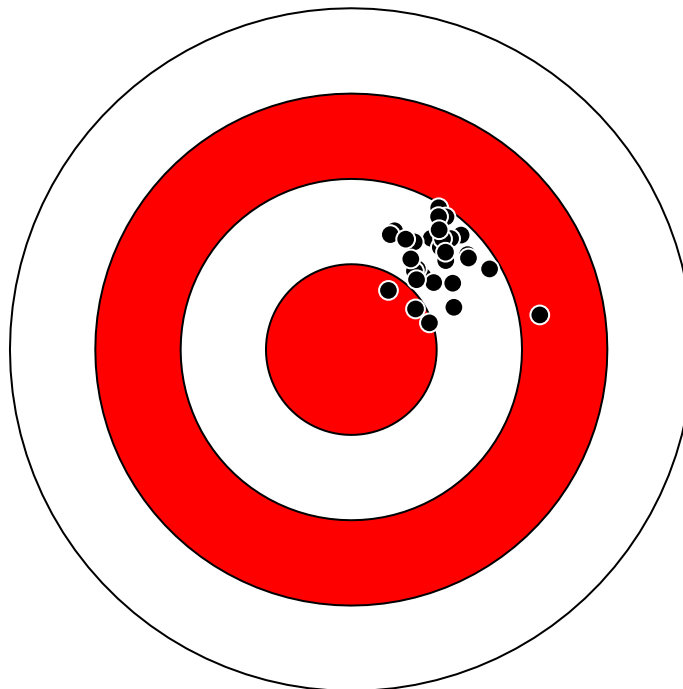
**High accuracy  
High precision**



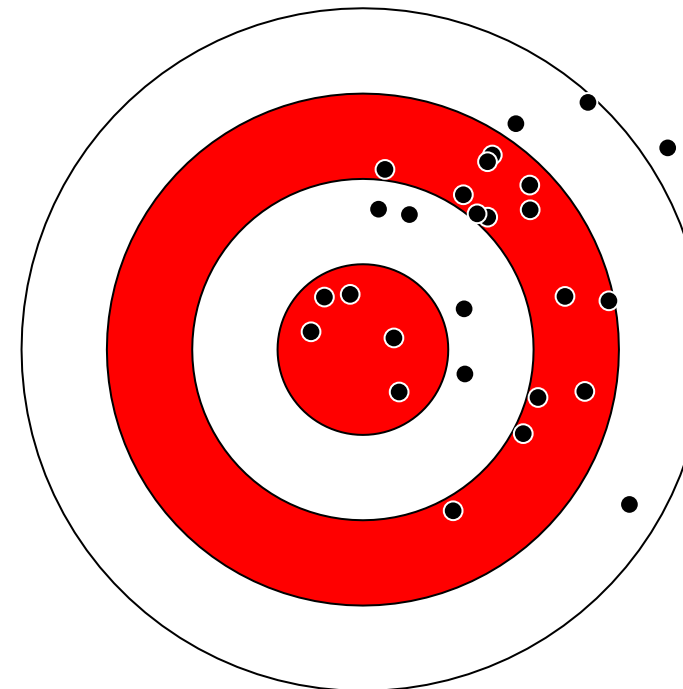
**High accuracy  
Low precision**



**Low accuracy  
High precision**

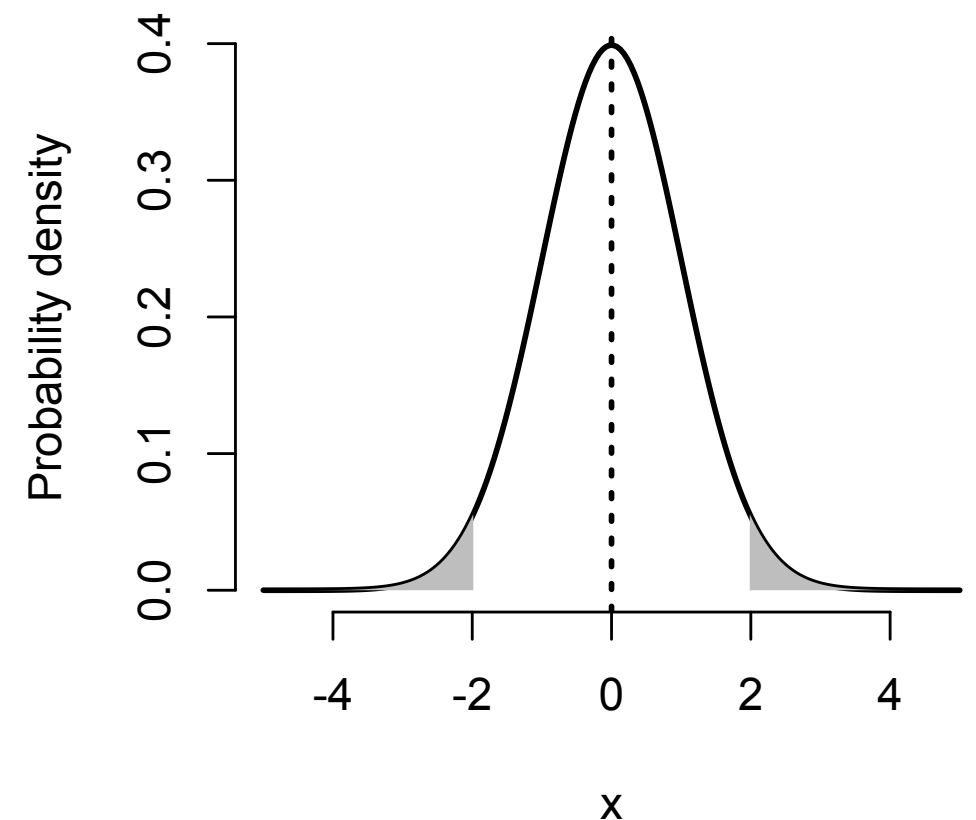


**Low accuracy  
Low precision**



# How Precise?

- Sampling error causes an estimate to differ from its true value, according its **sampling distribution**
- Measuring these deviations:
  - their standard deviation is called the **standard error**
  - the distribution can be used to compute **confidence limits**



**Confidence limits and hypothesis tests are related**

$(1 - \alpha)$  confidence intervals include all values for which a null hypothesis test is non-significant at a threshold of  $\alpha$



# Determining the Sampling Distribution

- How determine sampling distribution of a test statistic?
- Can be **derived mathematically** after making **assumptions** about population/data, such as:
  - normally distributed
  - equal variance
  - independent
- Non-parametric tests often relax assumptions about specific distributions, often with lower statistical power
- Several computationally intensive alternatives exist for generating a null / confidence interval: bootstrapping, permutation tests, parametric bootstrapping

### *Exercise 3. Point estimation*

1. Is the estimator of the sample mean unbiased? Write a script that uses simulation to generate many samples from a normal distribution in order to test this claim.
2. The standard error of estimates of a sample mean is known analytically to be  $s / \sqrt{n}$ , where  $s$  is the standard deviation of the sample and  $n$  is the number of observations. Check the results from your script above to see if they are consistent with this claim.
3. Simulate  $x$  from a standard normal with  $n = 2000$ . Now set  $y \leftarrow x + \text{rnorm}(2000, 0, 10)$ . Make a scatterplot of  $x$  and  $y$  and note the pattern. Now, do a linear regression of  $y \sim x$  and check the p-value. Are you surprised?

# Resampling Approaches

Useful when

- assumptions of parametric approaches are violated (non-normal, etc.)
- test statistic does not have known distribution

Unlike non-parametric approaches, resampling approaches often have statistical power comparable to standard parametric tests

Costs: computational time/effort, less accurate when assumptions strictly met

# Approaches to Resampling

method	nature of resampling	<b>R code</b> x = data vector n = sample size grp = grouping variable
Bootstrapping	sample, with replacement, n times from data	<code>sample(x, size = n, replace = TRUE)</code>
Permutation	shuffling, usually of group labels	<code>sample(grp) # default is permutation</code>
Parametric Bootstrapping (Monte Carlo simulation)	generating from a parametric null	<code>## for example x &lt;- rnorm(50, mean = 22.1, sd = 1.2)</code>

*see Kowalewski & Novack-Gottshall (2010) PS Short Course.*

# General Steps

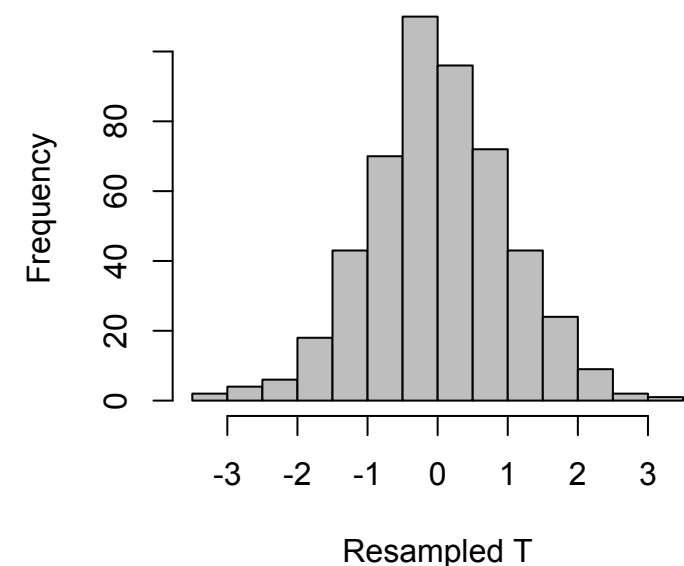
1. Identify the hypothesis
2. Choose the test statistic,  $T$
3. Calculate the observed test statistic,  $T_{obs}$
4. Produce a resampling distribution
5. Compute P-value or CI based on resampling distribution

**example**

$$H_0 : \mu_1 = \mu_2$$

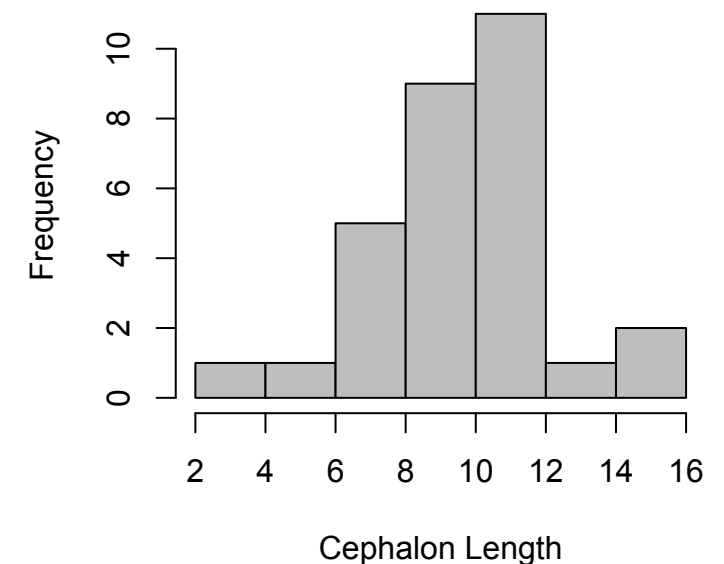
$$\bar{X}_1 - \bar{X}_2 \quad \text{or} \quad t$$

$$t_{obs} = 1.2$$



# Why Does this Work?

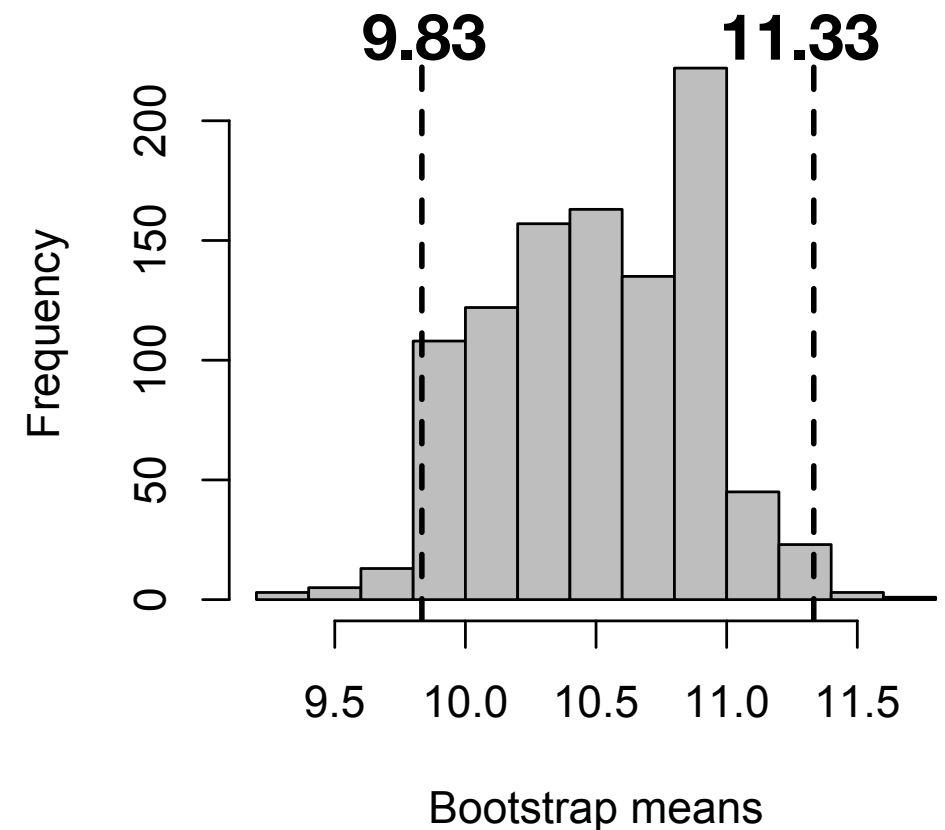
- Bootstrapping uses the data itself as a proxy for the population distribution
- Parametric bootstrapping, like null hypothesis testing, assumes a parametric form for the population. It just generates the sampling distribution of  $T$  through simulation rather than analytically.



