

Simulation and resampling analysis in R

Author: Nicholas G Reich, Jeff Goldsmith, Andrea S Foulkes, Gregory Matthews

*This material is part of the **statsTeachR** project*

Made available under the Creative Commons Attribution-ShareAlike 3.0 Unported License: <http://creativecommons.org/licenses/by-sa/3.0/deed.en-US>

Module learning goals

At the end of this module you should be able to...

- ▶ simulate data from a parametric distribution.
- ▶ Design and implement a resampling simulation experiment to test a hypothesis.
- ▶ Run simulations in parallel, when appropriate.

What is simulation?

Definitions

- ▶ Broadly: “The technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus, esp. for the purpose of study or personnel training.” (from the *OED*)
- ▶ In science: Creating a model that imitates a physical or biological process.
- ▶ In statistics: The generation of data from a model using rules of probability.

Simple examples of simulations

- ▶ Drawing pseudo-random numbers from a probability distribution (e.g. proposal distributions, ...).
- ▶ Generating data from a specified model (e.g. building a template dataset to test a method, calculating statistical power).
- ▶ Resampling existing data (e.g. permutation, bootstrap).

What simulations have you run?

Basic simulation: Random number generation in R

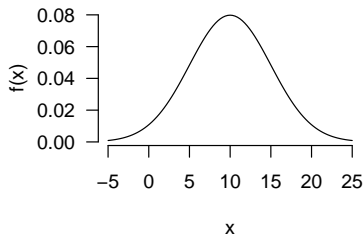
`rnorm()`, `rpois()`, etc...

Built-in functions for simulating from parametric distributions.

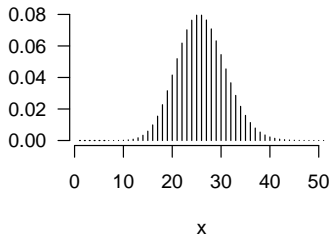
```
y <- rnorm(100, mean = 10, sd = 5)
(p <- rpois(5, lambda = 25))

## [1] 25 26 25 25 33
```

`dnorm(x, mean=10, sd=5)`



`dpois(x, lambda=25)`



Basic Simulation: Resampling data in R

`sample()`

Base R function for sampling data (with or without replacement).

```
p
## [1] 25 26 25 25 33

sample(p, replace = FALSE)

## [1] 25 26 33 25 25

sample(p, replace = TRUE)

## [1] 25 25 33 25 33
```

Generating data from a model

A Simple Linear Regression model

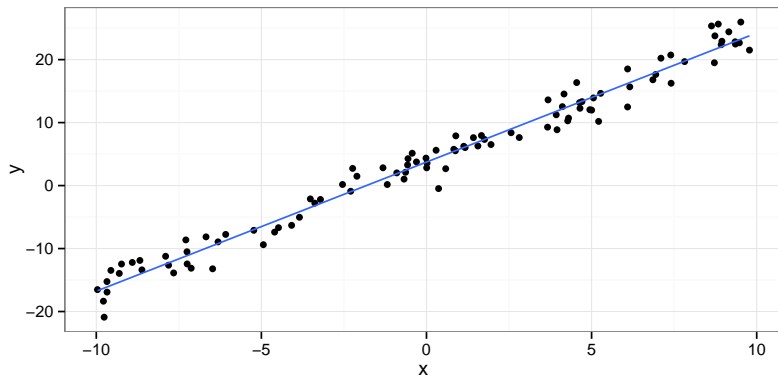
$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

What is needed to simulate data (i.e. Y_i) from this model?

- ▶ The X_i : fixed quantities.
- ▶ Error distribution: e.g. $\epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$.
- ▶ Values for parameters: $\beta_0, \beta_1, \sigma^2$.

Generating data from $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$

```
require(ggplot2)
n <- 100; b0=4; b1=2; sigma=2      ## define parameters
x <- runif(n, -10, 10)            ## fix the X's
eps <- rnorm(n, sd=sigma)         ## simulate the e_i's
y <- b0 + b1*x + eps              ## compute the y_i's
qplot(x, y, geom=c("point", "smooth"), method="lm", se=FALSE)
```



