Your Turn

This data set is the Puromycin data in R. The goal is to create a regression model about the rate of an enzymatic reaction as a function of the substrate concentration.

```
head(Puromycin)
```

```
##   conc rate   state
## 1 0.02   76 treated
## 2 0.02   47 treated
## 3 0.06   97 treated
## 4 0.06  107 treated
## 5 0.11  123 treated
## 6 0.11  139 treated
```

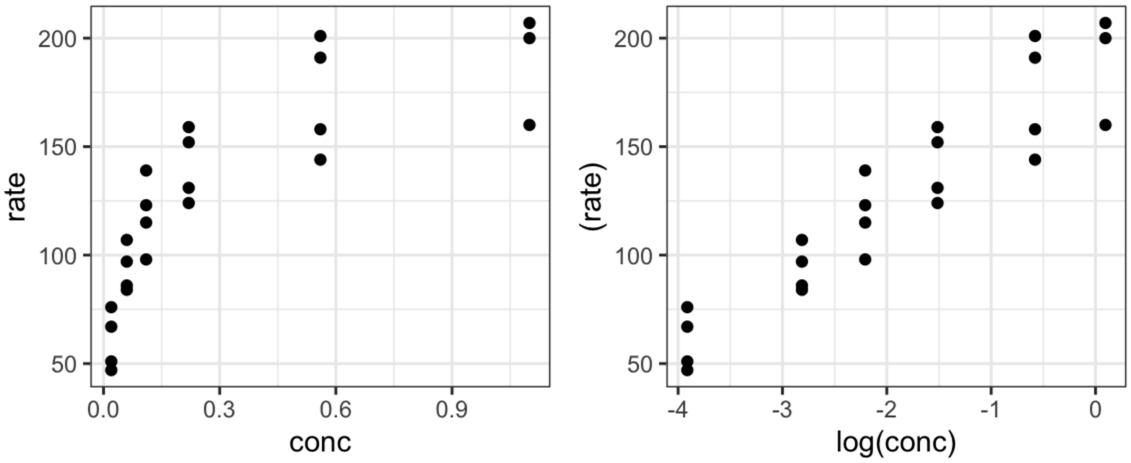
```
dim(Puromycin)
```

n=23

```
## [1] 23  3
```

```
ggplot(Puromycin) +
  geom_point(aes(conc, rate))

ggplot(Puromycin) +
  geom_point(aes(log(conc), (rate)))
```



treated

untreated

ignore for now

2.1.4 Standard regression

```
m0 <- lm(rate ~ conc, data = Puromycin)
plot(m0)
summary(m0)
```

```
##
## Call:
## lm(formula = rate ~ conc, data = Puromycin)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.861 -15.247  -2.861  15.686  48.054
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    93.92       8.00   11.74 1.09e-10 ***
## conc          105.40      16.92    6.23 3.53e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.82 on 21 degrees of freedom
## Multiple R-squared:  0.6489, Adjusted R-squared:  0.6322
## F-statistic: 38.81 on 1 and 21 DF,  p-value: 3.526e-06
```

```
confint(m0)
```

```
##                2.5 %   97.5 %
## (Intercept) 77.28643 110.5607
## conc        70.21281 140.5832
```

```
m1 <- lm(rate ~ log(conc), data = Puromycin)
plot(m1)
summary(m1)
```

```
##
## Call:
## lm(formula = rate ~ log(conc), data = Puromycin)
```