# Simple Bootstrap Example

**Confidence interval on sample mean:**

```
x <- c(9, 10, 10, 11, 11, 12)
```



(1) Generate 1000 bootstrap samples of  $x$

```
xb <- sample(x, replace = TRUE)
```

(2) Compute sample mean for each

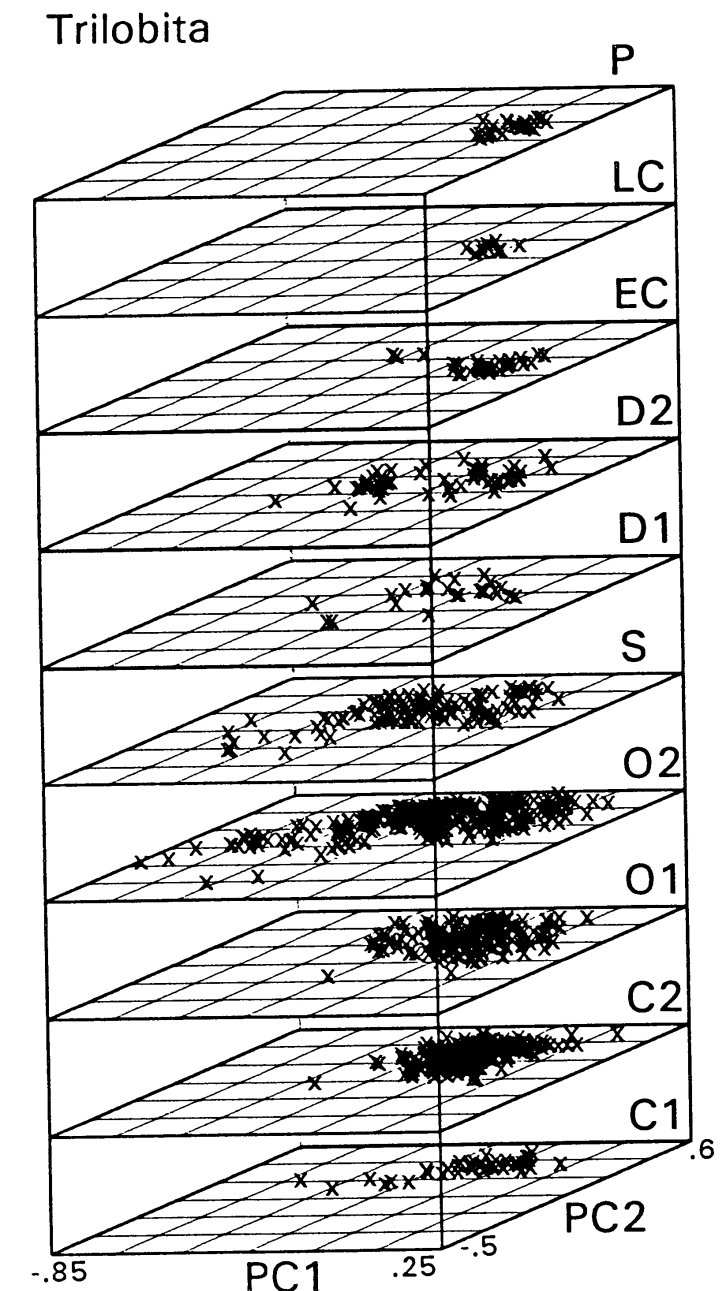
```
mb[i] <- mean(xb)
```

(3) CI from quantiles of resampled means

```
ci <- quantile(mb, prob = c(0.025, 0.975))  
# these are simple CI's; there are variants  
# of bootstrap CI's that are better
```

# Paleo Bootstrap Example

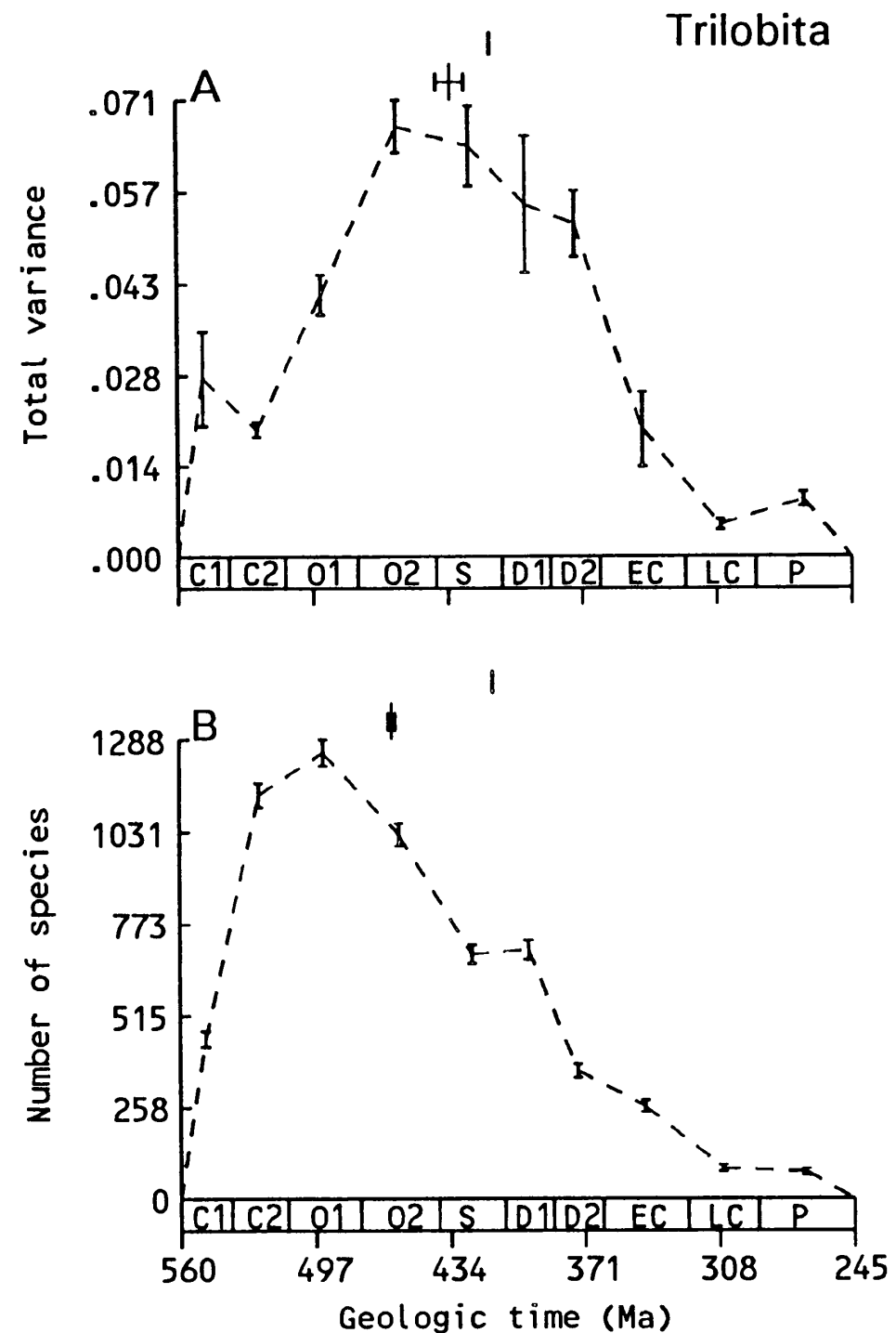
- Foote (1993) looked at morphological disparity in trilobites and blastoids over time.
- Disparity has unknown sampling distribution, so CIs were based on bootstrap resampling





# Paleo Bootstrap Example

- Foote (1993) looked at morphological disparity in trilobites and blastoids over time.
- Disparity has unknown sampling distribution, so CIs were based on bootstrap resampling



# Paleo Parametric Bootstrap Example

- Hunt & Chapman (2001) looked at evidence for instar clustering in trilobites using mixture models.
- Parametric bootstrapping was used to test sequentially the number of instar clusters.
- The test statistic (likelihood ratio) does not follow its expected distribution for mixture models.

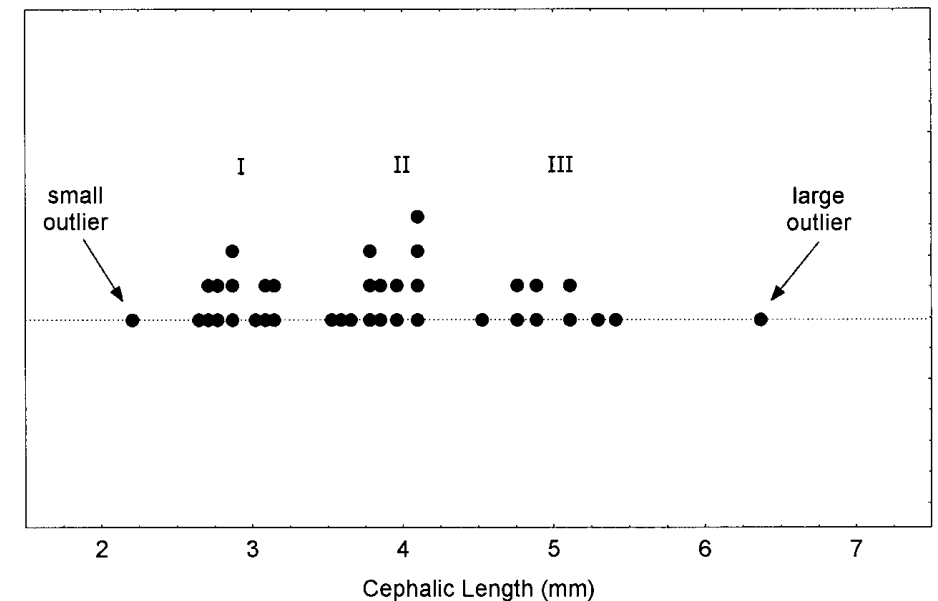


FIGURE 2. Dot plot of cephalic length for the *Piobaspis sellata* data. Each dot represents one specimen; specimens are binned over a small interval (0.065 mm). Arrows point to the smallest and largest specimens, which were excluded from the mixture analysis. Roman numerals I–III indicate putative clusters referenced in text.

No. of groups	Support	Bootstrap test ( $H_1$ vs. $H_2$ )	Observed $-2\lambda$	Bootstrap results	Bootstrap $p$ -value	Power
1	0	—	—	—	—	—
2	6.73	1 vs. 2 groups	13.46	183/3000	0.061	0.59
3	13.94	1 vs. 3 groups	27.89	57/2000	0.029*	0.87
4	19.06	3 vs. 4 groups	10.23	504/1000	0.504	0.10

## Exercise 4. Resampling approaches

1. One drawback of RMA is that it is not straightforward to compute confidence limits and standard errors for the slope. For this exercise, we'll make a new function, `RMAboot()`, to compute bootstrap standard errors on the RMA slope. You'll need to loop through a large number of replicates, and for each rep, take a bootstrap sample, compute the RMA slope for this bootstrap sample, and store it in an array. One of the tricky parts here is that the observations are paired, so you need to sample the indices; if there are  $N$  points, `sample(1:N, replace=TRUE)` will create a bootstrap sample of the indices.
2. Return to the cope data. We are going to test if populations from the species *P. pinto* and *P. species4* have different mean sizes. Extract the valve lengths from these two species into separate vectors, `pin` and `sp4`. Perform a `t.test` on these vectors.
3. The function `t.test` also allows one to specify the test with a formula, which will be more convenient for a permutation test. This formula is specified as `y ~ group vector`, where `y` is a vector that includes lengths from both species and `group` specifies which species they are from. Implement the same t-test as in #2 but use the formula interface.
4. A good way to visualize the distribution of small datasets of one variable is through `stripchart()`. Check out its help and figure out how to use formula notation to show the two species separately.
5. Now, write a script to perform a permutation test for these same data. Use the  $t$  statistic returned by `t.test` as your test statistic. With each iteration of the loop, you'll need to permute the `group` vector using the `sample` function.

# Outline

- Preliminaries: Probability and Samples
- Classical (frequentist) approaches
- Likelihood
- Bayesian Approaches (in part II)
- P-hacking and Crises in Replication (in part II)

# Likelihood

- Basis for statistical inference, developed by R. A. Fisher
- Likelihood ( $L$ ) of model parameters ( $\theta$ ) given data ( $x$ ), is the probability of the data, given the model:

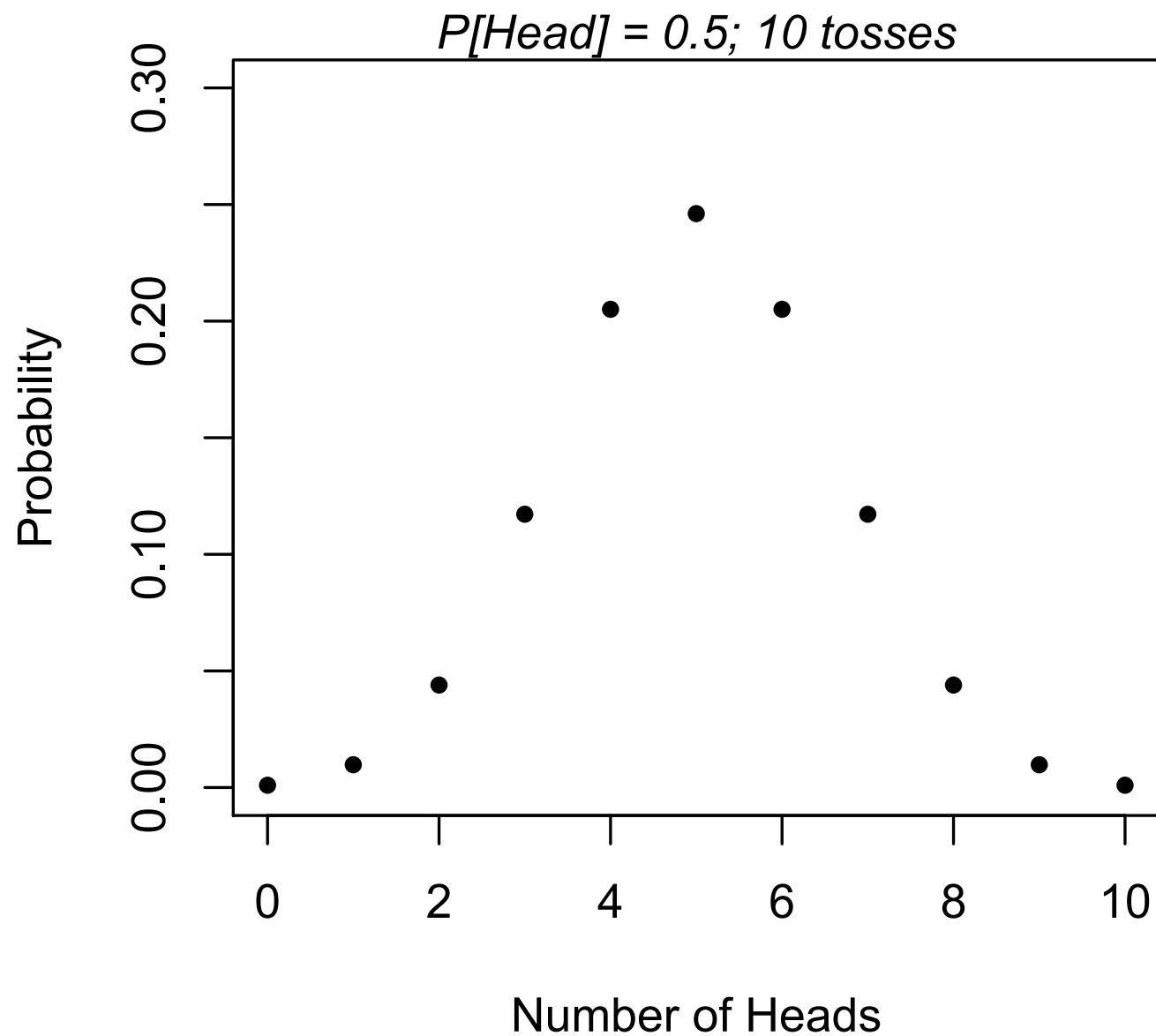
$$L(\theta | x) = Pr(x | \theta)$$

- Parameter values that yield the highest probabilities of producing observed data are favored
- Likelihood also provides a basis for evaluating the support different models receive

