S&DS 101
Intro Statistics: Life Sciences

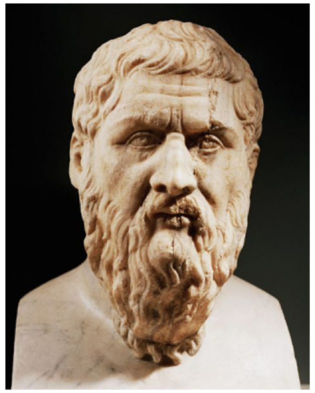
Overview

Quick review using the binomial and normal null distributions

For loops

Permutation tests for comparing 2 means

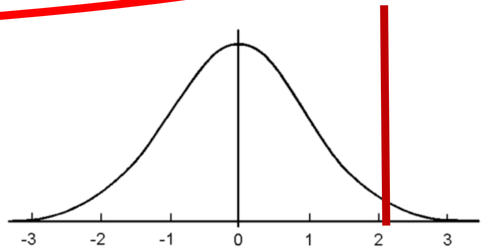
Five steps of hypothesis testing



1. State H_0 and H_A
   - Assume Gorgias (H_0) was right

2. Calculate the actual observed statistic

3. Create a distribution of what statistics would look like if Gorgias is right
   - Create the **null distribution** (that is consistent with H_0)



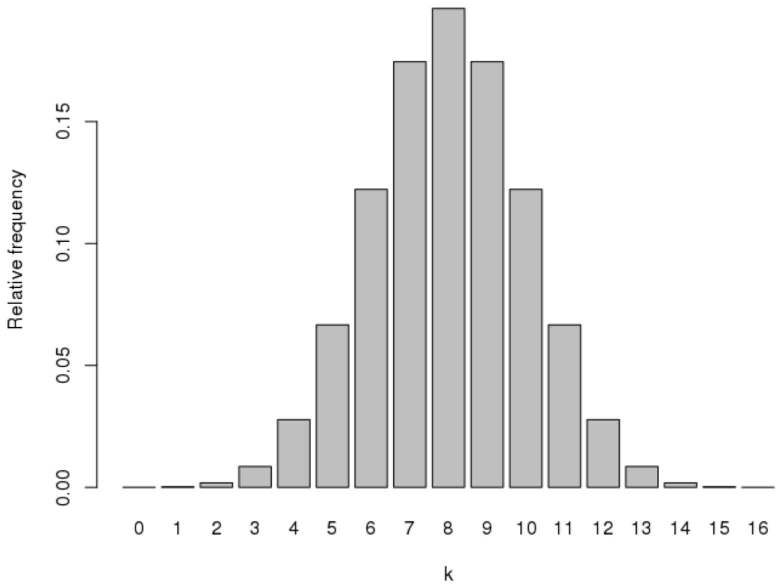
4. Get the probability we would get a statistic more than the observed statistic from the null distribution
   - p-value

