R Programming For Natural Resource Professionals

Week 14

Simulations: resampling/bootstrapping

The week ahead…

Learning objectives for this week

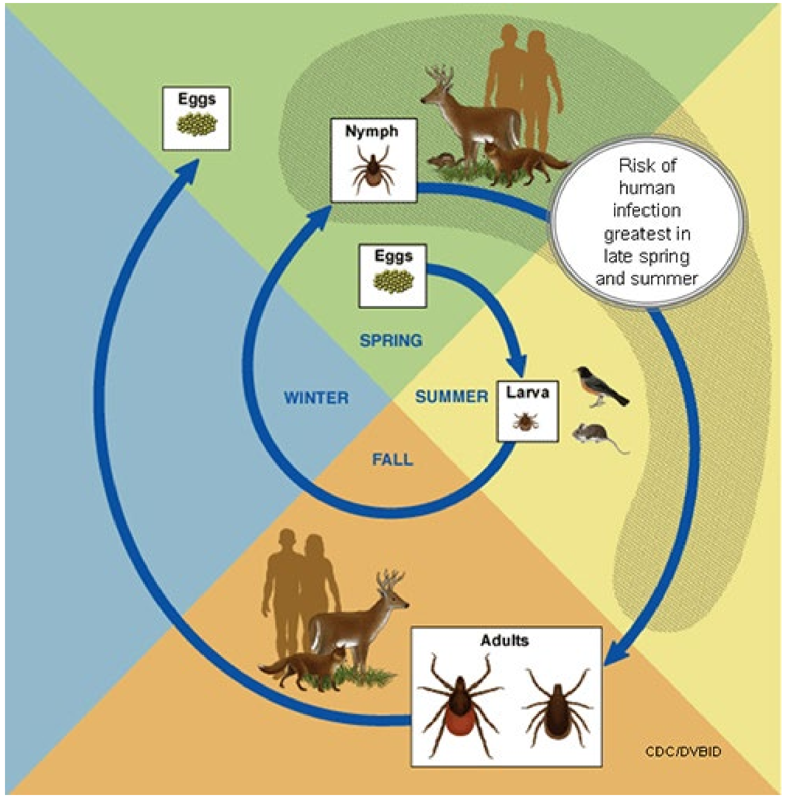
1. Understand motivators for simulation-based hypothesis testing.
2. Understand two simulation-based hypothesis testing algorithms.
3. Use a ‘tidy’ approach to permutation testing and bootstrapping

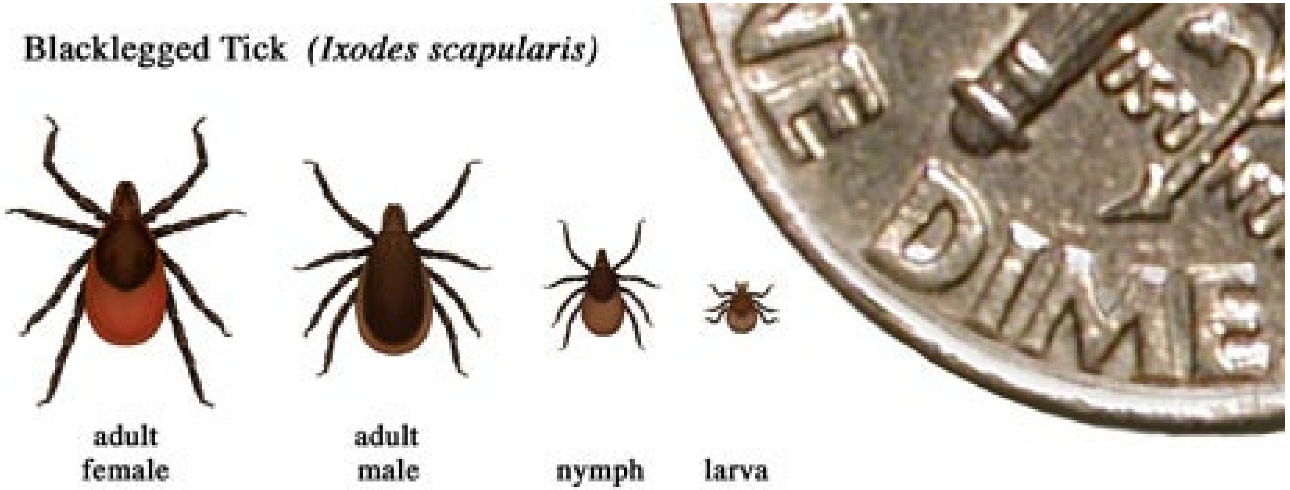
Rationale for simulation-based hypothesis testing

- Goal: Estimate the probability of the observed data
    - “How likely is the observed data to have occurred by random chance?”
- Simulation techniques make no assumptions about residual distributions
- Two approaches we’ll discuss
    - Permutation: Best for estimating a null distribution.
        - “Does variable A depend on variable B?”
    - Bootstrapping: Best for determining confidence intervals
        - “How confident am I in the relationship between variables?”

