3 Bootstrapping Dependent Data

Suppose we have dependent data $\boldsymbol{y} = (y_1, \ldots, y_n)$ generated from some unknown distribution $F = F_{\boldsymbol{Y}} = \underbrace{F_{(Y_1,\ldots,Y_n)}}_{\text{joint distribution}}$.

$Y_1, \ldots, Y_n$ no longer assuming independence, could be time series for example (or spatial, etc.)

Goal:

To approximate the dsn of a statistic $\theta = T(Y)$.

Challenge:

Since Y_i's dependent, it is inappropriate to use the bootstrap for iid data! Bootstrapped samples would no longer reproduce the data generating process!

If we used iid bootstrap for dependent data, then $\hat{\mathrm{var}}(\hat{\theta})$ based on bootstrap, would be wrong (too small). and any inference we make using this procedure would be invalid.

We will consider 2 approaches

1) model-based (parametric bootstrap)

2) block bootstrap (2 types) (nonparametric bootstrap).

3.1 Model-based approach

Example 3.1 Suppose we observe a time series $\boldsymbol{Y} = (Y_1, \dots, Y_n)$ which we assume is generated by an AR(1) process, i.e., "auto regressive" → "regressed on itself"

$$Y_t = \alpha Y_{t-1} + \varepsilon_t \qquad t = 1, \dots, n$$

turn our problem into iid bootstrap.

$$|\alpha| < 1 \quad \text{and} \quad \varepsilon_t \overset{iid}{\sim} (0, \sigma^2)$$

"innovations" mean 0 variance

If we assume an AR(1) model for the data, we can consider a method similar to bootstrapping residuals for linear regression.

1. Estimate $\hat{\alpha}$ from data (fit the model).
2. Define estimated innovations $\hat{e}_t = Y_t - \hat{\alpha} Y_{t-1}$, $t = 2, \dots, n$ (observed data; $\hat{\alpha}$ from ①)

   and $\bar{\hat{e}}_t = \frac{1}{n-1} \sum_{t=2}^{n} \hat{e}_t$ their sample mean
3. Define the residuals of the model as centered innovations

   $$\hat{\varepsilon}_t = \hat{e}_t - \bar{\hat{e}}$$
4. For $b = 1, \dots, B$
   a) Create the bootstrap sample $\hat{\varepsilon}^*_0, \dots, \hat{\varepsilon}^*_n$ by independently sampling $n+1$ (w/ replacement) values from the $n-1$ values $\hat{\varepsilon}_t$, $t = 2, \dots, n$.
   b) construct pseudo-data $Y^* = (Y_1^*, \dots, Y_n^*)$ as. $Y_0^* = \hat{\varepsilon}_0^*$,

   $$Y_t^* = \hat{\alpha} Y_{t-1}^* + \hat{\varepsilon}_t^*, \quad t = 1, \dots, n.$$
   c) define $\hat{\alpha}_b^*$ as the estimate of α from $Y_1^*, \dots, Y_n^*$.
5. The dsn of $\hat{\alpha}_1^*, \dots, \hat{\alpha}_B^*$ is used to estimate the sampling dsn of $\hat{\alpha}$.

Model-based – the performance of this approach depends on the model being appropriate for the data.

This may not always be a good assumption.

3.2 Nonparametric approach

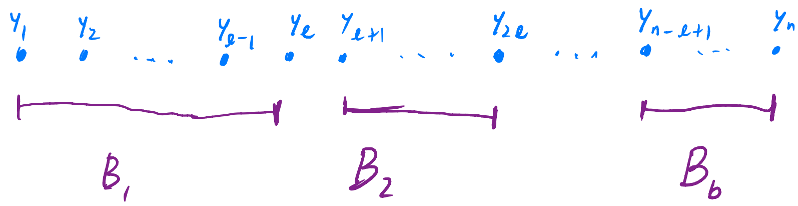
To deal with dependence in the data, we will employ a nonparametric *block* bootstrap.

Idea:

resample data in blocks to preserve the dependence structure within the blocks.

