Resampling methods: Bootstrap and Permutation tests

Julia Wrobel

Overview

```
library(tidyverse)
```

Today, we cover:

- Resampling methods
  - Bootstrap
  - Permutation tests

Announcements:

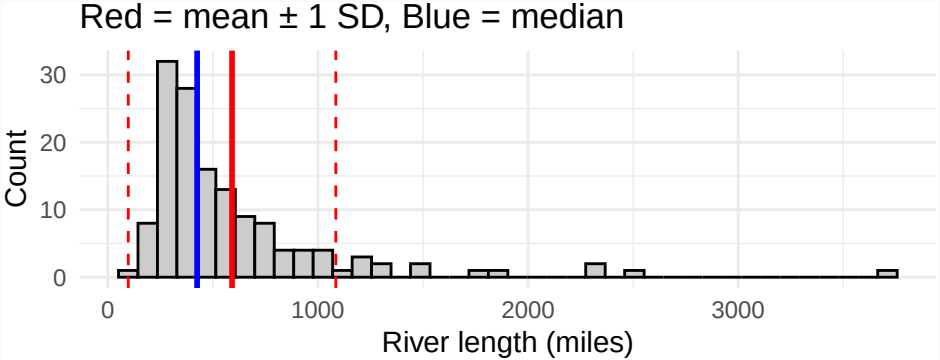
- HW2 posted and due 2/11 at 10:00AM
- Final project due 5/1 at midnight

Why do we need resampling methods?

- Inference requires understanding the **sampling distribution** of a statistic
- Classical tools rely on:
  - Known parametric models
  - Large sample (CLT-based) approximations
- In many modern settings:
  - Sampling distributions are unknown
  - Asymptotics may be inaccurate or hard to obtain
  - Deriving standard errors is difficult or impossible

Rivers t-test example

- `rivers` data: length (in miles) of 141 major North American Rivers



Rivers t-test example

Is the mean length of rivers in North America equal to 1000 miles?

- $H_0 : \mu = 1000$

```
    One Sample t-test

data:  rivers
t = -9.8293, df = 140, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 1000
95 percent confidence interval:
 508.9559 673.4129
sample estimates:
mean of x
 591.1844
```

Rivers t-test example

T-test assumptions

- Observations are independent
- CLT holds: sampling distribution of $\bar{Y}$ will be approximately normal even if the population isn't
- $\bar{Y}$ and s are approximately independent

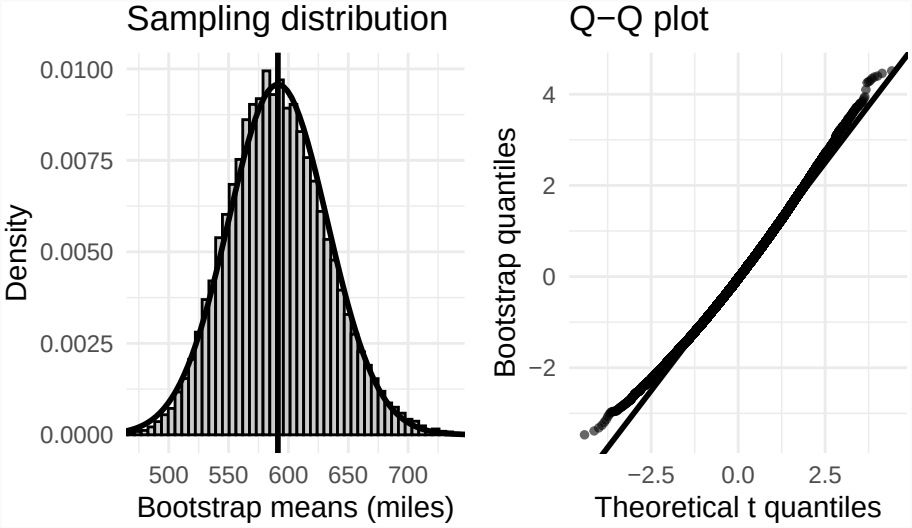
$$t = \frac{\bar{Y} - \mu}{s/\sqrt{n}}$$

- $t \sim t_{n-1}$ or $t \approx N(0, 1)$

Highly skewed data violates these assumptions

- Mean and variance are correlated
- Skewness is a property of the underlying distribution, not n, so increasing sample size won't necessarily fix the problem
    - Poor t-test coverage

Rivers t-test sampling distribution



Bootstrap: motivation

Suppose we observe data $Y_1 \dots Y_n \sim F$ and we want to compute a statistic $T(Y_1, \dots Y_n)$.