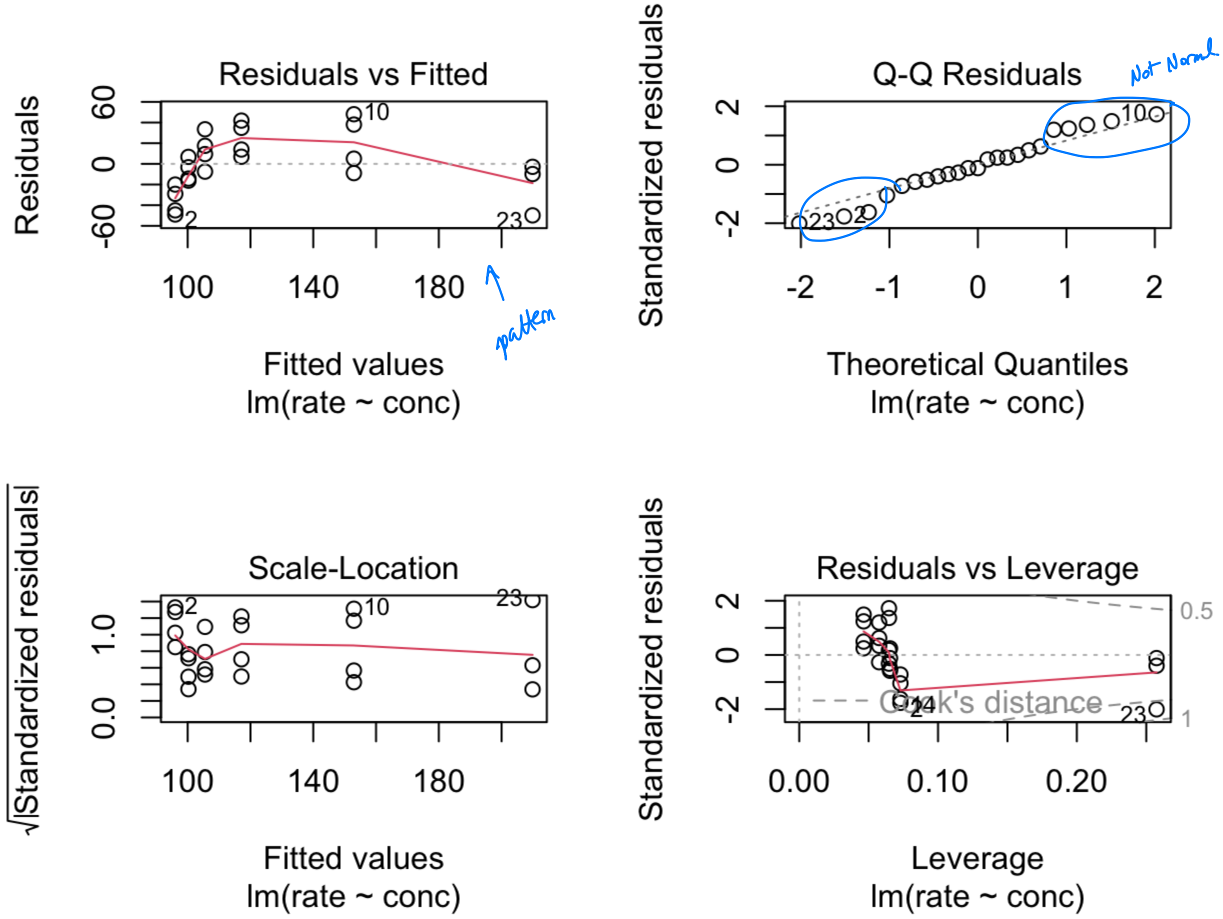
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.250 -12.753   0.327  12.969  30.166
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  190.085      6.332   30.02  < 2e-16 ***
## log(conc)     33.203      2.739   12.12 6.04e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.2 on 21 degrees of freedom
## Multiple R-squared:  0.875,  Adjusted R-squared:  0.869
## F-statistic: 146.9 on 1 and 21 DF,  p-value: 6.039e-11
```

