

# Lecture 1: Course Overview

Ciaran Evans

# “Statistical computing” vs. “computational statistics”

- ▶ **Statistical computing:** programming languages and computing tools for working with statistics (think: storing and accessing data, data transformations and wrangling, some data visualization, etc.)
- ▶ **Computational statistics:** the use of computational algorithms to implement statistical methods (think: simulating data from a distribution, fitting a regression model, performing hypothesis testing)

## Key points

- ▶ Focus on *how* to statistical methods work. E.g., how do we fit a logistic regression model? What is R actually doing “under the hood”?
- ▶ Focus on *implementation* – much of your work will be turning statistical methods and algorithms into code
- ▶ Focus on efficiency
  - ▶ Efficient algorithms (e.g. dynamic programming approaches)
  - ▶ Efficient approximations (e.g. integral and Hessian approximations)
  - ▶ Efficient languages (particularly the use of C++)
- ▶ Focus on iteration

# Tentative course outline

- ▶ Simulation (simulation studies, generating random numbers, simulating from a distribution)
- ▶ Model fitting (linear and generalized linear models, maximum likelihood, Newton and quasi-Newton methods)
- ▶ Missing data and EM algorithm (Gaussian mixtures, Hidden Markov Models)
- ▶ Integration (numerical integration, Monte Carlo integration, MCMC)
- ▶ Bootstrapping

# Roadmap for the first few weeks

- ▶ Motivation: simulation studies
  - ▶ Today: using simulations to answer questions about hypothesis tests
- ▶ Simulation: behind the scenes
  - ▶ Generating random numbers
  - ▶ Transformation methods
  - ▶ Acceptance/rejection sampling
  - ▶ Computing topics: iteration, functions, C++
  - ▶ Simulation for linear models and multivariate normal distributions
- ▶ Following unit: how do you actually fit a regression model? (linear, GLMs, GEEs, etc.)

## Recap

Suppose we have a sample  $X_1, \dots, X_n \stackrel{iid}{\sim} N(\mu, 1)$ . Want to test the hypotheses

$$H_0 : \mu = 0 \quad H_A : \mu \neq 0$$

How do I test these hypotheses? (What is my test statistic, and how do I make a decision?)

## Recap

Suppose we have a sample  $X_1, \dots, X_n \stackrel{iid}{\sim} N(\mu, 1)$ . Want to test the hypotheses

$$H_0 : \mu = 0 \quad H_A : \mu \neq 0$$

If actually  $\mu = 0$ , do we want to reject  $H_0$ , or fail to reject?

## Recap

- ▶ Either  $H_0$  is true or false (usually we don't know)
- ▶ We either reject or fail to reject  $H_0$

Possible outcomes of a hypothesis test:



## Recap

|                | $H_0$ is true    | $H_0$ is false   |
|----------------|------------------|------------------|
| fail to reject | correct decision | type II error    |
| reject         | type I error     | correct decision |

**Usual goal:** Minimize type II error, subject to control of type I error rate (i.e. want  $P(\text{reject } H_0 | H_0 \text{ true}) \leq \alpha$ )

## In R

```
mu_x <- 0
n <- 20
x <- rnorm(n, mean=mu_x, sd=1)
x
```

```
## [1] 0.6797336 -0.8548404 -1.3210164 -0.4273575 -1.0207
## [7] -0.4315816 -0.3921952 1.0212251 -0.1325580 -1.0810
## [13] 0.7787329 0.5623875 0.1579840 2.0061821 0.1557
## [19] -0.3435500 -0.2215555
```

```
t.test(x, alternative="two.sided", mu=0)
```

```
##
## One Sample t-test
##
## data: x
## t = 0.33834, df = 19, p-value = 0.7388
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
```

## In R

```
t.test(x, alternative="two.sided", mu=0)

##
## One Sample t-test
##
## data:  x
## t = 0.33834, df = 19, p-value = 0.7388
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.3297764  0.4569528
## sample estimates:
##  mean of x
## 0.06358819
```

Here we fail to reject  $H_0$ . Does this give us our type I error rate?

## In R

```
t.test(x, alternative="two.sided", mu=0)

##
##  One Sample t-test
##
## data:  x
## t = 0.33834, df = 19, p-value = 0.7388
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.3297764  0.4569528
## sample estimates:
##  mean of x
## 0.06358819
```

Here we fail to reject  $H_0$ . Does this give us our type I error rate?

No! Type I error rate =  $P(\text{reject } H_0 | H_0 \text{ true})$ . Need more than one observation to estimate a probability.

## Repeating many times

```
set.seed(379)
n <- 20
mu_x <- 0
nsim <- 1000 # number of times to repeat
test_results <- rep(NA, nsim) # vector to store the test results

for(i in 1:nsim){
  x <- rnorm(n, mean=mu_x, sd=1)
  test_results[i] <- t.test(x, alternative="two.sided", mu=0)
}

head(test_results)
```

```
## [1] 0.8213916 0.2525886 0.3818531 0.3760550 0.7342035 0.1111111
```

What is this code doing?

## Repeating many times

```
set.seed(379)
n <- 20
mu_x <- 0
nsim <- 1000 # number of times to repeat
test_results <- rep(NA, nsim) # vector to store the test results

for(i in 1:nsim){
  x <- rnorm(n, mean=mu_x, sd=1)
  test_results[i] <- t.test(x, alternative="two.sided", mu=0)
}

head(test_results)
```

```
## [1] 0.8213916 0.2525886 0.3818531 0.3760550 0.7342035 0.1111111
```

How do I find the fraction of times we rejected  $H_0$ ?

## Repeating many times

```
set.seed(379)
n <- 20
mu_x <- 0
nsim <- 1000 # number of times to repeat
test_results <- rep(NA, nsim) # vector to store the test results

for(i in 1:nsim){
  x <- rnorm(n, mean=mu_x, sd=1)
  test_results[i] <- t.test(x, alternative="two.sided", mu=0)
}

mean(test_results < 0.05)
```

```
## [1] 0.045
```

# Your turn: Practice questions

Practice questions on the course website:

[https://sta379-s25.github.io/practice\\_questions/pq\\_1.html](https://sta379-s25.github.io/practice_questions/pq_1.html)

- ▶ Experiment with changing  $\mu$  and  $n$ . How does the probability of rejecting change?
- ▶ Start in class. You are welcome to work with others
- ▶ Practice questions are to help you practice. They are not submitted and not graded
- ▶ Solutions are posted on the course website