We want:

- Standard errors
- Confidence intervals
- But F is unknown
- The distribution of T depends on F

How can we approximate the sampling distribution of T without knowing F?

Why classical asymptotics can fail

CLT-based inference assumes approximate normality of the test statistic. This can fail, even as $n \to \infty$ when:

- Data is skewed or heavy-tailed
- For nonlinear or non-smooth statistics (i.e. sample median vs. sample mean)

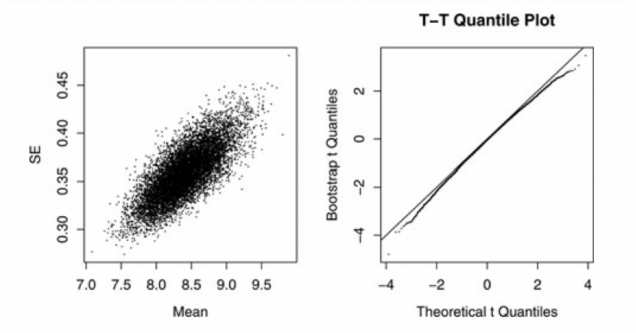
When these assumptions fail, asymptotics-based confidence intervals can have poor coverage.

- Also great for more complicated statistics where asymptotics can be hard to calculate

When t is not t

When Y_i come from a skewed distribution, the sample mean and its SE are correlated and their correlation does not decrease as $n \to \infty$

- $\implies$ t-statistic is not t-distributed



- If your test or CI assumes your statistic has a normal or t sampling distribution, **any deviation** from the diagonal implies your test/interval will perform poorly

When t is not t

Idea: confidence intervals for parameter estimates provide no robustness against non-normality!

- Coverage does not improve as $n \to \infty$!

Bootstrapping can be very useful for constructing better confidence intervals for these parameters.

A computational workaround

- We cannot repeatedly sample from the true population distribution F
- But, we *do* have data sampled from F

Idea: use the observed data to approximate F through simulation

Parameters as functionals

For some cumulative distribution function F, suppose we are interested in a parameter, $\theta \stackrel{\triangle}{=} T(F)$, written as a functional of F

- A **functional** is a function of a function
  - Mean of F: $T(F) = \int y dF(y)$
  - Median of F: $T(F) = inf\{y : F(y) \geq 0.5\}$
- The statistic $T(Y_1, ..., Y_n)$ is an estimate of $T(F)$

Inference about a statistic is inference about a **functional of an unknown distribution.**

Estimating the data-generating distribution

- The true distribution of F is unknown, so we have to estimate it
- A natural estimator is the **empirical distribution**:

$$\hat{F}_n(y) = \frac{1}{n} \sum_{i=1}^{n} I\{Y_i \leq y\}$$

- $\hat{F}_n$ places $1/n$ mass at each observed data point
- To estimate a functional $T(F)$, we plug in the empirical distribution, $\hat{F}_n(y)$, for $F(y)$:

The Bootstrap Principle

- Replace the unknown distribution F with its estimate $\hat{F}_n$
- Approximate the sampling distribution of $T(Y_n, \dots, Y_n)$ by:
  - Repeatedly sampling from $\hat{F}_n$
  - Recomputing the statistic

Idea: Bootstrap principle is used to estimate the sampling distribution of a *statistic* without relying on strong parametric assumptions about the underlying population distribution

What the bootstrap gives us

- Approximate sampling distribution
- Standard error estimates
- Confidence intervals

The Bootstrap Principle

- Nonparametric bootstrap
- Parametric bootstrap
- Wild bootstrap
- Smooth bootstrap
- Bag of little bootstraps

Nonparameteric bootstrap

A **bootstrap sample** is a random sample of size drawn **with replacement** from $\hat{F}_n$

- Bootstrap sample denoted $y^* = (y_1^*, y_2^*, \dots, y_n^*)$
- $\hat{\theta}^* \triangleq T(\hat{F}_n(y^*))$ is the corresponding bootstrap estimate using the sample y^*
  - We'll call this $T(\hat{F}_n^*)$ to simplify notation

Draw B bootstrap samples and calculate $\hat{\theta}_i^*, i = 1, \dots, B$.