# Probability vs. Likelihood

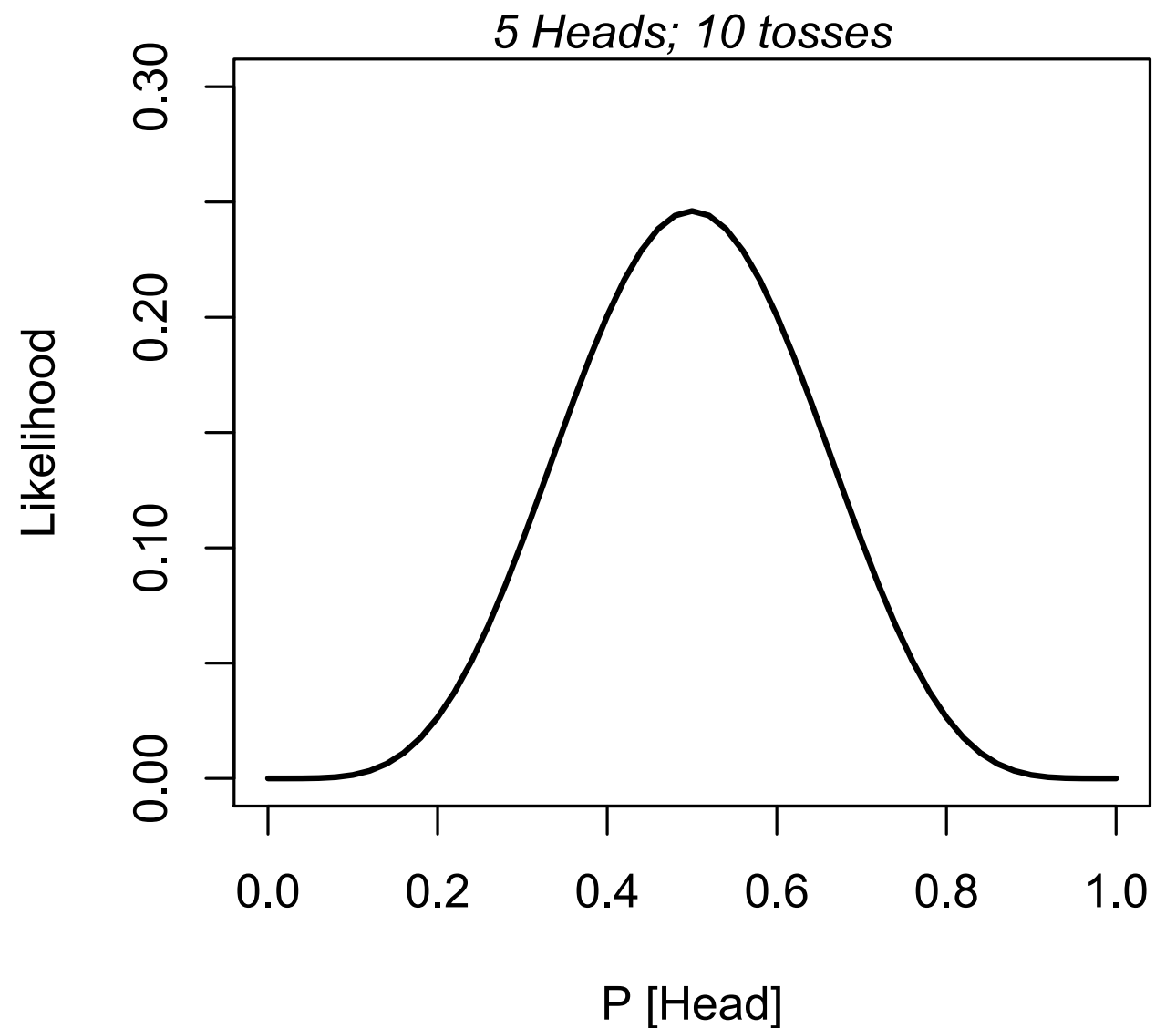
$$Pr(x | \theta)$$

**parameters are fixed,  
data are variables**



$$L(\theta | x)$$

**data are fixed,  
parameters are variables**



# Log-likelihood

- It is usually more convenient to work with log-likelihoods,  $\log L = \log(L)$
- With multiple, independent observations, the total likelihood is their product and the total log-likelihood is their sum

$$L(\theta | x_1, x_2) = L(\theta | x_1) \cdot L(\theta | x_2)$$

$$\log L(\theta | x_1, x_2) = \log L(\theta | x_1) + \log L(\theta | x_2)$$

# Example: coin tosses

- Flip a coin 10 times. Data ( $x$ ) are 7 heads 3 tails
  - $H_1$ : fair coin,  $\Pr[H] = p = 0.5$
  - $H_2$ : possibly unfair coin;  $p$  can vary
- Free parameters
  - $H_1$ : none
  - $H_2$ :  $p$



# Probability of the Data

- Binomial distribution: probability of  $x$  successes in  $n$  trials, given probability of success  $p$ :

$$Pr(x|p) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

- Likelihood for  $H_1$ : fill in  $x$  (7 Heads),  $n$  (10 trials), and  $p$  (0.5), compute result.

$$L(H_1 | x) = 0.117$$

```
dbinom(x = 7, size = 10, prob = 0.5)
```

$$\log L(H_1 | x) = -2.14$$

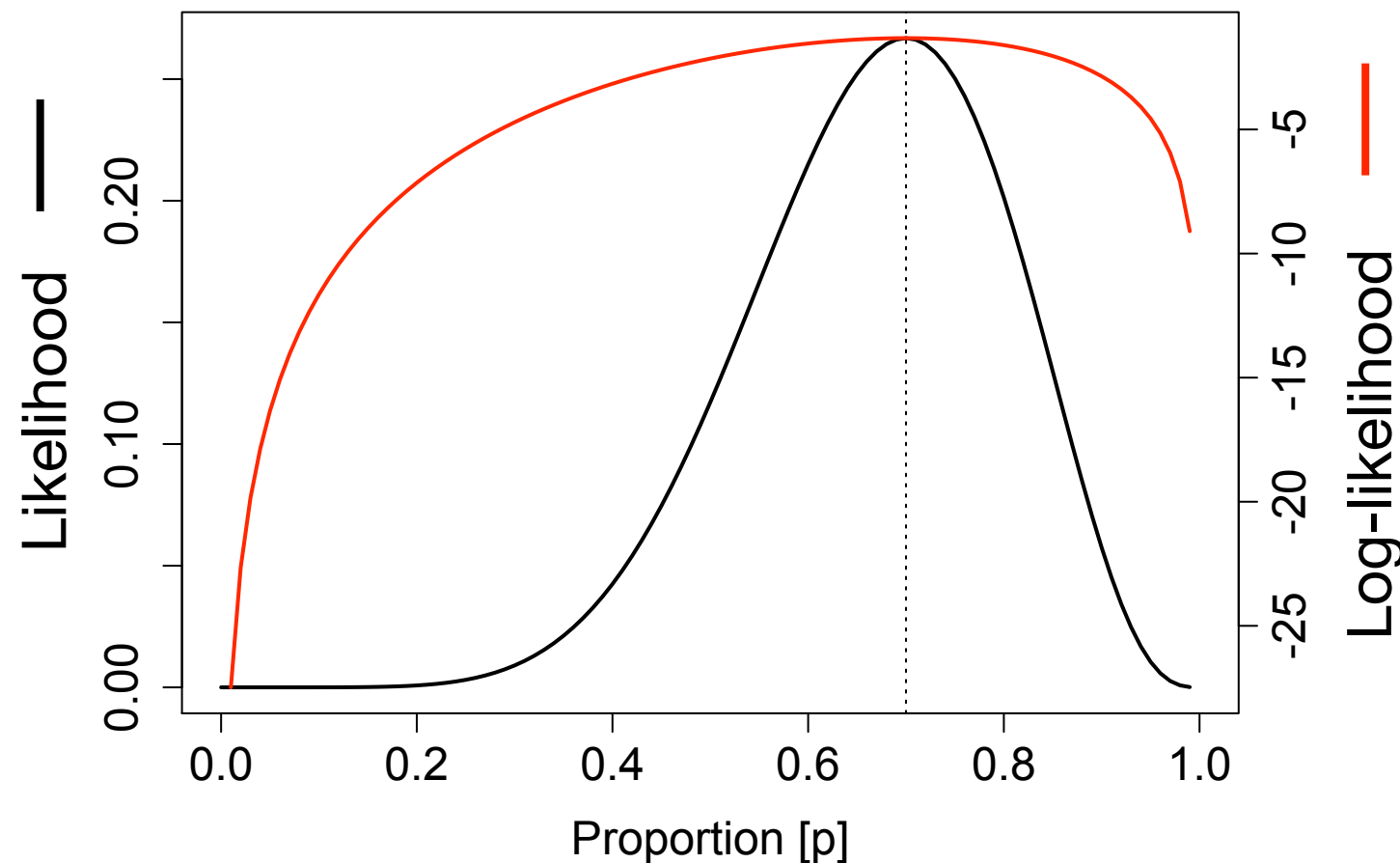
```
dbinom(x = 7, size = 10, prob = 0.5, log = T)
```

# Likelihood of $H_2$

For  $H_2$ ,  $p$  is freely variable, and unknown. We want to choose a value for  $p$  such that it maximizes  $L(H_2)$ , which is same as maximizing  $\log L(H_2)$ .

$L(p = 0.1)$	<code>dbinom(x = 7, size = 10, prob = 0.1)</code>	0.000
$L(p = 0.2)$	<code>dbinom(x = 7, size = 10, prob = 0.2)</code>	0.001
$L(p = 0.3)$	<code>dbinom(x = 7, size = 10, prob = 0.3)</code>	0.009
$L(p = 0.4)$	<code>dbinom(x = 7, size = 10, prob = 0.4)</code>	0.042
$L(p = 0.5)$	<code>dbinom(x = 7, size = 10, prob = 0.5)</code>	0.117
$L(p = 0.6)$	<code>dbinom(x = 7, size = 10, prob = 0.6)</code>	0.215
$L(p = 0.7)$	<code>dbinom(x = 7, size = 10, prob = 0.7)</code>	0.267
$L(p = 0.8)$	<code>dbinom(x = 7, size = 10, prob = 0.8)</code>	0.201
$L(p = 0.9)$	<code>dbinom(x = 7, size = 10, prob = 0.9)</code>	0.057

# Likelihood of H<sub>2</sub>



$$L(p | x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

For logL, get derivative and set equal to zero, solve for p.

$$\hat{p} = \frac{x}{n} = 0.7$$

The value of  $p$  that produces the highest likelihood is called the **maximum likelihood estimate** (MLE).

# Asymptotic Properties of MLEs

$$n \rightarrow \infty$$

- **unbiased:** the expected value is equal to the true parameter
- **efficient:** has the smallest variance among unbiased estimators
- **normal:** the sampling distribution is Gaussian
- **consistent:** becomes arbitrarily close to the true parameter

# Likelihood of $H_2$

- For  $H_2$ , the best estimate for  $p$  is 0.7. Substituting into the probability function gives  $L(H_2)$  and  $\log L(H_2)$ :

$$L(H_2) = \frac{n!}{x!(n-x)!} \hat{p}^x (1 - \hat{p})^{n-x}$$

- Likelihood for  $H_2$ : fill in  $x$  (7 Heads),  $n$  (10 trials), and MLE of  $p$  (0.7), compute result.

$$L(H_2 | x) = 0.266 \quad \text{dbinom}(x = 7, \text{size} = 10, \text{prob} = 0.7)$$

$$\log L(H_2 | x) = -1.32 \quad \text{dbinom}(x = 7, \text{size} = 10, \text{prob} = 0.7, \text{log} = \text{T})$$

# Choosing Among Models: LRTs

- $L(H_2)$  is higher than  $L(H_1)$ . How do we interpret this?
- Generally log-likelihoods increase with model complexity
- One can use a **likelihood ratio test** (LRT) to see if more complex model is significantly better than simpler one. The likelihood ratio is:

$$\Lambda = \frac{L(H_1)}{L(H_2)}$$

- Some models are **nested** — simpler one is special case of the more complex one. For nested models, -2 times the the log of the likelihood ratio is approximately distributed as a Chi-square, with df as the difference in the number of free parameters in  $H_1$  and  $H_2$ .

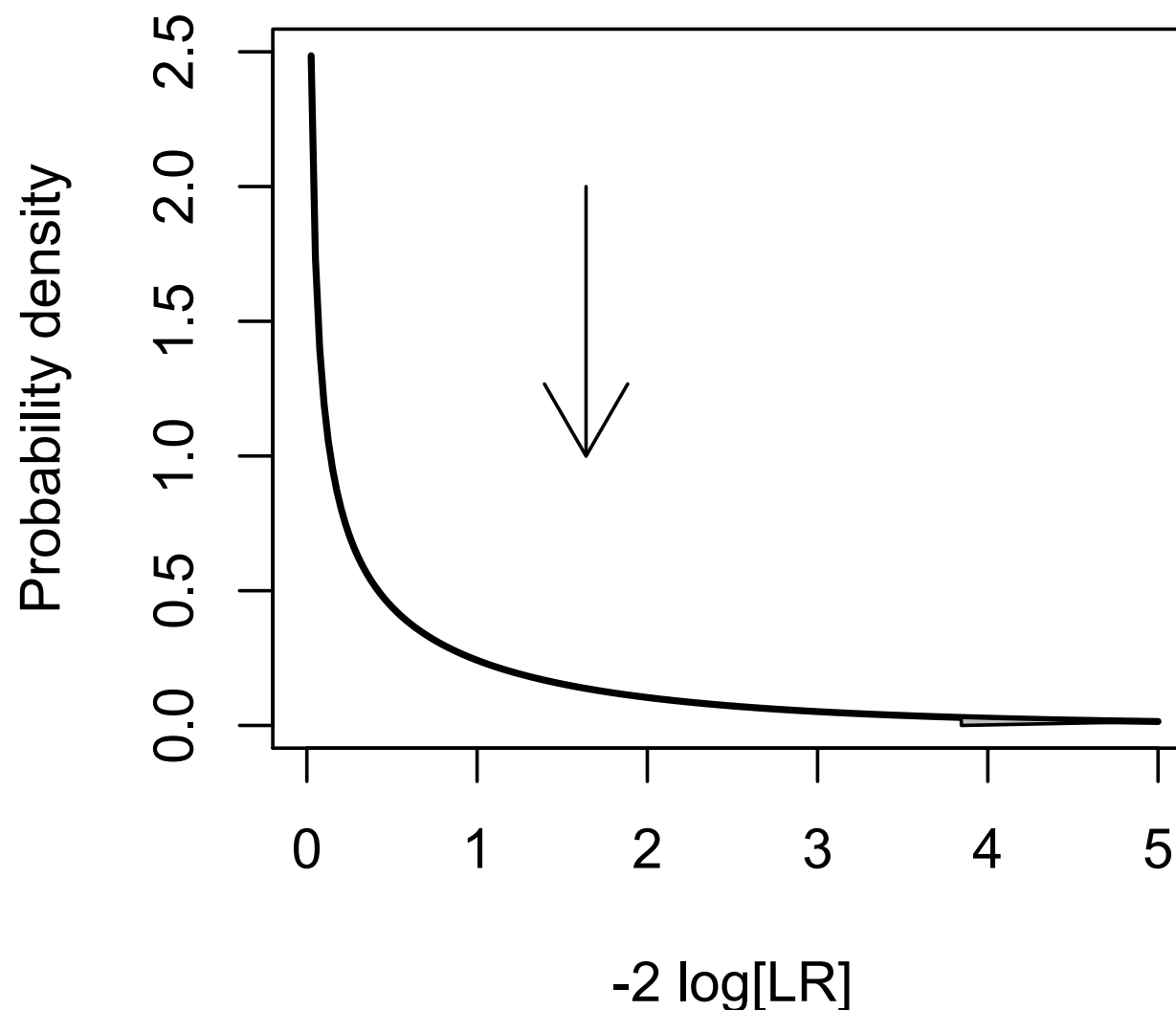
$$-2 \log \Lambda \sim \chi^2 \qquad -2 \log \frac{L(H_1)}{L(H_2)} \qquad -2 (\log L(H_1) - \log L(H_2))$$