5. Make a judgement
   - Assess whether the results are statistically significant

Binomial functions for the null distribution

$$Pr(X = k) =$$

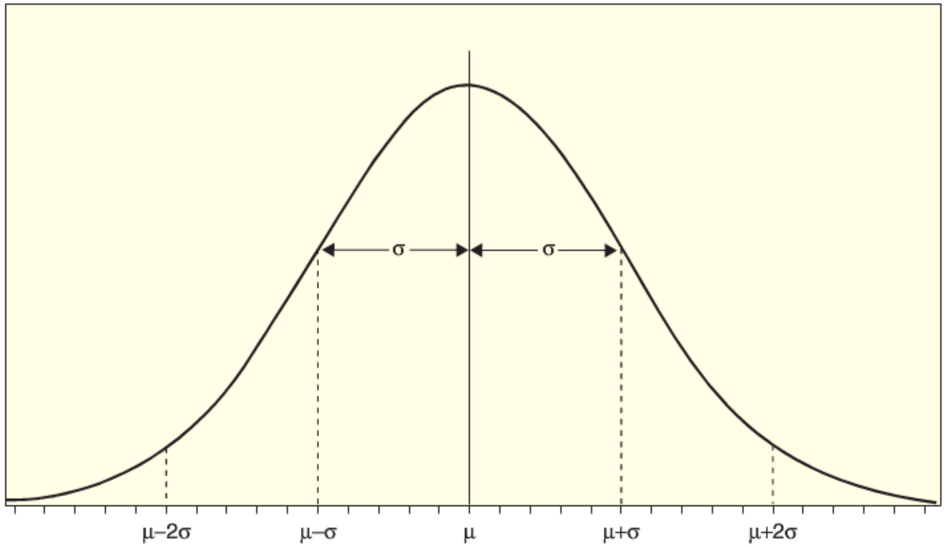
$$\binom{n}{k}\pi^k(1-\pi)^{n-k}$$



1. rbinom(): generate random numbers from a binomial distribution
2. dbinom(): create the binomial density function Pr(X = k; n, π)
3. pbinom(): create the cumulative distribution function Pr(X ≤ k; n, π)

Normal density as an approximation to the binomial density function

$$f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



$\mu = \pi_0$

$$\sigma_{\hat{p}} = \sqrt{\frac{\pi_0(1-\pi_0)}{n}}$$

1. rnorm(): generate random numbers from a normal distribution
2. dnorm(): create the normal density function $f(x; \mu, \sigma)$
3. pnorm(): create the cumulative distribution function $Pr(X < x; \mu, \sigma)$

For loops

For loops are useful when you want to repeat a piece of code many times under similar conditions

The syntax for a for loop is:

```
for (i in 1:100) {
        # do something
}
```

This is repeated 100 times
i is incremented by 1 each time

For loops

For loops are useful when you want to repeat a piece of code many times under similar conditions

The syntax for a for loop is:

```
for (i in 1:100) {
        print(i)
}
```

This is repeated 100 times
i is incremented by 1 each time

For loops

For loops are particular useful in combination with vectors that can store the results

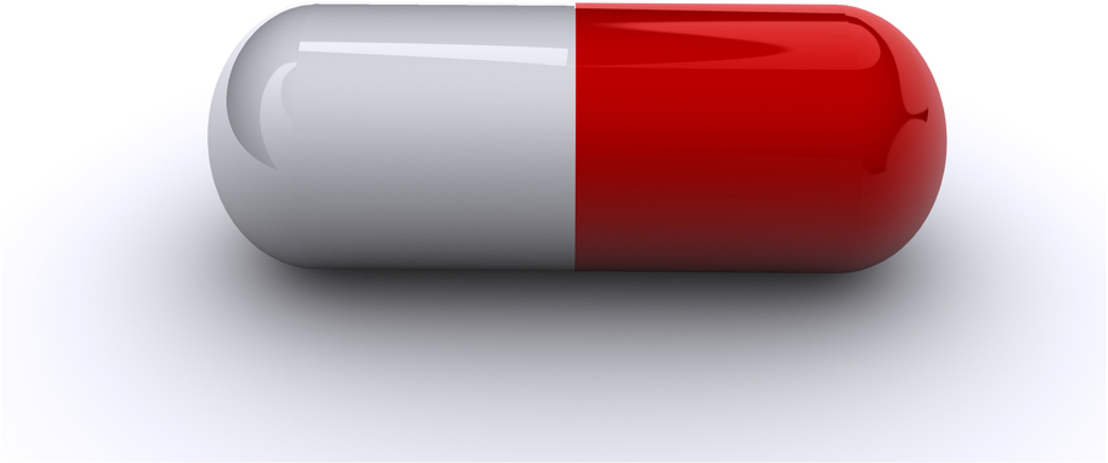
```
my_results <- NULL      # create an empty vector to store the results
for (i in 1:100) {
        my_results[i] <- i^2
}
```

Sometimes there are more efficient ways to do the same thing without for loops

```
        > (1:100)^2
```

Let’s try it in R

Hypothesis tests for comparing two means



Question: Is this pill effective?

Testing whether a pill is effective

How would we design a study?

What would the cases and variables be?

What would the parameter and statistic of interest be?