```
confint(m1)
```

```
##                 2.5 %   97.5 %
## (Intercept) 176.91810 203.2527
## log(conc)    27.50665  38.8987
```

← based on asymptotic normality of MLE + Fisher Information.

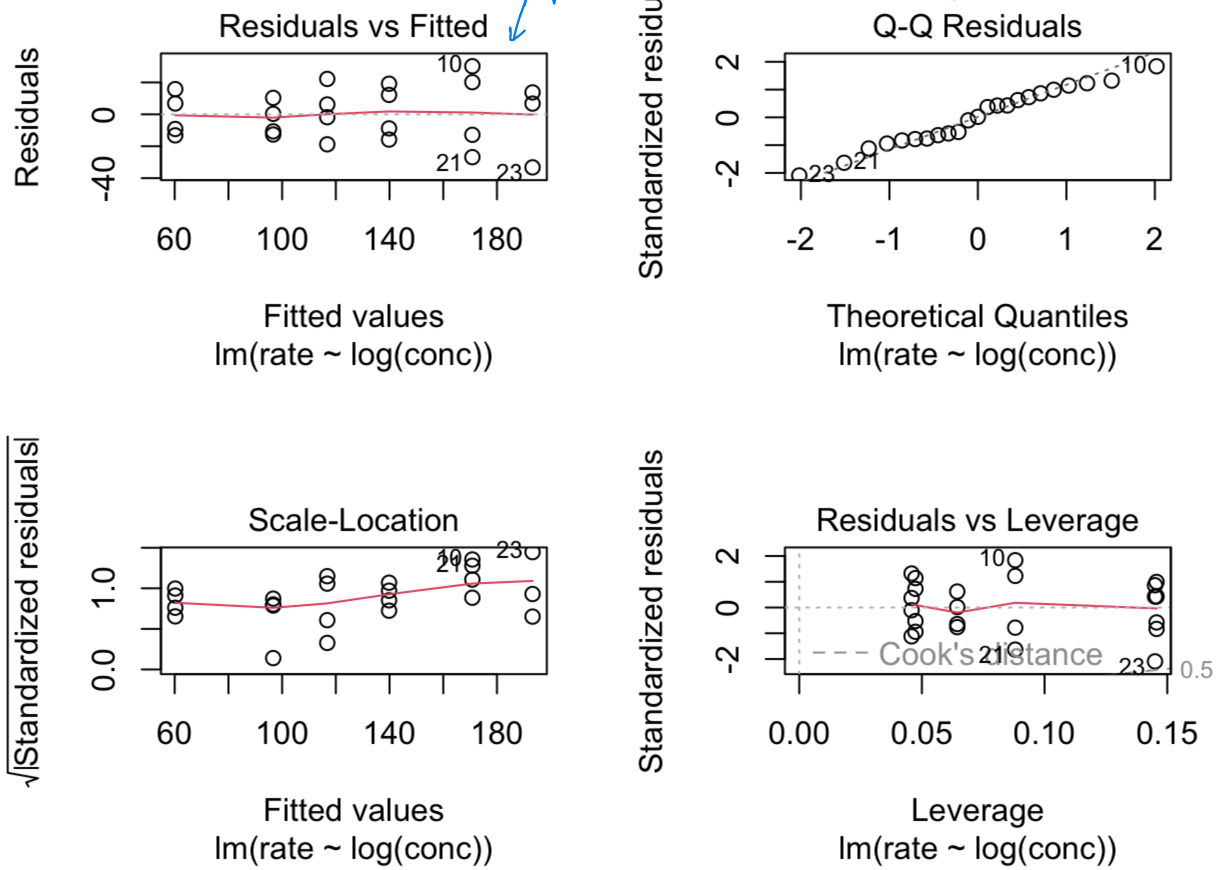
MO


Residuals vs Fitted
Residuals
-60
0
60
100
140
180
Fitted values
lm(rate ~ conc)
10
2
23
pattern
Q-Q Residuals
Not Normal
Standardized residuals
-2
0
2
-2
-1
0
1
2
Theoretical Quantiles
lm(rate ~ conc)
10
23
2
Scale-Location
√|Standardized residuals|
0.0
1.0
100
140
180
Fitted values
lm(rate ~ conc)
2
10
23
Residuals vs Leverage
Standardized residuals
-2
0
2
0.00
0.10
0.20
Leverage
lm(rate ~ conc)
Cook's distance
0.5
1
24
23

M1

slight heteroskedasticity.

looks better



2.1.5 Paired bootstrap

```r
# Your turn
library(boot)

reg_func <- function(dat, idx) {
  # write a regression function that returns fitted beta
}
                    or write your own is fine.
# use the boot function to get the bootstrap samples