# LRT for coin flip

For our coin toss,  $L(H_1) = 0.117$  and  $L(H_2) = 0.266$

$$-2 \log \Lambda = -2 \log \frac{0.117}{0.266} = 1.64$$

```
pchisq(q = 1.64, df = 1, lower=FALSE) # P = 0.20
```



# Information Criteria: AIC

- Can use **Akaike Information Criterion** to balance goodness-of-fit ( $\log L$ ) and model complexity ( $K$  = number of free parameters)

$$AIC = -2\log L + 2K$$

- Corrected version, for when sample sizes are not overly large ( $n/K < 40$ )

$$AIC_C = AIC + (2K[K + 1]) / (n - K - 1)$$

- AIC measures amount of information lost when approximating reality with a model; lower AIC scores are better



# Information Criteria: BIC

- Another criterion is the **Bayesian Information Criterion**, which tends to penalize model complexity more strongly than AIC:

$$BIC = -2 \log L + K \log(n)$$

- BIC can be considered a simple approximation to Bayesian approaches

# Information Criteria: AIC

- AIC and AICc scores are often converted to  $\Delta\text{AIC}$  scores by subtracting from each model the score of the best.
- $\Delta\text{AIC}$  are sometimes converted to **Akaike weights**, which sum to 1 across models, indicating proportional support.

$$\Delta_i = \text{AIC}_i - \min(\text{AIC})$$

$$w_i = \frac{\exp\left(-\frac{1}{2}\Delta_i\right)}{\sum_j \exp\left(-\frac{1}{2}\Delta_j\right)}$$

$\Delta\text{AIC}$	interpretation
< 2	substantial support
4	considerably less support
> 10	essentially no support

# Information Criteria: AIC

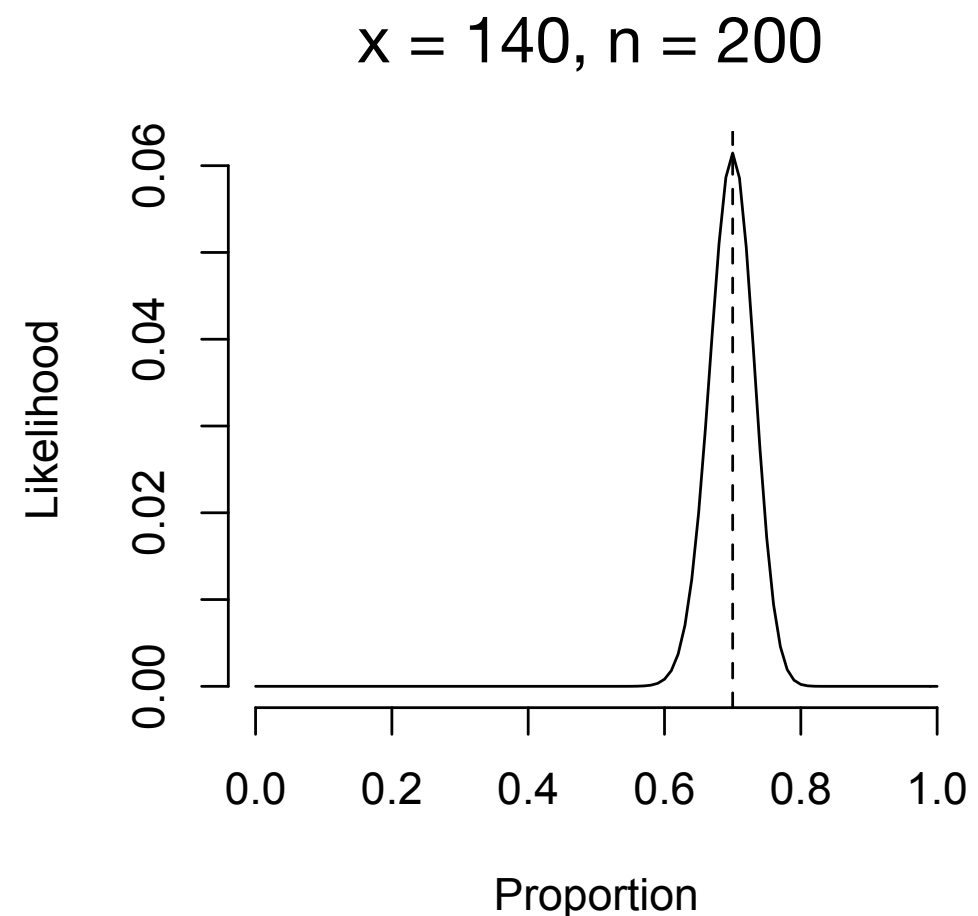
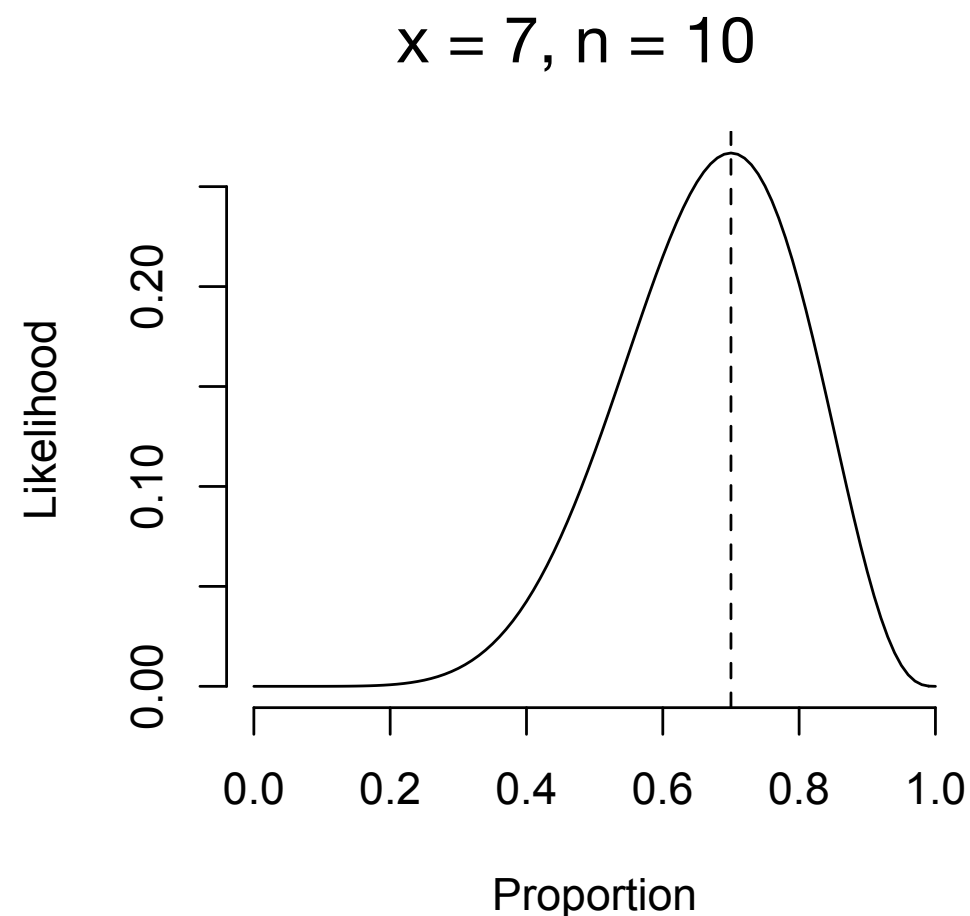
- AIC makes the most sense when all our models are much simpler than reality. If reality is simple enough to be included among our models, Bayesian approaches have better properties
- Because  $\Delta AIC$  determines interpretations, it is the additive difference between AIC scores that matters
- AIC measures relative, not absolute fit. The best model may still be poorly supported.
  - If possible, include a null model in the comparisons
  - Other approaches to model adequacy involve simulation under the null hypothesis and assessing if the observed data deviate from that

# Summary: Coin Toss

model	likelihood	K	AICc	$\Delta$ AICc	Akaike weight
H1: $p = 0.5$	0.117	0	4.29	0	0.61
H2: $p$ free	0.266	1	5.14	0.85	0.39

# Likelihood and Confidence Intervals

There is information in the shape of the likelihood function about precision of a MLE

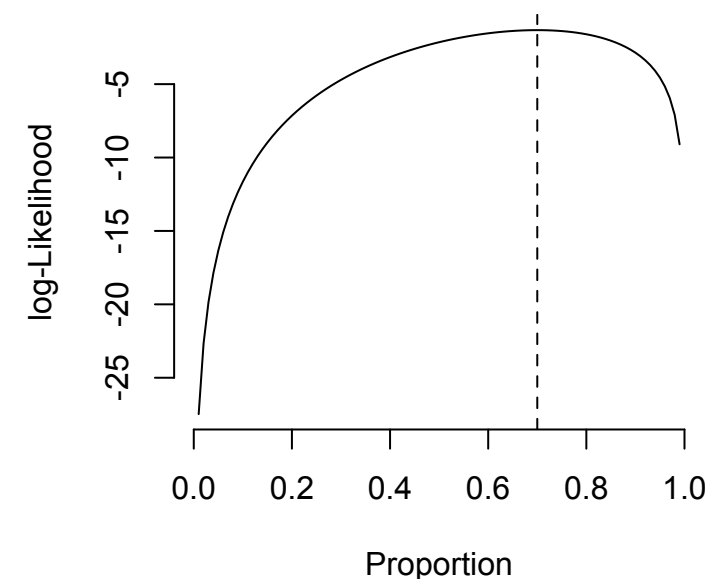


Two different approaches use information on how narrow the likelihood peaks are.

# Wald Confidence Intervals

- The standard error of a MLE is related to the second derivative of the log-likelihood function near the peak
- Can work out analytically, or if not solvable, numerically
- For binomial:  $SE(\hat{p}) = \sqrt{\frac{p(1-p)}{n}}$
- Under assumption of normality, confidence intervals can be computed from the percentiles of the standard normal distribution

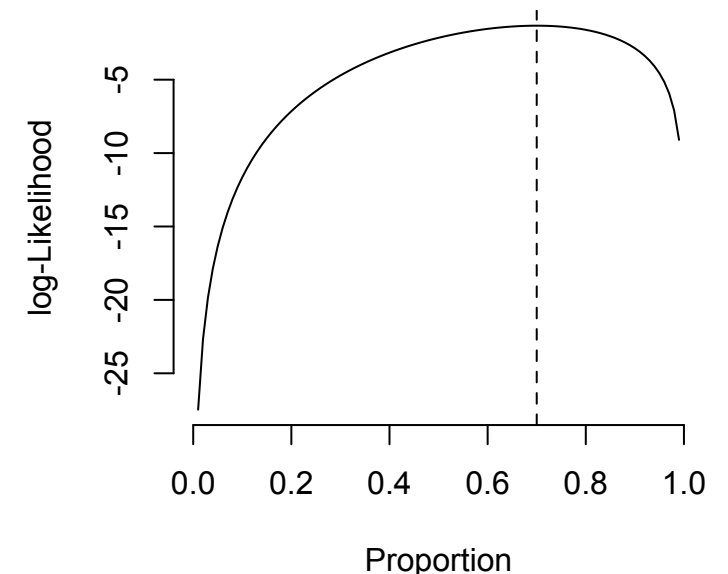
95% CI:  $\hat{p} \pm 1.96 \cdot SE(\hat{p})$



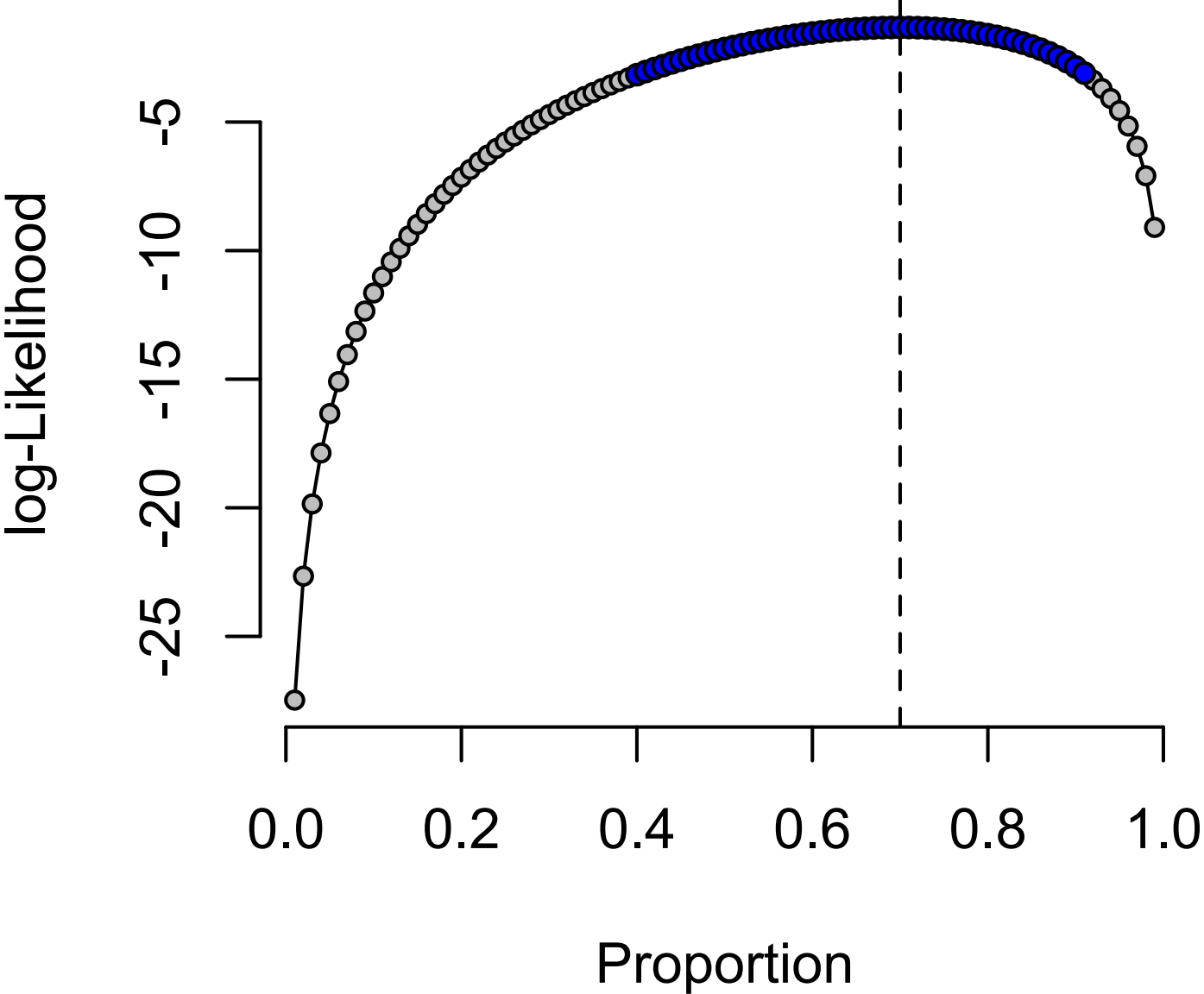
# Likelihood Ratio Confidence Intervals

- Likelihood ratio confidence intervals ( = profile likelihood CI's) can be thought of as applying a LRT to all parameter values and retaining those not significantly different from the MLE as delimiting the CI
- Upper 95% tail of the  $\chi^2$  (df = 1) is 3.84. Since twice the differences in log-likelihoods should be less than this, keep all parameter values with log-likelihoods within 1.92 units of the max log-likelihood