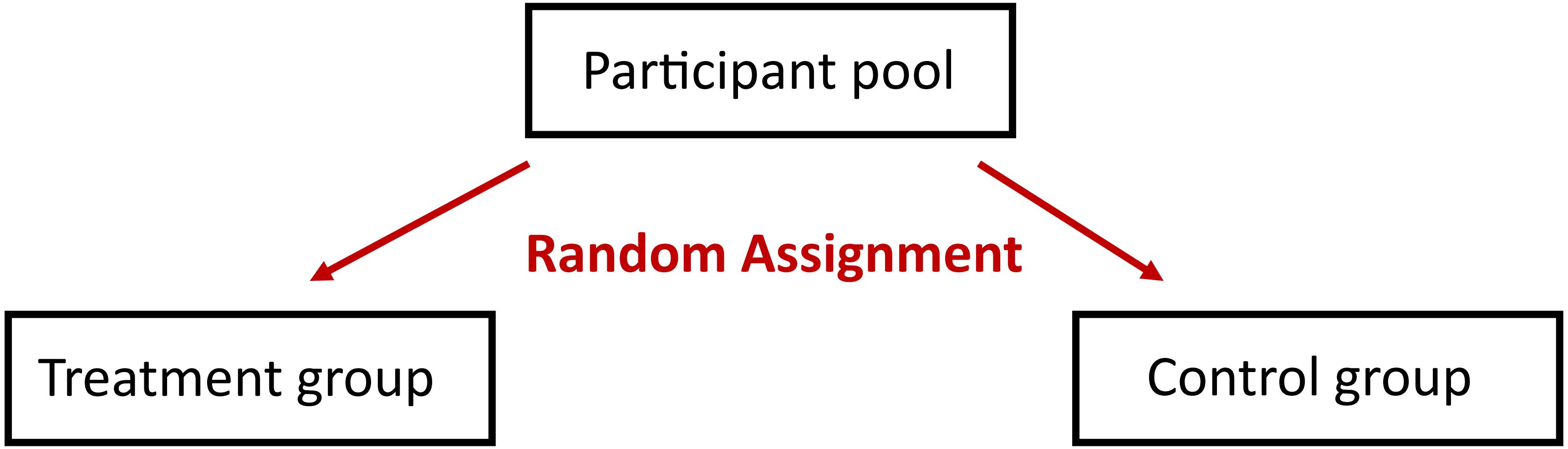
What are the null and alternative hypotheses?

- Assume we are looking for differences in means between the groups

Experimental design

Take a group of participant and ***randomly assign***:

- Half to a *treatment group* where they get the pill
- Half in a *control group* where they get a fake pill (placebo)
- See if there is more improvement in the treatment group compared to the control group



Participant pool

Random Assignment

Treatment group

Control group

Hypothesis tests for differences in two group means

1) State the null and alternative hypothesis

- H_0: $\mu_{\text{Treatment}} = \mu_{\text{Control}}$ or $\mu_{\text{Treatment}} - \mu_{\text{Control}} = 0$
- H_A: $\mu_{\text{Treatment}} > \mu_{\text{Control}}$ or $\mu_{\text{Treatment}} - \mu_{\text{Control}} > 0$

2) Calculate statistic of interest

- $\bar{x}_{\text{Effect}} = \bar{x}_{\text{Treatment}} - \bar{x}_{\text{Control}}$

Example: Does calcium reduce blood pressure?

A randomized by Lyle et al (1987) comparative experiment investigated whether calcium lowered blood pressure in African-American men

- A treatment group of 10 men received a calcium supplement for 12 weeks
- A control group of 11 men received a placebo during the same period

The blood pressure of these men was taken before and after the 12 weeks of the study

1) What are the null and alternative hypotheses?

- H_0: $\mu_{Treatment} = \mu_{Control}$ or $\mu_{Treatment} - \mu_{Control} = 0$
- H_A: $\mu_{Treatment} > \mu_{Control}$ or $\mu_{Treatment} - \mu_{Control} > 0$
  - i.e., a greater decrease in blood pressure after taking calcium

Does calcium reduce blood pressure?

Treatment data (n = 10):

Begin	107	110	123	129	112	111	107	112	136	102
End	100	114	105	112	115	116	106	102	125	104
Decrease	**7**	**-4**	**18**	**17**	**-3**	**-5**	**1**	**10**	**11**	**-2**

Control data (n = 11):

Begin	123	109	112	102	98	114	119	112	110	117	130
End	124	97	113	105	95	119	114	114	121	118	133
Decrease	**-1**	**12**	**-1**	**-3**	**3**	**-5**	**5**	**2**	**-11**	**-1**	**-3**

2) What is the observed statistic of interest?

- $\overline{x}_{\text{Effect}}$ = 5 - -.2727 = 5.273

3) What is step 3?

