Simulation and permutation tests in R

a **statsTeachR** resource

Module learning goals

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

- Simulate data from a probability distribution.
- Design and implement a resampling simulation experiment to test a hypothesis.

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?

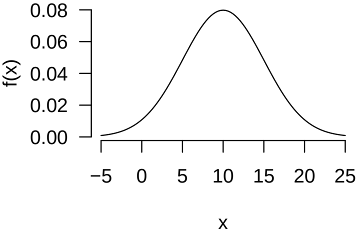
Random number generation in R

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

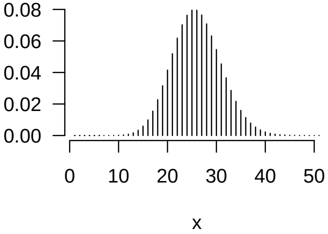
Built-in functions for simulating from parametric distributions.

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

## [1] 27 25 23 22 21
```

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



dpois(x, lambda=25)



Resampling data in R

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

```
p

## [1] 27 25 23 22 21

sample(p, replace=FALSE)

## [1] 23 27 21 22 25

sample(p, replace=TRUE)

## [1] 25 25 23 25 22
```

Generating data from a model

A Simple Linear Regression model

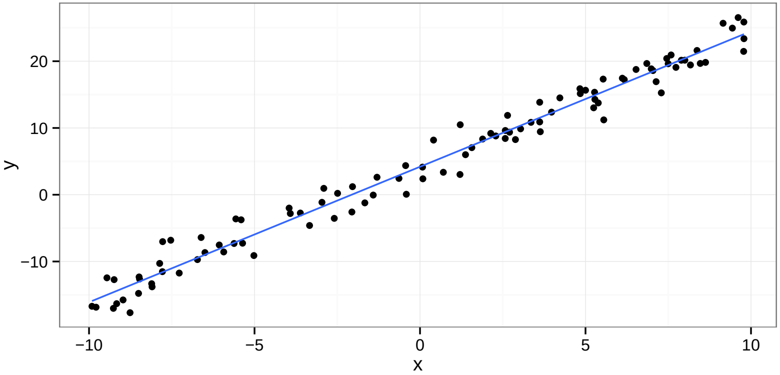
$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

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

- The X_i: fixed quantities.
- Error distribution: e.g. $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$.
- Values for parameters: β_0, β_1, σ^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

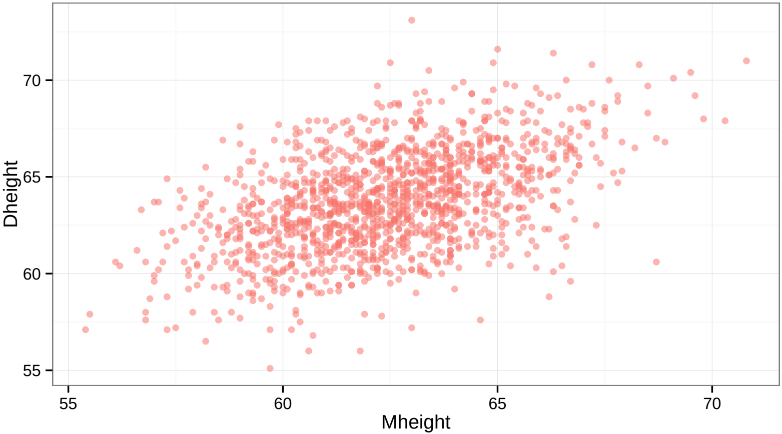
Heights of $n = 1375$ mothers in the UK under the age of 65 and one of their adult daughters over the age of 18 (collected and organized during the period 1893–1898 by the famous statistician Karl Pearson)

```r
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")
```



Are mothers' heights associated with daughters' heights?

Method 1: simple linear regression (must assume normality)

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

We can create a hypothesis test for the null hypothesis $H_0 : \beta_1 = 0$, which in words means "there is no association between mother and daughter heights."

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

##              Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 29.917437 1.62246940 18.43945 5.211879e-68
## Mheight      0.541747 0.02596069 20.86797 3.216915e-84
```

Are mothers' heights associated with daughters' heights?

Method 2: 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()`.

```r
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.
Permuting in essence “generates” new versions of data assuming that daughter heights are independent of mother heights, i.e.

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

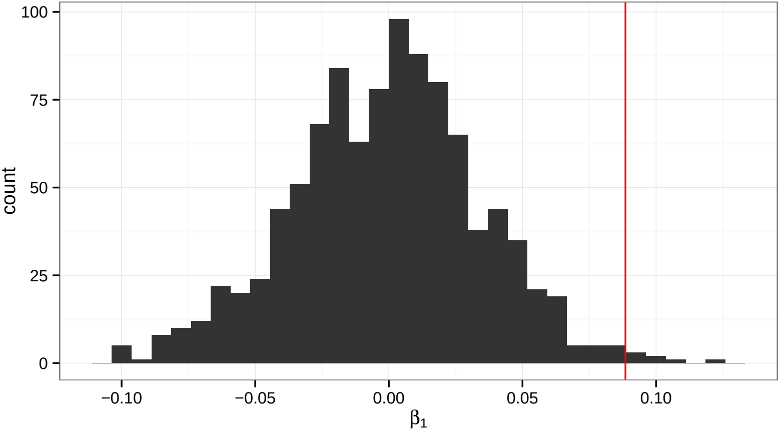
```
mod2 <- lm(resampDheight ~ Mheight, data=heights)
summary(mod2)$coefficients

##                Estimate Std. Error     t value      Pr(>|t|)
## (Intercept) 63.575982642 1.86206903 34.14265608 1.755141e-185
## Mheight      0.002803267 0.02979446  0.09408687  9.250539e-01
```

Permutation tests require repeated samples!

A permutation test algorithm

- Run original analysis (i.e. fit our linear model), store the parameter of interest (in our case, $\hat{\beta}_1$).
- For i in $1, 2, \ldots, N$:
  - Permute the Ys.
  - 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.

Permutation test results (1000 simulations)



Now that we have our distribution of β_1 under the null hypothesis, we can compare the estimate of β_1 from the real data analysis. Notice that we calculate a two-sided p-value by calculating how many $|\hat{\beta}_1^{(i)}| > \beta_1$. The resulting p-value is 0.013.

Homework: do this on your own!