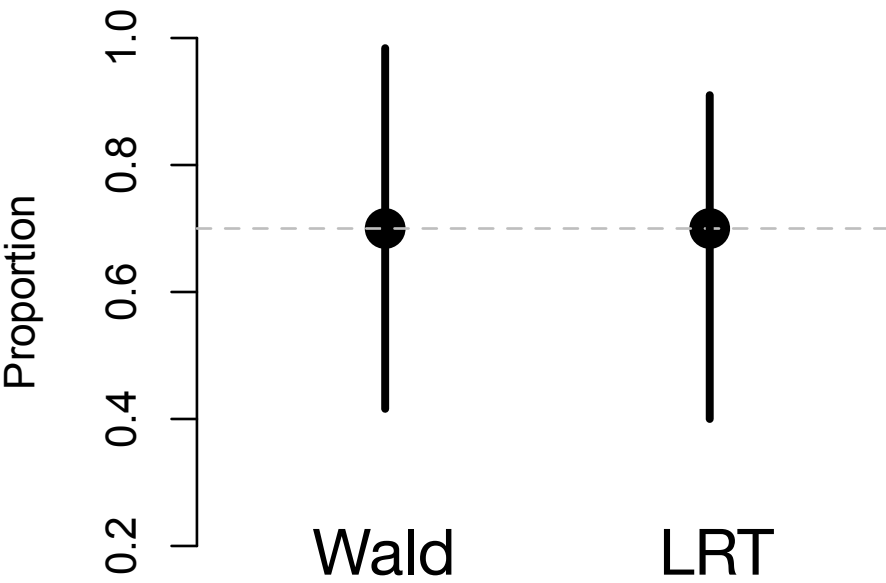
$$\begin{aligned} -2 \log \Lambda &\sim \chi^2 \\ -2 (\log L(H_1) - \log L(H_2)) \end{aligned}$$



# Likelihood Ratio Confidence Intervals

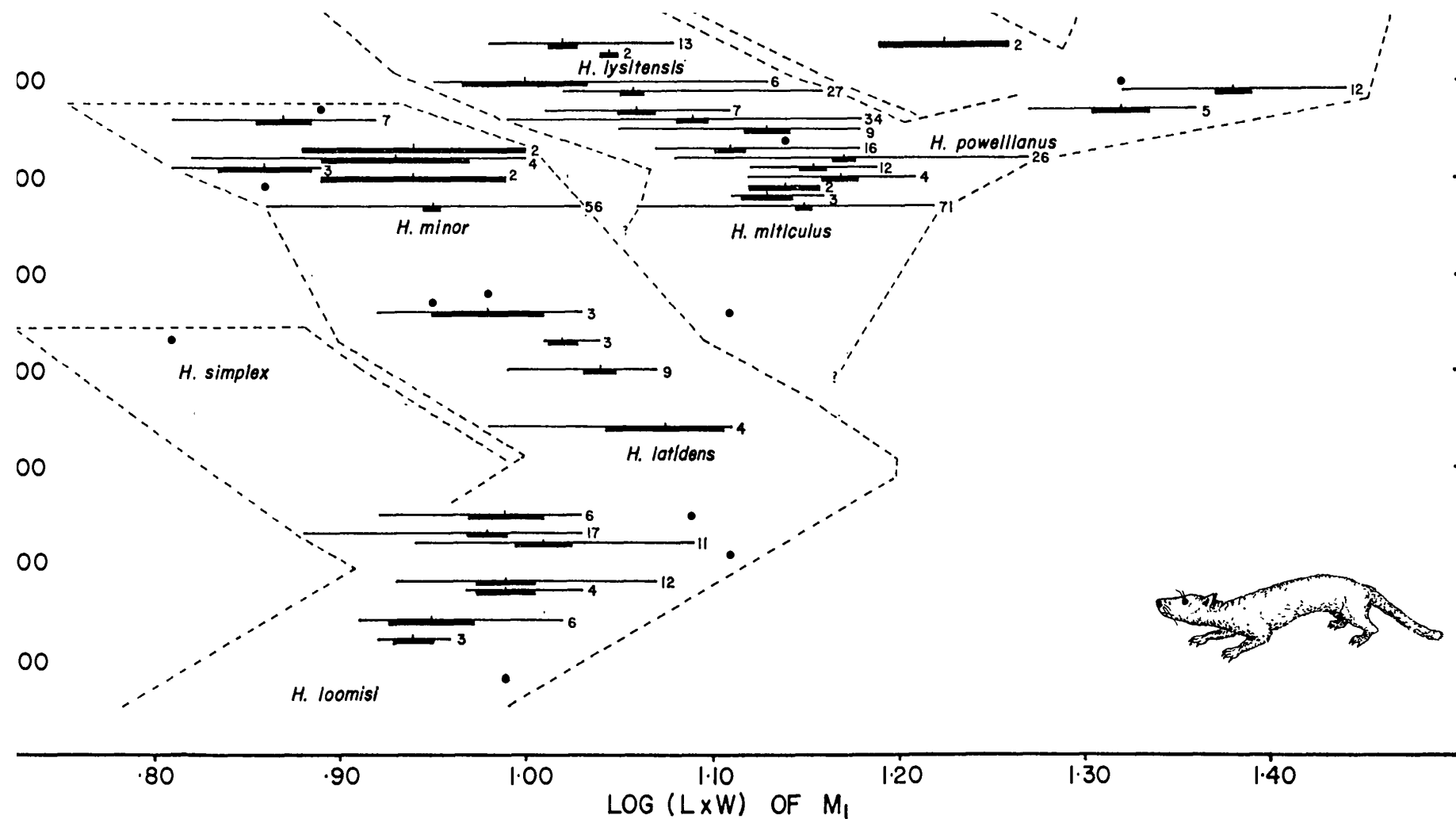


method	95% CI
Wald	(0.42, 0.98)
LRT	(0.40, 0.91)





# Application: Time-series of Traits

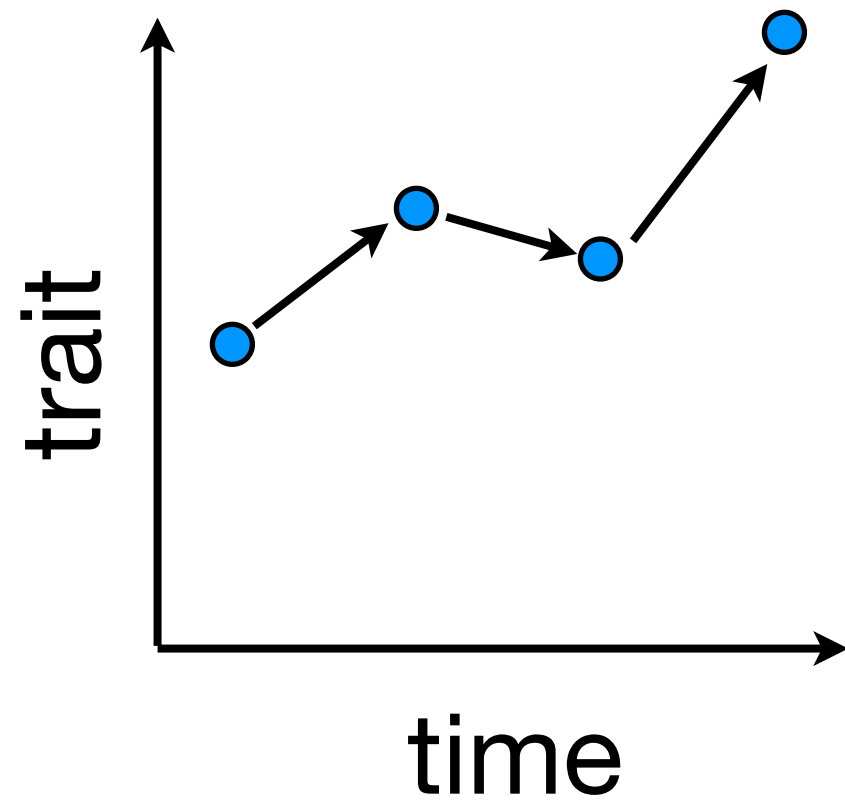


Gingerich (1976)

How should these data be interpreted?

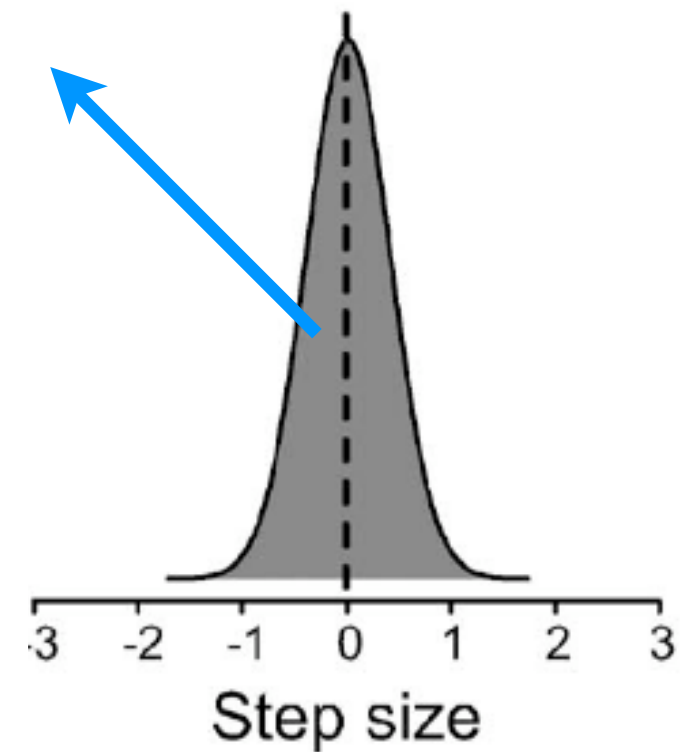
Qualitatively at first, then with null models.

# General Random Walk



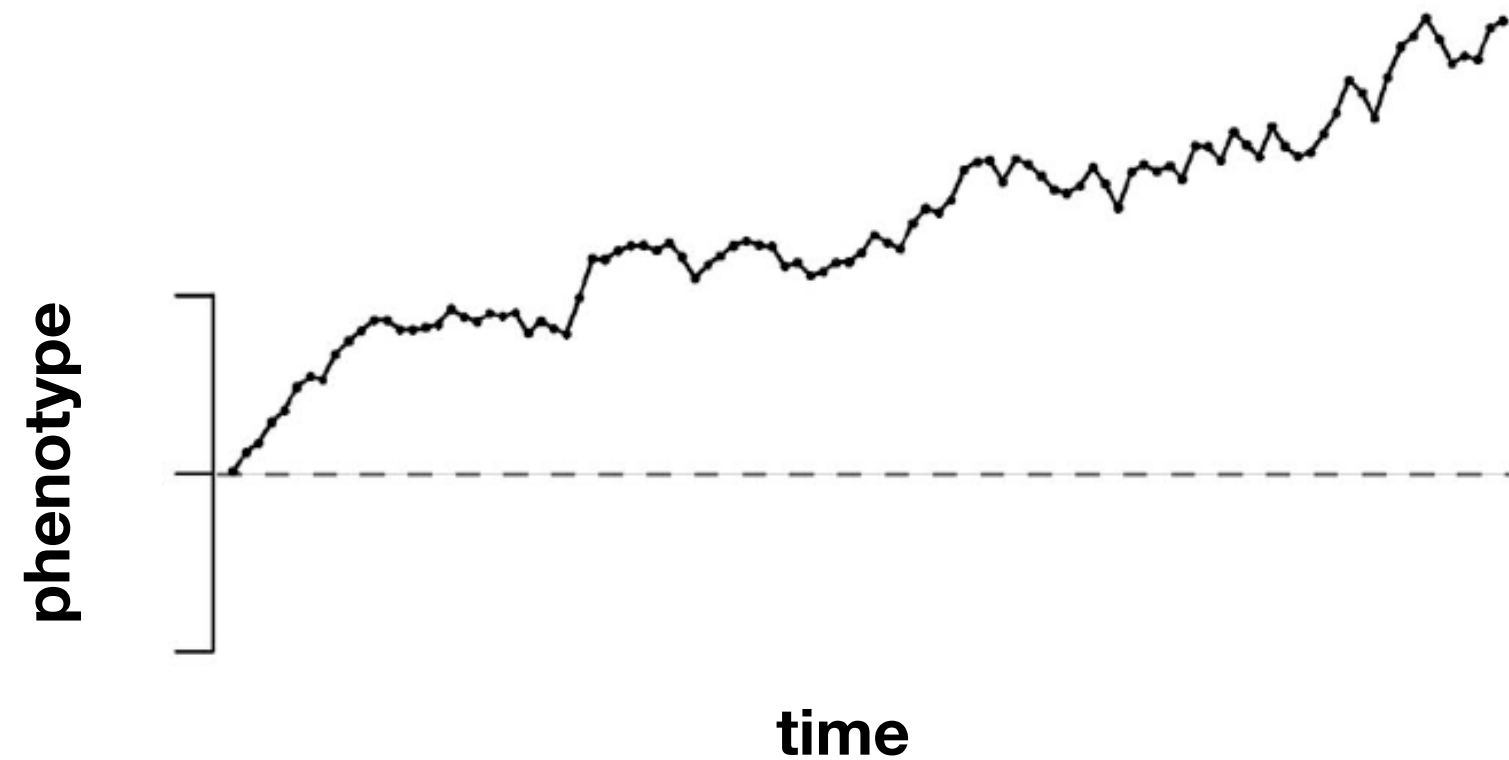
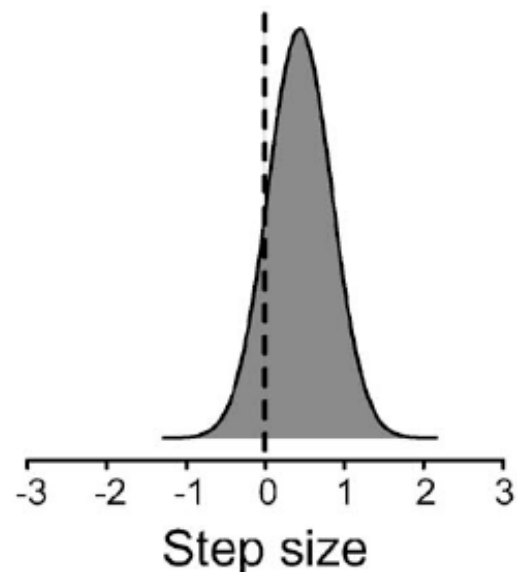
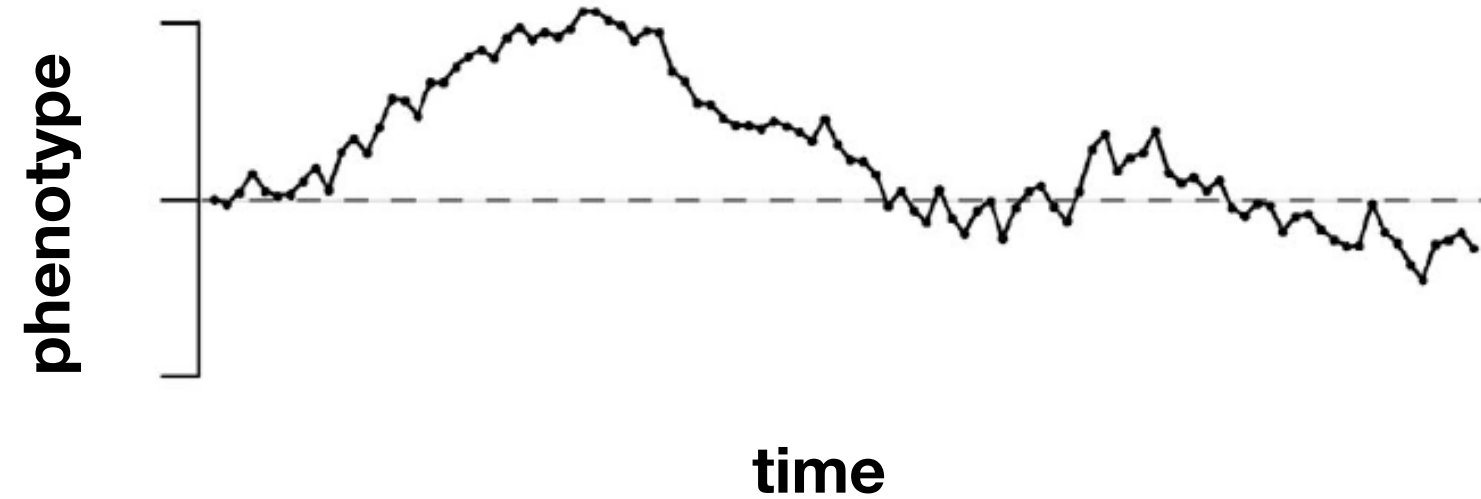
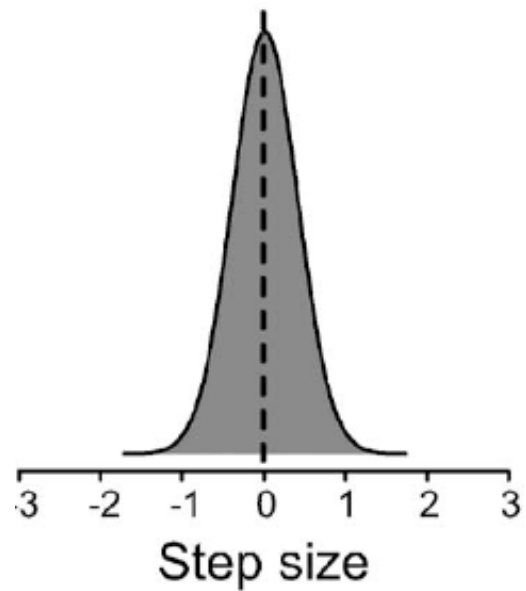
+1.0  
-0.2  
+2.1