# examing the bootstrap sampling distribution, make histograms

# get confidence intervals for beta_0 and beta_1 using boot.ci
```

2.1.6 Bootstrapping the residuals

```r
# Your turn
library(boot)

reg_func_2 <- function(dat, idx) {
  # write a regression function that returns fitted beta
  # from fitting a y that is created from the residuals

}

# use the boot function to get the bootstrap samples

# examing the bootstrap sampling distribution, make histograms

# get confidence intervals for beta_0 and beta_1 using boot.ci
```

3 Bootstrapping Dependent Data

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

No longer assuming $Y_1, \dots, Y_n$ independent.

↳ could be time series, spatial, network, etc.

Goal:

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

Challenge:

Since Y_i's are dependent it is inappropriate to use the iid bootstrap.

Bootstrapped samples would no longer reproduce the data generating process.

(and sampling independently from $\hat{F}_n$ no longer mimics drawing original sample from F).

We will consider 2 approaches

① Model-based (parametric).

② Block bootstrap (nonparametric).

Example 3.1 Suppose we observe a time series $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ which we assume is generated by an AR(1) process, i.e.,

$$Y_t = \alpha Y_{t-1} + \varepsilon_t \quad t = 1, \ldots, n$$

$$|\alpha| < 1 \text{ and } \varepsilon_1, \ldots, \varepsilon_n \overset{iid}{\sim} (0, \sigma^2).$$

Why not just move forward with our nonparametric bootstrap procedure?

Failure of nonparametric iid bootstrap for TS data.

"almost" independent

Let's suppose $\{X_t\}_{t \in \mathbb{Z}}$ is a stationary, m-dependent process with $EX_t = \mu$, $EX_t^2 < \infty$.

joint probability dsns don't change with time.

$$(X_{t_1}, \ldots, X_{t_k}) \overset{d}{=} (X_{t_1+h}, \ldots, X_{t_k+h})$$

for any $t_1, \ldots, t_k, h$

let $r(k) = \mathrm{Cov}(X_1, X_{1+k})$.

"m-dependent": $r(k) = 0$ for $k > m$.

This a stronger assumption than AR(1) on the dependence.

Say we want to approximate dsn of $T_n = \sqrt{n}(\bar{X}_n - \mu)$. Say we apply iid bootstrap:

Draw observations from $\{X_1, \ldots, X_n\}$ to get $T_n^* = \sqrt{n}(\bar{X}_n^* - E_* X_1^*) = \sqrt{n}(\bar{X}_n^* - \bar{X}_n)$ and approximate $P(T_n \le x)$ with $P_*(T_n^* \le x)$, $x \in \mathbb{R}$.

Th'm: If $X_1, X_2, \ldots$ stationary, m-dependent process w/ $\mathrm{Var}(X_1) = \sigma^2 < \infty$, then

$$\sup_{x \in \mathbb{R}} \left| P_*(T_n^* \le x) - \Phi\left(\frac{x}{\sigma}\right) \right| \equiv \Delta_n \to 0 \quad \text{as} \quad n \to \infty \quad \text{a.s.}$$

Proof is very similar to iid version (pg. 6-7). Just relies on M-Z SLLN to hold for m-dependent process, which it does.

Problem? If $\{X_t\}$ is stationary + m-dependent, then

comes from dominated convergence theorem

