Lecture 1
Introduction, motivation, puzzles & randomness

Dr. Franco Ubaudi

The Nature of Data
Western Sydney University

Spring 2021

Big picture

We have two combined classes:

- Undergraduates $->$ 301108 "Thinking About Data"
- Postgraduates $->$ 301114 "The Nature of Data"

Delivery

- Lecture: Tuesday 12 - 2pm $->$ 301108 and 301114
- Tutorial / Practical: Tuesday 2 - 4pm $->$ 301114
- Tutorial / Practical: Friday 3 - 5pm $->$ 301108

Big picture cont.

Main components of study for this unit:

- *R* programming language
  - *R* language
  - Jupyter Notebook
  - RStudio (IDE) *optional*
- Statistics / probability

Big picture cont.

Main components used in this unit:

- Moodle $->$ *ds-stats server*
  where most materials for this unit can be found
- Jupyter Notebook server
  place to do tutorials, quizzes, assignments & exam

Why do this?

- Data is growing at an alarming rate
- Useful things can be hidden in data
- Determining if there really is a difference
- Begin valuable to your employer

Can you do this? Can you succeed??

Yes of cause, but it takes the will and effort to succeed!

Five valuable resources

- The "Learning Guide"
- "A (very) short introduction to R"

  https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf
- **Practice, practice ...**
- Create a *single R* script file
  - why re-invent the wheel?
  - plus, exam is open book
- Seek help when you need it and don't wait until it is too late!

Lecture 1

Today we are going to start looking at a few different things:

- Randomness, what is it and how can it be measured
- Hands-on intro to *R*
- Hands-on intro to RStudio
- Hands-on intro to Jupyter Notebook

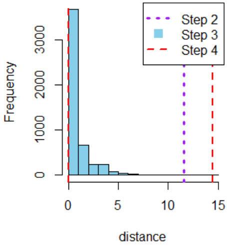
Notions of randomness

- Probability involves randomness
- Consider a coin
- Two things determine the outcome
  - Some process
  - Randomness or noise
  - Hence what we see always varies

R code simulation to determine if a coin is fair

Three key steps

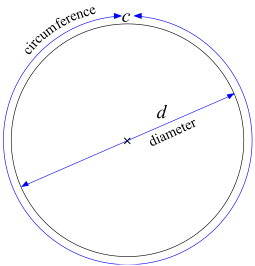
1. Determine what you *expect*; e.g. from a fair coin
2. Calculate / measure the distance between what was observed and what is expected
3. Determine distribution of expected since randomness causes variations
4. Compare results of steps 2 & 3



R code simulation to determine if a coin is fair, cont.

```
####################################
# foundations of hypothesis testing #
####################################

coin <- c('H', 'T')   # coin definition
flips <- 100      # Number of coin flips
trials <- 5000      # Number of trials or experiments to perform

# Secret code to create our unknown coin results
if(TRUE)
{ ; }

coinResults         # Unknown coin results
exp <- c(0.5, 0.5) * flips  # What is expected from a fair coin

coinResults; exp      # View variable contents

# Calculate difference between unknown and expected coins
cs <- sum((coinResults - exp)^2 / exp)
cs

# Simulate a fair coin
d <- replicate(trials,
               {
                 obs <- sample(coin, flips, replace = TRUE)
                 obs <- table(obs)

                 sum((obs - exp)^2 / exp)
               })

range(d)  # Limits of cs values resulting from trials

# Distribution of cs results for a fair coin
hist(d, col = 'skyblue', xlim = c(0, 15))

# Location of cs; unknown coin results
abline(v = range(d), col = 2, lwd = 2, lty = 2)
abline(v = cs, col = 'purple', lwd = 3, lty = 3)
```

Are the digits of π random?



π to 500 decimal places:

3.141592653589793238462643383279502884197169399375
10582097494459230781640628620899862803482534211706
79821480865132823066470938446095505822317253594081
28481117450284102701938521105559644622948954930381
96442881097566593344612847564823378678316527120190
91456485669234603486104543266482133936072602491412
73724587006606315588174881520920962829254091715364
36789259036001133053054882046652138414695194151160
94330572703657595919530921861173819326117931051185
48074462379962749567351885752724891227938183011949 16

- π cannot be expressed as a fraction
  $\therefore$ it is irrational
- Its decimal expansion goes on forever
- Given a sequence of digits, can we predict the next digit with any certainty?
  Or does the sequence seem random?

How do we measure randomness for a sequence of digits?