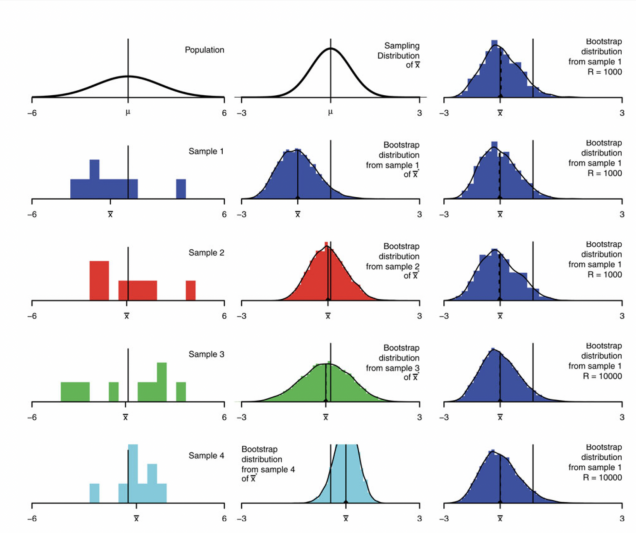
- Then, SE of $\hat{\theta}$ can be estimated using the standard deviation of the B replications:

$$\widehat{SE}(\hat{\theta}) = \sqrt{\frac{\sum_{i=1}^{B}(\hat{\theta}_i^* - \bar{\theta}^*)^2}{B-1}}$$

Why does it work?

- $\hat{F}_n \to F$ uniformly by Glivenko-Cantelli
- $\implies$ when n is large, sampling from $\hat{F}_n$ resembles sampling from F
- Potential sources of error:
  - Finite sample bias
  - Bootstrap distribution contains Monte Carlo error when exhaustive resampling is infeasible
    - For fixed n, number of possible bootstrap samples is $\binom{2n-1}{n}$

Sources of error



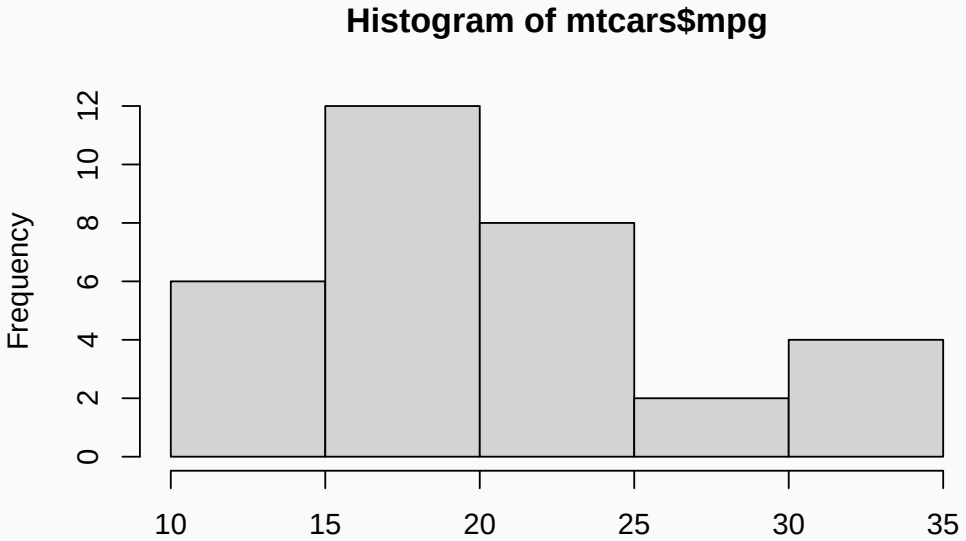
Example: nonparameteric bootstrap

Using `mtcars` dataset, we will look at the mean `mpg` for different cars

```
data(mtcars)
str(mtcars)
```

```
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Example: nonparameteric bootstrap



Example: nonparameteric bootstrap

