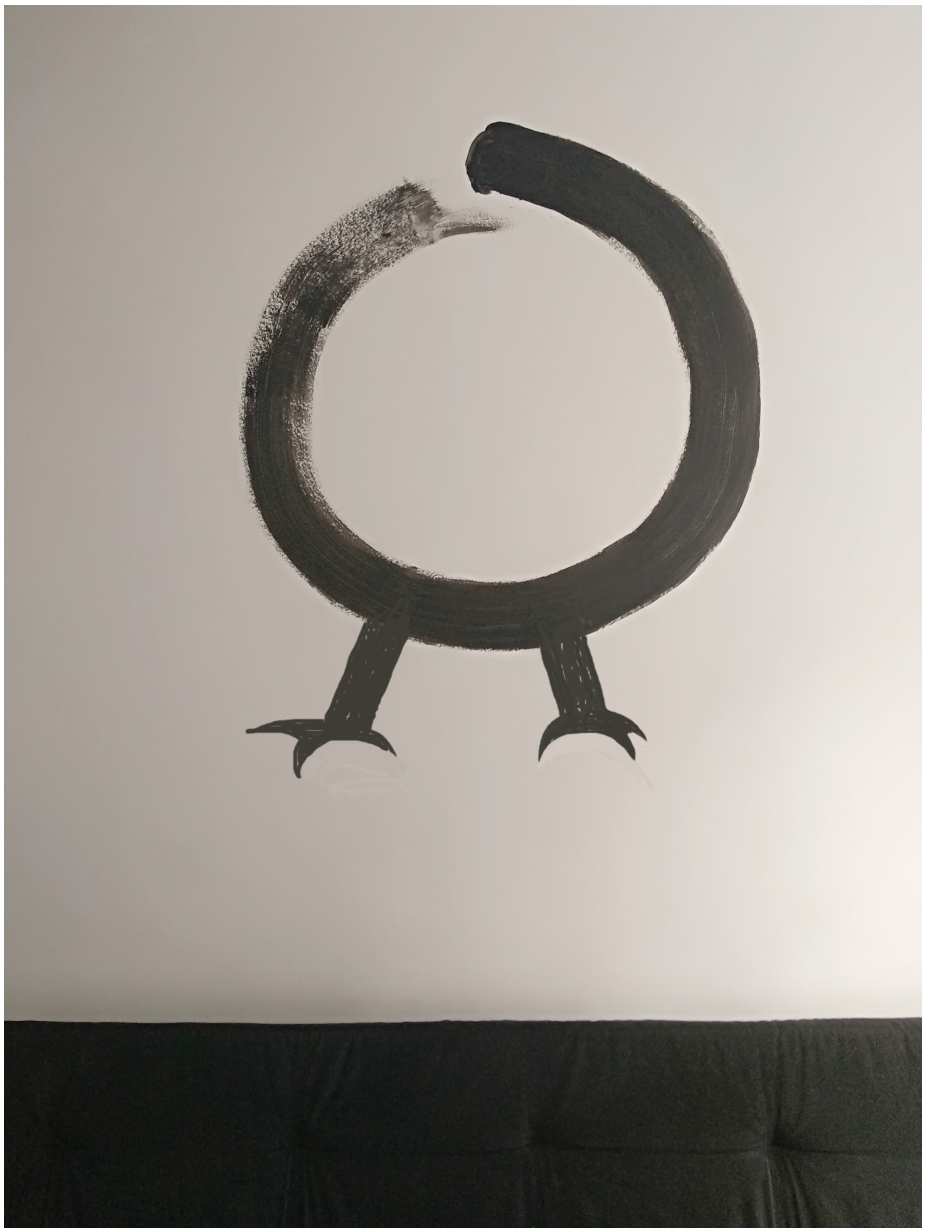


Inference for Science and Machine Learning

Philosophy and applications with statistics and probability.

Felipe Coelho Argolo felipe.c.argolo@protonmail.com

Londres, 8 de Julho de 2020
official website: https://www.leanpub.com/fargolo

Volume 1 Second edition

Preface

Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful
George Box & Norman R. Draper, Empirical Model-Building and Response Surfaces

---

When we entered the 21st century, the terms *artificial intelligence*, *quantitative analysis*, *machine learning*, *big data* and *data science* gained strong notoriety in due to unprecedented results in practical application problems. Technical advances in natural language processing, computer vision and other fields were quickly implemented by engineers and researchers in finance, industry and science. These solutions use statistical-probabilistic models to model empirical measures. A systematic study of the formalism and tools involved is voluminous (see List of *Recommended readings*).

Quod est inferius est sicut quod est superius. Et quod est superius est sicut quod est inferius, ad perpetranda miracula rei unius.
"What is inferior is like what is superior. And what is superior is like what is inferior, perpetuating the miracles of one thing. [1]

Preface to the second edition

Approximately one year has passed since the launch of the 1st edition. Some important changes have been incorporated.

Julia has been included as an alternative language to **R**. It is a language with a smaller community, but very promising. In addition to offering faster execution speed, it offers a more concise syntax for the examples.

A compilation with different applications of the concepts exemplified in the chapters is now available for both languages in a cookbook.

I got in touch with the work of Richard McElreath (Statistical Rethinking), which resulted in positive results: Chapter 1 includes a second perspective (maximum entropy) for the use of normal distribution in natural sciences. Chapter 4 has been restructured to include a more general approach to the study of causality with targeted graphs, using the **dagitty** package / software. Chapter 6 has excerpts related to the choice of priors and performance evaluation.

In the first edition, Ron Eglash's work in ethnomathematics influenced the use of green and yellow colors, associated with Orumla and the Yoruba divination, which uses binary numbers. One of the problems in the first version was to find

[1] http://catb.org/esr/writings/unix-koans/shell-tools.html

titles that synthesized each chapter. Finding the Adinkras and the concepts they represent was a fortuitous event.

Adinkras are symbols of Akan, incorporating abstractions linked to their names, forms and also to elements of culture, as popular sayings.

Introduction

Adinkras

The Adinkras, like the bird that illustrates the cover (Sankofa), are symbols in Akan culture. They represent specific concepts and popular knowledge, connected to their form. Theoretical physicists have also adopted the name for graphs representing the formal rules that govern particles in supersymmetric gravity models.

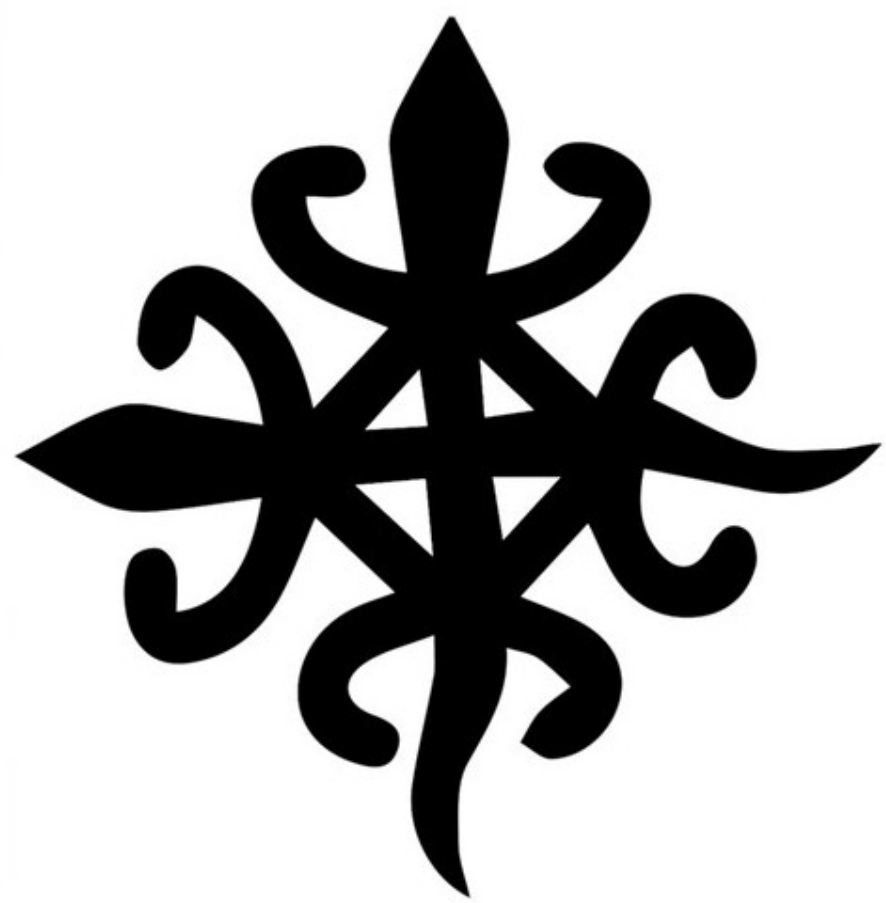
1 . **ADINKRA HENE** (Adinkra Leader / King)
Bases in descriptive statistics and the normal distribution

Formed by concentric circles, it is related to the inspiration and creation of the other Adrinkras. Basic intuitions are introduced in descriptive statistics and probability, linked to elementary concepts of physics. Starting from the study of Archimedes on levers, ways to describe samples and random variables using basic intuitions. It also addresses the relationship between empirical sciences and normal distribution.

2 . **DWENNIMMEN** (Ram Horns)
Hypothesis testing

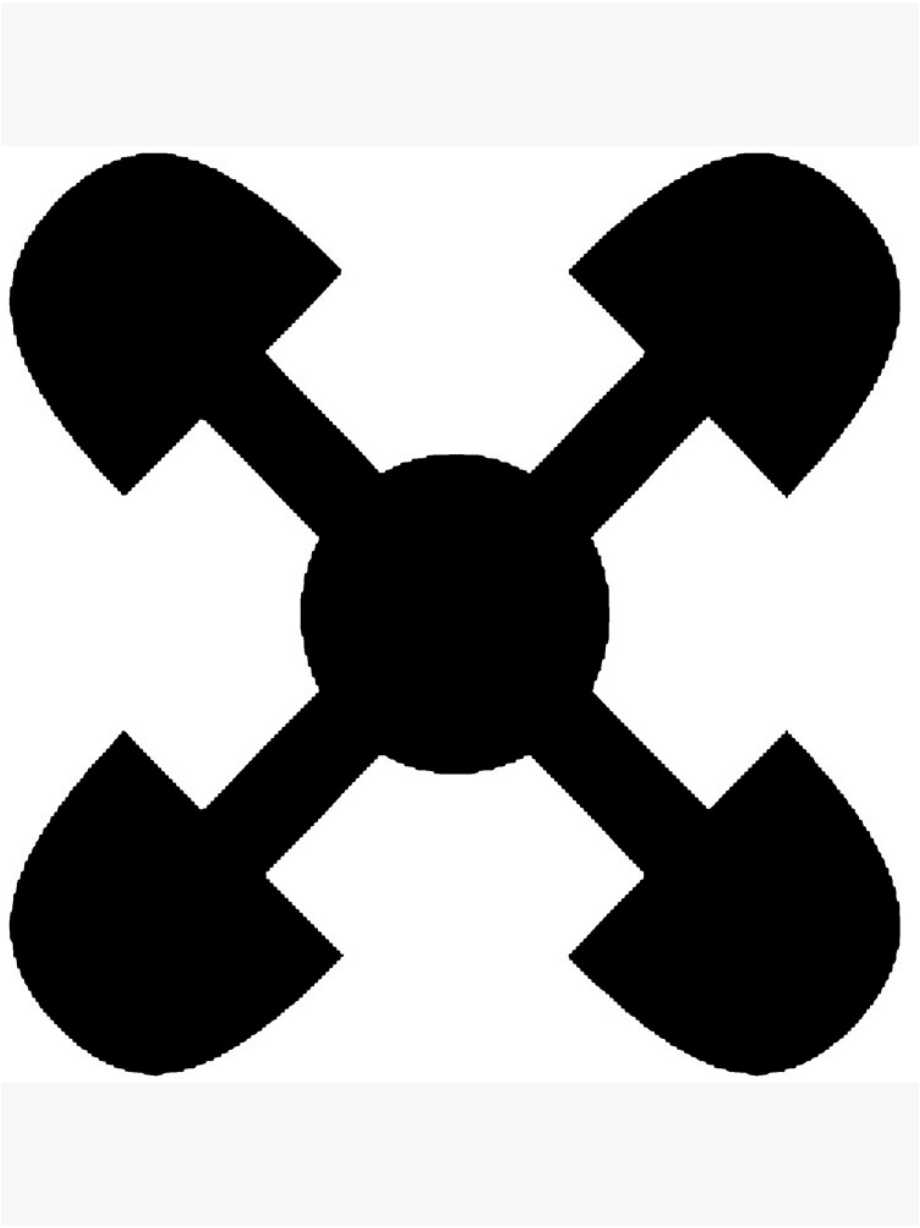
It represents the vision of two sheep fighting. The Sheep Horns symbolize strength and humility, as sheep fight fiercely against other pairs and predators, but accept death. The identity of science is strongly linked to the judicious use of experiments to test hypotheses. They make room for failure. The *second chapter* accompanies Charles Darwin in the Galapagos. Darwin waited 20 years between the conception of the theory and its publication. He worked tirelessly to investigate whether his impressions were not false. This chapter illustrates how the hypothetical-deductive rationale works to study scientific hypotheses. The Student's t test is applied to compare bird beaks in Gal?pagos.

3 . **FUNTUNFUNEFU-DENKYEMFUNEFU** (Siamese crocodiles)
Correlations and comparisons.



The crocodiles that share a stomach. They symbolize unity and cooperation. The *third chapter* highlights relationships between measures. We will learn linear correlations (Pearson's ρ) and effect size (*Cohen's D*). Nonparametric alternatives are also introduced: Spearman's ρ and Mann-Whitney U test). We use regression to make predictions using *closed forms*. Analyzing the model equations analytically, we found a unique estimate for the parameters involved.

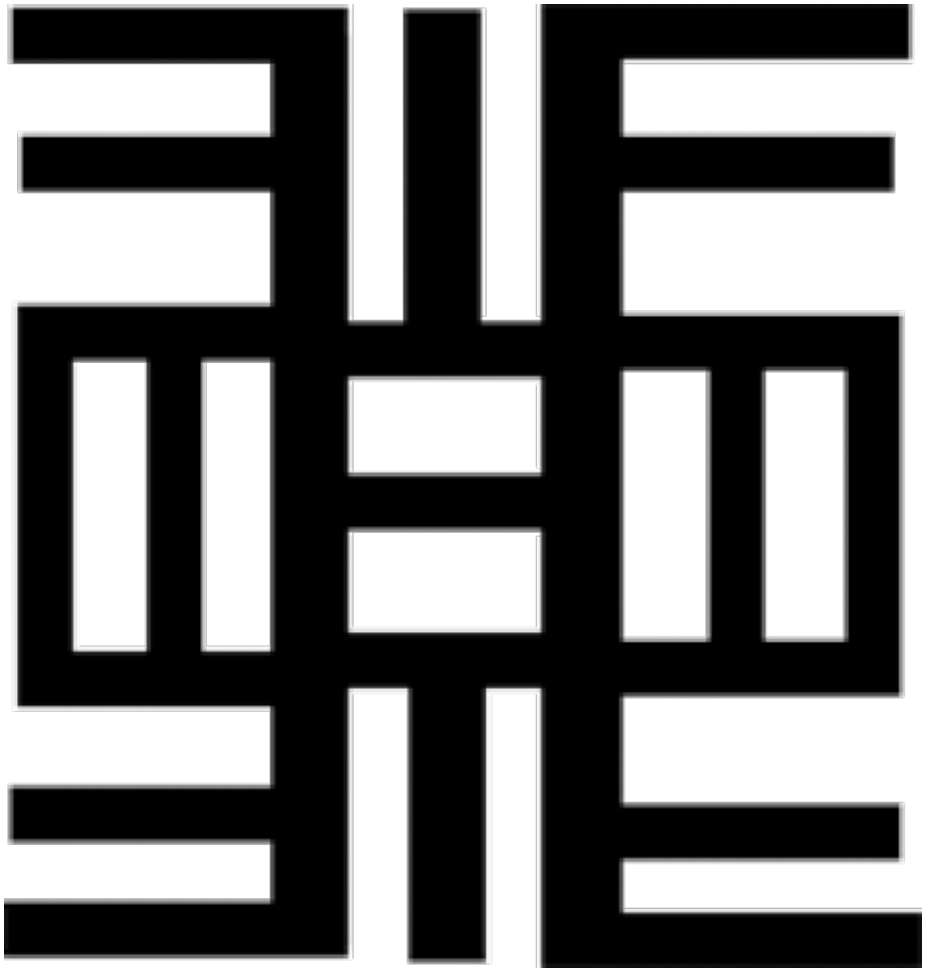
4 . **AKOMA NTOSO**(Linked Hearts)
Multivariate analysis, causal models, confounders, reduction of dimensions and structural equations.



Connected hearts symbolize mutual understanding and agreement. With many variables (multivariate analysis), graphs are the basic abstraction for relating concepts. We studied multiple regression and how to deal with covariates according to a causal diagram. The *fourth chapter* introduces a formal implementation of the comprehensive philosophical paradigm for **causality**. Collinearity, confounders, mediation and moderation. We also talked about reducing dimensions

and latent measures, with factor analysis, principal component analysis (PCA) and structural equations (SEM).