m-dependence

$$\lim_{n\to\infty} \mathrm{Var}(\sqrt{n}\bar{X}_n) = \lim_{n\to\infty} \mathrm{Var}(T_n) = \sum_{k=-\infty}^{\infty} r(k) = \sum_{-m}^{m} r(k) \equiv \sigma_\infty^2.$$

If $\sigma_\infty^2 > 0$, then $T_n \equiv \sqrt{n}(\bar{X}_n - \mu) \xrightarrow{d} N(0, \sigma_\infty^2)$. by a CLT.

So iid bootstrap will fail unless $\sigma_\infty^2 = \sigma^2 = r(0)$

In practice, $\sigma_\infty^2 > r(0)$ holds most often $\Rightarrow$ we are underestimating uncertainty w/ iid bootstrap!

This was for m-dependent process, which is a very strong assumption! Under more realistic process, may be even worse.

3.1 Model-based approach

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

↳ turn our problem into iid bootstrap.

Recall AR(1): $Y_t = \alpha Y_{t-1} + \varepsilon_t \quad t=1,\dots,n \quad |\alpha|<1$ and $\varepsilon_1,\dots,\varepsilon_n \overset{iid}{\sim} (0,\sigma^2)$. !!

① Estimate $\hat{\alpha}$ from data (fit the model).

② Define estimated "innovations" $\hat{e}_t = Y_t - \hat{\alpha} Y_{t-1}$, $t=2,\dots,n$

and $\bar{\hat{e}} = \frac{1}{n-1}\sum_{t=2}^{n} \hat{e}_t$

③ Define the residuals as centered innovations.

$$\hat{\varepsilon}_t = \hat{e}_t - \bar{\hat{e}} \qquad [E\varepsilon_i = 0]$$

④ For $r=1,\dots,R$

a) Create a bootstrap sample $\hat{\varepsilon}_0^*,\dots,\hat{\varepsilon}_n^*$ by randomly sampling $n+1$ values from the $n-1$ values $\hat{\varepsilon}_t$, $t=2,\dots,n$.

b) Construct pseudo data $Y^* = (Y_1^*,\dots,Y_n^*)$ from

$$Y_0^* = \hat{\varepsilon}_0^*, \quad Y_t^* = \hat{\alpha} Y_{t-1}^* + \hat{\varepsilon}_t^*, \quad t=1,\dots,n.$$

c) define $\hat{\alpha}_r^*$ as the estimate of α from $Y_1^*,\dots,Y_n^*$.

⑤ dsn of $\hat{\alpha}_1^*,\dots,\hat{\alpha}_r^*$ is bootstrap estimate of dsn of $\hat{\alpha}$.

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

As we know, this may not always be a good assumption.

3.2 Nonparametric approach

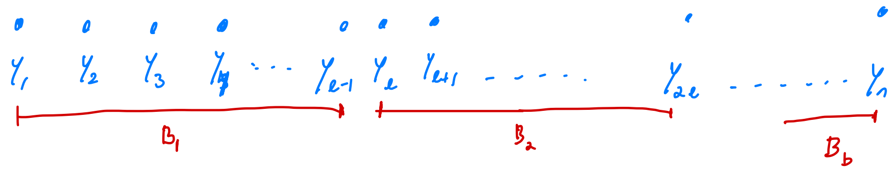
To deal with dependence in the data, we will employ a nonparametric *block* bootstrap.

Idea:

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

3.2.1 Nonoverlapping Blocks (NBB) Carlstein (1986).

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



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

① Sample nonoverlapping blocks $B_1^*, \ldots, B_b^*$ independently from $B_1, \ldots, B_b$ with replacement to form pseudo data set $Y^* = (B_1^*, \ldots, B_b^*)$.

② estimate statistic of interest from Y^* to get $\hat{\theta}^*$.

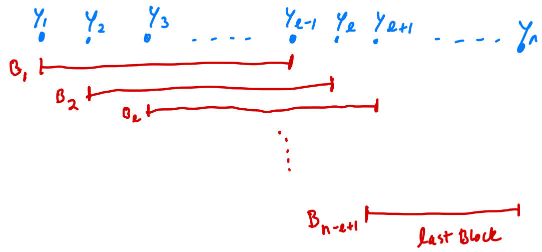
③ Repeat ①-② R times to obtain $\hat{\theta}^{*(1)}, \ldots, \hat{\theta}^{*(R)}$ to estimate dsn of $\hat{\theta}$.

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)

Künsch (1989)
Liu & Singh (1992).

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