3. Create the null distribution!

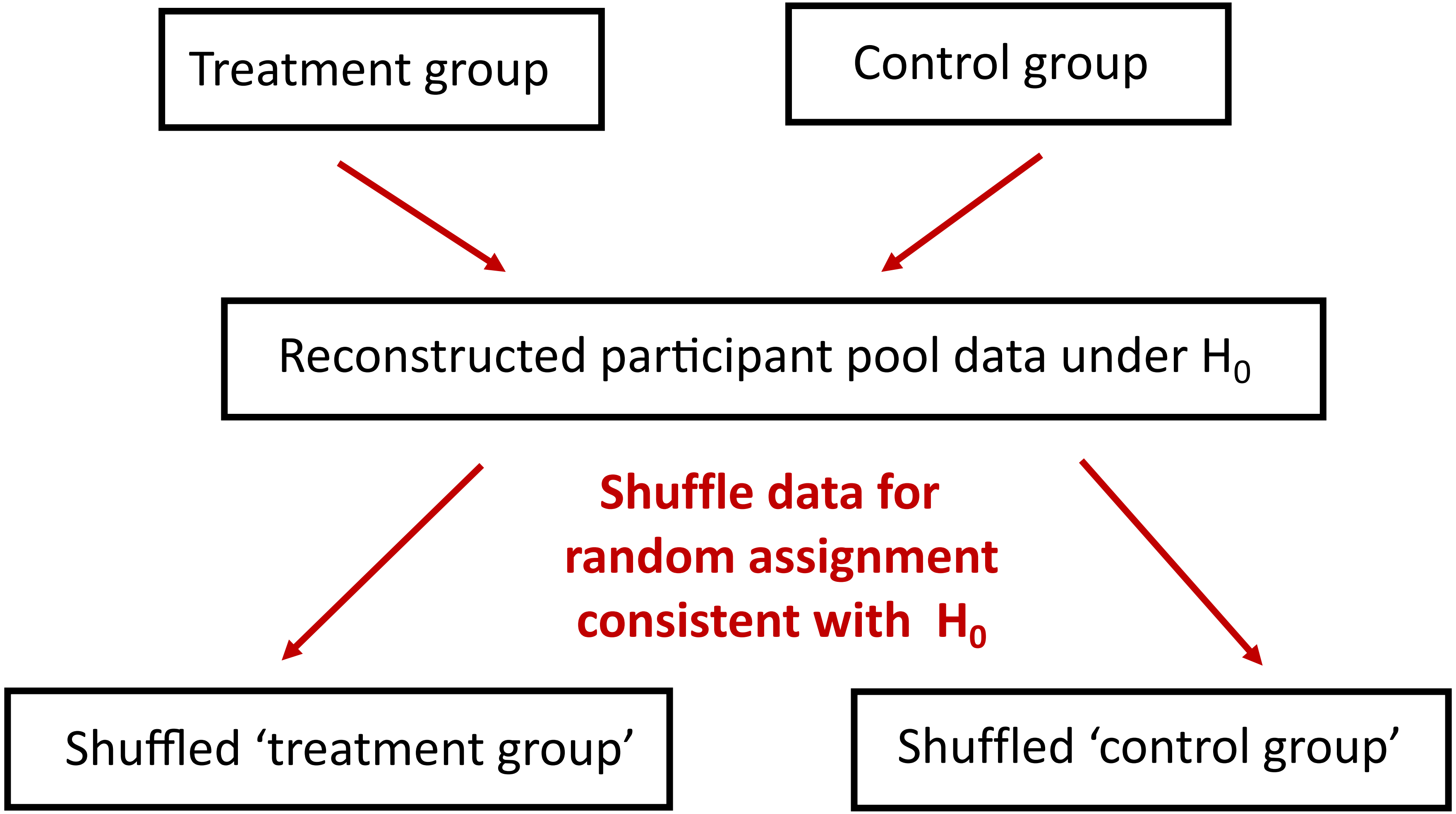
How could we create the null distribution?

Need to generate data consistent with H_0: $\mu_{\text{Treatment}} – \mu_{\text{Control}} = 0$

- i.e., we need fake $\overline{x}_{\text{Effect}}$ that are consistent with H_0

Any ideas how we could do this?