5 . **NEA ONNIM NO SUA A, OHU**
Neural networks



"He who does not know can know through learning". The *fifth chapter* introduces neural networks, which start from several simple, empty units, processing inputs to learn patterns. We started with the biological inspiration of artificial neurons and the first intelligent machine in history: the *Mark I Perceptron*. We coded a virtual Mark I, which uses a new way of estimating parameters: *gradient descent*. Instead of using a closed formula, we use derivatives to 'walk' towards the minimum progressively.

Neural networks expand the power of a neuron with multiple nodes to build complex predictive systems. Deep networks include successive layers, allowing

sequential transformations to solve more general classes of problems. We understand how neurons can propagate errors to others, optimizing gradients in conjunction with the *backpropagation* mechanism. We will also encode a neural network from scratch, Mark II.

6 . **SANKOFA** (San - Back; Ko - Go; Fa - Search, get)
Modelos Bayesianos

Figure 1: Sankofa in a gold weight. Displayed at the New York Museum. Made between the 18th and 19th centuries

This Adinkra is about returning to the past and learning from it. The proverb says "There is nothing wrong with learning from the past". Bayesian models incorporate prior information (*prior*) in their formulation. The *sixth chapter* discusses the supposed clash between the **frequentist** and **bayesian** probability schools. The context is given by alternatives to the hypothetical deductive

method: Carnap demonstrates the difficulty of refutations, Feyerabend proposes an epistemological anarchy supported by historical facts and W. van Quine paints an intertwined system for theories, hypotheses and observations. We reiterate some previous examples using Stan for Bayesian inference. We explore a third way of estimating parameters. Without closed formulas, we use the power of stochastic simulations (*Markov Chain Monte Carlo*).

Summary

Chapter 1 - ADINKRAHENE - Center and dispersion

- Center and dispersion
- Mean and variance
- Normal distribution
- Experimental science and the central limit theorem
- Moments

Chapter 2 - DWENNIMMEN (Strength and humility) - Deductive hypothetical method and Darwin's finches

- Birds in the Galapagos
- Hypothetical-deductive method and hypothesis tests
  - P-value
  - Student t distribution and t test

Chapter 3 - FUNTUNFUNEFU-DENKYEMFUNEFU (United crocodiles) - About associations

- Prelude: Who needs the p-value?
- Effect size: Cohen's D
- Linear correlations
  - Pearson's ρ correlation coefficient
  - Predictions with linear regression
- Correlations and non-parametric tests
  - Spearman's ρ
  - Mann Whitney U test

Chapter 4 - AKOMA NTOSO (Connected hearts) - Multivariate analysis, graphs and causal inference

- Multiple regression
  - Collinearity
- Graphs and causal trajectories
  - Mediation and moderation
  - Factor analysis
  - Structural equations

Chapter 5 - NEA ONNIM NO SUA A, OHU (He who does not know can know by learning) - Neurons

- Logistic regression
- An artificial neuron: The perceptron
  - History and implementation from scratch: Mark I
- Neural Networks and Deep learning (multiple layers)
- Gradient Descent
- Backpropagation

Chapter 6 - SANKOFA (Return and search) - Context and Bayesian inference

- Odds
  - Frequentists and Bayesians
- Many scientific methods: Feyerabend, Carnap and Quine
- Bayesian inference
  - Bayes' theorem
  - Intuitions: prior, likelihood, posterior and marginal probabilities
  - Comparison of samples with normal distribution
  - Linear correlation
- Markov Chain Monte Carlo Estimators and Methods
  - Closed solutions, Gradient Descent and MCMC

Prerequisites

All examples can be reproduced using free software.

Recommended reading:

Philosophy and scientific dissemination

- Surely You're Joking, Mr. Feynman
- The Demon-Haunted World - Carl Sagan
- The logic of scientific research - K. Popper
- The structure of scientific revolutions - Thomas Kuhn
- Against Method - Paul Feyerabend
- Two dogmas of empiricism - Willard van Quine
- Stanford Encyclopedia of Philosophy - https://plato.stanford.edu/
- The Open Handbook of Formal Epistemology - https://jonathanweisberg.org/post/open-handbook/

Neurosciences

- Principles of neural science - Eric Kandel

Mathematics / Computing

- Collection '*Fundamentals of elementary mathematics*' (Gelson Iezzi)
- Statistical Rethinking. A Bayesian Course with Examples in R and Stan, Richard McElreath.
- Biostatistics without secrets. Annibal Muniz.
- What is mathematics - Courant & Robbins
- Better Explained (https://betterexplained.com/)
- http://material.curso-r.com/
- R Graphics Cookbook
- R Inferno
- Learn you a Haskell for Great Good
- Layered Grammar of Graphics - Hadley Wickham.
- Algorithms unlocked
- Online: Statsexchange, stackoverflow, mathexchange, cross-validated.

Machine Learning

- An Introduction to Statistical Learning: with Applications in R
- Neural Networks and Learning Machines - Simon Haykin
- Stanford (computer vision): http://cs231n.stanford.edu/
- Oxford 2015 (Deep learning): (https://www.youtube.com/watch?v=dV80NAlEins&list=PLE6Wd9FR--EfW8dtjAuPoTuPcqmOV53Fu)

Thanks

My family, Suzana, Paulo, Isaac and Chris. Friends Gabriel, Guilherme, Pedro, Wei.

To the teachers: Carla Daltro, Anibal Neto, Lucas Quarantini, Luis Correia, Rodrigo Bressan, Ary Gadelha.

To colleagues Fatori, Luccas, Macedo, Walter, Rafael, Sato, Hiroshi, Lais, Luci, Davi, n3k00n3 (Fernando), Loli (Lorena).

For comments, criticisms, suggestions, or just say *hi*: felipe.c.argolo@protonmail.com.

Chapter 1: Center and dispersion

Introduction

In the opening chapter, we will get in touch with the basic elements of our journey. ADINKRAHENE is made of three concentric circles and this chapter has three parts.

First, we start from the physical concept of *moments* and learn about the center and dispersion of distributions and samples. We introduce **R** to calculate mean and variance.

Then, we will study the natural relationship between the normal distribution and the t distribution, which are related. The adoption of normal distribution in scientific works is quite popular. To understand the reasons, the Central Limit Theorem and the concept of entropy are fundamental. We use the language **Julia**.

The third part presents data visualization with a grammar graphics and has examples in both languages.

Part 1 - Center, dispersion, mean and variance

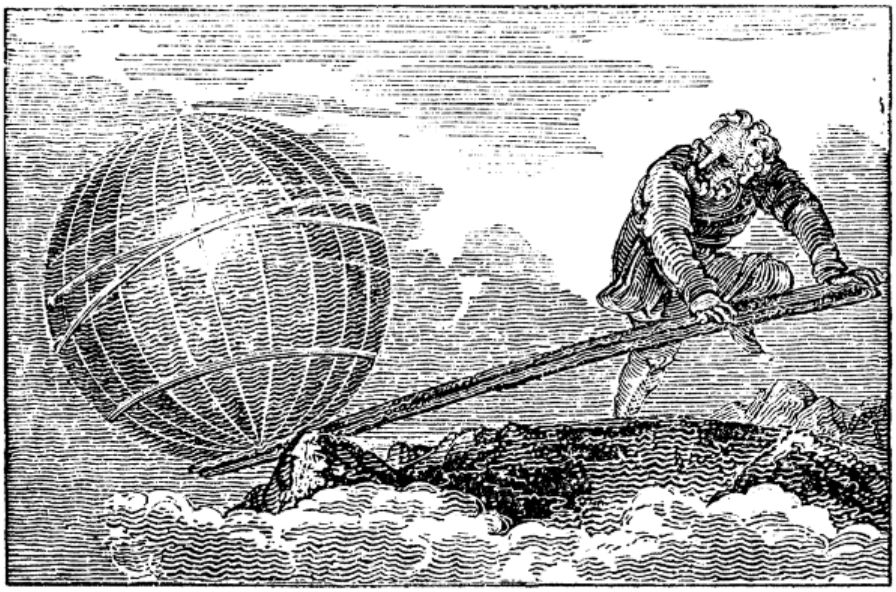


Figure 2: Give me a foothold and I will move the Earth

A brief dive into physics: Moments

[2]

The physical concept of moment was originally conceived by Archimedes. Although he did not invent the lever, he described the mathematical principles behind it. In *On the Equilibrium of Planes*, Archimedes declares that *Magnitudes are in balance when at a distance reciprocally proportional to their weights.*

This is the well-known law of the lever. Given a foothold and a plan on it, we apply force anywhere on the plan. The resulting moment (torque) is the result of multiplying the physical quantity (F) by the distance to a fixed point (d). $M = F * d$

Assuming a constant force, the further away from the a fixed point, the greater the resulting moment. Thereafter, physicists extended the concept to other domains. For example, an object containing parts with opposite charges $-q$ and $+q$ separated by a distance d has an analogous moment (electric dipole moment): $M = q * d$ In general, *we speak of moments when multiplying a physical quantity by a distance* .

[2] Pappus of Alexandria, Synagoge, Book VIII

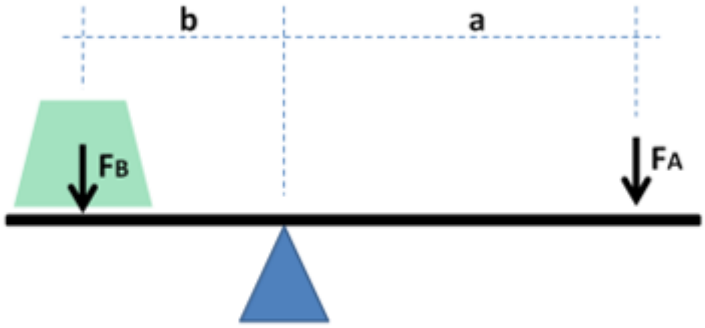


Figure 3: .

Resulting moment In the case of the lever, we saw that each force applied to the object is associated with a moment (torque). We know that gravity acts on each piece with mass making up the whole. We can then calculate the resulting moment by adding the moments of all N points. Let F_i be the function describing the force in each i-th:

$$M = \sum_{i=1}^{N} F_i d_i$$

A system, like the bird resting on the finger, is in equilibrium when the sum of the moments in relation to the fixed point is zero. For electrical charges, the system is nonpolar when the moment is zero. In the figure below, we see how the CO_2 molecule is nonpolar, while the water molecule is polar:

The moments described above are expressions of the *first moment*, since the quantity is multiplied by the distance with exponent 1: $d = d^1$.

We can calculate other moments by exponentiating the spatial component (distance). We will now study moments of mass of a one-dimensional object:

The **moment zero** of mass for an object is $M_0 = \sum_{i=1}^{N} m_i d_i^0$. Since $d^0 = 1$, we have $M_0 = \sum_{i=1}^{N} m_i$, which is simply the sum of the masses of all points. The zero moment is the **total mass**.

Figure 4: As the toy above is balanced on only one point?

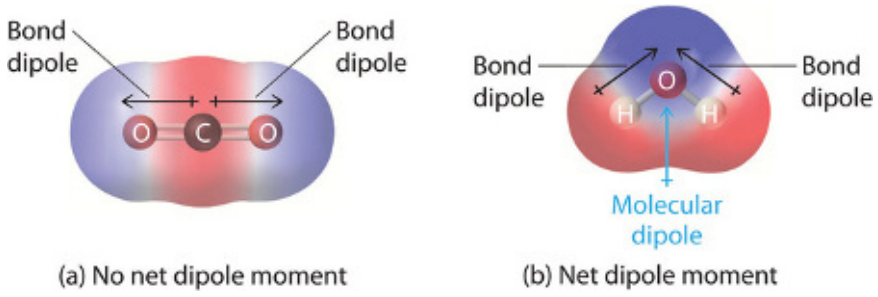


Figure 5: .

$$M_0 = m$$

The **first moment** of mass for an object is $M_1 = \sum_{i=1}^{N} m_i d_i^1$ and determines the **center of mass** in relation to the d dimension. It is the point on which the bird on the photo is balanced.

$$M_1 = C_m$$

The **second moment** of mass is $M_2 = \sum_{i=1}^{N} m_i d_i^2$ and is the **moment of inertia** . Corresponds to the system's resistance to rotations. Note that the terms d_i^2 would be present in the area of a circle with a center identical to the object and a radius equal to the distance to the center: πd^2. The total resistance to rotation is analogous to the resistance offered by the rays of these imaginary circles [3].

The nth moment is given by

$$M_n = \sum_{i=1}^{N} m_i d_i^n$$

.

[3] https://physics.stackexchange.com/a/371165/218274

Generalizing moments

We calculate some moments for physical objects (moment zero: total mass, first moment: center of mass, second moment: moment of inertia). We can generalize the concept and calculate moments of abstract entities, such as random variables. **The good news is: we've done this before!**

Let $f(x)$ be the function that describes a probability distribution for the variable,

Just as the **moment zero** represents the sum of the contribution of each point to the mass (total mass), here it represents the sum of the possible probabilities, the total probability (1).

The **first moment** corresponds to the center of mass in static mechanics. For probabilities, it is the center, the **average**.

The **second moment** corresponds to the inertial moment and it is the **variance**. The normalized **third** and **fourth** moments report asymmetries *(skewness)* and fat tails *(kurtosis)*.

Formally, let $d(x, x_0)$ be the distance to the center x_0 reference $(x - x_0)$, the nth moment μ_n is defined by:

$$\mu_n = \int_{-\infty}^{\infty} d(x, x_0)^n f(x) dx$$

The above integral corresponds to the continuous version of the sum of discrete parts presented before for a physical quantity, such as mass: $M_n = \sum_{i=1}^{N} d_i^n m_i$

Moment zero:

$$\mu_0 = \int_{-\infty}^{\infty} d(x, x_0)^0 f(x) dx$$

The sum of all probabilities of a distribution must add up to 1.

$$= \int_{-\infty}^{\infty} f(x) dx = 1$$

. **First moment**: $\mu_1 = \int_{-\infty}^{\infty} d(x, x_0) f(x) dx$, assuming center at 0 $(x_0 = 0)$, we have the average,

$$\mu_1 = \int_{-\infty}^{\infty} x f(x) dx$$

, also called expected value $E[X]$. It extends the intuition of adding all the measures and dividing them by the number of observations. We use an integral to add the infinitesimal possibilities for $f(x)$.

Second moment:

$$\mu_2 = \int_{-\infty}^{\infty} d(x, x_0)^2 f(x) dx$$

. The sum of the squares of the deviations, our variance,

$$\sigma^2 = E[(x - \mu)^2]$$

.

Computers

Master Foo and the Shell Tools[4]

---

*A Unix novice came to Master Foo and said: "I am confused. Is it not the Unix way that every program should concentrate on one thing and do it well?"

Master Foo nodded.

The novice continued: "Isn't it also the Unix way that the wheel should not be reinvented?"

Master Foo nodded again.

"Why, then, are there several tools with similar capabilities in text processing: sed, awk and Perl? With which one can I best practice the Unix way?"

Master Foo asked the novice: "If you have a text file, what tool would you use to produce a copy with a few words in it replaced by strings of your choosing?"

The novice frowned and said: "Perl's regexps would be excessive for so simple a task. I do not know awk, and I have been writing sed scripts in the last few weeks. As I have some experience with sed, at the moment I would prefer it. But if the job only needed to be done once rather than repeatedly, a text editor would suffice."

Master Foo nodded and replied: "When you are hungry, eat; when you are thirsty, drink; when you are tired, sleep."

Upon hearing this, the novice was enlightened.*

[4] http://catb.org/esr/writings/unix-koans/shell-tools.html

Throughout the text, we will use examples with software. Computers are useful for speeding up the calculations necessary for our purposes.

For millennia, man has used instruments, such as abacuses and tables, to perform extensive and accurate operations involving large numbers. Faced with a problem or operations to be computed, these mechanisms automate parts of the process due to the way they were built. The main difference from these tools to today's computers is that our machines can be programmed to do arbitrary computations.

Ada Lovelace (*10 December 1815 - 27 November 1852*) was the first to discover this possibility. Studying Charles Babbage's Analytical Machine, Ada devised a way of performing computations for which the machine was not originally designed. The program calculated the Bernoulli numbers. Arguably, changing the structure of simpler machines also involves reprogramming them.

Machines from that time weighed tons and were much slower. The advancing years have made technology more accessible, to the point of enabling high-powered, low-cost personal computers. In addition, instead of complex mechanical operations, we can use programming languages that translate English-based commands into machine instructions.

The programs presented here are written in R and Julia. They are languages aimed at statistical computing, having useful tools. Being 'high-level' languages, we have no cognitive overhead with memory and hardware handling for the programmer. Abstraction of physical details, such as CPU registers, is done automatically by the interpreter. The data visualization ecosystem has power and flexibility. The community of both grows fast, with large bases of support. Both support functional, object-oriented and imperative styles.

R Download and installation instructions can be found at: https://cloud.r-project.org/ On Windows, the process usually consists of clicking on the installation executable and agreeing to the prompts. For Linux, it involves adding CRAN to the list of repositories and downloading the *r-base* package or the source / tarball directly from the website.

Rstudio With R installed, I recommend using the RStudio development environment (https://www.rstudio.com/) to get some facilities. Among them: *vim* shortcuts, editor with syntax highlighting, autocomplete, real-time rendering of plots and animations, visualization of datasets, development environment, logs, markup languages support, such as Markdown, RMarkdown and Latex. [5]

---

[5] This text is written in Markdown and the source code can be found at https://github.com/fargolo/stat-learn

Mean and Variance

We can define the function of the mean for a vector of numbers, given by (1) sum divided by (2) size of the vector:

```
> mean_vec <- function (x) {
  sum (x) / length (x)
}
> mean_vec (b) # Previously defined by b <- c (2.2, 4.4, 5.5)
[1] 4.033333
```

sum (x) returns the sum of all elements of the vector x. length (x) returns the size (number of cells) of the vector x.