Y_1 Y_2 Y_3 .... $Y_{\ell-1}$ Y_ℓ $Y_{\ell+1}$ .... Y_n
B_1
B_2
B_ℓ
$B_{n-\ell+1}$
last Block

Now we have more blocks to choose from! $\left(N = n-\ell+1 \quad \text{vs.} \quad b = \lfloor \frac{n}{\ell} \rfloor\right)$.

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

get/collect blocks $\mathcal{C} \equiv \{B_1, \ldots, B_N\}$ sampling $B_1^*, \ldots, B_b^*$ from $\mathcal{C}$, $b = \lfloor \frac{n}{\ell} \rfloor$, put together to get $Y^* = (B_1^*, \ldots, B_b^*)$.

Alternative but equivalent formulation: let $I_1, \ldots, I_b$ be iid w/ $P(I_1 = j) = \frac{1}{N}$, $j = 1, \ldots, N$ set $B_i^* = B_{I_i}^*$ $i = 1, \ldots, b$.

Ex: let $\hat{\theta}_n = \bar{Y}_n$. Get MBB sample mean version $\bar{Y}_m^* = \sum_{i=1}^m Y_i^*/\underset{m = b\cdot\ell}{\underbrace{m}}$. Find $E_*(\bar{Y}_m^*)$ and $\text{Var}_*(\sqrt{m}\,\bar{Y}_m^*)$ which estimate $E(\bar{Y}_n)$ and $\text{Var}(\sqrt{n}\,\bar{Y}_n)$.

Note: $\bar{Y}_m^* = \frac{1}{b}\sum_{i=1}^{b} \bar{Y}_{B_i^*}^* \leftarrow$ sample mean of ith block B_i^*, $b = \lfloor \frac{n}{\ell} \rfloor$

$$① \; E_*(\bar{Y}_m^*) = \frac{1}{b}\sum_{i=1}^{b} E_*(\bar{Y}_{B_i^*}^*) \overset{\text{sample blocks iid}}{=} E_*(\bar{Y}_{B_i^*}^*) = \frac{1}{\underbrace{n-\ell+1}_{\text{uniform blocks}}} \sum_{i=1}^{\overset{\# \text{ blocks}}{n-\ell+1}} \left(\sum_{t=i}^{i+\ell-1} Y_t/\ell\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N} \bar{Y}_i \quad \text{where } \bar{Y}_i = \text{sample mean of block } B_i \text{ and } N = n-\ell+1.$$

$$\neq \bar{Y}_n$$

$$② \; \text{Var}_*(\sqrt{m}\,\bar{Y}_m^*) = \text{Var}_*\left(\sqrt{m}\,\frac{1}{b}\sum_{i=1}^{b} \bar{Y}_{B_i^*}^*\right) = \frac{m}{b^2}\sum_{i=1}^{b} \text{Var}_*(\bar{Y}_{B_i^*}^*) = \frac{\overset{b\cdot\ell}{m}}{b^2}\cdot b\, \text{Var}_*(\bar{Y}_{B_i^*}^*)$$

Bootstrap blocks sampled iid

$$= \ell\, E_*\left(\bar{Y}_{B_i^*}^* - E_*\bar{Y}_{B_i^*}^*\right)^2 = \ell\,\frac{1}{N}\sum_{i=1}^{N}(\bar{Y}_i - \hat{\mu})^2 \quad \text{where } \hat{\mu} \equiv \frac{1}{N}\sum_{i=1}^{N}\bar{Y}_i \text{ as above}$$

this looks like a sample variance of $\sqrt{\ell}\,\bar{Y}_1, \ldots, \sqrt{\ell}\,\bar{Y}_N$ of sample means from blocks.

This directly estimates the variance of sample mean of length ℓ block $\sqrt{\ell}\,\bar{Y}_i$.

$\Rightarrow \text{Var}_*(\sqrt{m}\,\bar{Y}_m^*)$ estimate $\text{Var}(\sqrt{\ell}\,\bar{Y}_1) = \ell\,\text{Var}\,\bar{Y}_1 \approx n\,\text{Var}\,\bar{Y}_n$ (target for MBB).

Both NBB and MBB fix the variance issue from page 35.

NOTE: The MBB version of $\sqrt{n}(\bar{Y}_n - \mu) = \sqrt{n}(\bar{Y}_n - E\bar{Y}_n)$ is NOT $\sqrt{m}(\bar{Y}_m^* - \bar{Y}_n)$!

is actually $\sqrt{m}(\bar{Y}_m^* - E_*\bar{Y}_m^*) = \sqrt{m}(\bar{Y}_m^* - \hat{\mu})$

$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N}\bar{Y}_i$.