3.2.1 Nonoverlapping Blocks (NBB)

Consider splitting $\boldsymbol{Y} = (Y_1, \dots, Y_n)$ in b consecutive blocks of length ℓ.



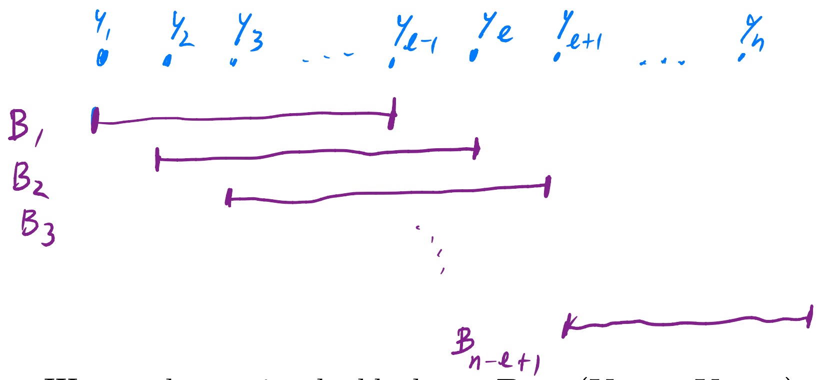
We can then rewrite the data as $\boldsymbol{Y} = (\boldsymbol{B}_1, \dots, \boldsymbol{B}_b)$ with $\boldsymbol{B}_k = (Y_{(k-1)\ell+1}, \dots, Y_{k\ell})$, $k = 1, \dots, b$. $b = \lfloor \frac{n}{\ell} \rfloor$ "floor function" = round down.

1. Sample nonoverlapping blocks $B_1^*, \dots, B_b^*$ independently from $B_1, \dots, B_b$ (with replacement) to form pseudodata $Y^* = (B_1^*, \dots, B_b^*)$.
2. Statistic of interest θ is estimated from Y^* to create $\hat{\theta}^*$.
3. Repeat 1-2 B times to obtain $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$ with which to estimate the dsn of $\hat{\theta}$.

Note, the order of data within the blocks must be maintained, but the order of the blocks that are resampled does not matter.

3.2.2 Moving Blocks (MBB)

Now consider splitting $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ into overlapping blocks of adjacent data points of length ℓ.



now we have more blocks to choose from!

We can then write the blocks as $\boldsymbol{B}_k = (Y_k, \ldots, Y_{k+\ell-1})$, $k = 1, \ldots, n - \ell + 1$.

$b = \lfloor \frac{n}{\ell} \rfloor$

① Create pseudo data by resampling independently w/ replacement the blocks $B_1^*, \ldots, B_b^*$ from $B_1, \ldots, B_{n-\ell+1}$ and form pseudo data $Y^* = (B_1^*, \ldots, B_b^*)$.

② Calculate $\hat{\theta}^*$ from Y^*

③ Repeat 1–2 B times to get $\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*$.

3.2.3 Choosing Block Size

If the block length is too short,

The resampling cannot capture the dependence ($l=1$ is the iid bootstrap)

If the block length is too long,

not many blocks to sample (does not resemble data generation)