3. Create the null distribution!



Treatment group

Control group

Reconstructed participant pool data under H_0

Shuffle data for random assignment consistent with H_0

Shuffled ‘treatment group’

Shuffled ‘control group’

One null distribution statistic: $\bar{x}_{Shuff\_Treatment} - \bar{x}_{Shuff\_control}$

3. Create a null distribution

1) Combine data from both groups

2) Shuffle data

3) Randomly select 10 points to be the ‘null’ treatment group

4) Take the remaining points to the ‘null’ control group.

5) Compute the statistic of interest on these ‘null’ groups

6) Repeat 10,000 times to get a null distribution

Let’s try it in R

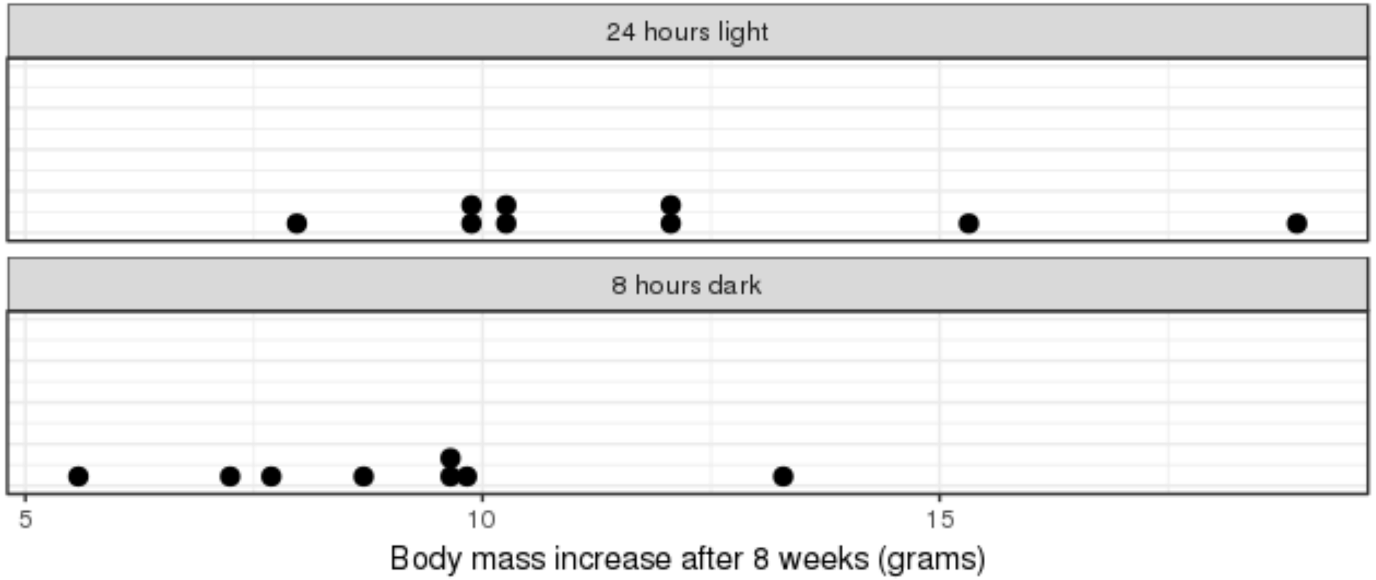
Do mice who eat late at night get fat?

A study by Fonken et al, 2010, wanted to examine whether more weight was gained by mice who could eat late at night

Mice were randomly divided into 2 groups:

- Dark condition: 8 mice were given 8 hours of darkness at night (when they couldn't eat)
- Light condition: 9 were constantly exposed to light for 24 hours (so they could always eat)

What's a good first thing to do when analyzing data?



Hypothesis tests for differences in two group means

1. State the null and alternative hypothesis
   - H_0: $\mu_{Dark} = \mu_{Light}$ or $\mu_{Dark} - \mu_{Light} = 0$
   - H_A: $\mu_{Dark} > \mu_{Light}$ or $\mu_{Dark} - \mu_{Light} > 0$

2. Calculate statistic of interest
   - $\overline{x}_{effect} = \overline{x}_{Dark} - \overline{x}_{Light}$

Do mice who eat late at night get fat?

