

# Objectives

## Bootstrapping

FW8051 Statistics for Ecologists

Department of Fisheries, Wildlife and Conservation Biology



- To understand how a bootstrap can be used to quantify uncertainty, particularly when assumptions of linear regression may not be met.
- To further your coding skills by implementing a cluster-level bootstrap appropriate for certain types of data sets containing repeated observations.

## RIKZ [Dutch governmental institute] data

Abundances from ~ 75 invertebrate species measured on various beaches along Dutch Coast.



Selected from: Zuur, A. F., E. N. Ieno, and G. M. Smith. 2009. Analyzing Ecological Data. Springer, New York.

## RIKZ data

### Sampling Effort:

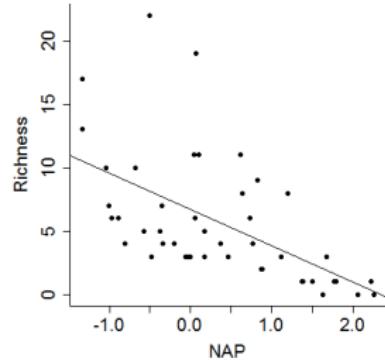
- 9 beaches (high, medium, low exposure)
- 5 stations at each beach.

### Variables:

- Richness = species richness (number of species counted).
- NAP = height of the sample site (relative to sea level).
  - Lower values imply more time spent under water.
  - Higher values typically found further up the beach.



Gerard Janssen, RIKZ



$$\hat{Y}_i = 6.886 - 2.867X_i + E_i$$

## Regression Output

```
lmfit.R<-lm(Richness~NAP, data=RIKZdat)
summary(lmfit.R)
```

Call:  
`lm(formula = Richness ~ NAP, data = RIKZdat)`

Residuals:  

Min	1Q	Median	3Q	Max
-5.0675	-2.7607	-0.8029	1.3534	13.8723

Coefficients:  

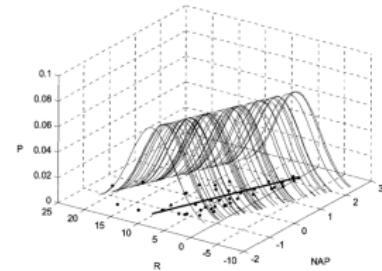
	Estimate	Std. Error	t value	Pr(> t )		
(Intercept)	6.6857	0.6578	10.164	5.25e-13 ***		
NAP	-2.8669	0.6307	-4.545	4.42e-05 ***		
---						
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1

Residual standard error: 4.16 on 43 degrees of freedom  
Multiple R-squared: 0.3245, Adjusted R-squared: 0.3088  
F-statistic: 20.66 on 1 and 43 DF, p-value: 4.418e-05

## Model for the Data

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2)$$



\end{center}

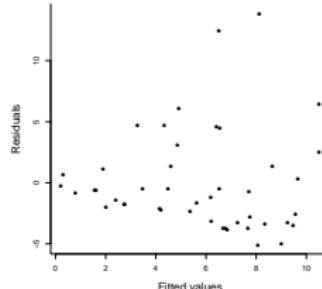
# Linear Regression

## Residual Plots: Constant Variance?

```
# Could use standardized residuals, rstandard(lmfit), instead
plot(lmfit.R$resid~lmfit.R$fitted, xlab="Fitted values",
     ylab="Residuals", pch=16, bty="L", cex.lab=1.4)
```

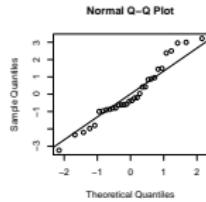
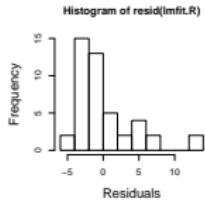
### Assumptions (HILE Gauss):

- Homogeneity of variance (constant scatter about the line);  
 $\text{var}(\epsilon_i) = \sigma^2$
- Independence: Correlation( $\epsilon_i, \epsilon_j$ ) = 0
- Linearity:  $E[Y_i | X] = \beta_0 + X_i\beta_1$
- Existence (we observe random variables that have finite variance)
- **Gauss:**  $\epsilon_i$  come from a Normal (Gaussian) distribution