```r
B = 10000
mpg = mtcars$mpg
boot_mean = rep(NA, B)
set.seed(222)
for(i in 1:B){
  ystar = sample(mpg, size = length(mpg), replace = TRUE)
  boot_mean[i] = mean(ystar)
}

paste0("mean: ", round(mean(mpg),3),
       "; bootstrapped mean: ", round(mean(boot_mean),3))
```

```
[1] "mean: 20.091; bootstrapped mean: 20.096"
```

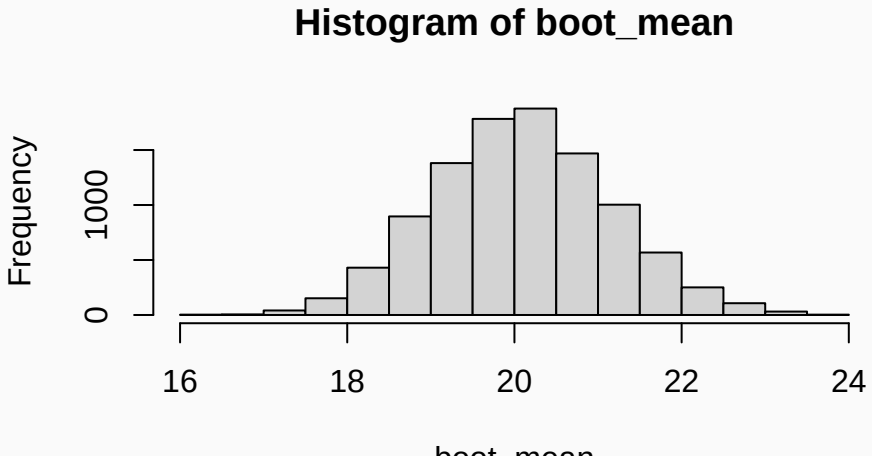
Example: nonparameteric bootstrap

```
se_mean = sd(mpg)/sqrt(length(mpg))
paste0("sd: ", round(se_mean,3),
       "; bootstrapped sd: ", round(sd(boot_mean),3))
```

```
[1] "sd: 1.065; bootstrapped sd: 1.056"
```

Example: nonparameteric bootstrap