You can get the data using:

```
>  download_class_data('mice.Rda')
```

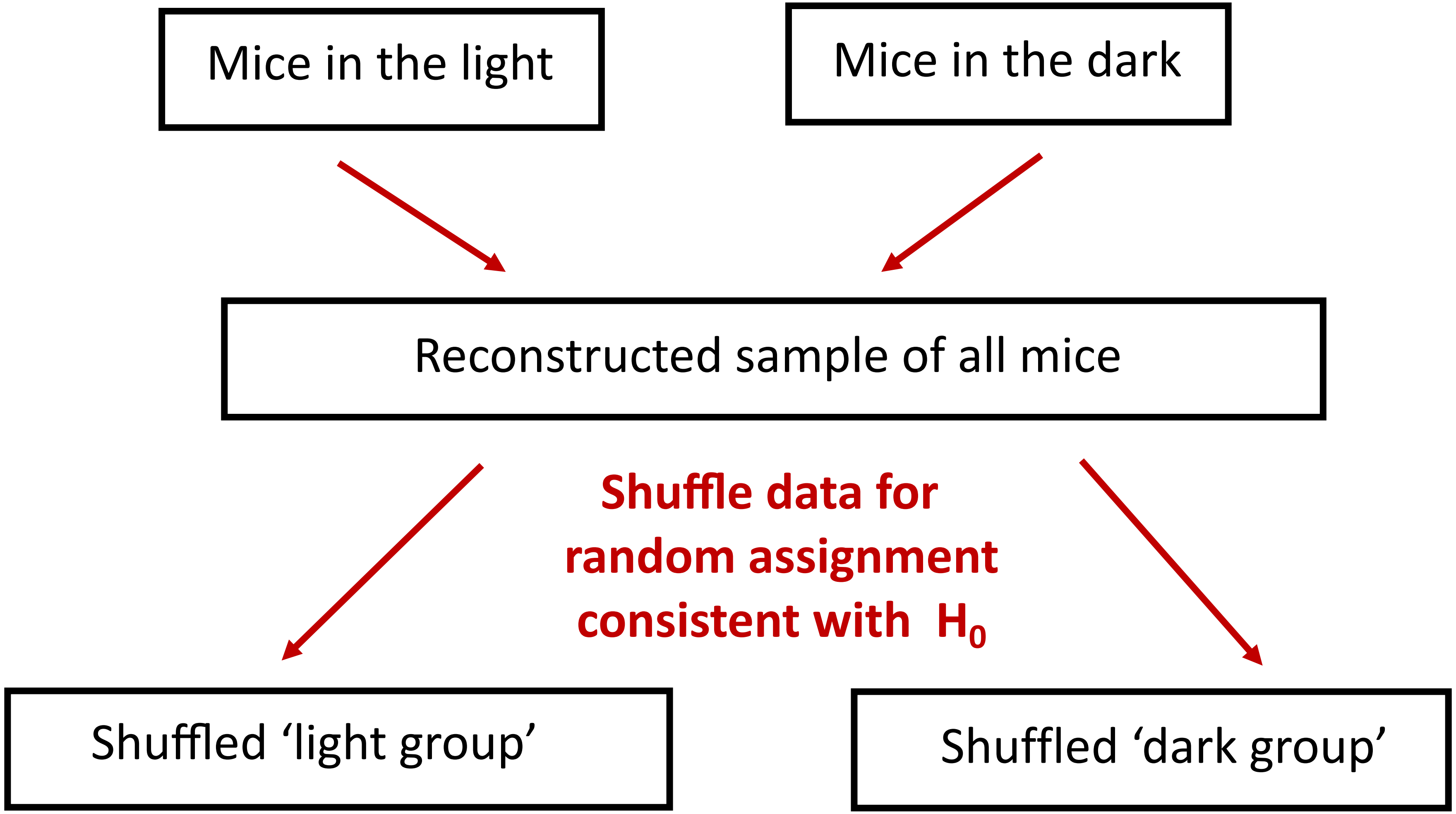
```
> dark_BM_increase       # length(dark_BM_increase)
> light_BM_increase      # length(light_BM_increase)
```

Can you calculate the observed statistic (step 2)?

```
    > obs_stat <- mean(light_BM_increase) - mean(dark_BM_increase)
```

What’s next?