As previously described, the mean is a measure of central tendency for a set of observations. It is the closest point to all the others.

Many ways to calculate variance We can also calculate a measure related to how far our values deviate from the center.

First, we calculate a distance between each element x and the average of μ observations. The notion of distance implies that it must be a positive value. Assuming that x and μ are measured in an ordered space, we can use the module of the difference between the values: $\|x - \mu\|$. Also, we can use the difference square: $d_i = (x_i - \mu)^2$.

The σ^2 variance of the observations is a measure of the dispersion of the entire sample. To calculate σ^2, we add all the distances d_i and divide the result by $n - 1$.

```
>var_2 <- function(x) sum((x - mean(x))^2) / (length(x) - 1)
>var_2 (b)
[1] 2.823333
```

Being proportional to the distances from the mean, the variance σ^2 tends to be greater when the values are very different from each other:

```
>c <- c(100,200,1,45,-24)
>var_2(c)
[1] 7966.3
```

Another measure of dispersion, given in the original units of the observed measure, is the standard deviation σ, given by the root of the variance σ^2.

```
>var_2(b) %>% sqrt
[1] 1.680278
```

The division by $(n - 1)$ instead of n is a correction applied in order not to underestimate the population value of the variance ($\frac{\sum_1^n (x-\mu)^2}{n}$) when we used a

sample ($\frac{\sum_1^n (x-\mu)^2}{n-1}$).

Vectors, loops and recursions

Previously, we defined the function to calculate variance as:

```
>var_2 <- function(x) sum((x - mean(x))^2) / (length(x) - 1)
```

This is only possible because R automatically applies functions to vectors. Thus, the expression *(x - mean(x))^2* subtracts the average of each element of the vector x. Usually, it is necessary to use recursive structures for this. The for loop defines a sequence of defined size n and repeats a block of commands n times. If we want to print numbers between 1 and 10:

```
>for (i in 1:10) print(i)
[1] 1
[1] 2
[1] 3
(...)
[1] 8
[1] 9
[1] 10
```

The instruction evaluates print (i) for values i = 1,2,3 .., 10 repeatedly. Let's rewrite our function to calculate variance σ^2 using a loop. We can define a loop with the size of the vector x and square the difference in each element. Like this,

```
var_3 <- function (x) {
  # empty vector to store distances
  accumulator <- numeric ()
  # loop starts at 1 goes up to the given vector size
  for (i in 1: length (x))
  # calculates and stores distances.
accumulator [i] <- (x [i] - mean (x)) ^ 2
#calculate media
  return (sum (accumulator) / (length (x) - 1))
}
```

Both definitions have the same result as the native implementation:

```
> var(b)
[1] 2.823333
> var_2(b)
[1] 2.823333
> var_3(b)
[1] 2.823333
```

Yet, one way to manipulate many elements is through high-order functions.

These functions are given other functions as arguments. An example is the map function from lib purrr. We define a function for the distance, $f(x) = (x - \mu)^2$, and apply it to all elements. Only then do we add the results and divide by n-1.

```
> y_mean <- mean (a)
# Apply distance function and store values
> sq_dists <- purrr :: map_dbl (.f = function (x) (x - y_mean) ^ 2,
.x = a)
# Sums distances and divides by n-1
> sum (sq_dists) / (length (a) - 1)}
```

When using the pipe, the period character (.) Refers to the value provided as input by the previous pipe. Thus, sum (.), In the following example, adds the values passed by the *map* function. Our function can be written:

```
var_4 <- function(arg){
y_mean <- mean(arg)
purrr::map_dbl(.f = function(y) (y - y_mean)^2, .x = arg) %>%
sum(.)/(length(arg) - 1) }
> var_4(b)
[1] 2.823333
```

Exercises

1. What is the difference between compiled and interpreted languages?
2. A program written in R can be written in any other language. Is this statement true? Why?
3. Name 3 resources that an IDE provides the programmer.
4. Change the RStudio background theme to a dark color (less light for the eyes :)).
5. Apply the sd, mean and var functions to normal random samples of n = 10, 30, 100 and 300. The rnorm (n, mean, sd) function can help. Compare the values of the source distribution with those obtained.
6. *UnLISP it!* Transform the following expressions, replacing nested parentheses with the pipe operator (%>%) when deemed convenient: *> round (mean (c (10, 2, 3)))* > round (mean (rnorm (n = ceiling (runif (1,0,10))))) *> paste ("a", seq (1: max (sample (1:10))))* > round (nrow (iris) + exp (1), digits = ceiling (runif (1,0,10)))
7. Using the code for the functions var_2 (vectorized), var_3 (for loop) and var_4 (high-order function map)

- Write the corresponding functions (sd_2, sd_3, sd_4) for standard deviation and compare with the standard function of R (sd). Tip: Just apply square root to the final value previously returned!

8. Using the iris dataset

- Select only those examples with a petal size greater than 4.
- Select the 10 largest copies. Suppose the size is given by the average of the 4 measurements provided.
- Calculate the mean and standard deviation for two measurements in each species.
- Make a scatterplot between two measurements
  - Add colors according to the species
  - Add the text label to one of the points
  - Change titles (main, x and y axes, legend)
  - Change the background theme. Tip: try lib *ggthemes* themes

9. Using loops, write a function that returns an approximation of e.

- Remember that $e = \lim_{n\to\infty}(1 + \frac{1}{n})^n$.

Part 2 - The normal distribution and a curious theorem

In empirical studies, it is common to assume that measures of a random variable come from a population with a normal distribution. Next, we will study the behavior of this probabilistic function.

Abraham de Moivre (26 May 1667 - 27 November 1754), without lacking funding for studies and research, provided secondary services. Among them, probability calculations for clients in gambling. In 1733, de Moivre realized that the probabilities of a binomial distribution, such as the $(p(cara) = p(coroa) = 0.5)$ coin flip, approach a smooth (continuous) curve as the number of events increases.

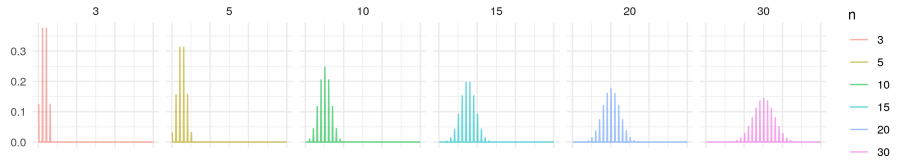


Figure 6: Binomial distributions for different numbers of launches with p = 0.5 (e.g., one coin flipping). For n> 1, extreme values indicate results with only heads (left tail, 0000...) or crowns (right tail, 1111...)

Bernoulli's distribution describes the possibility of two events, such as the coin toss. Taking the different values of heads (0) and crowns (1), the observation is 1 with probability p and 0 otherwise $(1 - p)$. For an honest currency, we have a uniform probabilistic distribution over the domain, $X = 0, 1$: $P(1) = P(0) = 0.5$.

If we add Bernoulli distributions, we obtain the binomial distribution. Each observation is a set of releases. Taking $p = 0.5$, more frequent results are similar numbers of heads (0) and crowns (1).

For $n = 10$, it is much more likely to get a number of heads close to 5 (center of the curves) than a result with 9 or 10 equal throws. It is possible to demonstrate that increasing the value of n causes the distribution to approach the following continuous curve:

De Moivre noted that the distribution of binomials with many launches approached that of a smooth function. He sought an approximation in terms of the exponential function [natural] e^x.

But what are the parameters of the curve?

First, de Moivre deduced the solution to the $(p = \frac{1}{2})$ currency problem. The following general expression describes the probability $P(x)$ corresponding to the curve we are looking for, known as *Gaussian*.

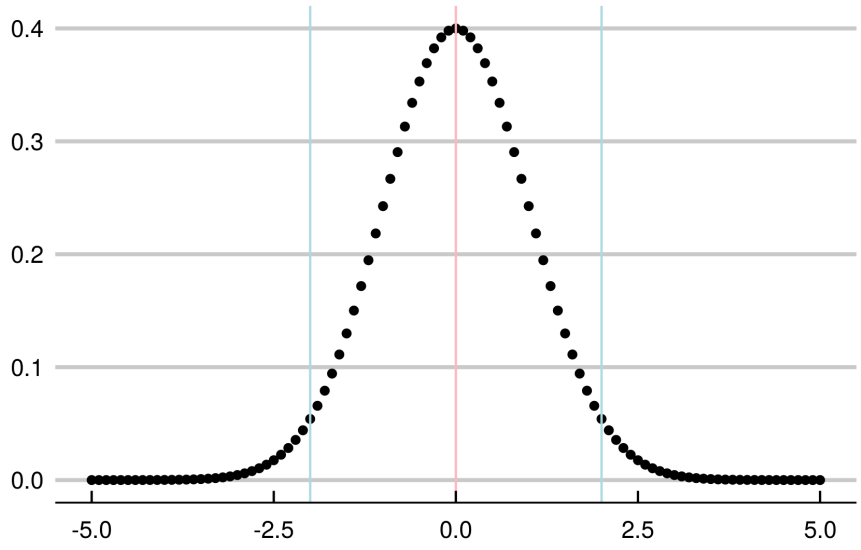


Figure 7: Normal (Gaussian) distribution, whose shape resembles that of a bell

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Where e is Euler number ($e \sim 2.72\ldots$).

The value $\frac{1}{\sqrt{2\pi}}$ appears as a normalizer for evaluating the function as a probability density (The integral of $-\infty$ a $+\infty$ must be 1). The value π arises from the Gaussian integral for e^{-x^2} and results from the fact that $2\pi i$ is a period of the function e^x:

$$\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$$

Intuitions The definition has a constant factor $\frac{1}{\sqrt{2\pi}}$ (approximately 0.4), multiplying the exponential result in the format e^{-x}. In Julia, we can define the function and observe the probability associated with some points around the maximum ($f(0) = 0.4$):

```
mgauss(x) = 0.4*exp((-1)*(x^2)/2)
mgauss(-2), mgauss(-1), mgauss(0) , mgauss(1) , mgauss(2)
    (0.054134113294645084, 0.2426122638850534,
    0.4, 0.2426122638850534, 0.054134113294645084)
```

Then, obtain some values in the range $[-5, 5]$ and plot them, giving rise to the previous Gaussian curve.

```
using Plots
gauss_values = map(mgauss,-5:0.1:5)
plot(gauss_values,xaxis=("Gaussian"),leg=false)
```

We observe how the distribution occurs from the equation.

Since x^2 returns only positive values, $-x^2$ always returns negative ones. The function returns values between 0 and 1 exponentiating ($e \sim 2.718...$) to a negative quadratic factor ($y \sim 0.4 * e^{-x^2/2}$).

We also noticed that values close to the center ($x \sim \mu = 0$) cause the exponent to approach 0, maximizing our function: $f(0) = 0.4 * e{-x^2}/2 = 0.4 * e^0 = 0.4$). The obtained value (0.4) corresponds to the top of the curve in the graph above (pink line).

We observed the curve symmetrically approaching the maximum at values close to 0.

This directly reflects the fact that values close to the average are more likely and extreme values less likely. Strictly speaking, the probability for any value among the possible infinites is zero.

It is possible to evaluate the probability of any event related to the interval between points a and b by the integral of $f(x)$ over the interval $[a, b]$:

$$P(A_{a,b}) = \int_a^b f(x)dx$$

For example, an event (A) related to 'values less than or equal to zero' on a scale is in the range $[-\infty, 0]$:

$$P(A) = \int_{-\infty}^0 f(x)dx$$

The quadratic term makes the distribution symmetrical for values opposite to the mean. $P(A) = P(-A)$. As we calculated $f(2)$ before, we know that: $f(-2) = f(2) = 0.05$ for $\mu = 0$. It is also likely to find values two units larger or two units smaller than the average. These points are marked by a blue line in the figure.

We can work with normal curves with centers (average μ) shifted to the left (μ <0) or to the right (μ> 0), subtracting the term from x in our exponent. In addition, different variances (σ^2) reflect the frequency of values far from the average and how far from it they are. Visually, it determines the size of the bell base in the illustration (Figure 3).

We use the notation $N \sim (\mu, \sigma^2)$ to describe a Gaussian distribution with average μ and σ^2 arbitrary variance:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Why do we use The Normal Distribution? Are large binomial distributions and currencies so important? The launches are an example of a larger class of phenomena. Each result series is made up of many almost identical events (individual launches).

Entropy In the natural sciences, we rarely know in advance the mechanisms by which observations are generated. Consequently, we do not know the probabilistic distribution that they obey. A fair dice has equivalent probabilities among the possible values (uniform distribution). A rigged dice tends to fall more frequently at certain values (peaks and valleys). As we saw earlier, we can describe a distribution through the relationships between possible values and the center. These are the *moments*. The first moment reflects the relative position of the center (mean), while the second reflects the dispersion of the values (variance). Knowing the natural source mechanism would allow specifying distributions directly, however we need to face the limitations of the real world.

If we only know the center (mean) and the dispersion (variance) of a distribution, what is the most conservative guess possible?

Considering real numbers (domain in $[-\infty, +\infty]$), the normal distribution is that with maximum entropy in relation to the others. Roughly speaking, this means that it is the description using less information when compared to any other distribution obeying these restrictions (defined mean and variance).

Gaussian is the one that introduces less extra information in relation to possible true distributions. By the *Maximum Entropy Principle* , it best describes observations a priori, when we only have an idea of the mean and variance. This is a reasonable justification for adopting Gaussians as tools.

The proof is reasonably complex, involving calculus of variations to optimize the expression:

$$H(x) = -\int_{-\infty}^{+\infty} p(x) - \log\, p(x) dx$$

The Central Limit Theorem Another connection between normal distribution and the natural sciences is the central limit theorem. If we add many

distributions from the same family, the resulting distribution approaches a normal one. Without much explanation, we assumed that this was true for coins. Some examples may help to gain intuition. When rolling a 6-sided fair dice, we have a probability of $\frac{1}{6}$ in each result.

A uniform discrete distribution, where $P(1) = P(2) = P(3) = P(4) = P(5) = P(6)$ is defined for natural numbers between 1 and 6: $X \sim U_{discr}(1, 6)$.

The average for many launches, or expected value, is given by: $E(X) = E(U(1, 6)) = (1 + 6)/2 = 3.5$

Let's do a virtual experiment using 100 rolls of 11 dice.

Julia Download and installation instructions can be found at: https://julialang.org/ For Linux, it involves downloading the binary / source code / tarball directly from the website.

IDE With Julia installed, I recommend using the Juno development environment (http://docs.junolab.org/stable/) to get some facilities. Among them: shortcuts, editor with syntax highlighting, autocomplete, real time rendering of animations and plots, visualization of datasets, development environment, logs, support for markup languages.

Final notes on the Central Limit Theorem

We can better understand the central limit theorem. The information provided by the moments is valuable: a probability function is fully defined by its moments. The Central Limit Theorem, which we talked about earlier, is proven to show equivalence between moments of the normal curve and the sum of n identical distributions through other tools.

We can create a *Moment generating function*, $M_X(t) = E[e^{tX}]$ where t is a fixed value. We call it that, because its polynomial form via Taylor expansion corresponds to a series that contains all the moments M_n: $1 + tX + \frac{t^2 M_2}{2!} + \frac{t^3 M_3}{3!} + ...$, since $\frac{de^x}{dx} = e^x$ and the order derivative n multiplies the order $n - 1$:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$$

$$E[M_X(t)] = 1 + tE[X] + \frac{t^2 E[X^2]}{2!} + \frac{t^3 E[X^3]}{3!} + ...$$
$$= 1 + tM_1 + \frac{t^2 M_2}{2!} + \frac{t^3 M_3}{3!} + ...$$

The *Characteristic function* is the Fourier transform of the density function, associating values with periodic components in the imaginary plane. It involves multiplying t by the unit in the definition of the moment generating function $M_X(t) = E[e^{tX}], \phi_X(t) = M_X(it) = E[e^{itX}]$. It is possible to use the characteristic function to show that the moments in the sum of similar distributions converge to the moments of a Gaussian distribution. That is: $\phi_{\sum X_n}(t) \sim \phi_{N(\mu,\sigma)}(t)$ for similar X_n [6].

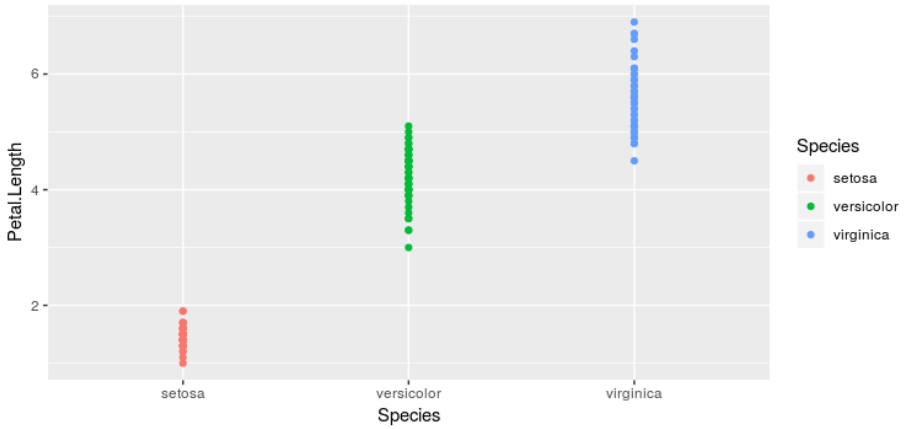
[6] The first tests assumed X_n identical, but more general versions were demonstrated. Two Proofs of the Central Limit Theorem, Yuval Filmus, 2010. http://www.cs.toronto.edu/~yuvalf/CLT.pdf

##Part 3 - Graphics grammar

Bertin [7] outlined this approach, which consists of mapping data characteristics to visual elements following a consistent syntax.