```
hist(boot_mean)
```



Bootstrap confidence intervals

- When it works, the bootstrap can be very useful for constructing confidence intervals (CIs) for a parameter θ
- Multiple methods for constructing bootstrap CIs exist!
  - Percentile intervals
  - t intervals with bootstrap SE (also called Wald-type)
  - bootstrap t
  - BCa intervals
- Quality of different approaches generally depends on n and the skewness of the original data

Order of accuracy

- Recall "big-O" notation for non-negative functions f, g:
  - $f(x) \in O(g(x))$ iff $f(x) \leq hg(x) \forall x \geq x_0$ and some $h > 0$
- A confidence interval is **first-order accurate** if the non-coverage probability differs from the nominal value by $O(n^{-1/2})$:
  - $Pr(\theta < \theta_{lb}) + P(\theta > \theta_{ub}) = \alpha + O(n^{-1/2})$
- A confidence interval is second-order accurate if the non-coverage probability differs from the nominal value by $O(n^{-1})$:
  - $Pr(\theta < \theta_{lb}) + P(\theta > \theta_{ub}) = \alpha + O(n^{-1})$

Percentile method

The simplest approach to constructing a bootstrap CI is the **percentile method**.

- Let $\widehat{\theta}_1^*, \ldots, \widehat{\theta}_B^*$ be a bootstrap sample of point estimators
- Then, a two-sided $100 \times (1-\alpha)\%$ bootstrap percentile CI is

$$(\xi_{\alpha/2}^*, \xi_{1-\alpha/2}^*)$$

- where ξ_p^* is the p^{th} percentile of the bootstrap samples

Percentile method

Pros:

- Simple to implement and easy to understand
- It is **transformation invariant**: the percentile method confidence interval for a monotone transformation of θ can be obtained by transforming the endpoints of the interval for θ
- Works better than some other methods when data are skewed

Cons:

- Tends to be too narrow with small sample sizes
- The percentile method is only first-order accurate

t intervals with bootstrap SE

Recall that if $(\hat{\theta} - \theta)/\hat{se}(\hat{\theta}) \to N(0, 1)$, then the (Wald) approximate CI for θ is

$$\hat{\theta} \pm Z_{1-\alpha/2} \times se(\hat{\theta})$$

- This interval will closely agree with the percentile bootstrap CI when $\hat{\theta}$ is approximately normal
- What if we don't know $se(\hat{\theta})$ or $se(\hat{\theta})$ is difficult to calculate?

t intervals with bootstrap SE

When $se(\widehat{\theta})$ is unknown or cumbersome, we can use a bootstrap estimate of the standard error of $\widehat{\theta}$ instead:

$$\widehat{\theta} \pm Z_{1-\alpha/2} \times \widehat{se}_b(\widehat{\theta})$$

- For smaller samples, replace the Z quantile by a t quantile:

$$\widehat{\theta} \pm t_{1-\alpha/2,n-1} \times \widehat{se}_b(\widehat{\theta})$$

- If you can estimate $se(\widehat{\theta})$ directly, there is no real benefit of using $\widehat{se}_b(\widehat{\theta})$

t intervals with bootstrap SE

Pros:

- Simple to implement and easy to understand
- Can be applied to situations where $se(\widehat{\theta})$ is difficult to derive

Cons:

- **Biased**: comparable to using the MLE $\sqrt{\frac{1}{n}\sum_i (x_i - \bar{x})^2}$ instead of the unbiased estimate $\sqrt{\frac{1}{n-1}\sum_i (x_i - \bar{x})^2}$
- Can perform poorly if distribution is skewed
- Tends to be too narrow with small sample sizes
- Only first-order accurate

Bootstrap t method

The **bootstrap-t** method uses simulation to estimate the t quantile for the interval.

- For **each** bootstrap sample, calculate t_b^*, where:
  - $t_b^* = \frac{\hat{\theta}_b^* - \hat{\theta}}{se(\hat{\theta}_b^*)}$
  - $\hat{\theta}$ is estimated from the original sample
  - $\hat{\theta}_b^*$ is estimated from the b^{th} bootstrap sample
  - $se(\hat{\theta}_b^*)$ is the SE of the b^{th} estimate

Bootstrap t method

The **bootstrap-t** method uses simulation to estimate the t quantile for the interval.

- For **each** bootstrap sample, calculate t_b^*, where:
  - $t_b^* = \frac{\hat{\theta}_b^* - \hat{\theta}}{se(\hat{\theta}_b^*)}$
  - $\hat{\theta}$ is estimated from the original sample
  - $\hat{\theta}_b^*$ is estimated from the b^{th} bootstrap sample
  - $se(\hat{\theta}_b^*)$ is the SE of the b^{th} estimate
- Since $se(\hat{\theta}_b^*)$ is unknown, must estimate it...

Bootstrap t method

Note that calculating $\widehat{se}(\hat{\theta}_b^*)$ for each bootstrap estimate requires (nested) bootstrapping!

- Obtain a bootstrap estimate of $se(\hat{\theta}_b^*)$ for **each** $b = 1, \dots, B$