3. Create the null distribution!



Mice in the light

Mice in the dark

Reconstructed sample of all mice

Shuffle data for random assignment consistent with H_0

Shuffled ‘light group’

Shuffled ‘dark group’

One null distribution statistic: $\bar{x}_{Shuff\_Dark} - \bar{x}_{Shuff\_Light}$

Do mice who eat late at night get fat?

What is the first thing we need to do for creating the null distribution?

```
combo_data <- c(light_BM_increase, dark_BM_increase)
```

How do we create one point in our null distribution?

```
# shuffle the data
shuff_data  <- sample(combo_data)

# create fake light and dark data
shuff_light <- shuff_data[1:9]
shuff_dark <- shuff_data[10:17]

# compute fake statistic
mean(shuff_light) - mean(shuff_dark)
```

Do mice who eat late at night get fat?

How do we create a full null distribution?

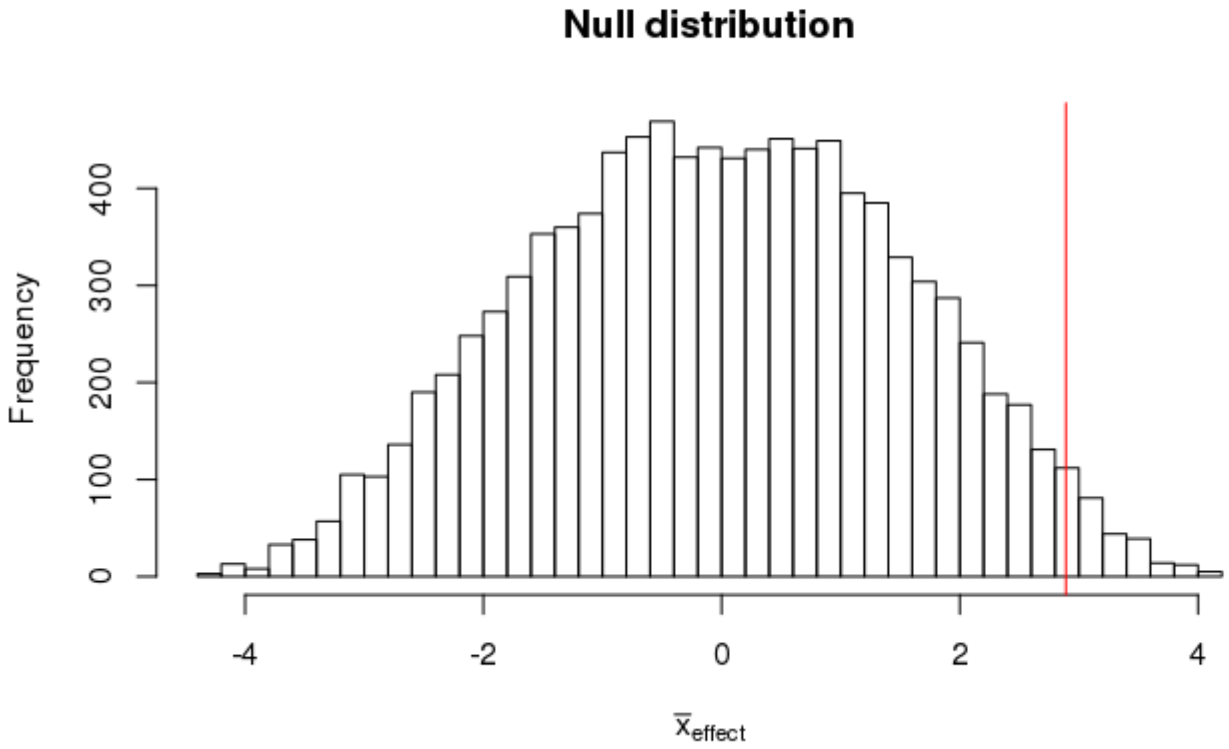
```
null_dist <- NULL
for (i in 1:10000) {

        shuff_data <- sample(combo_data)
        shuff_light <- shuff_data[1:9]
        shuff_dark <- shuff_data[10:17]
        null_dist[i] <- mean(shuff_light) - mean(shuff_dark)

}
```

Do mice who eat late at night get fat?

Plot the null distribution: hist(null_dist, nclass = 50)



What do we do next?

Do mice who eat late at night get fat?

Get the p-value

```
p_val <- sum(null_dist >= obs_stat)/10000
```

p-value = 0.02