Example data: heights of mothers and daughters

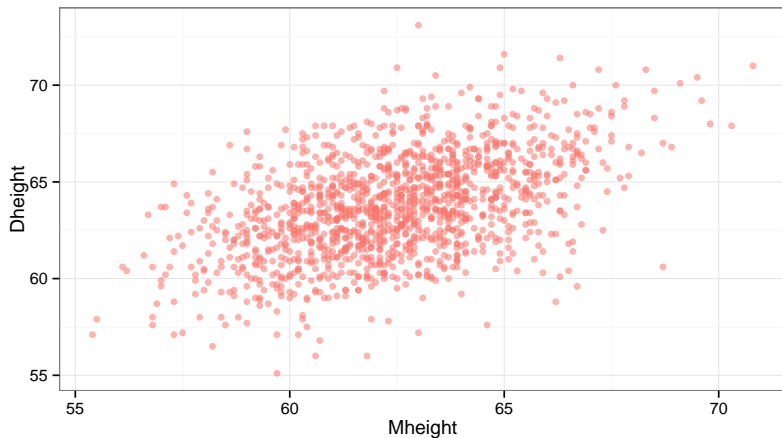
Heights of $n = 1375$ mothers in the UK under the age of 65 and one of their adult daughters over the age of 18 (collected and organized during the period 1893–1898 by the famous statistician Karl Pearson)

```
require(alr3)
data(heights)
head(heights)
```

```
##      Mheight Dheight
## 1      59.7    55.1
## 2      58.2    56.5
## 3      60.6    56.0
## 4      60.7    56.8
## 5      61.8    56.0
## 6      55.5    57.9
```

Example data: heights of mothers and daughters

```
qplot(Mheight, Dheight, data=heights, col="red", alpha=.5) +  
  theme(legend.position="none")
```



One way to draw inference about height association

Using normality assumptions and simple linear regression

$$Dheight_i = \beta_0 + \beta_1 \cdot Mheight_i + \epsilon_i$$

```
mod1 <- lm(Dheight ~ Mheight, data = heights)
summary(mod1)$coefficients
```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  29.9174    1.62247   18.44 5.212e-68
## Mheight      0.5417     0.02596   20.87 3.217e-84
```

Inference without Normality

What if normality assumptions are not valid?

- In large samples, $\hat{\beta}$ is approximately Normal even if the errors are not
- In smaller samples, especially if Normality may not be justified, the tests we've developed are not valid
- **Permutation tests** can be used to test hypotheses without distributional assumptions
- **Bootstrapping** is a computational method for obtaining empirical distributions for unknown parameters
- Note – bootstrapping can also give empirical distributions for parameters whose true distribution is hard to derive

Permutation tests

Using a simulation-based permutation test

- ▶ This can evaluate evidence for/against a null hypothesis.
- ▶ We are interested in $H_0 : \beta_1 = 0$, i.e. there is no relationship between heights of mother and daughter.
- ▶ The trick: we can easily simulate multiple sets of data that we know have no association!
- ▶ All we need is `sample()`.

```
resampDheight <- sample(heights$Dheight, replace = FALSE)
```

Single permutation results

We can then fit this model

$$Dheight_i^* = \beta_0 + \beta_1 \cdot Mheight_i + \epsilon_i$$

where $Dheight_i^*$ are the permuted daughter heights.
This essentially “generates” data from the null model:

$$Dheight_i^* = \beta_0 + 0 \cdot Mheight_i + \epsilon_i$$

```
mod2 <- lm(resampDheight ~ Mheight, data = heights)
summary(mod2)$coefficients

##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  64.5815    1.86194  34.6851 7.732e-190
## Mheight      -0.0133    0.02979  -0.4463 6.554e-01
```

Permutation tests require repeated samples!

A permutation test algorithm

- ▶ Run original analysis (i.e. fit our linear model), store $\hat{\beta}_1$.
- ▶ For i in $1, 2, \dots, N$:
 - ▶ Permute the Y s.
 - ▶ Re-run original analysis, store $\hat{\beta}_1^{(i)}$.
- ▶ Calculate fraction of the $\hat{\beta}_1^{(i)}$ as or more “extreme” than $\hat{\beta}_1$, from our “null distribution” of $\hat{\beta}_1$ s.

Hands-on exercise

- ▶ We have provided code for you to adapt and run a permutation test.
- ▶ [permutationLab.Rmd](#)

Bootstrap

- The basic idea is that the observed data mimics the underlying distribution, whatever that may be
- Drawing samples (**with replacement**) from the observed data mimics drawing samples from the underlying distribution
- Recalculating regression parameters for the “new” samples gives an idea of the distribution of regression coefficients
- Can use estimates from bootstrap analyses to compute “bootstrap” confidence intervals.

Bootstrapping requires repeated samples!

A bootstrapping algorithm

- ▶ Run original analysis (i.e. fit our linear model), store $\hat{\beta}_1$.
- ▶ For i in $1, 2, \dots, N$:
 - ▶ Resample the Y s **with replacement**.
 - ▶ Re-run original analysis, store $\hat{\beta}_1^{(i)}$.
- ▶ Compute a $(1-\alpha)\%$ confidence interval where the bounds are the $(\frac{\alpha}{2})$ and $(1 - \frac{\alpha}{2})$ percentiles of the distribution of the $\hat{\beta}_1^{(i)}$.

Key differences between permutation tests and bootstrapping

- Both are useful when parametric assumptions may not hold.
- Both are called “resampling” approaches to data analysis.
- Permutation tests permute the data; bootstrapping resamples the data with replacement.
- Permutation tests are used for hypothesis testing; bootstrapping characterizes uncertainty in your estimate.