One of the prominent tools in the R ecosystem is *ggplot**. It provides a very powerful and flexible syntax for plotting visualizations. The secret lies in its design, which uses graphics grammar (* *G rammar of G raphics Plot). Lib ggplot implements a layered grammar, allowing overlays for complex graphics. The Julia implementation is at Gadfly.*

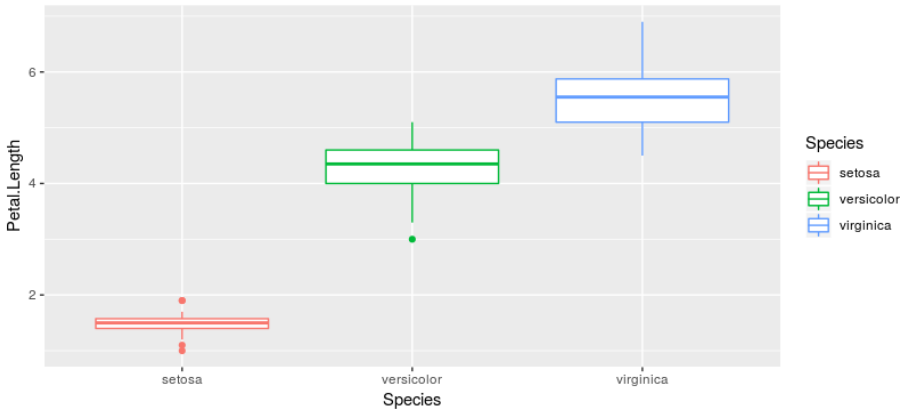
```
>library(ggplot2)
>ggplot(data=iris,aes(y=Petal.Length,x=Species,color=Species))+
  geom_point()
```



To illustrate the flexibility of the library, note that changing only the geometric object (geom), we obtain a different graph, keeping data and relationships (mappings) the same:

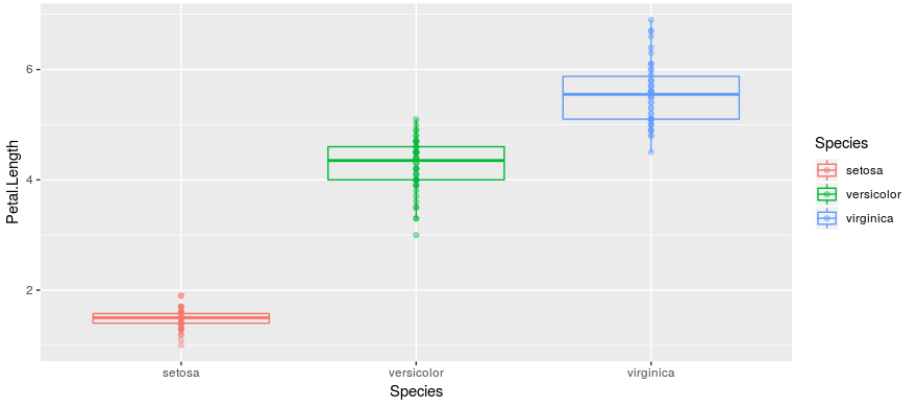
```
>ggplot(data=iris,aes(y=Petal.Length,x=Species,color=Species))+
 geom_boxplot()
```

[7] Wilkinson L. The grammar of graphics. InHandbook of Computational Statistics 2012 (pp. 375-414). Springer, Berlin, Heidelberg. Bertin, J. (1983), Semiology of Graphics, Madison, WI: University of Wisconsin Press.



The above figures are known as boxplots. The center corresponds to the median (50th percentile), the borders correspond to the 25th (lower) and 75th (upper) percentiles. The wires, known as "whiskers", extend up to 1.5 * IQR (where IQR = 75th percentile - 25th percentile). It is possible to add layers and they can overwrite information from previous layers. This makes the syntax of ggplot highly modular. Next, we overlap points and boxplot:

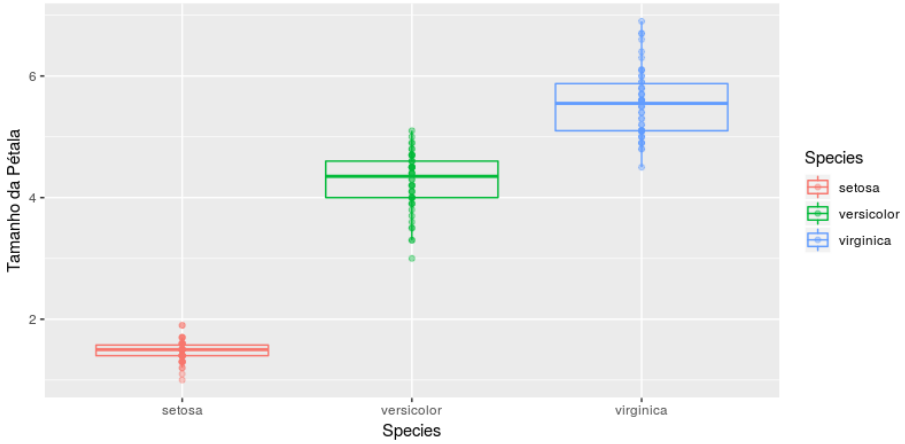
```
> ggplot (data = iris, aes (y = Petal.Length, x = Species, color = Species)) +
  geom_point (alpha = 0.4) + # layer 1
  geom_boxplot (alpha = 0) # layer 2
```



The *alpha* parameter regulates the transparency of objects. We place the boxplot with full transparency (alpha = 0), giving visibility to the points (alpha = 0.4). We added some degree of transparency so that overlapping dots are darker than individual dots. We will add a third layer, which replaces the y-axis label for a Portuguese caption:

```
> ggplot (data = iris, aes (y = Petal.Length, x = Species, color = Species)) +
  geom_point (alpha = 0.4) + # layer 1
```

```
    geom_boxplot (alpha = 0) + # layer 2
    ylab ("Petal Size") # layer 3
```



Still, there are themes ready to change the overall style of the image:

```
> ggplot (data = iris, aes (y = Petal.Length, x = Species, color = Species)) +
  geom_point (alpha = 0.4) + # layer 1
  geom_boxplot (alpha = 0) + # layer 2
  ylab ("Petal Size") # layer 3
  theme_bw () # layer 4: theme
```

! [] (images / chap0-gg-five.png)

```
> ggplot (data = iris, aes (y = Petal.Length, x = Species, color = Species)) +
  geom_point (alpha = 0.4) + # layer 1
  geom_boxplot (alpha = 0) + # layer 2
  ylab ("Petal Size") # layer 3
  theme_economist_white (gray_bg = F) # layer 4: theme
```

! [] (images / chap0-gg-six.png)

In Gadfly (Julia), the syntax is similar:

```
julia> using Plots, Gadfly, RDatasets, ColorSchemes
julia> plot (iris,
    layer (y =: PetalLength, x =: Species, color =: Species,
      Geom.point, Geom.boxplot),
    Guide.ylabel ("Petal Size"))
```

Petals with density (`Geom.violin`) and Sepals in boxplot (`Geom.boxplot`).

```
julia> plot (iris,
    layer (y =: PetalLength, x =: Species, color =: Species,
      Geom.violin),
```

```
    layer (y =: SepalLength, x =: Species, color =: Species,
      Geom.boxplot))
```

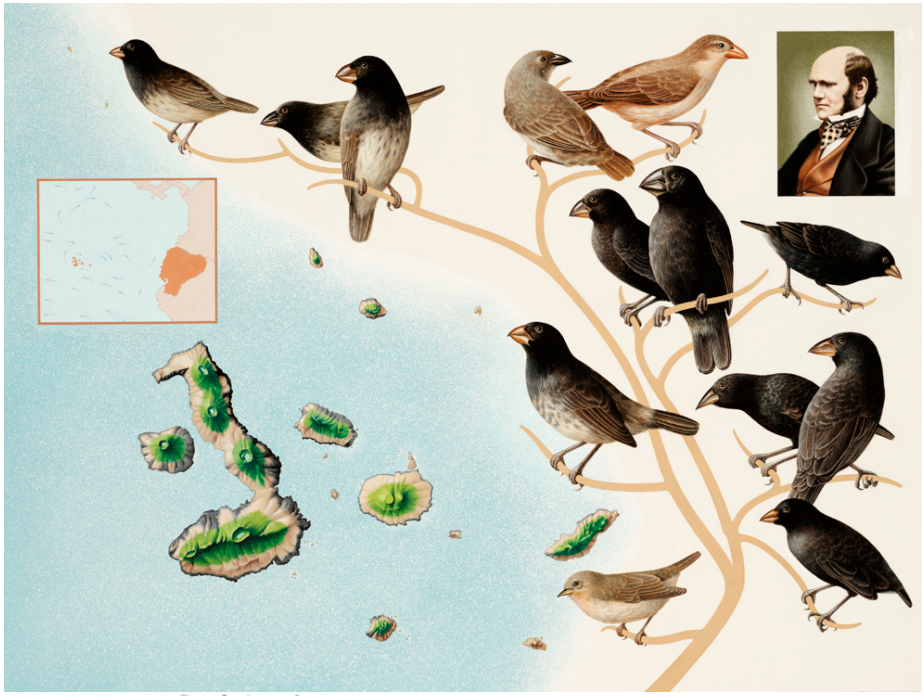
Chapter 2: Hypothetical-deductive method and Darwin's finches

Statistical tests and probabilistic distributions

Part 1 - Introduction

Charles Darwin observed that finch birds on the Galapagos islands had varieties of beak shape and size. His intuition about the origin of varieties from a common ancestor was one of the most scathing in "On the Origin of Species" (1859).

In this chapter, we will simulate data for a quantitative approach to the problem. We will study finch beak measurements in small samples from each island and make inferences about the populations of origin (different species).



South American Finch

Food source: small seeds

Three Types of Galápagos Finches

Food source: small insects

Food source: cactus flowers, fruit, and nectar

Food source: large seeds

Galapagos Islands

On his trip aboard the Beagle, Darwin described a group of birds that inhabit the Galapagos Islands, an archipelago located approximately 900 km off the coast of Ecuador (South America). The variety in beak sizes drew attention: *It is quite remarkable that an almost perfect gradation in the structure of this group can be traced in the shape of the beak, from one exceeding the dimensions of the largest of the big beak sparrows, to another differing little from the papa. blackberries.* [8]

He noted that the variety of beaks was adapted to the diet of each group: fruits, nuts, insects. The pointed beak can eat fruit and aril from the seed of the cactus, while the short beak shatter the base of the cactus and eat its pulp.

Before the publication of The Origin of Species, the case of finches (the name of these birds) already contained an embryo of the natural selection process. In the second edition, in 1845, he speculated about a common ancestral group shaped by specific environments:

(. . .) [when] seeing this gradation and diversity in structure in a small, closely related group of birds, it is possible to imagine that, from a few birds in this archipelago, a species was chosen and modified for certain purposes. [9]

Doubts - Hypotheses and observations

Darwin took approximately 20 years between the initial iception of the idea (1838) and the publication of the work (1859). Aware that similar proposals were ridiculed, he was meticulous in defending his theory about the origin of species.

Bird watching on the island was evidence, but it did not confirm the theory. Darwin then devised an investigation plan to test several different consequences of the theory. Geographic distribution, phenotypic variability (hybridization and cross-fertilization), variation under domination ... Would experiments in these areas obey predictions?

The two decades were dedicated to contacting and interacting with specialists from different areas (from botany to the breeding of pigeons and rabbits). The accumulated evidence spoke strongly in favor of the Darwinian explanation, which described different fields in a comprehensive and simple model. The ant's

[8] *"It is very remarkable that a nearly perfect gradation of structure in this one group can be traced in the form of the beak, from one exceeding in* ***dimensions*** *that of the* ***largest*** *gros-beak, to another* ***differing*** *but little from that of a warbler".*[^4] Tradução livre. The Voyage of the Beagle (1839).

[9] *"Seeing this gradation and diversity of structure in one small, intimately related group of birds, one might really fancy that from an original paucity of birds in this archipelago, one species had been taken and modified for different ends."[^5].*Darwin, Charles (1845), Journal of researches into the natural history and geology of the countries visited during the voyage of H.M.S. Beagle round the world, under the Command of Capt. Fitz Roy, R.N (2nd. ed.), London: John Murray

job was to explore data and convince other scientists to accept the idea. This lasted until Alfred Wallace anticipated some of the most striking consequences in 1855, which Darwin had avoided attacking directly. ("On the Law which has Regulated the Introduction of New Species", Annals and Magazine of Natural History).

Charles Lyell was a geologist, a friend of Darwin's, and he was the one who strongly encouraged the publication of a solid exposition of the theory. The theory conceived in 1938 for the origin of species could be wrong, even though the Beagle's evidence was promising. The study of secondary hypotheses would clarify the veracity of the theory. The experimental confirmations provided security for a convincing defense.

Odds
It is interesting to note that the language used to denote differences is eminently quantitative (in the excerpt above: *dimensions*, *largest*, *differing*).

Darwin observed the suitability of the beaks to the diet through his intuition, without taking measures.
Visual inspection by a trained naturalist was able to detect these nuances. In his perception, there were a total of **3 species* *on 4 islands: 1 on Charles Island, 1 on Albemarle Island and 1 on James and Chatham Islands. Initially, he noticed that the birds were similar to those seen in Chile. Darwin collected 26 birds and took them back for an ornithologist to study in more detail. The expert (John Gould) suggested that the 26 birds represented 12 completely new species, a number that later rose to 25. Today, taxonomists suggest a number of* *15 species**.

Like the naturalist, we will examine the differences for different groups. However, we will use statistics and probabilities (normal distribution and Student's t) to test hypotheses and make more accurate conclusions about the measures.

Falsifiability and hypotheses

Philosophers of science study characteristics in the modus operandi of other scholars. What is there in common between the procedures employed by biologists and geologists? What distinguishes Charles Darwin and Paul Dirac from John Dee and Edward Kelley? What works in different areas of human knowledge?

We adopted the collective name of "sciences" for some areas of knowledge. In addition, we associate them with common characteristics in procedures and internal structure. Somehow, scientificity communicates credibility. In recent decades, philosophers have discussed the validity of the problem of demarcating science from pseudoscience and non-science. [10] In this chapter, we will stick to an older and arguably influential conceptual paradigm.

The hypothetical-deductive method was popularized in the 20th century as an identification flag associated with scientific work. A cycle that consists of formulating theories, designing experiments, testing falsifiable hypotheses, verifying results and repeating the process in an iterative way.

The rationale for using testable hypotheses is that valid propositions about a system contain information that helps to predict it. Thus, "it is sunny or not sunny tomorrow" is a useless proposition, while "it is sunny tomorrow" is a useful proposition. Note that "it is sunny tomorrow" is a testable (falsifiable) hypothesis, while "it is sunny or not sunny tomorrow" is a true hypothesis regardless of observations.

Theophrastus (c. 371 – c. 287 BC) and Eudemus (400 BC) originally outlined *modus tollens*.

1 . If the theory is true, fact X will happen, 2 . X did not happen, *soon* the theory is false.

K. Popper was a leader in the revitalization of the hypothetical deductive method in the past century. For him, the difficulty in generating testable and falsifiable hypotheses signaled an evident weakness in theories. The freedom to test the veracity of theories is a hallmark of science as opposed to esoteric and / or authority-based practices.

Popper severely attacked Karl Marx's dialectical materialism, as well as Charles Darwin's theory of evolution by natural selection and psychoanalysis.

These branches of human knowledge encountered difficulties with the proposed demarcation criterion. Marx predicted that the revolution would happen in an industrialized nation, through the working class and other events that did not materialize. His followers used *ad-hoc* hypotheses to justify their observations while maintaining predictions made in the light of dialectical materialism. Darwin's theory of evolution by natural selection was supported by many examples of impossible reproduction (e.g. recomposition of the evolutionary trajectory in

[10] Massimo Pigliucci - Philosophy of Pseudoscience: Reconsidering the Demarcation Problem

fossils). Psychoanalysis has also suffered harsh criticism, due to the irrefutability of its central pillars.

As we will discuss in the following chapters, hypotheses are not essential in the scientist's life. However, falsifiable predictions are extremely useful to demonstrate the usefulness of a theory. A specialist has an extremely increased credibility when he / she usually makes guesses in uncertain situations and the same goes for scientific theorists.

Hypothesis tests can be formalized through probabilities and statistics, which incorporate quantitative aspects. We calculate the probability associated with observations, considering the scenario of a hypothesis (falsifiable). This rationale adapts robust mathematical tools to the hypothetical-deductive epistemological platform, being a dominant model of production in experimental sciences.

Hypothesis tests We usually start from a base hypothesis, called null hypothesis, which describes the least interesting scenario, that is, the inexistence of the phenomena proposed by the scientist.
It is common to compare two groups, A and B, as to the result of an intervention. The null hypothesis usually assumes that the groups present equal results. We want to study the size of the bird beaks of islands A and B. The null natural hypothesis assumes that the species are the same: There is no difference between the beaks of birds of type A and B.
We measure the beak of some birds in the two groups and calculate the probabilities of finding these measurements considering that A and B come from equal populations. If the differences are large, the probability is very low. We reject our hypothesis and accept the alternative (there is a difference between A and B).