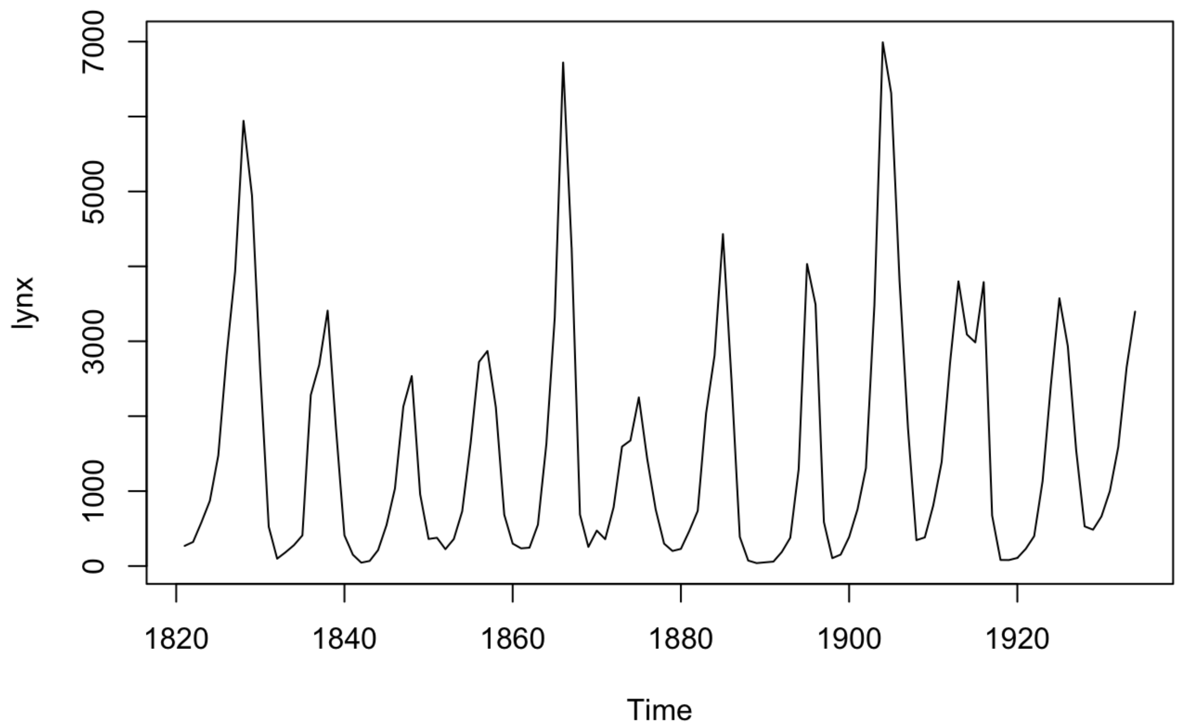
Asymptotic result: block length should increase w/ length of the time series. If so, MBB and NBB produce consistent estimators of moments, correct coverage probabilities for CIs and correct error rates for tests.

There are practical methods for choosing l (Lahiri, 2003)

Your Turn

We will look at the annual numbers of lynx trappings for 1821–1934 in Canada. Taken from Brockwell & Davis (1991).

```
data(lynx)
plot(lynx)
```



Goal: Estimate the sample distribution of the mean $\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} Y_i$

```
theta_hat <- mean(lynx)
theta_hat
```

```
## [1] 1538.018
```

