```
course = "Simulated data & Monte-Carlo simulation studies"


lesson_iteration = 1
lesson_title = "orientation + foundational concepts"


auth = "Ian Hussey"
dept = "Psychology of Digitalisation"
```

Background

of the course

Why I’m here

- ‘Violating assumptions of statistical tests is bad’
  - How bad?
  - Under what circumstances?

- Bad at math
  - I don’t have the matrix algebra to solve problems
  - but I can brute force it with code

- ‘How do I learn this new method while also not conditioning my results on my data?

Background

of the course

You

- Why this workshop?

- What are your wants and needs from it?

What is a Monte Carlo simulation?

- There is no consensus on how Monte Carlo should be defined!
  - Monte Carlo methods for quantitative (social) science methods research
    - This course
  - Monte Carlo methods as part of data analysis (e.g., MCMC in Bayesian data analysis)
  - Monte Carlo methods for the solution of general numerical problems (e.g., Monte Carlo integration)
    - Not this course

Core components of a simulation

1. Generate pseudo-random data set with known properties

2. Analyse data with a statistical method

3. Repeat 1 & 2 many times (‘iterations’)

4. Summarize results across iterations

5. Make it an experiment
   - Systematically vary parameters in Step 1 (between factor)
   - Compare different ways to do Step 2 (within factor)

Foundational Concepts

in Monte Carlo Simulation Studies

Simulations to increase understanding

What is the distribution of *p* values under the null hypothesis?

Simulations to increase understanding

What is the distribution of *p* values under the null hypothesis?

```
res ← replicate(10000, t.test(rnorm(n = 50, m = 0, sd = 1), rnorm(n = 50, m = 0, sd = 1))$p.value)

res ▷ hist()
```

Foundational Concepts

in Monte Carlo Simulation Studies

Simulations to increase understanding

What is the distribution of *p* values under the null hypothesis?

do it many times

analyze

```
res ← replicate(10000, t.test(rnorm(n = 50, m = 0, sd = 1), rnorm(n = 50, m = 0, sd = 1))$p.value)

res ▷ hist()
```

generate

generate

summarise across iterations

Foundational Concepts

in Monte Carlo Simulation Studies

Simulations to increase understanding

What is the distribution of *p* values under the null hypothesis?

do it many times

analyze

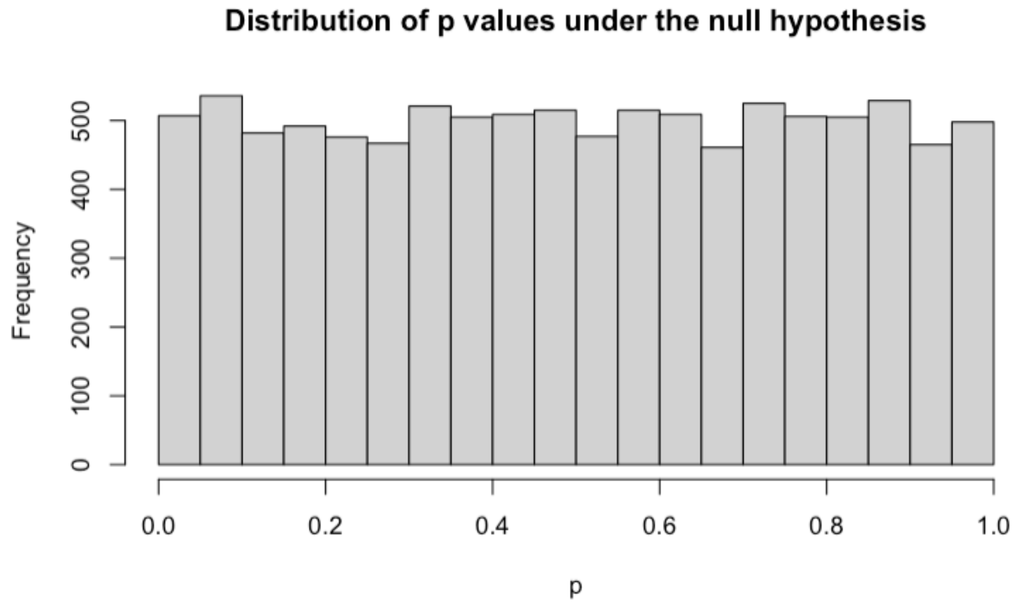
```
res ← replicate(10000, t.test(rnorm(n = 50, m = 0, sd = 1), rnorm(n = 50, m = 0, sd = 1))$p.value)

res ▷ hist()
```

generate

generate

summarise across iterations



tidy workflow

maximising transparency and reusability

But how to increase our understanding of simulations themselves?

```
res ← replicate(10000, t.test(rnorm(n = 50, m = 0, sd = 1), rnorm(n = 50, m = 0, sd = 1))$p.value)

res ▷ hist()
```

This code does not read in a linear fashion, eg top to bottom or left to right.

It makes its internal workings invisible - you cannot view the data or any intermediate steps

aut = "Ian Hussey";

dept = "Psychology of Digitalisation || Digitalisation of Psychology"

aut = "Ian Hussey";

dept = "Psychology of Digitalisation || Digitalisation of Psychology"

aut = "Ian Hussey";

dept = "Psychology of Digitalisation || Digitalisation of Psychology"

tidy data, tidy workflow

tidy workflow

maximising transparency and reusability

tidy inputs, tidy outputs

maximise code & workflow reusability

maximise transparency of intermediate steps

```
        Two Sample t-test

data:  rnorm(100) and rnorm(100)
t = 0.90203, df = 198, p-value = 0.3681
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.1392104  0.3739291
sample estimates:
   mean of x    mean of y
-0.002680745 -0.120040082
```

A tibble: 1 × 10

estimate <dbl>	estimate1 <dbl>	estimate2 <dbl>	statistic <dbl>	p.value <dbl>	parameter <dbl>	conf.low <dbl>	conf.high <dbl>	method <chr>	alternative <chr>
0.01230826	0.08646291	0.07415465	0.08756086	0.9303142	198	-0.2648943	0.2895108	Two Sample t-test	two.sided

aut = "Ian Hussey";

dept = "Psychology of Digitalisation || Digitalisation of Psychology"

Core components of a tidy simulation

1. Generate tidy data

2. Analyse tidy data and return tidy results

3. Repeat 1 & 2 many times (‘iterations’) in a tidy way

4. Summarize results across iterations

5. Make it an experiment
   - Systematically vary parameters in Step 1 (between factor)
   - Compare different ways to do Step 2 (within factor)

Foundational Concepts

in Monte Carlo Simulation Studies

Example of Different Ways to Combine Factors in the Design of a Simulation Study.