Example scenario

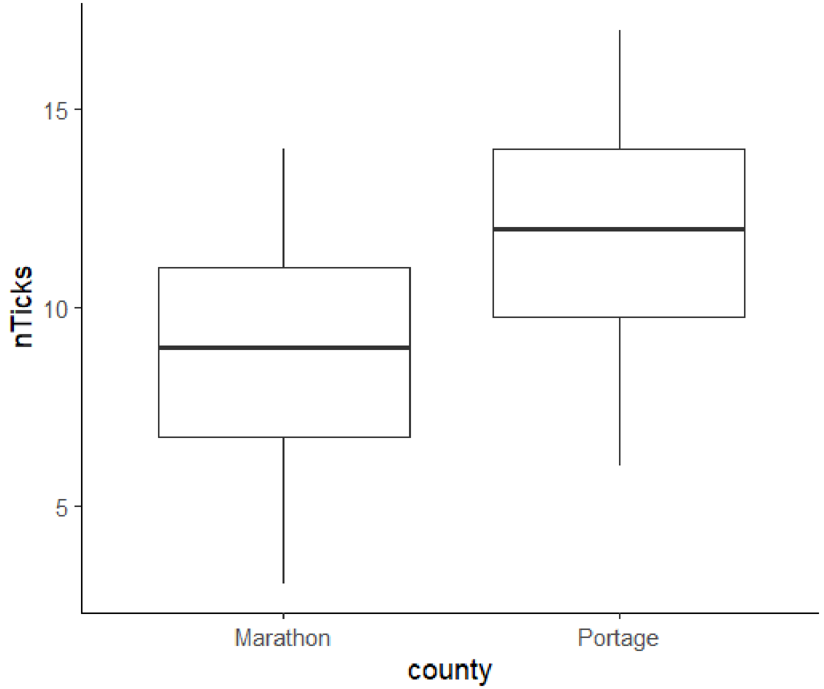
- Compare the number of blacklegged ticks found on harvested deer across two counties.





Example scenario

- Compare the number of blacklegged ticks found on harvested deer across two counties.
- On average, a Marathon County deer had 3.03 ticks fewer than a Portage County deer