step  
distribution

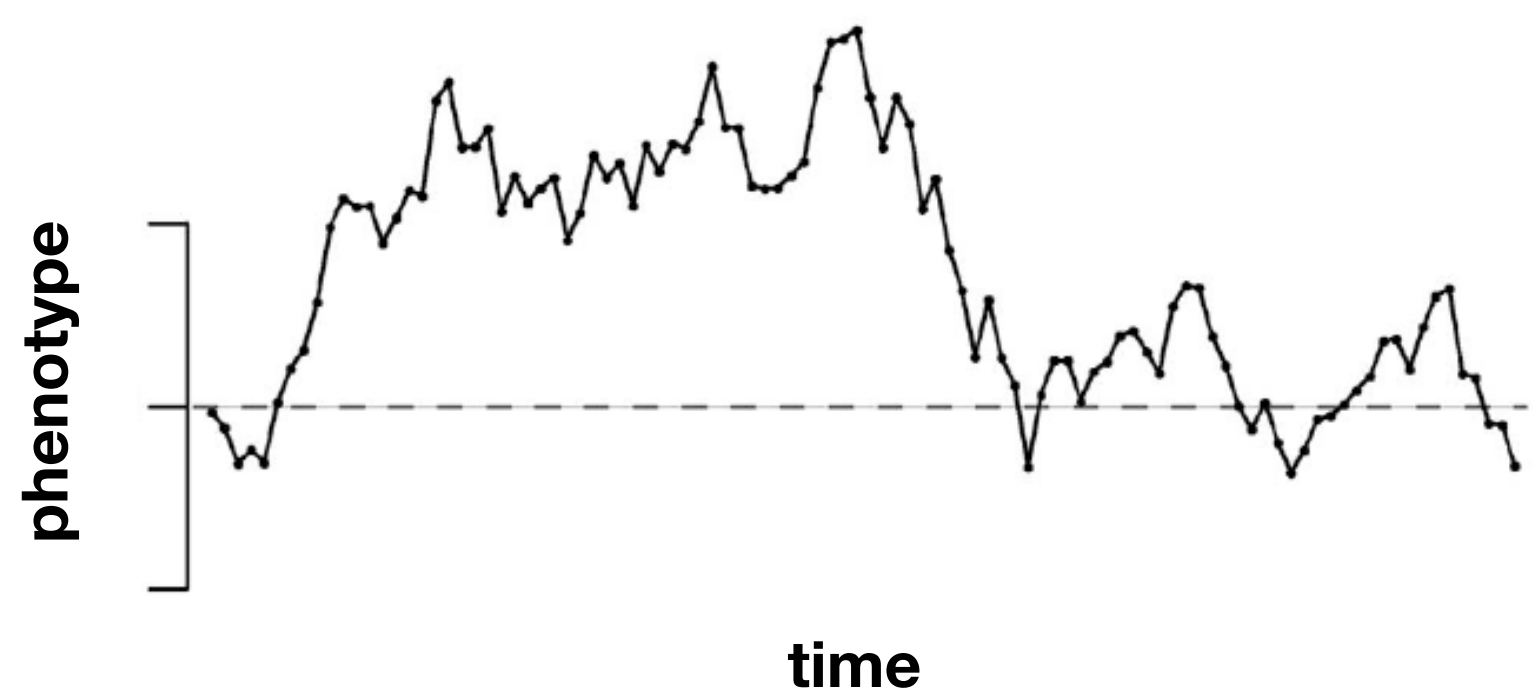
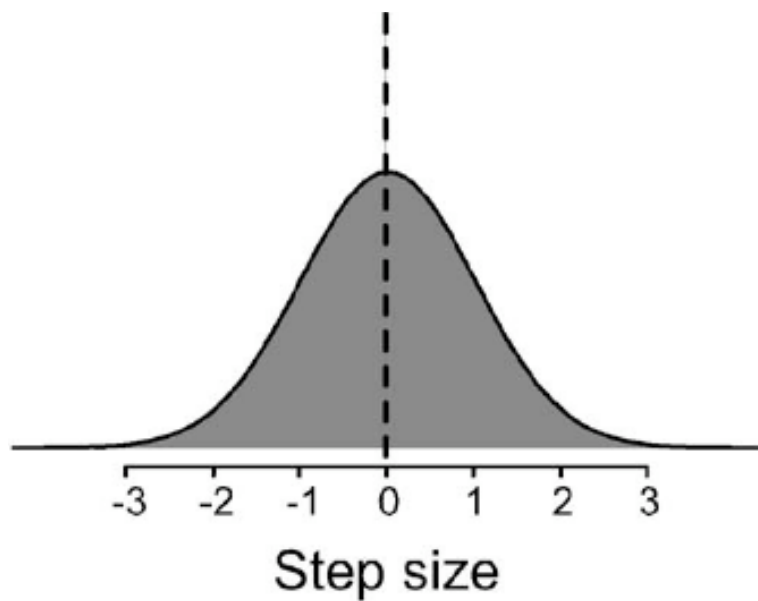
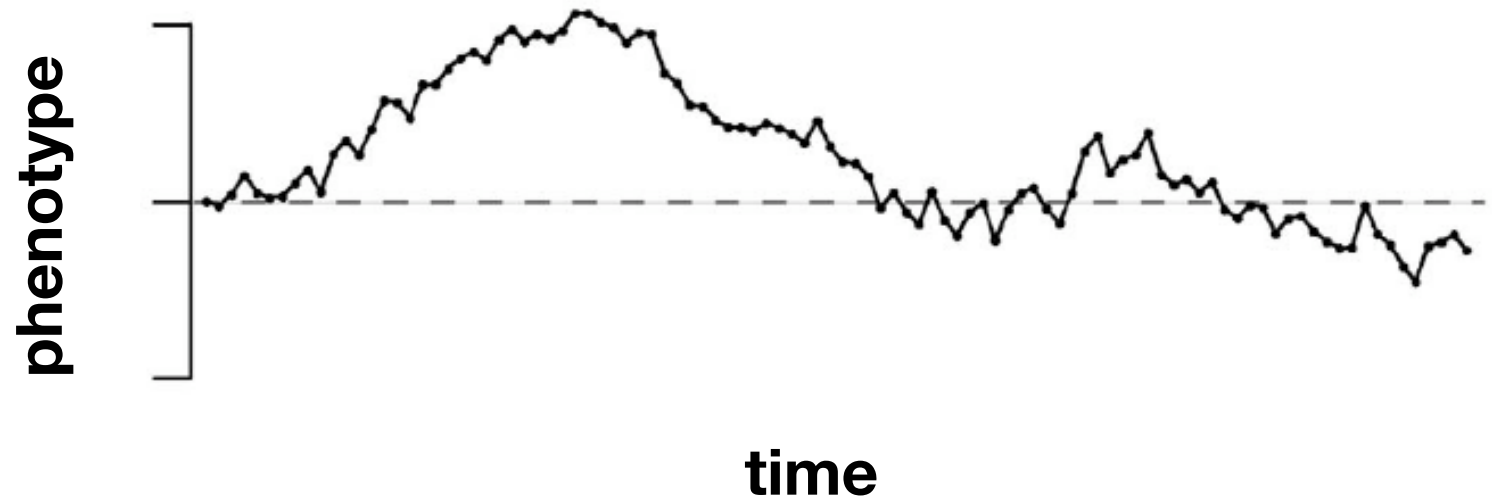
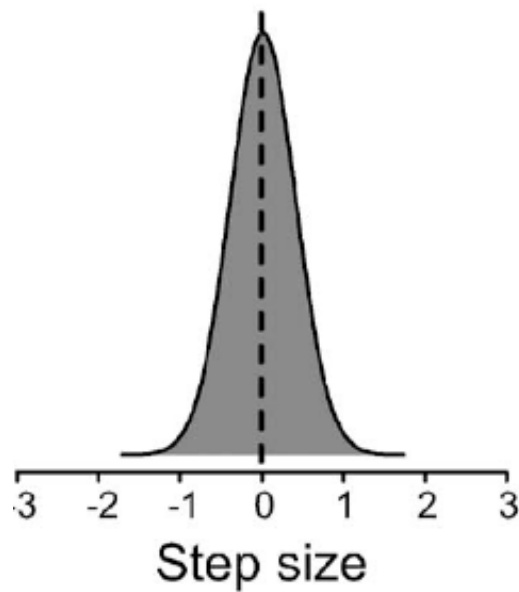


mean ( $\mu_s$ ) = **directionality**  
variance ( $\sigma^2_s$ ) = **volatility**

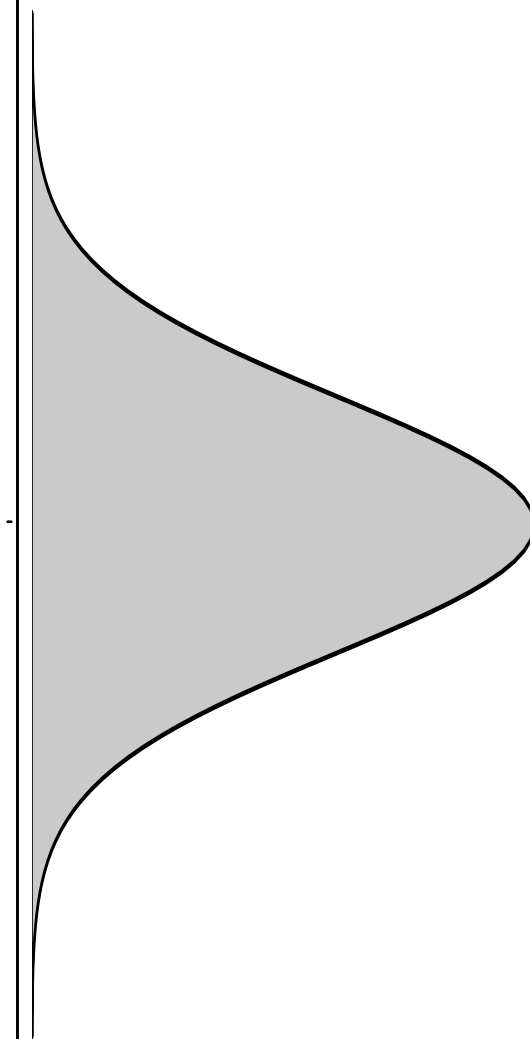
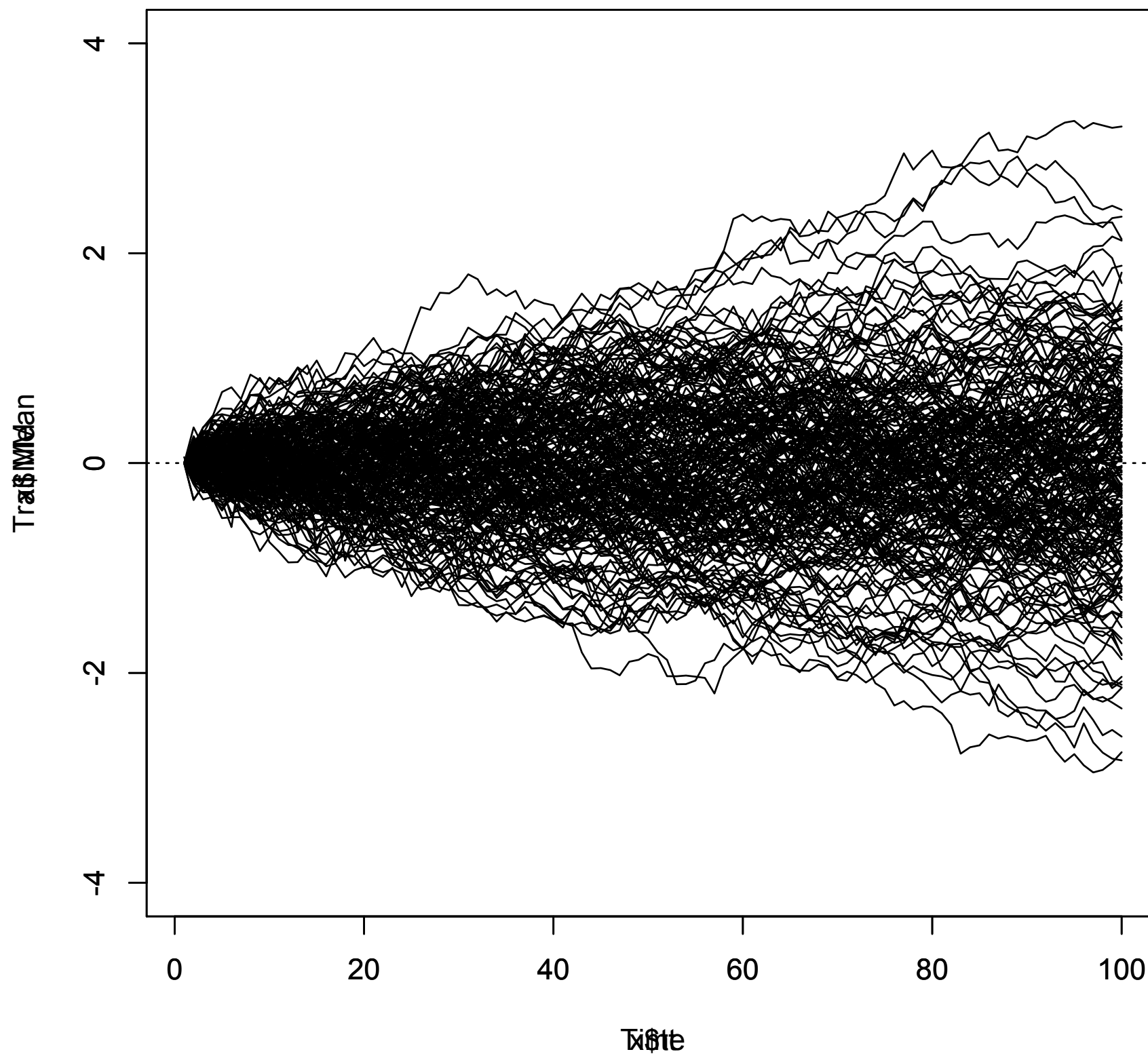
# Mean step ( $\mu_s$ ) $\Rightarrow$ directionality