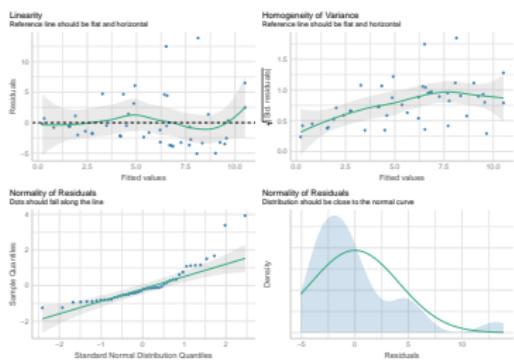
### Normality?

```
par(mfrow=c(1,2))
# Could use Studentized residuals, rstudent(lmfit), instead
hist(resid(lmfit.R), xlab="Residuals", pch=16, bty="L", cex.lab=1.4)
qqnorm(resid(lmfit.R))
qqline(resid(lmfit.R))
```

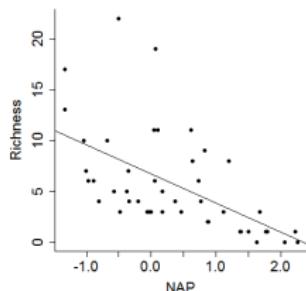


### Assumptions

```
performance::check_model(lmfit.R,
  check = c("linearity", "homogeneity", "qq", "normality"))
```



## Linearity?



$$\widehat{\text{Richness}}_i = 6.886 - 2.867 * \text{NAP}_i + \epsilon_i$$

What if  $\text{NAP} > 2.3$ ?

## Independence

Are 2 observations from the same beach more alike than 2 observations from 2 different beaches (after accounting for NAP)?

Consequence? SE's for  $\beta$  too small?

## Solutions?

- Transformations (constant variance, normality, linearity)
  - $\log(\text{Richess}) = \beta_0 + \beta_1 \text{NAP}$  ?
- Add other variables (can help if patterns in residuals or non-constant variance)?
- Can model mean and variance using "Generalized least squares" (glu) (Ch 4 of Zuur et al.)

$$Y_i = \beta_0 + X_i\beta + \epsilon_i, \text{ with } \epsilon_i \sim N(0, \sigma^2 f(X_i; \theta))$$

- Bootstrap (to deal with non-constant variance, clustering)

## Assumptions

Linear regression relies on least-squares:

$$\hat{\beta} \text{ minimizes: } \sum_{i=1}^n (Y_i - [\beta_0 + X_i\beta_1])^2.$$

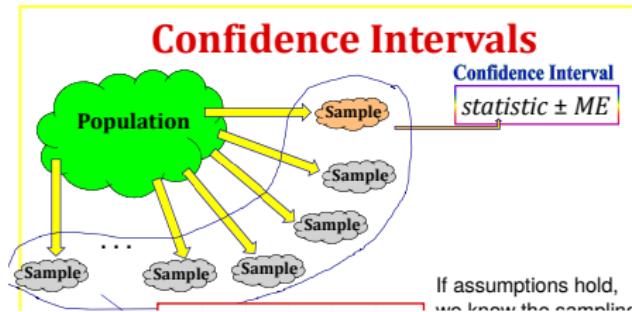
Least-squares can be motivated from just the linearity and independence assumptions:

- $Y_i = X_i\beta + \epsilon_i$
- $\epsilon_i$  is symmetric about 0 (possibly not-normally distributed, non-constant variance)

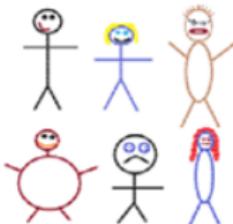
Normality is needed for inference (sampling distribution of  $\frac{\hat{\beta} - \beta}{SE(\beta)} \sim t_{n-p}$ )

To relax these assumptions, we will consider using a bootstrap for inference.

## Reality



Suppose we have a random sample of 6 people:

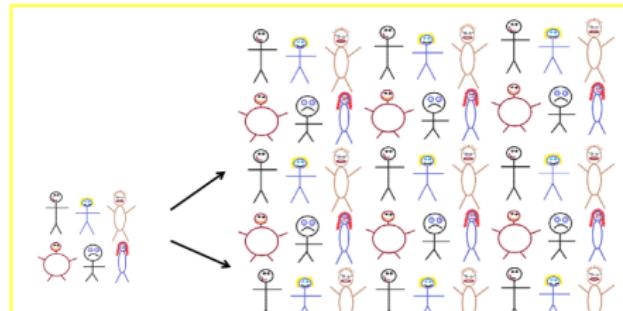


We only have 1 sample, what do we do? **BOOTSTRAP**

- Imagine the “population” is many, many copies of the original sample
- Resample this population many times and fit the model to each data set

What do you have to assume?