  - For each bootstrap sample b, run $k = 1, \dots, K$ bootstrap simulations to obtain $\hat{\theta}_{b,k}^*$ and use these to estimate $se(\hat{\theta}_b^*)$ for calculating t_b^*
- Calculate lower and upper quantiles of the *bootstrap t distribution*, $t_{1-\alpha/2}^*$ and $t^*\alpha/2$ and construct the CI:

$$\left(\hat{\theta} - t_{1-\alpha/2}^* \times se(\hat{\theta}), \hat{\theta} - t_{\alpha/2}^* \times se(\hat{\theta})\right)$$

- $se(\hat{\theta})$ is the SE of the estimate from the original sample, or can be estimated using the top-level bootstrap

Bootstrap t method

Pros:

- Second-order accurate
- Usually outperforms the other methods discussed so far, especially for non-normal populations.
- For skewed data, much more asymmetric than the percentile bootstrap interval which corrects for the possibility that we observed too few observations from the tail

Cons:

- Requires an iterated bootstrap
- The bootstrap-t interval is not transformation invariant. Must reconstruct the CI for a function of the parameter $g(\theta)$

BCa Bootstrap

The bias-corrected accelerated (BCa) method is an improvement over the percentile method, though still does not work well in small sample sizes.

- BCa intervals use percentiles of the bootstrap distribution, but not necessarily the $100 \times \alpha$-th and $100 \times (1 - \alpha)$-th percentiles
- Adjusts for **bias** in the bootstrap distribution and variability in the precision of the estimate (aka **acceleration**)

First, estimate a bias correction factor $\hat{z}_0$:

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#\{\hat{\theta}^* < \hat{\theta}\}}{B}\right)$$

- The acceleration parameter measures how the estimator's variability changes with the true parameter

BCa Bootstrap

Next, estimate the acceleration parameter $\hat{a}$:

$$\hat{a} = \frac{\sum_{i=1}^{n} (\hat{\theta}_{(.)} - \hat{\theta}_{(i)})^3}{6 \left(\sum_{i=1}^{n} (\hat{\theta}_{(.)} - \hat{\theta}_{(i)})^2 \right)^{3/2}}$$

- $\hat{\theta}_{(i)}$ is the value of the statistic with the i^{th} observation removed (i.e. the i^{th} jackknife estimate)
- $\hat{\theta}_{(.)}$ is the mean of the n jackknife estimates
- adjusts for the rate of change in the standard error as the estimate changes

BCa Bootstrap

After estimating the bias correction $\hat{z}_0$ and acceleration $\hat{a}$, compute:

$$\alpha_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{\alpha/2}}{1 - \hat{a}(\hat{z}_0 + z_{\alpha/2})}\right)$$

$$\alpha_2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{1-\alpha/2}}{1 - \hat{a}(\hat{z}_0 + z_{1-\alpha/2})}\right)$$

- Φ is the standard normal CDF (`pnorm`)
- z_a is the $100 \times \alpha$-th percentile of the standard normal
- Use these new percentiles to construct a CI
  - The BCa interval is the same as the percentile interval when $\hat{a} = \hat{z}_0 = 0$

BCa Bootstrap

Pros:

- Second-order accurate and transformation invariant
- Works well for a variety parameters
- For skewed data, the percentile bootstrap is not asymmetric enough and the BCa bootstrap addresses this limitation
- Implemented in the `bcanon()` function in the `bootstrap` R library

Cons:

- Estimation of the acceleration parameter requires the jackknife
- Less intuitive than other methods
- Still doesn't work great for small sample sizes

Parametric bootstrap

If we have information about the population distribution, this can be used in resampling for bootstrap inference

- If the assumption about the population distribution is correct then the parametric bootstrap will perform better than the nonparametric bootstrap
- If the assumption not correct, then the nonparametric bootstrap should perform better.

Parametric bootstrap example

Suppose we have data from a $N(\mu, \sigma^2)$ distribution, $Y = \{y_1, \dots, y_n\}$

- Interested in obtaining an estimate of the standard error of the trimmed mean with 10% trimmed from each tail
- To employ the parametric bootstrap, we would start by generating n values from a $N(\hat{\mu}, \hat{\sigma}^2)$
  - $\hat{\mu}, \hat{\sigma}^2$ are the ML estimates
- Then, compute trimmed mean using the simulated sample
  - repeat B times

Only difference between parametric and non-parametric bootstrap is the way to generate data!

Bootstrapping regression models

Regression model:

$$E(Y_i|\mathbf{X}_i) = f(,\mathbf{X}_i), i = 1, \dots, n$$