# Step variance ( $\sigma_s^2$ ) $\Rightarrow$ volatility



# Expected Change over Time



$$\Delta z \sim N(t\mu_s, t\sigma_s^2)$$

# General Random Walk (GRW)

- Same general model can be used for two modes:

Mode	Other names	
Directional Evolution	General RW Directional RW Biased RW	$\mu_s \neq 0$
Random Walk	Symmetric RW Unbiased RW	$\mu_s = 0$

# Summary So Far...

Mode	Distribution of AD differences ( $\Delta z$ )
Directional Evolution	$\Delta z \sim N(t\mu_s, t\sigma_s^2)$
Random Walk	$\Delta z \sim N(0, t\sigma_s^2)$

We want to be able to **fit** these models -- determine best parameter values, and measure of model support

## Likelihood

# Goals

- Use likelihood approaches to get best parameter estimates  $(\mu_s, \sigma_s^2)$ , and to compare fit of directional change and random walk models.
- Likelihood calculations involve the probability density of the Normal distribution

$$Pr(\Delta z) = \frac{1}{\sqrt{2\pi V}} \exp \left( -\frac{(\Delta z - M)^2}{2V} \right)$$

Mode	Distribution of AD differences ( $\Delta z$ )
Directional Evolution	$\Delta z \sim N(t\mu_s, t\sigma_s^2)$
Random Walk	$\Delta z \sim N(0, t\sigma_s^2)$



# Example Calculation

- Example, for General Random Walk:  $M = t\mu_s$ ,  $V = t\sigma_s^2$

$$Pr(\Delta z) = \frac{1}{\sqrt{2\pi V}} \exp \left( -\frac{(\Delta z - M)^2}{2V} \right)$$

$$L(M_{RW}; \Delta z) = \frac{1}{\sqrt{2\pi t\sigma_s^2}} \exp \left( -\frac{(\Delta z - t\mu_s)^2}{2t\sigma_s^2} \right)$$

$$\log L(M_{RW}; \Delta z) = \frac{1}{2} \log(2\pi t\sigma_s^2) - \frac{(\Delta z - t\mu_s)^2}{2t\sigma_s^2}$$

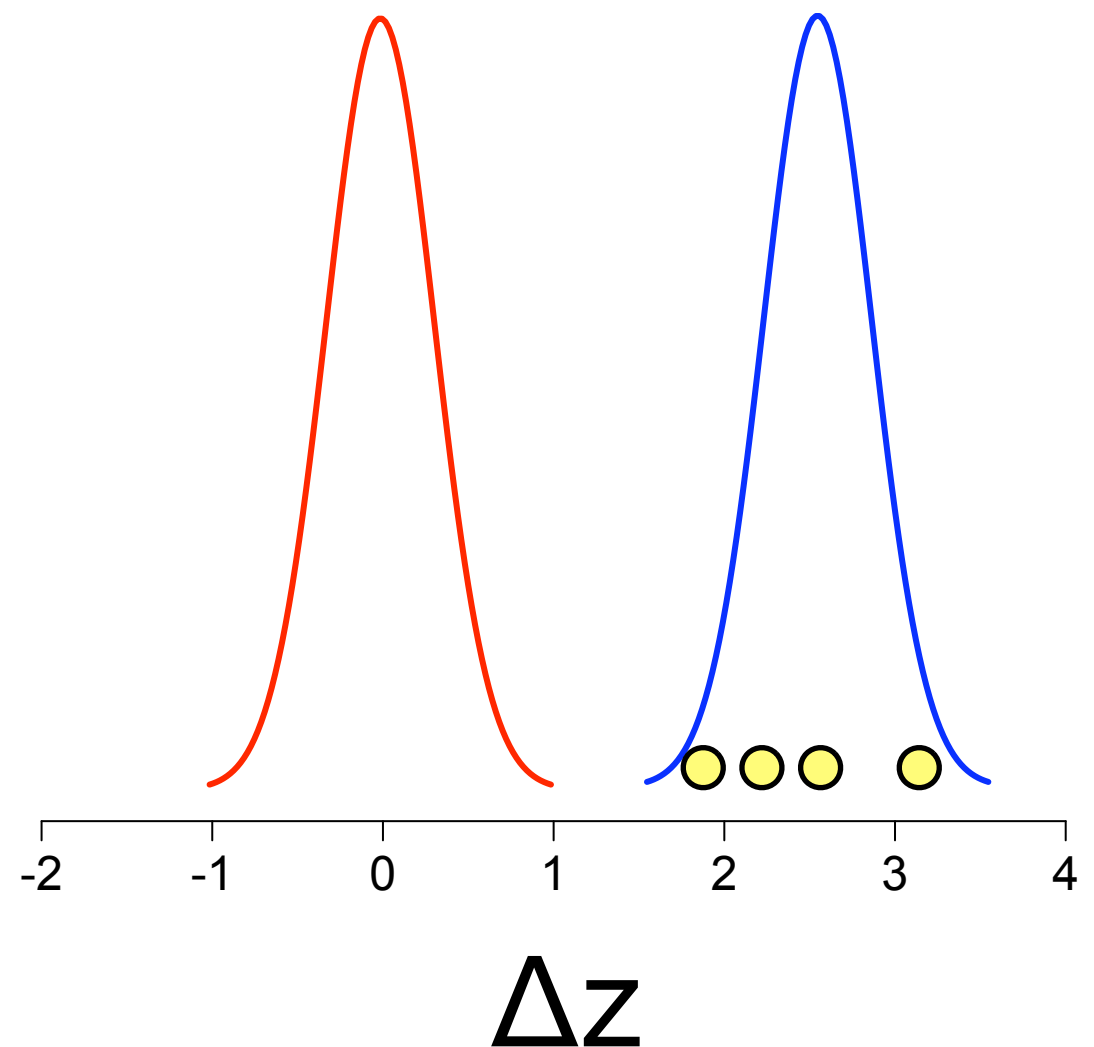
- For multiple  $\Delta z_i$ , simply sum  $\log L$  over all  $\Delta z_i$

# Example Calculation

Assume  $t = 10$  steps,  
compare two sets of  
parameter estimates for  
Directional model:

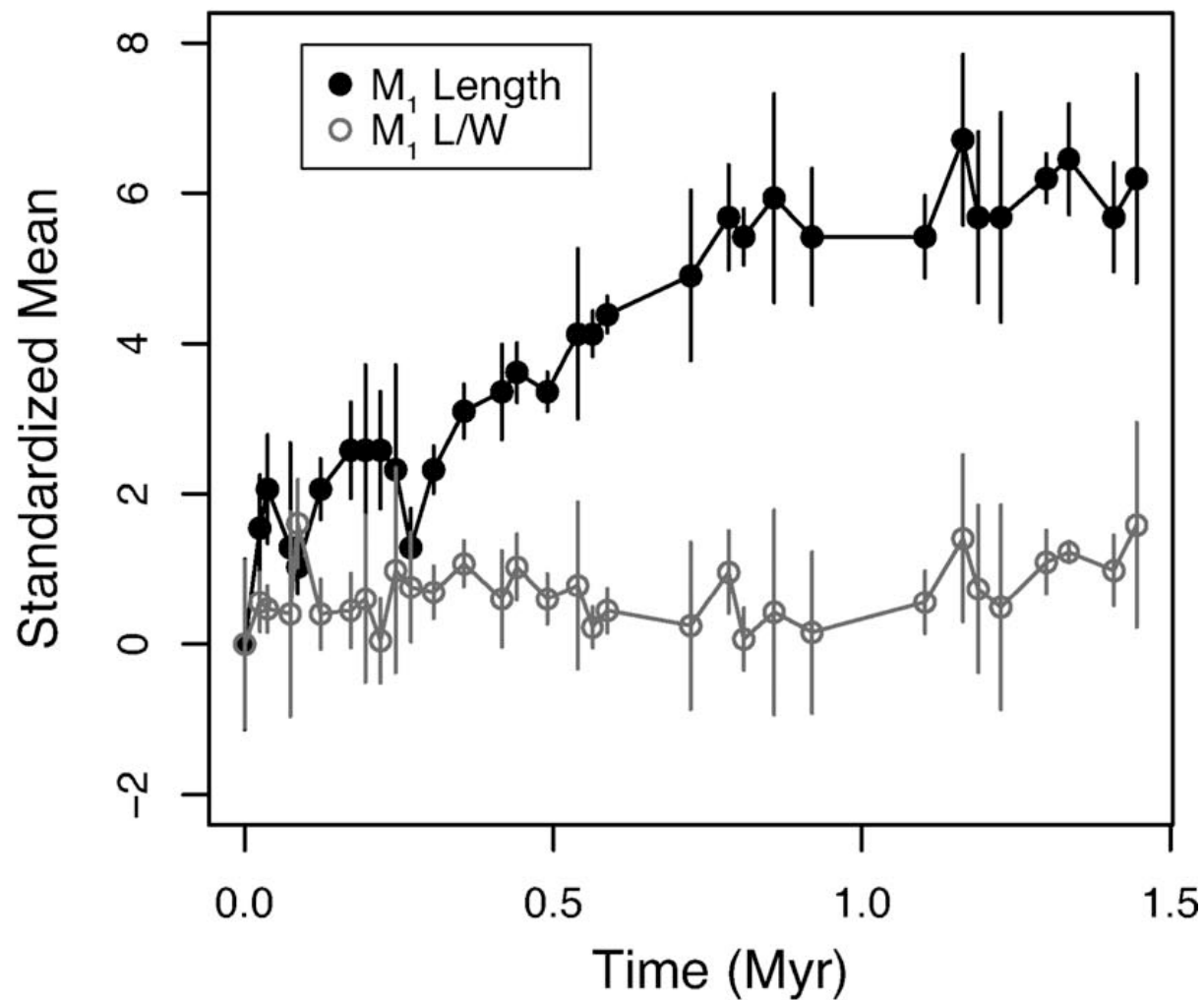
$$\begin{array}{ll} \mu_s = 0 & \mu_s = 0.25 \\ \sigma_s^2 = 0.01 & \sigma_s^2 = 0.01 \end{array}$$

$$N(0, 0.1) \quad N(2.5, 0.1)$$

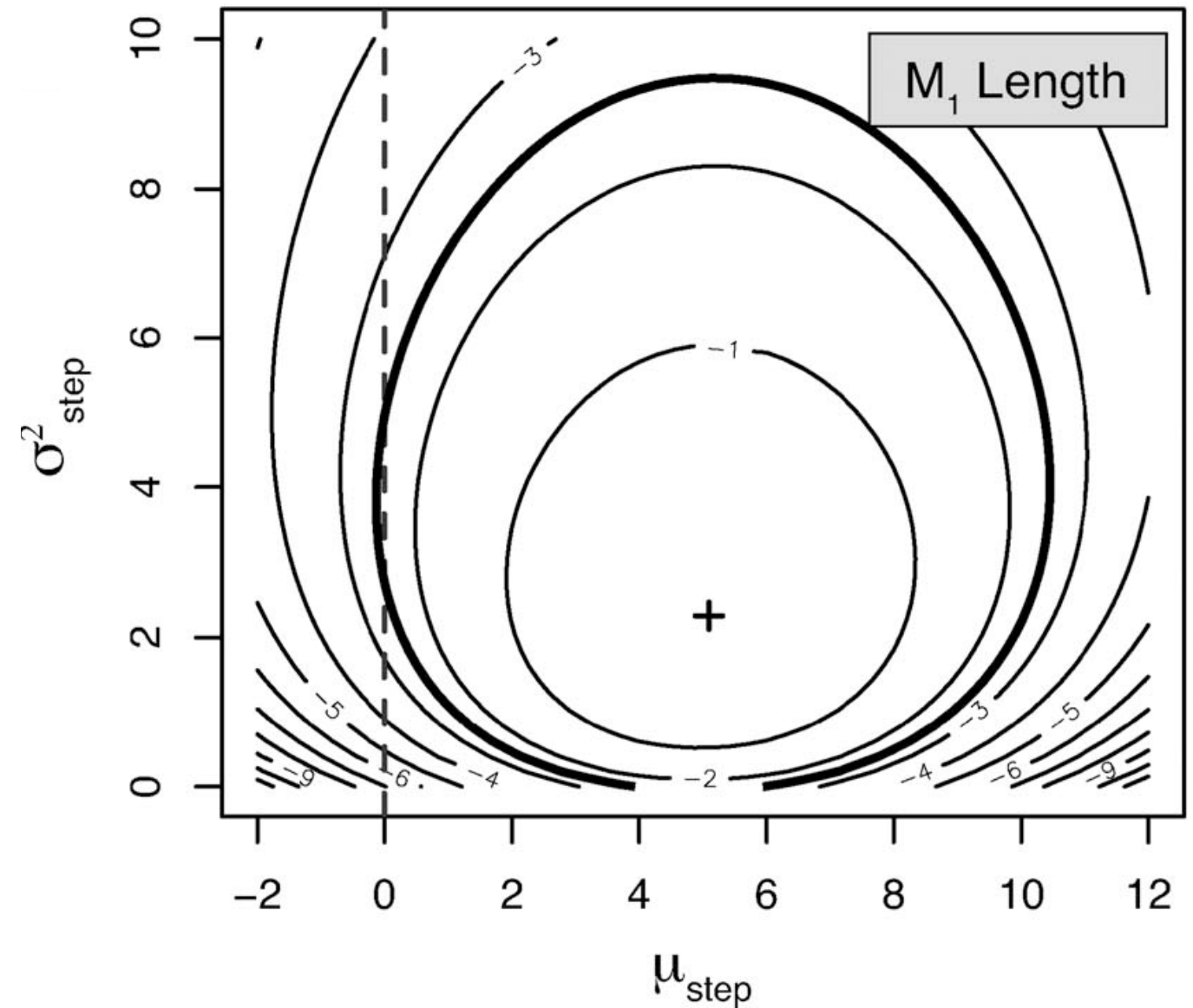


$$\Delta z \sim N(t\mu_s, t\sigma_s^2)$$

# Log-likelihood Surface



data from:  
*Clyde & Gingerich (1994)*



Use hill-climbing algorithm to  
find peak (= MLE)

## Exercise 5. Likelihood

1. Let's look at an example time-series analysis using the functions in the `paleoTS` package. First, install the package if necessary and then load it with the `library` function. Next, read in Gingerich's *Cantius* data with `cantius <- read.paleoTS("Cantius_lowerL.txt")`. If this function is called without any argument, the user is prompted to navigate to the file location. First plot the data. Finally, fit some models to these data with the function `fit3models(cantius)`. Take a look at the printed results. Which model is best, and which is the worst? How do you interpret the strength of evidence favoring the best model? Does this match the visual impression of the data?
2. Return again to the two regressions you fit to the cope data yesterday (`valve.length ~ mg.temp`, `valve.length ~ mg.temp + depth`). Re-run them from your script. Add an additional model that just has an intercept (`valve.length ~ 1`). Use the `AIC()` function to get the AIC score for each model, and then compute  $\Delta\text{AIC}$  values. How do you interpret these results in terms of the support of these models?
3. Figure out how to determine the LRT confidence interval for the coin tossing example. Start with a set of proportions to evaluate: `pp <- seq(0.01, 0.99, 0.01)`. Compute the log-likelihood scores from these using `dbinom()`. Finally, figure out how to select all the `pp` that are within 1.92 units of  $\log L$  of the maximum.