- The sample is representative of the population (otherwise, it's pointless to try to generalize from sample to population)



## Sampling with Replacement

- To simulate a sampling distribution, we can just take repeated random samples from this “population” made up of many copies of the sample
- In practice, we can't actually make infinite copies of the sample...
- ... but we can do this by sampling with replacement from the sample we have (each unit can be selected more than once)

## Bootstrap Sample

Your original sample has data values

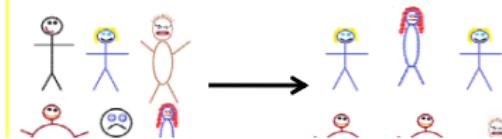
18, 19, 19, 20, 21

Is the following a possible bootstrap sample: 18, 19, 20, 21, 22 **No**,  
**22 is not a value from the original sample**

Is the following a possible bootstrap sample: 18, 19, 20, 21 **NO**,  
**Bootstrap samples must be the same size as the original sample**

Is the following a possible bootstrap sample: 18, 18, 19, 20, 21 **YES**.  
**Same size, could be gotten by sampling with replacement**

**Bootstrap Sample:** Sample with replacement from the original sample, using the same sample size.



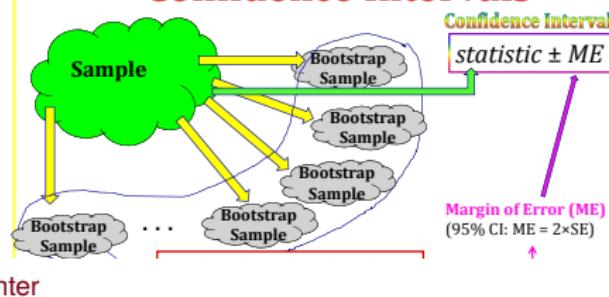
## Bootstrap

- A **bootstrap sample** is a random sample taken with replacement from the original sample, of the same size as the original sample
- A **bootstrap statistic** is the statistic computed on a bootstrap sample
- A **bootstrap distribution** is the distribution of many bootstrap statistics

## Why “Bootstrap”?



### Confidence Intervals



“Pull yourself up by your bootstraps”

- Lift yourself in the air simply by pulling up on the laces of your boots
- Metaphor for accomplishing an “impossible” task without any outside help

### Standard Error

- The sampling distribution is centered on the population parameter
- The bootstrap distribution is centered on the sample statistic
- Luckily, we don't care about the center...we care about the **variability!**

- The variability of the bootstrap statistics should be similar to the variability of the sample statistics
- The standard error of a statistic can be estimated using the standard deviation of the bootstrap distribution!

## Bootstrap Sampling Must Mimic Original Sampling Design

What do we do if we have clustered data (multiple obs. on same sampling unit)?

- Sample clusters with replacement (keeping all observations associated with the cluster)
- Number of clusters in the bootstrap sample is the same as in the original sample
- Treats "clusters" as independent

## Cluster-level bootstrap

This approach is most appropriate when:

- clusters have equal numbers of observations (all have 5 in this case)
- our interest lies in predictors that do not vary within a cluster

## RIKZ Data - Bootstrapping

Cluster-level bootstrapping

Field Plot Layout

A	B	C
D	E	F



Original Sample

Plot A (4, 2, 1, 6, 2)  
Plot B (7, 3, 2, 1, 1)  
Plot C (4, 2, 3, 2, 4)  
Plot D (6, 4, 3, 3, 1)  
Plot E (5, 4, 2, 3, 1)  
Plot F (2, 4, 3, 5, 6)

## Cluster-Level Bootstrapping Example

Bootstrap Sample  
Plot B (7, 3, 2, 1, 1)  
Plot D (6, 4, 3, 3, 1) →  
Plot A (4, 2, 1, 6, 2)  
Plot C (4, 2, 3, 2, 4)  
Plot E (5, 4, 2, 3, 1)  
Plot F (2, 4, 3, 5, 6)

## Cluster-level Bootstrap: Sample Beaches with replacement

The code below shows how you can get 1 bootstrap data set using cluster resampling:

```
# let's use the bootstrap to deal with non-constant variance and clustering
uid<-unique(RIKZdat$Beach)
nBeach<-length(uid)

# Single bootstrap sample & fit
(bootids<-data.frame(Beach=sample(uid, nBeach, replace=T)))

Beach
1   3
2   6
3   4
4   8
5   1
6   5
7   7
8   5
9   1

bootdat<-merge(bootids, RIKZdat)
table(bootdat$Beach)
```

1 3 4 5 6 7 8  
10 5 5 10 5 5 5