Structuring the steps:

1. We define the null hypothesis (H_0) and at least one alternative hypothesis(H_1).

- H_0: Birds on islands A and B have beaks of equal size.
- H_1: Birds have different sized beaks.

Then, we can do an experiment, collecting experimental measurements for the nozzle length. These measures, together with reasonable mathematical premises, allow us to speculate: what is the probability p of obtaining our observations considering equal distributions between A and B? That is, considering H_0 true, would our results be rare or common?

If p is less than a predefined threshold (conventionally, 0.05), we reject H_0. The probability is very small for H_0 to be true.

The domain of hypothetical-deductive procedures in the sciences has produced interesting results. Especially in the work axis called by Thomas Kuhn "normal science", focused on accumulating evidence and testing hypotheses. The ideal of

designing an impartial experiment, with the possibility of failure, sharpened the perception of researchers for the fallibility of ideas. The degree of sophistication in reproducibility of procedures has been amplified.

Note *We use the lower limit of 0.05 as a criterion to reject the null hypothesis, which may seem arbitrary. And is. The p-values were interpreted according to their magnitude and statistics based on which they were calculated. It was Ronald Fisher, in Statistical Methods for Research Workers (1925), who proposed (and later popularized) the number: "The value for which* $p = 0.05$*, or 1 in 20, is 1.96 or nearly 2; it is convenient to take this point as a limit in judging whether a deviation ought to be considered significant or not.* [11] *"*

Part 2 - Darwins's Finches and a parametric test

We will show how the individual contribution of genes with additive effects of uniform distribution results in approximately normal measurements for the birds' beaks.

We will simulate the beak measurements in 4 samples (n = 150) of birds. The size of the beaks is given by the additive effect of many similar genes, so we expect their distribution to be normal by the Central Limit Theorem.

A copy of the gene adds x millimeters to the final size. The value of x is drawn from a random variable of uniform distribution, $X \sim U(0, 1)$. Birds have a fixed number of n of additive genes in each sample, drawn in the range between 80 and 100. The final measurement of the beaks is given by the sum of the effects of the n genes. This number is fixed for each population and varies between populations.

To simulate the data with the above conditions:

```
using Distributions , DataFrames , Random
Random.seed!(5)
n_birds = 150 # sample_size
genes_low = 80 # lower bound on number of genes
genes_hi = 100 # upper bound on numbe
n_islands = 4 #samples
# Function that adds uniform distributions (n = n_genes)
function unif_sum(n_genes)
          gene_samples = [rand(Uniform(0,1),100) for _ in 1:n_genes]
          effects_sums = sum(gene_samples)
          return effects_sums
        end
```

[11] The value [of the z statistic on a normal curve] for which $p = 0.05$, or 1 in 20, is 1.96 or approximately 2; it is convenient to take this point as a limit when judging when a deviation should be considered significant or not.

```
# Function to generate n_pop individuals with n_genes
function generate_pop(;n_pop,n_genes)
          pupulation = [unif_sum(n_genes) |> mean for _ in 1:n_pop]
        end
# Generate random samples with n_birds;
# n_genes of each island varies between genes_low genes_hi
galapagos_birds = map(x -> generate_pop(n_pop=n_birds, n_genes=x) ,
    rand( DiscreteUniform(genes_low,genes_hi), n_islands)) |> DataFrame
```

As expected, we found that the histogram of the final measurements is close to that of a Gaussian.

```
using StatsPlots
@df stack(galapagos_birds) groupedhist(:value, group = :variable,
      bar_position = :dodge,bins=50,title=("Darwin Finches"),
    xlabel="Beak Size",ylabel="Count",legend=false)
```

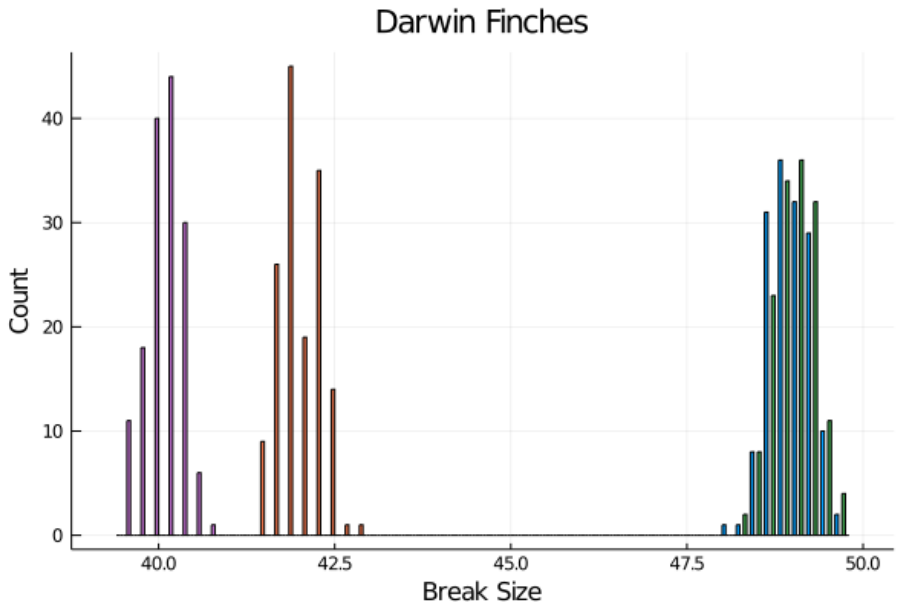


Figure 8: Figure 4. Distribution of nozzle measurements in simulated populations for genes with an additive effect.

The random numbers generated using the suggested seed (`Random.seed!(5)`), line 4 of the code above) are similar to Darwin's assumption: 4 islands (samples) and three species (nozzle distributions). We noticed that there are two samples (purple, red) of very similar measures and two separate ones (green, blue). Assuming that we measure the beaks of some birds, how do we know if the groups are different?

By calculating the differences between distributions, we can infer whether two samples have the same number of underlying genes! For this, we will use a rationale and some new tools.

Student's t test and t-distribution: A practical example

To statistically test whether the measurements are different, we will perform a t test to compare the groups.

The t distribution arises when we want to understand how unlikely our estimates (μ') are by assuming a hypothetical real average (μ) of origin in an unknown normal distribution variable.

Example: We measured the beaks of 30 birds. We obtained a sample mean of $\mu' = 38$ mm and a standard deviation of $\sigma' = 0.3$ mm. **Problem**: Assuming that the real average (μ) of the population is 40 mm, what is the probability of obtaining $\mu' = 38$ mm in a random sample, as happened in our experiment? Understanding the inaccuracy of an average estimate was the main axis for the description of this distribution by William Gosset. Under the pseudonym Student, the statistician, who worked for the Guinness brewery, published in Biometrika (1908) the famous article *The probable error of a mean*.

To understand the inaccuracy, we need a measure of the dispersion of these measures. We assume samples taken from a random variable with normal distribution with average μ and standard deviation σ. We can take j samples of size n and average these samples $\mu'_1, \mu'_2, ..., \mu'_j$. The sample averages μ' are estimates of the real average μ.

What is the dispersion of the estimates $\mu'_1, \mu'_2, ..., \mu'_j$?

For a set of estimates $\mu'_1, \mu'_2, ..., \mu'_j$, we call **standard error of the mean* the population standard deviation σ divided by the square root of the size of the sample family in question ($std.err. = \sigma/\sqrt{n}$). Since we do not know the standard deviation in the population, we approximate it using the sample standard deviation σ'.

Student proposed using a quantity to estimate the probability of a μ' estimate given a hypothetical μ center. This pivotal amount is the ratio between *(1)* distance from the estimates and the real average, μ' - μ, and *(2)* the standard error. The t statistic:

$$t = \frac{Z}{s} = (\mu' - \mu)/\frac{\sigma}{\sqrt{n}}$$

Thus, the t statistic for our example (μ'= 38; $\mu = 40$; n = 30; σ' = 0.3) is:

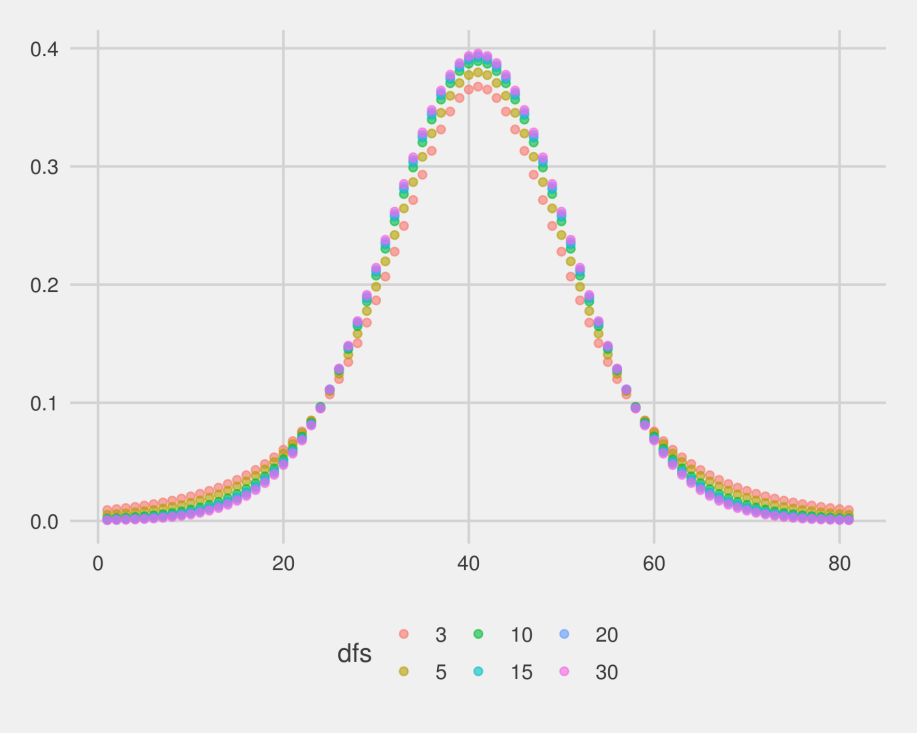
$$t = \frac{(38 - 40)}{\frac{0.3}{\sqrt{30}}}$$

Student (Gosset) showed that this statistic follows a probabilistic distribution (Student's t) defined by:

$$f(t) = \frac{1}{\sqrt{\nu}\,\mathrm{B}(\frac{1}{2}, \frac{\nu}{2})} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

B is the Beta function [ˆ9] and v are degrees of freedom. It has a density similar to that of the normal distribution, but with greater probabilities for extreme values. The parameter ν (degrees of freedom) expresses this characteristic. It is empirically estimated by the size of the samples used in the μ' estimate. We associated a sample (size n) taken from a normal population (arbitrarily high size,$n \to \infty$) to a t distribution with $n - 1$ degrees of freedom. In our example, $n = 30$, so $\nu = n - 1 = 29$.

[ˆ 9]: The Beta function accepts two arguments (x, y) and its result is the ratio is between (1) product of the functions $\Gamma(x)\Gamma(y)$ and (2) gamma function of the sum $\Gamma(x + y)$. The Γ function generalizes the concept of factorials (product of the predecessors).



Higher values correspond to larger samples and bring the t-distribution closer to a normal distribution. In an extreme case, we have $n_{samples} = n_{pop}$ and the samples are identical to the source distribution.

Knowing the t-statistic (-36.51) and the degrees of freedom for our sample family ($\nu = 29$), we can use the expression $f(t)$ to know the probability of obtaining

our average 38 mm in a sample (n = 30) if the population average is 40 mm.

For that, we add the probabilities of extreme values lower than the t-statistics provided.

$$\int_{-\infty}^{-36.51} f(t)dt$$

In julia, the native function *cdf* does the dirty work of calculating the integral:

```
using Distributions
cdf(TDist(29),-36.51)
    4.262182718504655e-2
```

This value reflects the probability of negative t values that are more extreme (smaller) than our ($t < -36.51$).

Two-tailed test It seems to be our p-value, but it needs an adjustment: we want to know the probability associated with obtaining such extreme values in general, not restricting ourselves to extremely smaller values.

Since the distribution is symmetrical, the tail on the left (negatives) is identical to the tail on the right (positives). Extreme values (negative or positive) in relation to the average are twice as likely as negative extreme values. We consider significant t-values much higher (right) or lower (left) than the average. So, our threshold must be robust to the possibility of extremes greater than the positive t symmetric statistic.

The value $t = 36.51$ it would be the resultant statistic of a sample with symmetric mean (42 mm) in relation to the mean (40 mm). Remember that the original measurement was 38 mm.

($t_{min} = -36.51; t_{max} = 36.51$). When making this adjustment, we call the two-tailed test.

Knowing the symmetry in the t-distribution, we can then use the following trick:

```
2*cdf(TDist(29),-36.51)
  8.52436543700931e-26
```

It is not possible to directly calculate the probabilities for t = 36.51, as Julia approximates the integral above ($p \sim 1 - 4.262^{-26}$) ~ 1.

```
cdf(TDist(29),36.51)
    1.0
```

Note *A common misconception about the t distribution is that it describes small samples taken from a population with a normal distribution. Any sample taken from a normal distribution variable will, by definition, have a normal*

distribution, even though it is composed of 1 or 2 observations. What follows t-distribution is the pivotal amount described above.

In section IX of the article, Student (Gosset) demonstrates how his insight can be used to test the effect of scopolamine isomers as a sleep inducer. [12] Two samples are used (levos and hyoscyamine hydrobromide dextro).

Additional hours' sleep gained by the use of hyoscyamine hydrobromide.

Patient	1 (Dextro-)	2 (Laevo-)	Difference (2–1)
1.	+ ·7	+1·9	+1·2
2.	−1·6	+ ·8	+2·4
3.	− ·2	+1·1	+1·3
4.	−1·2	+ ·1	+1·3
5.	−1	− ·1	0
6.	+3·4	+4·4	+1·0
7.	+3·7	+5·5	+1·8
8.	+ ·8	+1·6	+ ·8
9.	0	+4·6	+4·6
10.	+2·0	+3·4	+1·4
	Mean + ·75	Mean +2·33	Mean +1·58
	S. D. 1·70	S. D. 1·90	S. D. 1·17

Figure 9: Taken from The probable error of a mean, pag. 20. The data is available in the base library of R, under the name 'school'.

Using data from 10 patients who used both substances and measures of the additional amount of observed hours of sleep, "Student" calculates: (1) the probability of the data assuming an average of 0 in each group and (2) the probability of the data assuming that the difference in means is 0.

The first procedure is identical to the one we performed with the nozzle measurement and is called a single sample t-test. Hypothetizing a value for the mean (e.g. $\mu_{bico} = 40mm; \mu_{sonoadicional} = 0horas$), we calculate the probabilities of our estimate.

The second procedure is called the t-test of independent samples. We hypothesize a value for the difference in means between two populations ($\mu_a - \mu_b = 0$) and calculate the probability of our estimate. Practical example: is there a difference in weight between the beaks of birds A and B?

[12] https://atmos.washington.edu/~robwood/teaching/451/student_in_biometrika_vol6_no1.pdf

Applications

Returning to our Galapagos example, we will do a t test of independent samples.

1. The measurements in A and B are samples of random variables with normal distribution.
2. We define the null hypothesis and at least one alternative hypothesis.
   - H_0: Birds from islands A and B have beaks of equal size.
   - $\mu_a - \mu_b = 0$ B. H_1: Birds have different sized beaks.

The procedure is similar to the previous one. We calculate an intermediate quantity that follows t-distribution using the sample estimate of the difference and associated standard error. So, we can speculate: how likely is it that someone will get our observations considering distributions of equal averages ($\mu_a = \mu_b$)? This test infers the probability for the populations from which the samples came.

If p is less than an arbitrarily predefined threshold (conventionally, 0.05), we reject H_0 The probability of looking at the data is small if H_0 is true. We obtain the p-value by adding the probability values corresponding to the differences obtained or more extreme values. If the difference between values is large, the value of the statistic will grow. This implies a low probability of observing those results if the samples were similar (coming from the same distribution).