Measuring randomness

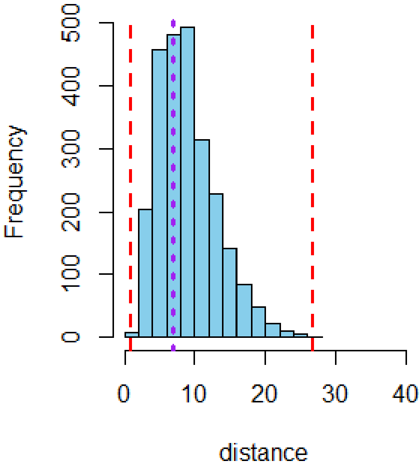
- What do we expect *if* the digits of π are random?
- Expect a uniform distribution of digit counts
- Hence every digit occurs the same number of times

Determine if digits of π are random

Use same three key steps for determining if a coin is fair

```
# randomness of digits of pi
trials <-  2500

df <- read.csv('pi500.csv')
head(df)

digitCount <- nrow(df) / length(obs)

digitsOfPi <- table(df$pi.digits)
exp <- rep(1, 10) * digitCount

digitsOfPi; exp

# Calculate difference between observed and expected
cs <- sum((digitsOfPi - exp)^2 / exp)
cs

# Simulate a random digits
d <- replicate(trials,
               {
                 obs <- sample(0:9, nrow(df), replace = TRUE)
                 obs <- table(obs)

                 sum((obs - exp)^2 / exp)
               })

range(d)  # Limits of cs values resulting from trials

# Distribution of cs results for random digits
hist(d, col = 'skyblue', xlim = c(0, 40),
     main = '', xlab = 'distance')

# Location of cs & limits for random digits
abline(v = range(d), col = 2, lwd = 2, lty = 2)
abline(v = cs, col = 'purple', lwd = 3, lty = 3)
```

Determine if digits of π are random cont.



The Birthday Problem

The "Birthday Problem" is an interesting problem that fits well within the objectives of this unit.

Imagine this scenario:

- A gathering of people
- How many people in the gathering are needed so
  - there is a probability of at least 50%
  - in finding at least two people with the same birthday?
- Wording is important!
- We want *at least one group of two people*
- But it could be one group of two and one group of three, ...
- How big a gathering do you think we need?

The Birthday Problem cont.

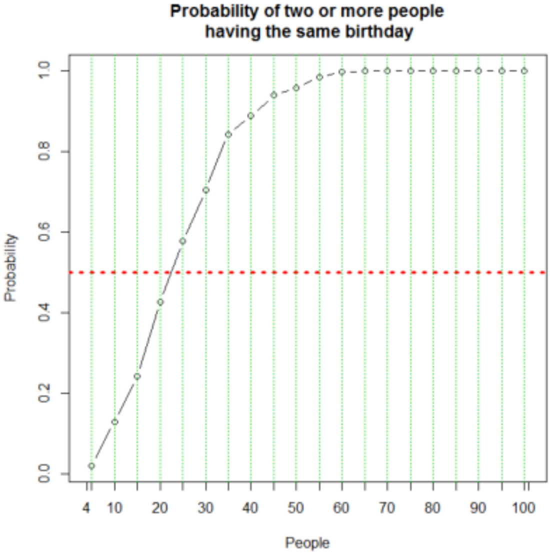
How big a gathering do you think we need?

$P(A) + P(B) = 1$

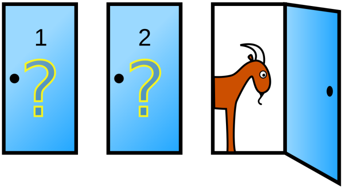
$A \equiv$ someone shares the same birthday
$B \equiv$ no one has the same birthday

$P(A) = 1 - P(B)$

The Birthday Problem cont.

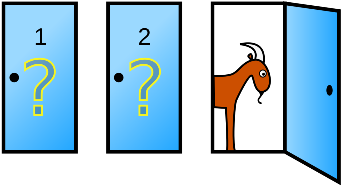


Monty Hall Problem



- Three doors
- Only one door has a car behind it
- Remaining two doors have a goat behind each

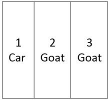
Monty Hall Problem cont.



- In the simplest version of the game, you have 1 chance in 3 of winning a car
- But Monty makes it more interesting:
  - after you make a choice
  - he opens a door
  - then he asks if you want to change your mind

Monty Hall Problem cont.

Imagine the following



Scenario I

- you happen to choose door 1
- say Monty opens door 3 and asks if you want to change your mind
- you swap to door 2 and lose
- BUT you will lose 1/3

Scenario II

- you happen to choose door 2
- say Monty opens door 3 and asks if you want to change your mind
- you swap to door 1 and win
- BUT you will win 2/3