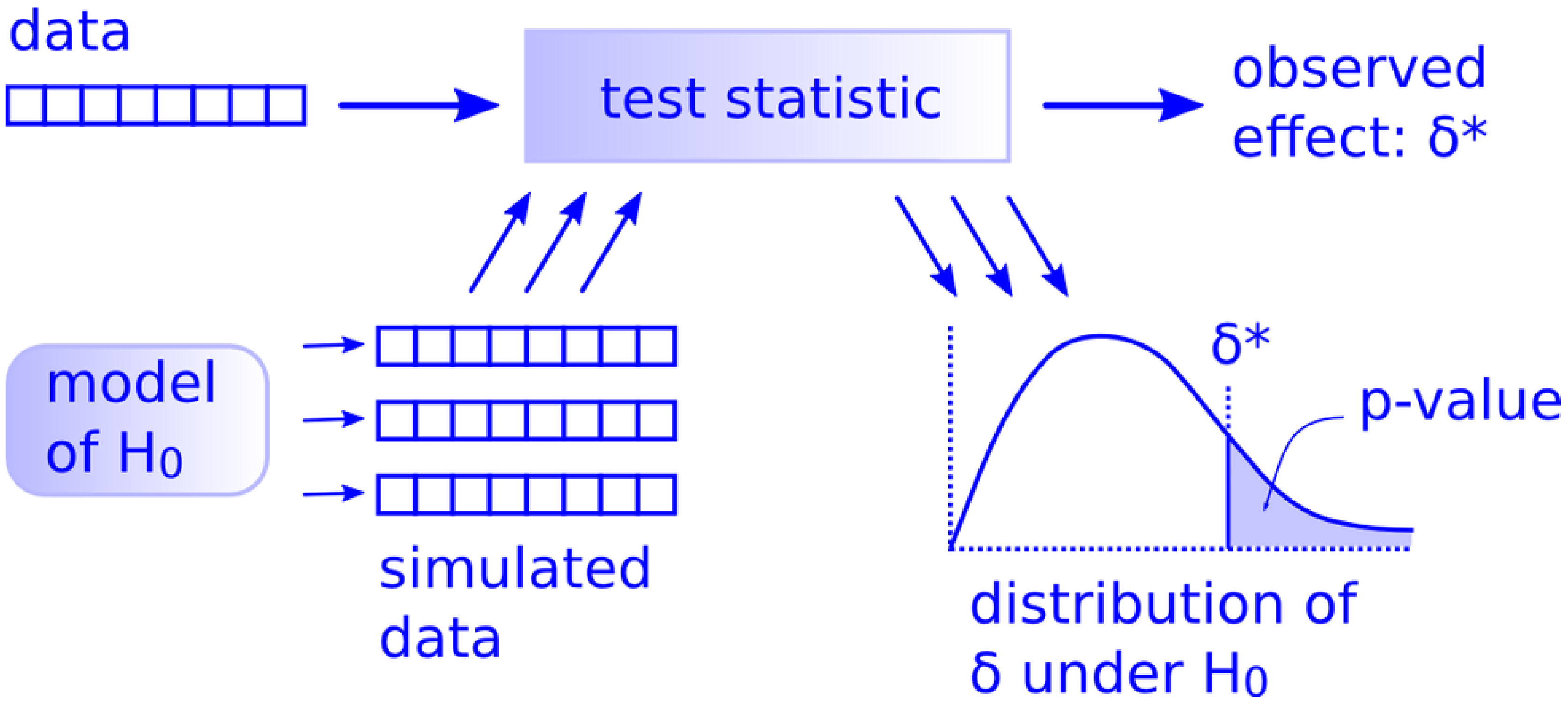
Example scenario

- Compare the number of blacklegged ticks found on harvested deer across two counties.
- On average, a Marathon County deer had 3.03 ticks fewer than a Portage County deer
  - What is likelihood this observation occurred due to chance?

Example scenario

- Compare the number of blacklegged ticks found on harvested deer across two counties.
- On average, a Marathon County deer had 3.03 ticks fewer than a Portage County deer
  - What is likelihood this observation occurred due to chance?
- Test by:
  1. Removing the “county” designation from deer’s tick count.
  2. Re-assign “county” designations to each value by drawing from the pooled values <u>without replacement.</u>
  3. Calculate the difference in mean tick values per county.
  4. Repeat many times.
  5. Compare observed statistic to the generated distribution of the statistic.

Overview of permutation test routine



Simulation-based hypothesis testing using:

1. specify(): allows you to specify the variable, or relationship between variables, that you’re interested in
2. hypothesize() allows you to declare the null hypothesis
3. generate() allows you to generate data reflecting the null hypothesis
4. calculate() allows you to calculate a distribution of statistics from the generated data to form the null distribution

Simulation-based hypothesis testing using:

```
Data %>%
        specify() %>%
        hypothesize() %>%
        generate() %>%
        calculate()
```

Simulation-based hypothesis testing using:

specify(): allows you to specify the variable, or relationship between variables, that you’re interested in.

State the model’s formula in the usual R syntax

- Response ~ predictor
- Response ~ predictor1 + predictor2
- Etc.

Simulation-based hypothesis testing using:

hypothesize() allows you to declare the null hypothesis

Two supported options: “point” and “independence”

- Point: Does the distribution differ from a point estimate (1 sample t-test)
- Independence: Are the variables related to one another?

Simulation-based hypothesis testing using:

generate() allows you to generate data reflecting the null hypothesis

Reps: Number of replicates to generate

Type: “permute” or “bootstrap”

Simulation-based hypothesis testing using:

calculate() allows you to calculate a distribution of statistics from the generated data to form the null distribution

Stat: "mean", "median", "sum", "sd", "prop", "count", "diff in means", "diff in medians", "diff in props", "Chisq" (or "chisq"), "F" (or "f"), "t", "z", "ratio of props", "slope", "odds ratio", or "correlation".

Order: How to order variables for the calculation. E.g., c(“var1”, “var2”)

Bootstrapping

- Purpose: Determine confidence of a parameter estimate
- There are a variety of bootstrap methods, but at their core is a common process:
    1. Begin with an observed sample of size *N*
    2. Generate a simulated sample of size *N* by drawing observations from your observed sample independently and with replacement.
    3. Compute and save the statistic of interest
    4. Repeat this process many times (e.g. 1,000)
    5. Treat the distribution of your estimated statistics of interest as an estimate of the population distribution of that statistic.

Bootstrapping

```
Data %>%
        specify() %>%
        generate(type = “bootstrap”) %>%
        calculate() %>%
        get_confidence_interval()
```