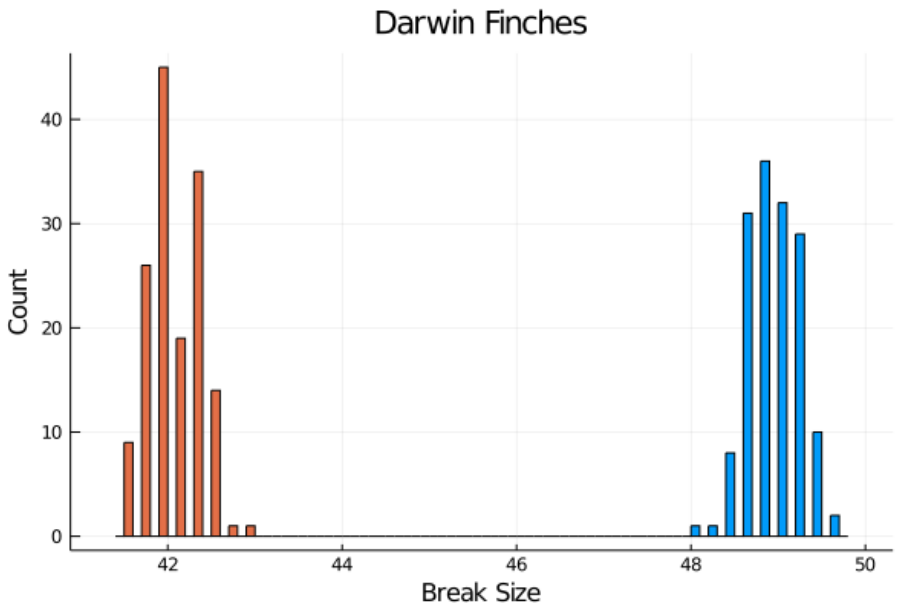
Student t test with Julia We will compute a t test for 2 independent samples. The t statistic is calculated with some changes. The degrees of freedom are added together and the standard error (dispersion of estimates) is balanced through the weighted average (by degrees of freedom, n-1) between samples.

$$t = \frac{X_1 - X_2}{\sigma_{pooled}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$\sigma_{pooled} = \sqrt{\frac{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}{(n_1 - 1) + (n_2 - 1)}}$$

Considering $(n_1 - 1) + (n_2 - 1)$ degrees of freedom, we calculate the t-statistic and the corresponding p-value for our degrees of freedom. Using the samples created earlier, corresponding to the gray (A) and blue (B) bars, we will plot the histograms.

```
@df stack(galapagos_birds[:,[:x1,:x2]]) groupedhist(:value, group = :variable,
    bar_position = :dodge,bins=50,title=("Darwin Finches"),
    xlabel="Beak Size",ylabel="Count",legend=false)
```




```
# Ajustes nos dados
a = galapagos_birds[:,:x2]
b = galapagos_birds[:,:x4]
sd_a = std(a)
sd_b = std(b)
```

Here, instead of comparing the estimates of the t-distribution means for samples A and B. We calculate the (1) Expected difference in the validity of the null hypothesis ($diff_{H_0} = 0$), (2) estimate of the difference ($diff = \mu_A - \mu_B$), degrees of freedom (df) and balanced standard error (se_{pooled}) for the distribution of differences in means.

```
expected_diff = 0
mean_diff = mean(a) - mean(b)
6.963886183171148
len_a , len_b = length(a) , length(b)
# balanced degrees of freedom
df_pool = len_a + len_b - 2
# balanced standard deviation
sd_pool  = sqrt(((len_a - 1) * sd_a^2 + (len_b - 1) * sd_b^2) /
  df_pool)
```

The t statistic corresponding to the observed difference, considering a t distribution with the parameters calculated above.

```
# Difference divided by standard error
# t-statistic
t = (mean_diff - expected_diff) /
      (sd_pool * sqrt(1/length(a) + 1/length(b)))
```

P-value for two-tailed hypothesis (extreme results considering the possibility that the difference is greater or less than 0):

```
p = 2*cdf(TDist(df_pool),-abs(t))
```

Finally, adding the summary of the results (averages A and B, difference verified, resulting t-statistic, p-value):

```
results = Dict(
                "Mean Difference" =>mean_diff,
                "t"=>t,"p value" => p,
                "Mean in A" => mean(a),"Mean in B" => mean(b))
Dict{String,Float64} with 5 entries:
  "Mean Difference" => 0.46412
  "t"               => 16.7569
  "p value"         => 7.28163e-45
  "Mean in A"       => 42.9969
  "Mean in B"       => 42.5328
```

We obtained a significant p-value ($p < 0.001$) using $n = 150$. The degrees of freedom are 149 (150 - 1) in each sample, with 298 in total. We can automate the process in 1 line:

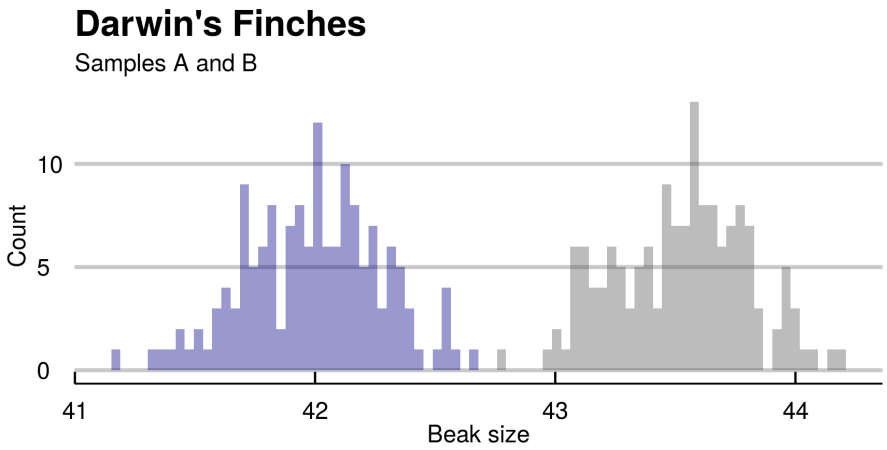
```
using HypothesisTests
UnequalVarianceTTest(a, b)
```

```
Two sample t-test (unequal variance)
------------------------------------
Population details:
    parameter of interest:   Mean difference
    value under h_0:         0
    point estimate:          0.4641197814036815
    95% confidence interval: (0.4096, 0.5186)

Test summary:
    outcome with 95% confidence: reject h_0
    two-sided p-value:           <1e-44

Details:
    number of observations:   [150,150]
    t-statistic:              16.756889937632724
    degrees of freedom:       297.5604548062057
    empirical standard error: 0.02769725069097449
```

The t-statistics and degrees of freedom presented by the implementation are identical to those found by performing the procedure step by step. Instead of the exact value ($p = 1.23^{-179}$), we received the information that $p < 1e^{-99}$. Given the p-value obtained, we would conclude that the distribution of data as observed is unlikely if the null hypothesis $ H_ {0} $ is true that the difference between samples is 0

Report example The estimated difference of beak mean sizes among samples A and B was significantly (p<0.05) different from zero (t = 47.28, df = 298)

	Sample A	Sample B	valor p
Mean (μ)	43,52	41,99	<0,001
Std. Dev. (σ)	0,28	0,28	

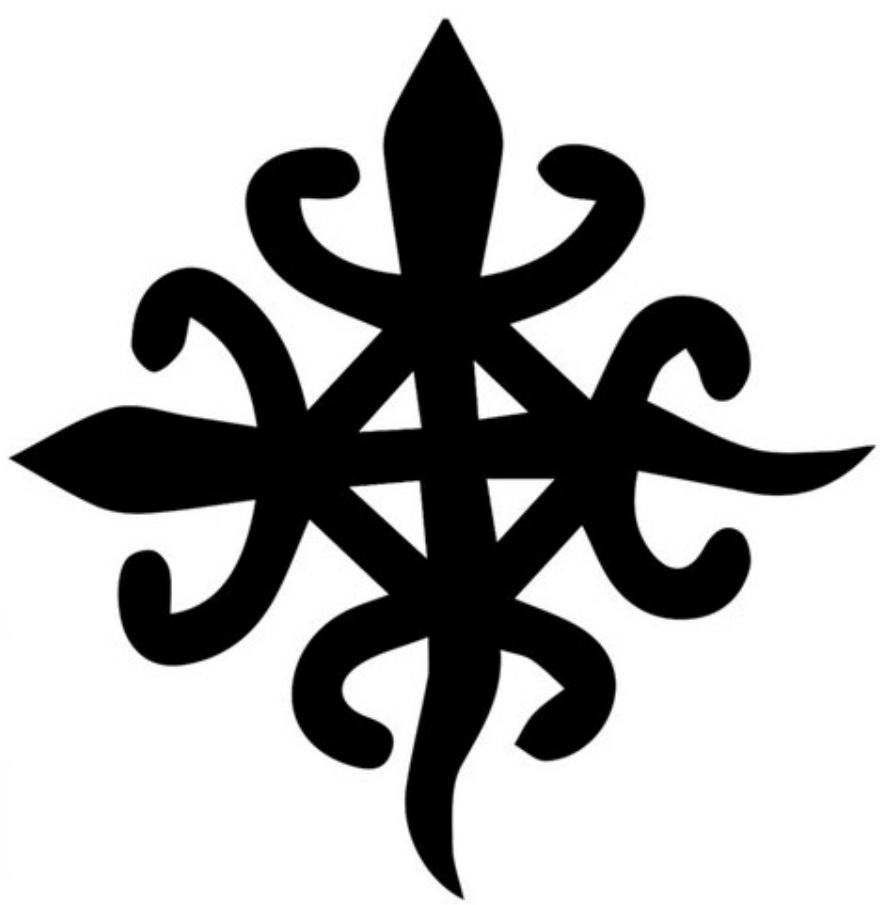

Darwin's Finches
Samples A and B
Count
10
5
0
41
42
43
44
Beak size

Note ###Exercises

1. Using the simulated dataset in the chapter:

- a. Perform T test for each pair of samples
- B. Which tests have p <0.05?
  - i. Describe t-statistics, degrees of freedom and p-value.
    - 1. How are the degrees of freedom of the different tests?
    - 2. Were these values expected for our samples?
  - ii. Using ggplot, plot histograms for all pairs compared in just one panel. Hint: grid.arrange
  - iii. Plot boxplots for one of the comparisons.
  - iv. From the previous graphic, add a layer with transparent violin plots (geom_violin) (alpha = 0).

2. Using the iris dataset
   - a. Choose two species and two measures.
   - B. Perform t-tests for both measures
   - ç. Report the results in a table, including mean and standard deviation of both measurements in both species.
3. The data used by Student for scopolamine are included in the base library of R.
   - a. Examine the data by invoking "sleep":> sleep
     - i. Plot histograms for measurements in both groups
     - ii. Run a t test assuming zero population mean ($\mu = 0$).
     - iii. Run a t test between samples, assuming the same average ($H_0 : \mu_1 = \mu_2$).
4. Generating the t distribution:
   - a. Simulate a set of many measures (suggestion: 100,000) from a normal distribution ($\mu = 0, \sigma = 1$).
   - B. Take 200 samples of n = 30 and save the 200 averages (sample function).
   - ç. Divide the values by the standard error, $\frac{\sigma}{\sqrt{n}}$.
   - d. Take 200 samples from a t-distribution with 29 degrees of freedom (rt function)
   - and. Plot the superimposed histogram of the obtained distribution and the theoretical distribution



Chapter 3: About associations

Prelude: *Hypotheses non fingo*?

I have not yet been able to discover the reason for these properties of gravity , and I make no assumptions. *Anything that is not deduced from the phenomenon can be called a hypothesis; and hypotheses, whether metaphysical or physical, or based on hidden qualities, or mechanical, have no place in experimental philosophy. In this philosophy, particular propositions are inferred from the phenomenon, and then generalized by induction.*

The rationale presented in the previous chapter is directly related to the hypothetical-deductive method and its philosophical principles. Although suitable for this scenario, the interpretation of the p-value is not very intuitive. It

involves *measuring how unlikely observations are in a hypothetical scenario under the null hypothesis*. His most popular (wrong) translation is that it represents *"the chance that the result of this study is wrong"*.

The framework described in the previous chapter is sufficient to produce a cryptic scientific work for laypeople.

When following pre-defined recipes (formulation of H_0 and H_1, calculation of statistics and p-values), a text seems to conform to academic standards, even if the elementary hypothesis around the research object is simplistic. Thus, inadvertently, we prioritize the form and relegate the core of scientific proposals to the background.

Another side effect is the search for p-values that reject H_0, disregarding theoretical precedents and probabilistic assumptions (multiple tests).

The difficult interpretability of the p-value and the frequent pitfalls involved in the inference process led the scientific community to question the hegemony of this parameter. There is a present tendency to abandon the p value and the limit $p < 0.05$ as canonical criteria.

We will learn about formal arguments against the hypothetical deductive method in science. For now, just know that it is always advantageous to obtain other information, complementary or alternative.

In this chapter, we will learn how to estimate (1) the magnitude of the difference between two samples and (2) how related are paired values (e.g. weight and height).

I have not as yet been able to discover the reason for these properties of gravity from phenomena, and I do not feign hypotheses. For whatever is not deduced from the phenomena must be called a hypothesis; and hypotheses, whether metaphysical or physical, or based on occult qualities, or mechanical, have no place in experimental philosophy. In this philosophy particular propositions are inferred from the phenomena, and afterwards rendered general by induction. *Isaac Newton (1726). Philosophiae Naturalis Principia Mathematica, General Scholium. Third edition, page 943 of I. Bernard Cohen and Anne Whitman's 1999 translation, University of California Press ISBN 0-520-08817-4, 974 pages.*

Effect size

The effect size helps us to express magnitudes. Returning to the previous example, what is the use of a significant difference between the size of the birds' beaks, if it is 0.00001 mm?

Still, there are cases in which small studies suggest important effects, but the sample size does not provide enough statistical power to reject the null hypothesis.

In addition to knowing how unlikely the difference is observed, it is natural to imagine how big it is.

A very popular measure is Cohen's *D (Cohen's D)*.

It is a parameter that expresses the magnitude of the difference without using units of measurement.
A soccer fan tells (happily) to a friend that her favorite team won with a score of 4×1 (goals).However, this friend accompanies basketball and is used to scores like 102×93 (baskets). How is it possible to compare goals with baskets? Which win represents the most disparate scores: 4×1 or 102×93?

The problem here is that scores behave differently between sports. Basketball scores have much higher averages and dispersions. Cohen's D consists of expressing this difference in standard deviations. Simple enough:

$$D_{cohen} = \frac{\mu_1 - \mu_2}{\sigma_{pooled}}$$

Using the * effects * library, we can directly calculate:

```
library(effects)
# O dataset galapagos_birds was created in chapter 1
>cohen.d(galapagos_birds$X1,galapagos_birds$X2)

Cohen's d

d estimate: -5.460017 (large)
95 percent confidence interval:
    lower     upper 1
-5.954047 -4.965987
```

Cohen proposed some tracks to classify the magnitude of these effects:

	Small	Medium	Big
Cohen's D	0-0.2	0.2-0.5	0.5 - 0.8

Thus, we can update our previous results, also reporting the effect size of the

difference and its confidence interval. If the distributions are from the same family, we have a comparable estimate between contexts.

Correlations

In the scientific endeavor, we don't just stick to comparisons. A more noble objective is to describe exactly how the relationship between studied entities occurs.

As we know, there are many classes of functions to express relationships between variables / sets. In the previous chapters, we used some functions, such as $y = \sqrt{x}$ and $y = e^x$.

Several natural laws have become particularly known, such as the relationship between force, mass and acceleration, elucidated by Newton:

$$\vec{F} = m\vec{a}$$

And the relationship between mass and energy for an object at rest, discovered by Einstein:

$$E = mc^2; c^2 \sim 8.988 * 10^{16} \frac{m^2}{s^2}$$

The above equations describe a linear relationship between quantities.

Linear relations

A linear relationship between two variables indicates that they are correlated in a constant proportion for any interval.

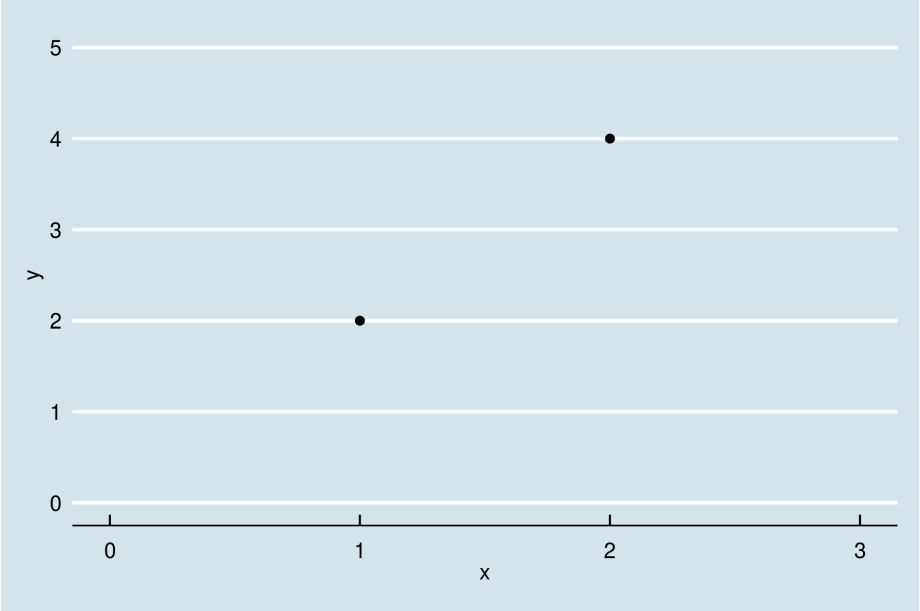
That is, higher mass values correspond to a proportional increase in energy. The value of c^2 expresses this constant proportion.

Example: a water molecule weighs approximately $m_{H_2O} = 2.992 \times 10^{-23} g$. Therefore, the associated energy is $E_{H_2O} = 2.992 \times 10^{-23} \times 8.988 \times 10^{16} \sim 2.689^{-6} J$. If we triple the number of water molecules, the same will happen with the associated energy: $E_{3H_2O} = 3 \times E_{H_2O}$.

If the correlation is positive, increments in x will be proportional to increments in y. If the correlation is negative, increments in x will be proportional to decreases in y.