- f is a known function, $\mathbf{X}_i$ a vector of parameters
- Interested in estimating a SE or CI for a function of parameters

Bootstrapping regression models

Three common methods:

1. Bootstrap the pairs $(Y_i, \mathbf{X}_i)$ and generate bootstrap parameter estimates that can be used to obtain bootstrap CIs
2. Bootstrap the **residuals** of the original model
3. Estimate the residual variance and simulate residuals

Bootstrapping regression models: residuals

Let's take linear regression as an example,

$$Y_i = \mathbf{X}_i^T + \epsilon_i; \epsilon_i \sim N(0, \sigma^2)$$

To fit a bootstrap on the residuals:

- Fit a model on the original data to obtain residuals $\hat{\epsilon}_i$
  - $\hat{\epsilon}_i = Y_i - \mathbf{X}_i^{\mathcal{T}}$
- Resample residuals $\hat{\epsilon}_i$ with replacement and repeat B times:
  - Compute $Y_i^b = \mathbf{X}_i^{\mathcal{T}} + \hat{\epsilon}_i^b$ for all i
  - Refit model using $Y_i^b, i = 1, \ldots, n$ to estimate $\hat{\beta}^b$
- This approach assumes errors are **identically distributed**

Bootstrapping regression models: residual variance

- Estimate residual variance from sample, $\hat{\sigma}^2$
- Repeat B times:
  - Generate a residual $\epsilon_i^b \sim N(0, \hat{\sigma}^2)$ and associated outcome $Y_i^b = \mathbf{X}_i^T + \epsilon_i^b$ for $i = 1, \ldots, n$
  - Refit model using Y_i^b, $i = 1, \ldots, n$ to estimate $\hat{\beta}^b$

This approach assumes the residual errors are **normally distributed**

Wild bootstrap

$$Y_i = \mathbf{X}_i^T + \epsilon_i$$

The models previously discussed fail if there is **heteroskedasticity** - i.e., if ϵ_i are not identically distributed

The **wild bootstrap** is a modification of the residual bootstrap intended to handle heteroskedasticity.

- Fit a model on the original data to obtain residuals $\hat{\epsilon}_i$
- Repeat B times:
  - Multiply residuals by random weights to obtain $\hat{\epsilon}_i^b$
  - Compute $Y_i^b = \mathbf{X}_i^T + \hat{\epsilon}_i^b$ for all i
  - Refit model using $Y_i^b, i = 1, \dots, n$ to estimate $\hat{\beta}^b$

Wild bootstrap

Common options for weights:

1. Multiply $\widehat{\epsilon}_i$ by -1 or 1 with equal probability
2. Multiply $\widehat{\epsilon}_i$ by a standard normal random variable

When does the bootstrap fail?

We must assume that the original sample is representative of the population so that $\hat{F}_n$ is a good estimate of F

The bootstrap may fail when:

- The support of F depends on the parameter of interest
- The true parameter sits on the boundary of the parameter space
- The parameter of interest is nonregular, i.e., there does not exist an estimator that converges to it uniformly in distribution over the parameter space

Permutation tests

Typically bootstrap is used for CI rather than hypothesis testing. For hypothesis testing and p-values, we can use a **permutation test**. - Idea: use resampling to generate a **null distribution** for a test statistic, then compare it to the one you observe in the real data

- **Null distribution**: the distribution of a quantity of interest (i.e. $\hat{\beta}$) if the null hypothesis H_0 is true
- The null distribution is available theoretically in some cases. For example, assume $Y_i \sim N(\mu, \sigma^2), i = 1, \dots, n$.
  - Under $H_0 : \mu = 0$, we have $\bar{Y} \sim N(0, \sigma^2/n)$
    - Test H_0 by comparing $\bar{Y}$ with $N(0, \sigma^2/n)$
- Use **permutation test** when null distribution cannot be obtained theoretically

Permutation tests

The basic procedure of permutation test for H_0:

- Permute data under H_0 B times. Each time recompute the test statistics. The test statistics obtained from the permuted data form the null distribution.
- Compare the observed test statistics with the null distribution to obtain statistical significance.

Permutation test example

Assume there are two sets of independent normal r.v.'s with the same known variance and different means:

- $X_i \sim N(\mu_1, \sigma^2)$
- $Y_i \sim N(\mu^2, \sigma^2)$

Our goal is to test $H_0 : \mu_1 = \mu_2$. Define test statistic: $t = \bar{X} - \bar{Y}$. Permutation test steps:

1. Randomly shuffle labels of X and Y
2. Compute $t^* = \bar{X}^* - \bar{Y}^*$
3. Repeat `nperm` times. Resulting t^* values form the **empirical null distribution** of t.
4. To compute p-values calculate $Pr(|t^*| > |t|)$