3.2.4 Independent Bootstrap

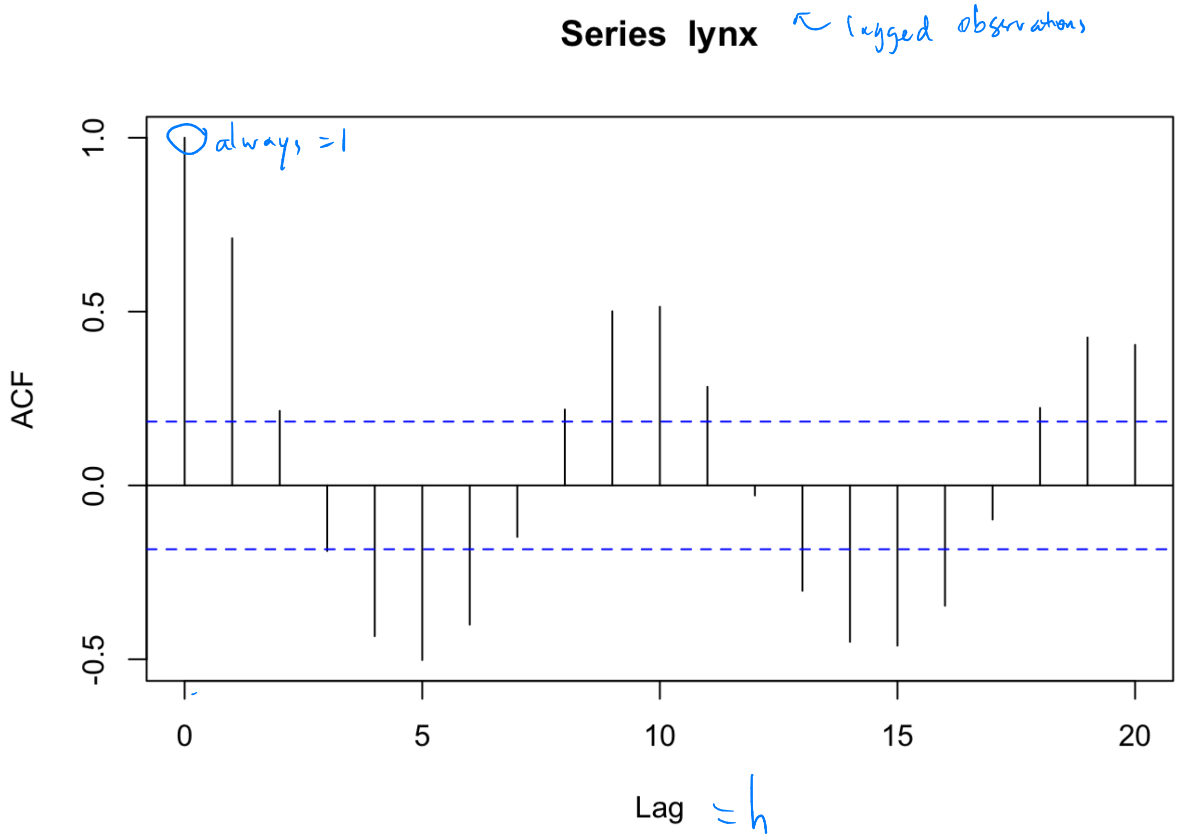
```r
library(simpleboot)
B <- 10000

## Your turn: perform the independent bootstrap
## what is the bootstrap estimate se?
```

We must account for the dependence to obtain a correct estimate of the variance!

```r
acf(lynx)
```

Auto correlation function $\gamma(h) = \mathrm{Cor}(y_t, y_{t+h})$



The acf (autocorrelation) in the dominant terms is positive, so we are *underestimating* the standard error.

3.2.5 Non-overlapping Block Bootstrap

```
# function to create non-overlapping blocks
nb <- function(x, b) {
  n <- length(x)
  l <- n %/% b

  blocks <- matrix(NA, nrow = b, ncol = l)
  for(i in 1:b) {
    blocks[i, ] <- x[((i - 1)*l + 1):(i*l)]
  }
  blocks
}

# Your turn: perform the NBB with b = 10 and l = 11
theta_hat_star_nbb <- rep(NA, B)
nb_blocks <- nb(lynx, 10)
for(i in 1:B) {
  # sample blocks
  # get theta_hat^*
}

# Plot your results to inspect the distribution
# What is the estimated standard error of theta hat? The Bias?
```

3.2.6 Moving Block Bootstrap

```
# function to create overlapping blocks
mb <- function(x, l) {
  n <- length(x)
  blocks <- matrix(NA, nrow = n - l + 1, ncol = l)
  for(i in 1:(n - l + 1)) {
    blocks[i, ] <- x[i:(i + l - 1)]
  }
  blocks
}

# Your turn: perform the MBB with l = 11
mb_blocks <- mb(lynx, 11)
theta_hat_star_mbb <- rep(NA, B)
for(i in 1:B) {
  # sample blocks
  # get theta_hat^*
```

```
}

# Plot your results to inspect the distribution
# What is the estimated standard error of theta hat? The Bias?
```

3.2.7 Choosing the Block size

```
# Your turn: Perform the mbb for multiple block sizes l = 1:12
# Create a plot of the se vs the block size. What do you notice?
```

4 Summary

as opposed to Bayesian inference

Bootstrap methods are simulation methods for frequentist inference.

Bootstrap methods are useful for

especially when model assumptions are invalid.

Bootstrap methods can fail when

We have extremes or heavy tailed dsns.

can be be computationally intensive (slow).

need to be careful w/ dependence.