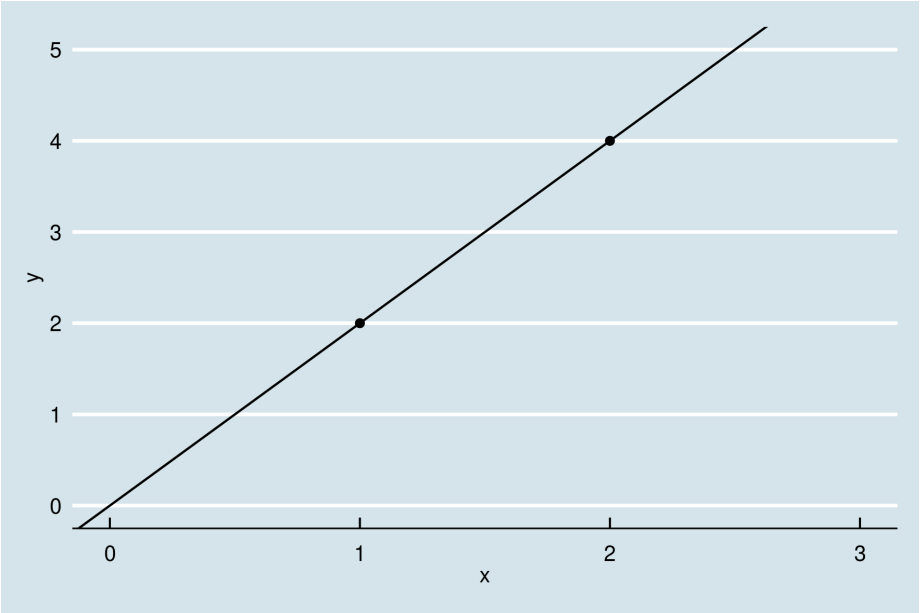
In a perfect scenario, if we know that there is a linear relationship between variables, we need only two observations to find out the proportion between them. This problem is identical to that of finding the slope of the line that passes through two points. It is easy to solve using elementary techniques.

```
>library(ggplot2)
>ggplot()+
geom_point(mapping=aes(x=1,y=2))+
geom_point(mapping=aes(x=2,y=4))+
xlim(0,3)+ylim(0,5)+
theme_economist()
```



$y = \beta * x$
$a = (1, 2); b = (2, 4) \rightarrow \beta = 2$

```
>ggplot()+
geom_point(mapping=aes(x=1,y=2))+
geom_point(mapping=aes(x=2,y=4))+
xlim(0,3)+ylim(0,5)+
geom_abline(slope = 2)+
theme_economist()
```



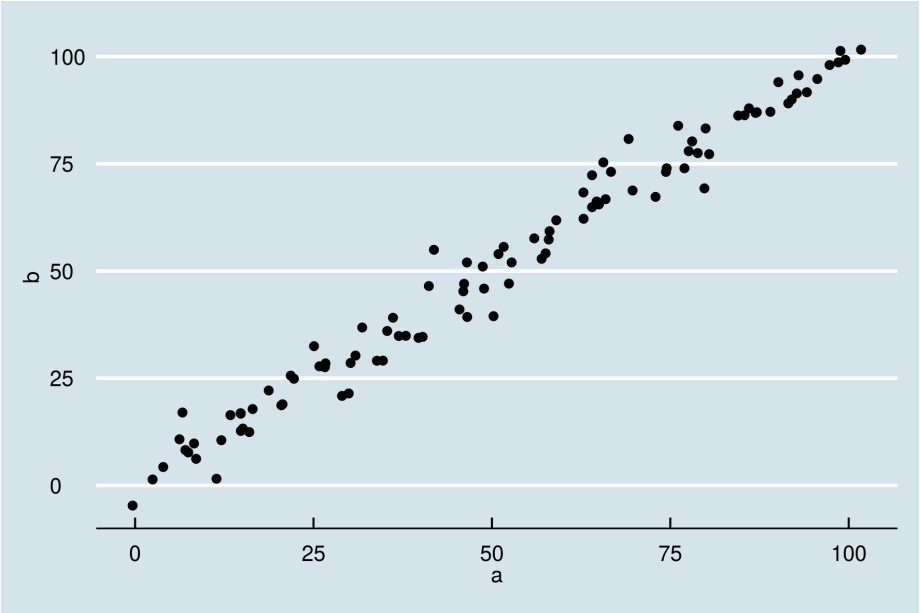
Errors and randomness

Controlling experimental factors, the relationships described are quite accurate. In a scenario without friction with surfaces and air, the measurement errors obtained with $\vec{F} = m\vec{a}$ are very low. However, this is not always true. First, we may experience interference from unknown variables.

Imagine a set of anthropometric measures, such as the height and weight of individuals. A human's height is expected to be related to his weight. However, other unmeasured characteristics, such as the percentage of total fat, may interfere with the final values. We normally treat these fluctuations as random errors [^ 11].

We can simulate this scenario starting from identical variables and adding random noise.

```
>set.seed(2600)
>a <- seq(1:100)+rnorm(n=100,sd=3)
>b <- seq(1:100)+rnorm(n=100,sd=3)

>cor_data <- data.frame(a,b)
>ggplot(cor_data,aes(x=a,y=b))+
geom_point()+theme_economist()
```



The result suggests that there is a strong linear relationship between x and y. On the other hand, we note that it is impossible for a line to cross all points. Next, we will investigate how to quantify the linear correlation, as well as find the line that minimizes the distance for all observations.

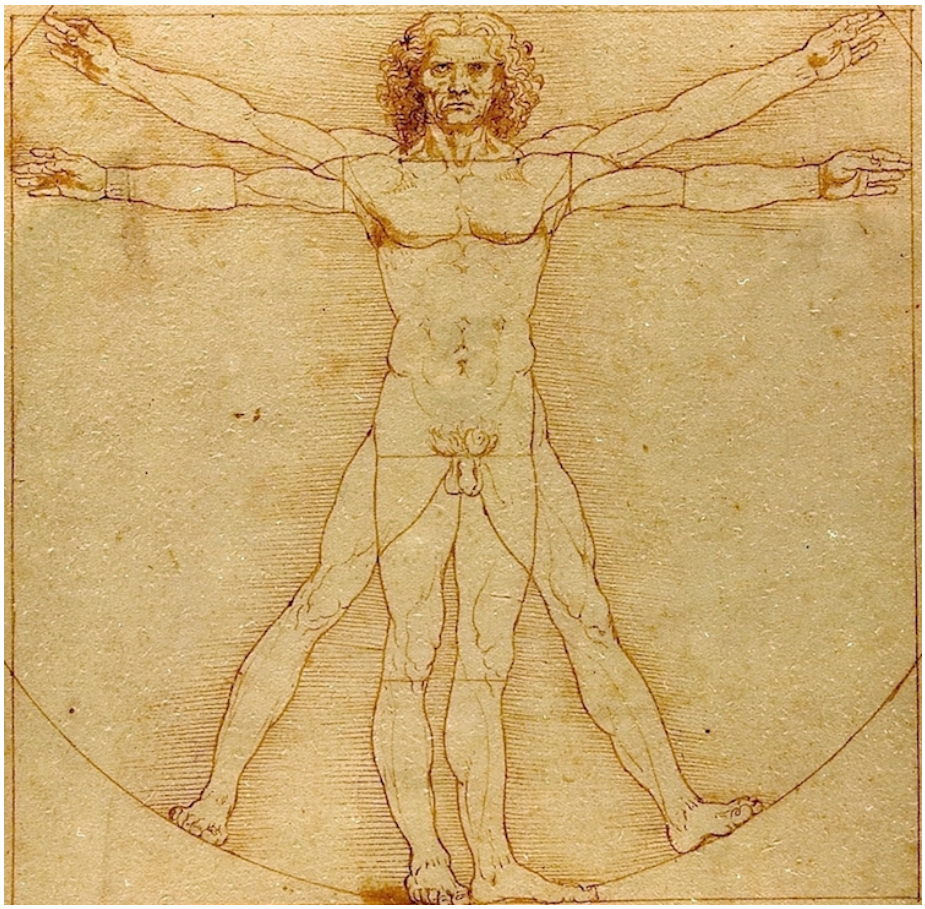
With these tools, we can extend our inferences. In addition to comparisons, we will have notions about the magnitude of a relationship, as well as we can predict the expected value for new observations.

[^ 11]: The nature of randomness is a philosophical question. Ultimately, we can imagine that it would be possible to explain random fluctuations through unknown variables (*hidden variables*). This is true of most natural phenomena. However, recent experimental findings in quantum physics (*Bell's inequality experiment*) suggest that hidden variables cannot explain the probabilistic nature of observations.

Pearson's product-moment correlation coefficient, or simply Pearson's (ρ).

Pearson's (rho) correlation coefficient is a real number guaranteed [^ 12] between -1 and 1. Expresses the magnitude and direction of a linear relationship, with -1 being a perfect inverse relationship and 1 being a direct relationship perfect.

For the data we generate, the correlation is almost perfect: $\rho = 0.989$. The coefficient has *product-moment* in its name, because it uses an abstraction originally used in physics, which we studied in the previous chapter: the moment (torque).



Calculating linear correlations

The notion of **distance**or* *deviation **was repeated many times. In fact, the linear correlation coefficient was born when Francis Galton (1888) numerically studied two apparently distinct problems in anthropometry** [13]**: 1.** Anthropology: **If we recovered from an ancient tomb only one bone of an individual's thigh (femur), what could we say about its height? 2.** Forensic science:** In order to identify criminals, what can be said about different measures by the same person?

Galton realized that he was actually dealing with the same problem. Given paired measures, (x_i, x_i'), what does the x_i deviation tell you about the x_i' deviation?

The femur recovered from a pharaoh's skeleton is 5 cm larger than the average.

[13] Francis Galton's account of the invention of correlation. Stephen M. Stigler. Statistical Science. 1989, Vol. 4, No. 2, 73-86.

How far from the average do we expect your height to be? Naively, we can think that if one of the measures is 1% higher than the average, the other will also be 1% higher. Galton realized that there was a trap in that thought.

Although there is a relationship between the measures, there are also random fluctuations: part of the deviation results from this. We need to understand the degree of correlation to make a good guess.

Then, he proposed a coefficient measuring the relationship between deviations of variables. If femur size and height are closely related, a large femur suggests an equally tall individual. Otherwise (low correlation), a large femur (high deviation) does not imply great stature.

To quantify the relationship, we multiply the deviations for each pair of measures:

$$Cov(X, X') = \sum_{i=1}^{N} (x_i - \mu_x)(x'_i - \mu_{x'})$$

The above formula expresses **covariance** between X and X' and will be useful in other contexts. The expression resembles the calculation of the first moment, but each deviation is multiplied by the corresponding deviation of the paired measure. Hence the name product-moment correlation coefficient.

Note that if both deviations agree in the direction (sign), the result of the multiplication will be positive. Consistently matching pairs increase the value of the final sum. If both deviations disagree in the direction (sign), the result will be negative. Consistently discordant pairs decrease the value of the final sum.

Thus, we can have highly correlated variables positively or negatively, as long as the sense of the association is constant. On the other hand, if the measures are at times inconsistent and at other times concordant, the values tend to cancel each other out in the sum and the result approaches zero.

Observing only the covariance is dangerous, as the values depend on the unit of measurement and data dispersion.

We calculated Pearson's correlation coefficient, normalizing [^ 17] the covariance by dividing it by the product of standard deviations:

$$\rho_{XX'} = \frac{cov(X, X')}{\sigma_X \sigma_{X'}}$$

Extensively:

$$\rho_{XX'} = \frac{\sum_{i=1}^{N} (x_i - \mu_x)(x'_i - \mu_{x'})}{\sqrt{\sum_i^N (x_i - \mu_x)^2} \sqrt{\sum_i^N (x'_i - \mu_{x'})^2}}$$

Uma boa notícia: ρ follows a known distribution, the t distribution, with n-2 degrees of freedom. We can use the previous tools to test hypotheses.

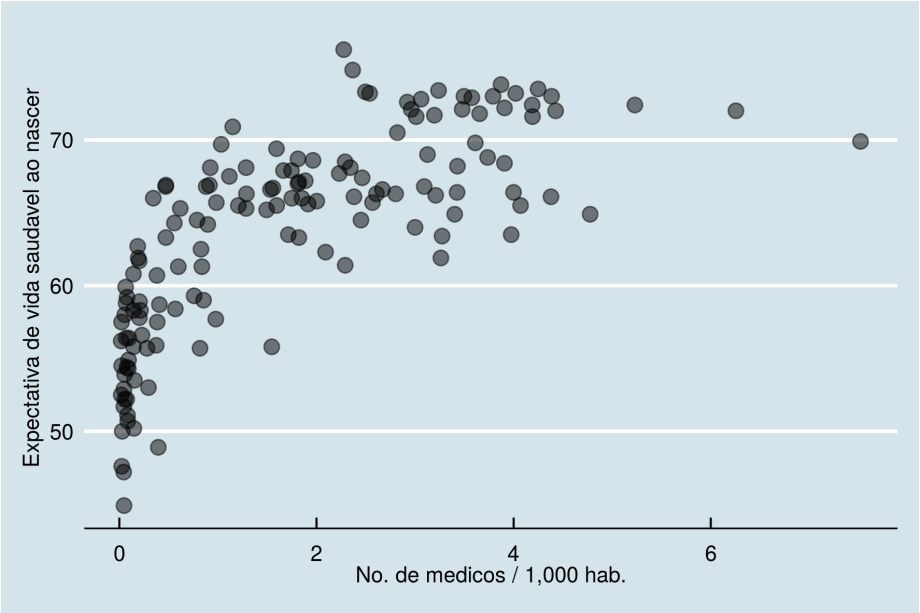
Practical example

The following example was a happy find. At the time, the Brazilian government was discussing the need to increase the number of doctors to improve health care. Some argued that it was the right decision, while others advocated that investments should be made in other areas of health.

Out of curiosity, I accessed the WHO (World Health Organization) and World Bank (World Bank) data on the number of doctors per country and health indicators. My expectation was to find at least a timid relationship between indicators. More than that, understand the location of Brazil in relation to other countries. I was surprised by a strong correlation, which we will explore next.

We adopted countries as an observational unit with measures x, the number of doctors 1,000 inhabitants, and y, the expected life expectancy at birth. Using data obtained from the WHO and World Bank portals, we plot the points on the Cartesian plane.

```
# http://apps.who.int/gho/data/view.main.HALEXv
# https://data.worldbank.org/indicator/SH.MED.PHYS.ZS
>library(magrittr)
>library(ggplot2)
>library(dplyr)

>worldbank_df <- read.csv("data/API_SH.MED.PHYS.ZS_DS2_en_csv_v2_10227587.csv",
                         header = T,skip = 3)
>colnames(worldbank_df)[1] <- "Country"

>worldbank_df$n_docs <- sapply(split(worldbank_df[,53:62], #lists of values
                                     seq(nrow(worldbank_df))),
    function(x) tail(x[!is.na(x)],1)) %>% #last non-null values
  as.numeric

>who_df <- read.csv("data/who_lifeexpect.csv",skip=2)
>who_df$hale <- who_df$X2016
>uni_df <- left_join(worldbank_df[,c("Country","n_docs")],
                    who_df[,c("Country","hale")],by="Country")

>ggplot(uni_df,aes(x=n_docs,y=hale))+
  geom_point(alpha=0.5,size=3) +
  xlab("No. of doctors / 1,000 inhab.")+
  ylab("Healthy life expectancy at birth")+
  theme_economist()
```



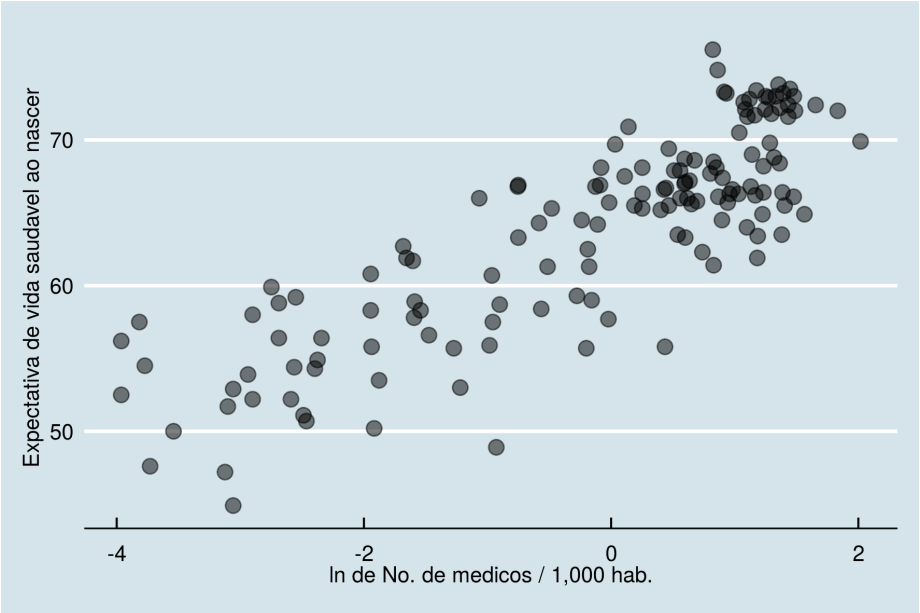
It is clear that the pattern is not random. Visually, we noticed that the value of life expectancy increases with a greater number of doctors. Still, we noticed an initially rapid increase until it reached a plateau. The pattern is similar to that of a logarithmic curve.

$y = log(x)$ or $HALE = log(N_{médicos})$

If this hypothesis is true, transforming the number of doctors using a logarithmic function will make the relationship linear with the transformed variable:
If $y = log(x)$, we do the replacement $x' = log(x)$ to get $y = x'$.

Then life expectancy becomes linearly correlated with the logarithm of the number of doctors.

```
>uni_df$log_docs <- log(uni_df$n_docs)
>ggplot(uni_df,aes(x=log_docs,y=hale))+
  geom_point(alpha=0.5,size=3) +
  xlab("ln No. of doctors / 1,000 inhab.")+
  ylab("Healthy life expectancy at birth")+
  theme_economist()
```



In fact, we see a notable linear trend for points.