## *Exercise 6. Additional Exercises*

1. Here you will expand on the study of Type II error from Exercise 2-3. First, take your work and turn it into a function that returns the probability of Type II error. Have this function accept as arguments the means, standard deviations, and sample sizes of samples 1 and 2, as well as the number of replications to perform. Set a sensible default for this last argument.
2. Use your new function from question 1 to explore Type II error with respect to changing sample size for a range of  $n$ 's from 10 to 100. Keep all other parameters as they were for Exercise 2-3. Plot the probability of Type II error with respect to sample size.
3. Do a similar study as question 2, but keep sample sizes at 20 and instead vary the mean of sample 2 from 10.1 to 12. Plot the probability of Type II error as a function of the mean of sample 2. Note that statistical power is equal to  $1 - \text{Pr}(\text{Type II error})$ , so you can plot power from these results instead.

## Exercise 6. Additional Exercises (continued)

4. There are some ideas that the permutation test we did in Ex. 4-5 may be less appropriate than what is called a two-sample bootstrap test, which you will perform now. For this, the data from both samples are combined under the null hypothesis that there is no difference between the groups. Next, with each replication, a bootstrap sample is drawn from these combined data for each treatment (species, in this case); these bootstrap samples must be of the same size as the original samples. Otherwise, the procedure is the same as the permutation test.
5. There is a file in the Dropbox folder called “Fossil positions.csv” that gives the position in two dimensions (x, y) of 50 fossils found in life position on a feeding plane. Let’s call them oysters (the data are simulated). Find this file and read it in to R. Plot their positions. Does it appear that these fossils are clumped, random or overdispersed spatially?
6. We are going to test your observation using something that is called *nearest neighbor distance (NND)*. The nearest neighbor to each fossil is the other fossil that is closest in space to it. This mean of such NN distance is sometimes used as a test statistic to detect deviations from a uniform distribution. The first step is to make a function that, for an assemblage of points, computes the NND for each point and returns their mean. You’ll want to use the `dist()` function to get the pairwise distances and then convert it to a matrix for easier manipulation. You can get the NND for each specimen using `apply` on this matrix, but you have to deal with the fact that there is zero distance between each specimen and itself; the `diag()` function can help with this.
7. Mean NND has some known theoretical properties, but there are complications in applying them, so we will use parametric bootstrapping to generate its null distribution under a uniform distribution of fossils. Make a function that accepts as an argument an observed distribution of fossils and returns this null distribution. Note that the simulation of the uniform distribution should use for the min and max arguments the observed min and max of the fossils in the x and y dimensions.
8. Is the observed mean NND higher or lower than is typical for the null? Does this mean that the sample is clumped or overdispersed compared to the uniform null? Is the difference significant?

**Exercise answers follow**

## Exercise Answers

Ex 1: (1) 10/36 # from counting outcomes (2) `rnorm(n = 50, mean = 100, sd = 10)` (3) `pnorm(q = 12, mean = 10, sd = 1); pnorm(q = 12, mean = 10, sd = 1, lower.tail=FALSE)` (4) `qnorm(p = c(0.25, 0.75), mean = 10, sd = 1)`

Ex. 2: `x1 <- rnorm(50); x2 <- rnorm(50); t.test(x1, x2)` (2, 3) See following. (4) Prob of Type II is ~0.14 with  $n = 20$ , ~0.44 when  $n = 10$ , and ~0.001 when  $n = 50$ .

Ex. 3: (1) See following (2) `sd(mu_hat)` # very close to  $s/\sqrt{N}$

Ex. 4: (1) See following (2)(3)(4) See following

Ex. 5: (1) `library(paleoTS); cant <- read.paleoTS("Cantius_lowerL.txt"); fit3models(cant)` # GRW model is best, substantially better than URW; support for stasis is negligible. (2) See following. (3) See following

Ex. 6: See following pages for all



```
# test Type I error (Ex. 2-2)
```

```
nrep <- 1000
NN <- 50 # get same result, regardless of NN
pv<- array(dim = nrep)
for(i in 1:nrep){
  x1<- rnorm(NN)
  x2<- rnorm(NN)
  w<- t.test(x1, x2)
  pv[i]<- w$p.value
}

hist(pv, col="tan") # P-values are uniform (0, 1)
num.sig <- sum(pv <= 0.05) # counts # significant
print(num.sig/nrep) # about 5%, same as alpha
```

```
# test Prob of Type II error (Ex. 2-3)
```

```
nrep <- 1000
mu1 <- 10
mu2 <- 11
s1 <- 1
s2 <- 1
NN <- 20
pval <- array(dim = nrep)
for(i in 1:nrep){
  x1 <- rnorm(NN, mu1, s1)
  x2 <- rnorm(NN, mu2, s2)
  w <- t.test(x1, x2)
  pval[i] <- w$p.value
}
probII <- sum(pval > 0.05) / nrep
```

```
# Standard error of the mean (Ex. 3-1)
```

```
nrep <- 10000
N <- 100
ss <- 1
mu_hat<- array(dim = nrep)
for(i in 1:nrep){
  x <- rnorm(N, mean = 0, sd = ss)
  mu_hat[i] <- mean(x)
}
hist(mu_hat, col="tan")
abline(v = 0, lwd = 3, col="blue") # true mean in simulations
print(mean(mu_hat)) # very close to true value of zero
```

```
## P-value and effect size are not the same (Ex. 3-3)
```

```
NN <- 2000
x <- rnorm(NN)
y <- x + rnorm(NN, 0, 10)
plot(x, y, pch = 21, col = "white", bg = "tan")
w <- lm(y ~ x)
summary(w) # highly significant!
abline(w) # but slope very low
abline(h = 0, lty = 3)
```

```
# RMA and RMAboot functions (Ex. 4-1)
```

```
RMA<- function(x,y)
{
  # compute needed summary statistics
  mx <- mean(x)
  my <- mean(y)
  sx <- sd(x)
  sy <- sd(y)
  rxy <- cor(x,y)

  # compute slope and intercept
  b1 <- sy / sx * sign(rxy)
  b0 <- my - b1 * mx

  # combine slope and intercept into a vector
  result <- c(b0,b1)
  return(result)      # note abline(result) works!
}
```

```
RMAboot<- function(x, y, nrep = 1000, alpha = 0.05)
# compute bootstrap SE on RMA slope
{
  w <- RMA(x,y)
  N <- length(x)
  boot.slope<- array(dim=nrep) # array to hold bootstrap slopes

  for (i in 1:nrep)
  {
    ii <- sample(1:N, replace=T) # bootstrap sample of the indices
    wb <- RMA(x[ii], y[ii])      # RMA of bootstrap sample
    boot.slope[i]<- wb[2]         # second element is the slope
  }

  se <- sd(boot.slope) # standard error is SD of bootstrap sample
  ci <- quantile(boot.slope, probs = c(alpha/2, 1 - alpha/2))

  result<- list(estimates = w, se = se, ci = ci)

  return(result)
}
```

```
# Permutation test (Ex. 4-2, 4-3, 4-4)
```

```
# Ex 4-2
pin <- cope$valve.length[cope$species == "pintoi"]
sp4 <- cope$valve.length[cope$species == "species4"]
t.test(pin, sp4)

# Ex. 4-3
pinsp4 <- c(pin, sp4)
gg <- c(rep("pin", 15), rep("sp4", 8))
wt <- t.test(pinsp4 ~ gg)
tobs <- wt$stat

# Ex. 4-4
stripchart(pinsp4 ~ gg, group.names = c("pintoi", "species4"))

# Ex. 4-5
nrep <- 1000
res <- array(dim = nrep)
for(i in 1:nrep){
  ggp <- sample(gg) # by default, sample permutes
  wt.p <- t.test(pinsp4 ~ ggp)
  res[i] <- wt.p$stat
}

# compute p-value from permuted results more extreme
# than observed t -- in both directions!
# tobs is negative
num.extreme <- sum(res < tobs) + sum(res > -tobs)
p.value <- num.extreme/nrep # very similar to t.test
```

```
# Regression and AIC (Ex. 5-2)
```

```
# cope already read in
w.0 <- lm(valve.length ~ 1, data = cope) # intercept only model
w.t <- lm(valve.length ~ mg.temp, data = cope)
w.td <- lm(valve.length ~ mg.temp + depth, data = cope)
a.0 <- AIC(w.0)
a.t <- AIC(w.t)
a.td <- AIC(w.td)
aa<- c(a.0, a.t, a.td) # combine all AIC into a vector
names(aa) <- c("intercept", "mg.temp", "mg.temp+depth")
Daa <- aa - min(aa) # delta AIC
```

```
## LRT CIs (Ex. 5-3)
```

```
# x = 7, n = 10, for logL CI
pp <- seq(0.01, 0.99, 0.01)
logl <- dbinom(7, 10, prob=pp, log = TRUE)
plot(pp, logl, xlab = "Proportion", ylab = "log-likelihood", type="l")
abline(v=0.7, lty=2)
```

```
# want all logL within 1.92 units of max logL
yys <- yy - max(yy) # rescale so max logL is zero
good <- yy > -1.92 # want all those with rescaled logl > -1.92
ci.lrt <- range(xx[good]) # reports the range of these points = CI
points(xx[!good], yy[!good], pch=21, cex=0.7, bg="grey")
points(xx[good], yy[good], pch=21, cex=0.8, bg="blue")
```

```
## Ex. 6-1, 6-2, 6-3: Expanded power analysis
```

```
probTypeII <- function(m1, m2, s1, s2, n1, n2, nrep = 1000){
  pval<- array(dim=nrep)
  for(i in 1:nrep){
    x1 <- rnorm(n1, m1, s1)
    x2 <- rnorm(n2, m2, s2)
    w <- t.test(x1, x2)
    pval[i] <- w$p.value
  }
  probII <- sum(pval > 0.05) / nrep # Pr(type II)
  return(probII)
}
```

```
## Ex. 5-2
```

```
## Get series of Pr(type II) with varying n
```

```
m1 <- 10
m2 <- 11
s1 <- 1
s2 <- 1
Nvec <- seq(10, 100, 10)
beta <- array(dim = length(Nvec))
for(i in 1:length(Nvec)){
  beta[i] <- probTypeII(m1, m2, s1, s2, Nvec[i], Nvec[i])
}
plot(Nvec, beta, type="b", xlab = "n", ylab = "Pr(Type II)")
```

```
## Ex. 5-3
```

```
## Instead, now vary m2
```

```
m1 <- 10
m2vec <- seq(10.1, 12, 0.1)
s1 <- 1
s2 <- 1
n1 <- 20
n2 <- 20
beta <- array(dim = length(m2vec))
for(i in 1:length(m2vec)){
  beta[i] <- probTypeII(m1, m2vec[i], s1, s2, n1, n2)
}
plot(m2vec, beta, type="b", xlab = "m2", ylab = "Pr(Type II)")
```

```
## Two sample bootstrap (Ex. 6-4)
## assumes code for Ex. 4-2 has been run

npin <- length(pin)
nsp4 <- length(sp4)

nrep <- 1000
res <- array(dim=nrep)
for(i in 1:nrep){
  b.pin <- sample(pinsp4, size = npin, replace = T)
  b.sp4 <- sample(pinsp4, size = nsp4, replace = T)
  wt.b <- t.test(b.pin, b.sp4)
  res[i] <- wt.b$stat
}
# compute p-value from permuted results more extreme
# than observed t -- in both directions!
# tobs is negative
num.extreme <- sum(res < tobs) + sum(res > -tobs)
p.value <- num.extreme/nrep
### resulting p very similar to t-test and permutation
```

```
# Testing spatial clumping using NND (Ex. 6-5, 6-6,
6-7, 6-8)
# read in data
xy <- read.csv("Fossil positions.csv")
plot(xy) #looks clumped
```

```
# NND continued

# function to return average nearest-neighbor (NN) distance for a set of
points
meanNN <- function(xy){
  dd <- dist(xy) # pairwise distance matrix between points
  ddm <- as.matrix(dd) # convert to matrix
  diag(ddm) <- 1e6 # need to change diagonal so don't see NN as self
  minD <- apply(ddm, 2, min) # distance to NN for each point
  mnNN <- mean(minD) # average over all NN's
  return(mnNN)
}

genNNnull <- function(xy, nrep = 1000){
  # get min and maxes for simulated uniform distribution
  xmin <- min(xy[,1])
  xmax <- max(xy[,1])
  ymin <- min(xy[,2])
  ymax <- max(xy[,2])
  N <- nrow(xy)

  mNN.b <- array(dim = nrep)
  for (i in 1:nrep){
    xb <- runif(N, xmin, xmax)
    yb <- runif(N, ymin, ymax)
    xy.b <- cbind(xb, yb)

    mNN.b[i] <- meanNN(xy.b)
  }
  return(mNN.b)
}

# analyze given data set
NN.obs <- meanNN(xy)
NN.null <- genNNnull(xy)
hist(NN.null, col="grey", main = NA, xlab = "Null distrib. of NN")
ci <- quantile(NN.null, prob = c(0.025, 0.975))
abline(v = ci, lty= 2, lwd = 2)
arrows(NN.obs, 200, NN.obs, 100) # signif. bc outside of middle 95% of null
```