Using the native implementation in R for Pearson's coefficient:

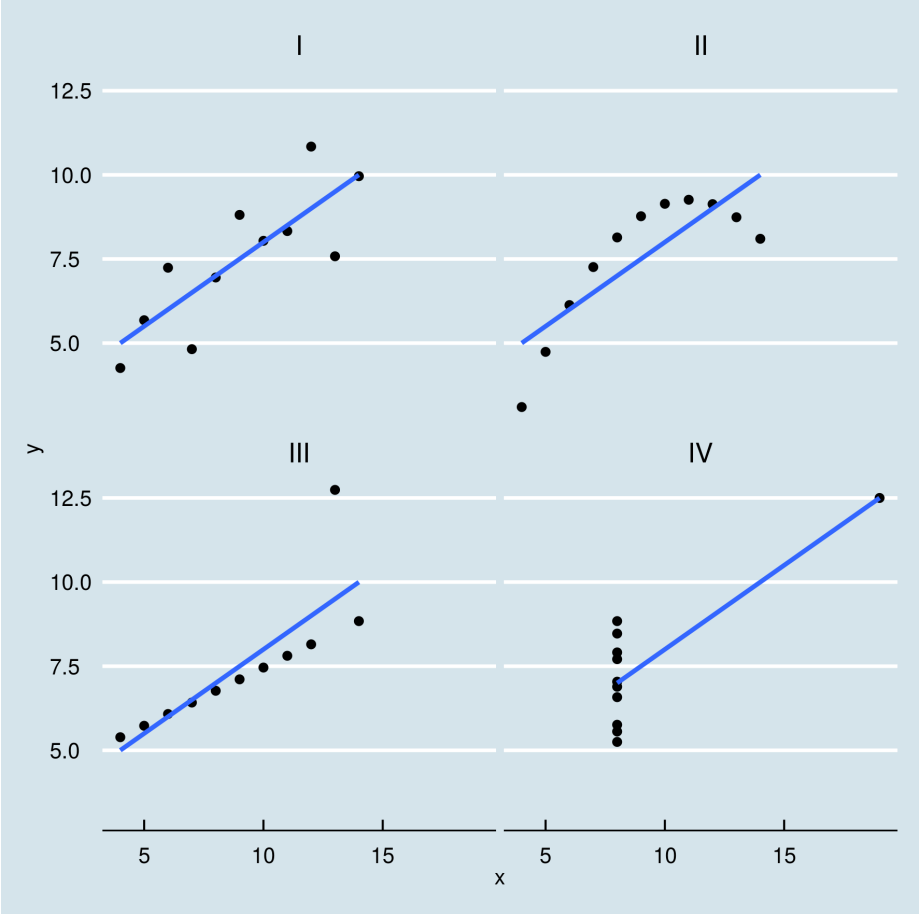
```
>cor.test(uni_df$log_docs,uni_df$hale)
Pearson's product-moment correlation
data:  uni_df$log_docs and uni_df$hale
t = 18.572, df = 143, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval: 0.7854248 0.8828027
sample estimates:
    cor
0.8407869
```

The linear correlation obtained for our sample of countries is surprisingly large, as suggested by the visualization ($\rho \sim 0.841$).

The p value is low ($p < 0.001$) considering the null hypothesis H_0 of $\rho = 0$. We conclude then that there is a significant linear relationship of strong magnitude between the logarithm of the number of doctors and the life expectancy of the countries in our sample.

It is really curious that there is such an evident mathematical relationship between tenuously connected constructs. The average time that an organism takes between birth and death and the number of professionals working. It is virtually impossible to spell out each causal relationship behind that relationship, which manifests itself robustly through the sum of many related factors.

Note *It is customary to state that there is no relationship between variables if the relationship coefficient does not prove to be important. As we have seen, this indicator reports only on linear relationships between variables. Data visualization can be of great help in inferring the nature of relationships. Data with very different distributions can result in equal coefficients, as shown by the classic Anscombe quartet. The 4 samples below show the same correlation coefficient.*

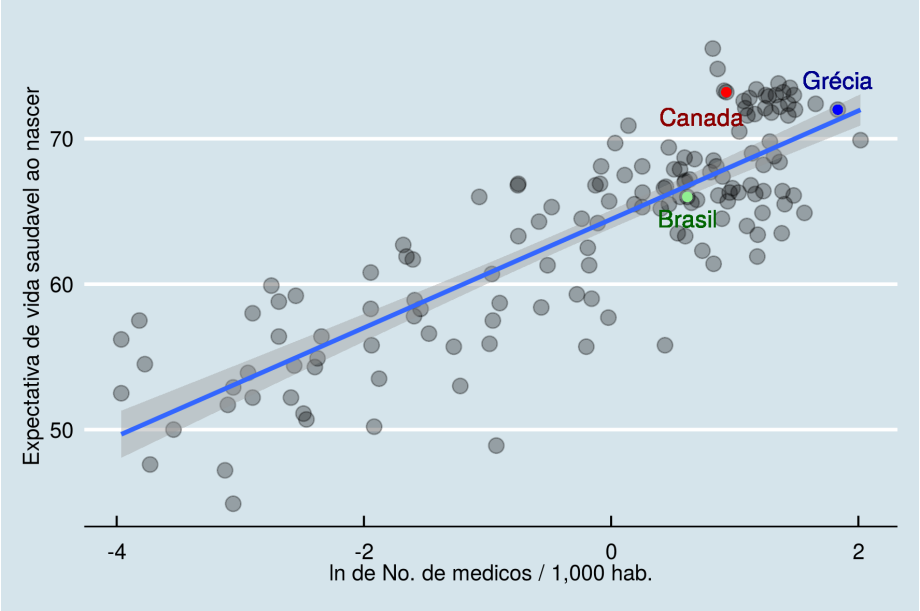


##Forecasts

We now know that it is reasonable to assume a linear relationship between these variables. As stated before, we can then find the line that minimizes the distance for observations.

The equation that describes this line tells us the expected value for life expectancy given the number of doctors.

```
>uni_df$log_docs <- log(uni_df$n_docs)
>ggplot(uni_df,aes(x=log_docs,y=hale))+
   geom_point(alpha=0.3,size=3) +  geom_smooth(method="lm")+
   geom_point(y=66.0,x=0.61626614,color="light green")+
   geom_text(y=64.5,x=0.61626614,label="Brazil",color="dark green")+
   geom_point(y=73.2,x=0.93177030,color="red")+
   geom_text(y=71.5,x=0.73177030,label="Canada",color="dark red")+
   geom_point(y=72.0,x=1.833381,color="blue")+
   geom_text(y=74.0,x=1.833381,label="Greece",color="dark blue")+
   xlab("ln No. of doctors / 1,000 inhab.")+
   ylab("Healthy life expectancy at birth")+
   theme_economist()
```



Biases must be addressed before conclusions are reached, but the model is sufficiently interpretable to make decisions. z A good policy can compare the investment value by sectors with other countries under similar conditions and different results. Assuming that there is really a linear relationship, we see that Brazil is quite close to what was expected for the number of doctors [ˆ 18]. If the strategy is to hire more people, we can look at programs in countries with more doctors per capita and positive results (e.g. Greece). If the strategy is to save on payroll and prioritize investment in structure, we can use countries with high life expectancy for the expected number of professionals (e.g. Canada).

[ˆ 18]: It is practically a consensus among specialists that Brazil has a problem with the distribution of professionals, with a shortage of doctors in poorer and less populated areas.

Predictions with linear models

How to guess one measure based on the other? Considering the linear relationship previously discovered, we can create a function that receives as input the value of a variable (number of doctors) and returns the expected value for life expectancy as an output.

Finding the equation that describes this function consists of finding the line that best fits the point cloud, as in the previous figure.

For this, we calculate the slope (β_1) and the vertical adjustment (β_0) that minimize the sum of the distances between the line and the observations. The term ϵ corresponds to errors, with normal distribution of mean 0 and standard deviation σ.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon$$

We adjust the model using the R lm (linear model) function:

```
# log_docs : x' = log(x)
>lm(hale ~ log_docs, data=uni_df)

Call:
lm(formula = hale ~ log_docs, data = uni_df)

Coefficients:
(Intercept)     log_docs
      64.46         3.73
```

We have $\beta_0 \sim 64.46$ and $\beta_1 \sim 3.73$.
Our estimate for healthy life expectancy "starts" at 64.46 years and increases with the number of doctors in the country. Specifically, it increases by 3.73 for each unit of our transformed variable ($log(x)$).
In our dataset, Brazil has 1,852 doctors / 1,000 inhabitants. Our prediction then is:
$\hat{y}_{Brasil} = log1.852 * 3.73 + 64.46 \sim 66.8$, which is very close to the real number (66).

Estimators

There is more than one way to estimate these parameters. One of particular interest, which will also serve in other contexts, is that of Maximum likelihood.

First, we determine a function that describes the probability of observation on the target variable (y_i) measurements of the predictor variables occur (x_i) and a set of parameters (β_k).

We can adopt as a likelihood function *(likelihood function)* for the values y_i a Gaussian probability distribution whose mean is given by the line $\mu_{yi} = \beta_0 + \beta_1 * x_i$.

Thus, the probability of each value y_i is given by a Gaussian, according to the deviation to the value predicted by the line.

$$L \sim N(\mu_{yi}, \sigma^2)$$

.

Assuming that the observations are independent, the probability of the set of observations is given by their product.

$$L = \prod_{i=1}^{n} P(y_i|x_i; \beta_0, \beta_1, \sigma^2)$$

Replacing the values of μ for the Gaussian by the line's predictions:

$$f(y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-\mu)^2}{2\sigma^2}}$$

$$L(\beta_0, \beta_1, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y_i-(\beta_0+\beta_1 x_i)^2}{2\sigma^2}}$$

This is our likelihood function and expresses the probability of observing the measures y_i given the measures x_i and considering a set of parameters (β_0, β_1).

The objective then is to find parameters that maximize this function. For convenience, we apply a logarithmic transformation to this function (*log likelihood function*). This transforms our product into a summation and we pass the counterdomain of the interval $[0; 1]$ for $[-\infty, 0)$.

$$\begin{aligned}
\text{log likelihood}(\beta_0, \beta_1, \sigma^2) &= log \prod_{i=1}^{n} P(y_i|x_i; \beta_0, \beta_1, \sigma^2) \\
&= \sum_{i=1}^{n} logP(y_i|x_i; \beta_0, \beta_1, \sigma^2) \\
&= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2
\end{aligned}$$

The parameters that maximize the likelihood function (max. Likelihood, ML) are the same as those that maximize the logarithm of the likelihood function (log-likelihood).

We introduce the rationale of the ML estimator as it will be useful in the future. In fact, it is easy to understand the closed formulas for our parameters, as they only express the linear relationships explored [14]:

$\hat{\beta}_1$ expresses the magnitude of the correlation between X and Y.It is natural that its value is the covariance normalized by the variance of the predictor.

$$\hat{\beta}_1 = \frac{cov(XY)}{\sigma_x^2}$$

$\hat{\beta}_0$ is our intercept, so it's the difference between predicted averages and predictions considering the average value in X.

$$\hat{\beta}_0 = \mu_y - \hat{\beta}_1 \mu_x$$

Finally, the variance of errors $\hat{\sigma^2}$ is given by the square of the deviations from the predictions in relation to the measures.

$$\hat{\sigma^2} = \frac{1}{n} \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

The solutions above provide the best estimates we can obtain by minimizing the distance from the line to the points. We must then be concerned with whether the linear model found is good in predicting the data.

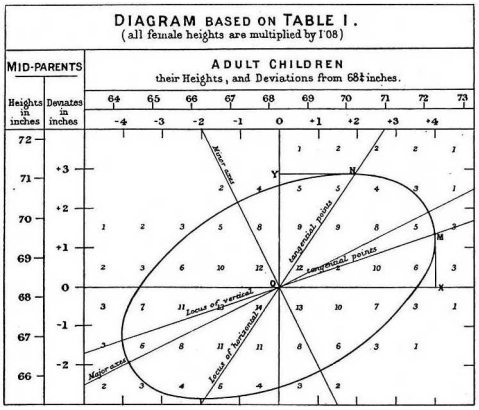


Figure 10: The first linear regression graph. Illustration by Francis Galton (1875) relationship between height of parents and children.

[14]Detalhes das deduções dos estimadores OLS and Max. Likelihood: https://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/05/lecture-05.pdf ; https://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/06/lecture-06.pdf

Evaluating performance There are different parameters to evaluate the performance of a model. In general, they seek to quantify how far the model results differ from ideal results.

For linear regression, the R^2 (coefficient of determination) is a widely used coefficient. Express the proportion between **(1)** variance explained by the model and **(2)** total variation. We call residual (or error) the difference between predicted values and real values.

(1) To capture the magnitude of model errors, we add the square of all residuals *(sum of squared residuals, SSR)* in relation to the predicted values. Be y_i the observations and $\hat{y}_i$ predictions:

$$SSR = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

(2) The total variability is quantified by adding the squared deviations from the mean *(total sum of squares, TSS)*, a term we saw in the variance calculation (second moment):

$$TSS = \sum_{i=1}^{n} (y_i - \mu_y)^2$$

So the fraction $\frac{SSR}{TSS}$ is the desired proportion. We define R^2 like:

$$R^2 = 1 - \frac{SSR}{TSS}$$

An intuitive view of SSR and TSS:

```
>source("aux/multiplot.R")
>doc_lmfit <- lm(hale ~ log_docs, data=uni_df)
>uni_df$preds[complete.cases(uni_df)] <- predict(doc_lmfit)
>uni_df$hale_mean <- mean(uni_df$hale,na.rm = T)
>ssr_res <- ggplot(uni_df,aes(x=log_docs,y=hale))+
    geom_point(alpha=0.5,size=3) +
    geom_segment(aes(xend = log_docs, yend = preds)) +
    geom_smooth(method="lm")+
    xlab("")+
    ylab("Healthy life expectancy at birth")+
    ggplot2::ggtitle("SSR") + theme_economist()

>tss_res <- ggplot(uni_df,aes(x=log_docs,y=hale))+
    geom_point(alpha=0.5,size=3) +
    geom_segment(aes(xend = log_docs, yend = hale_mean)) +
    geom_abline(slope = 0,intercept = 63.28165)+
```

```
        xlab("ln No. of doctors / 1,000 inhab.")+
        ylab("Healthy life expectancy at birth")+
        ggplot2::ggtitle("TSS")+theme_economist()

>multiplot(ssr_res,tss_res)
```

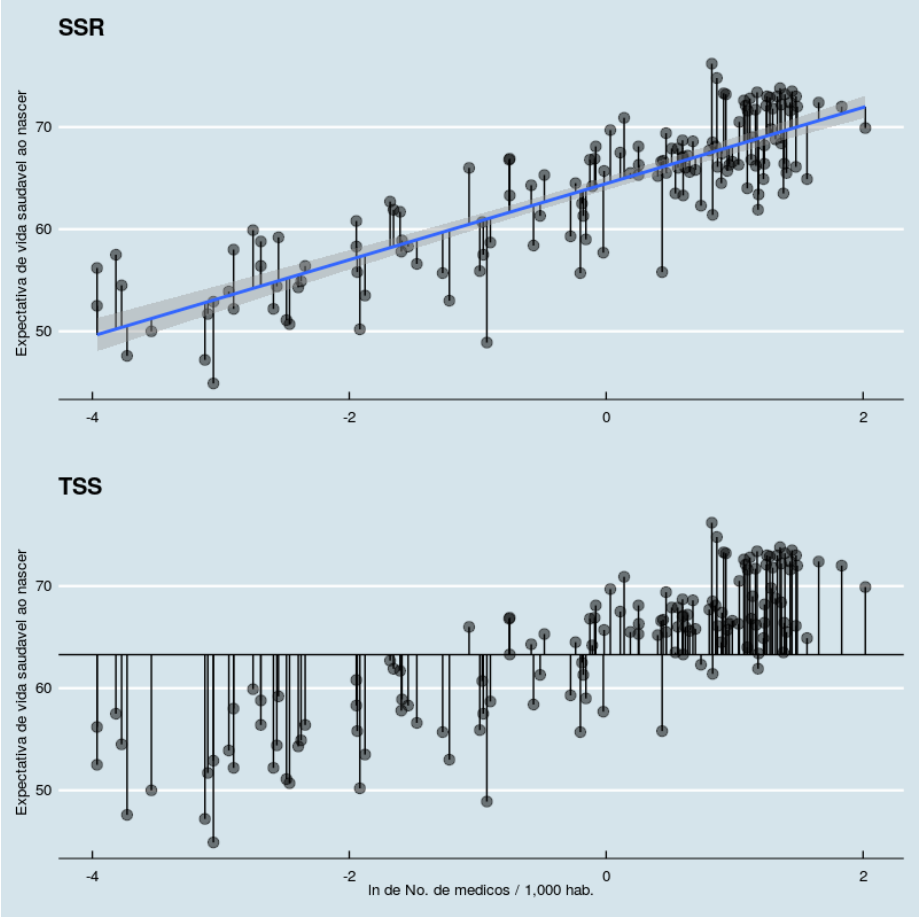


Figure 11: The square of the distance between a point and the line corresponds to a residue. We obtain SSR and TSS by adding all the residues in the upper and lower figures, respectively.

Values of R^2 close to 1 indicate residue sum (SSR) similar to 0. Using the line as a guide accumulates almost zero errors. Values of R^2 close to 0 indicate $\frac{SSR}{TSS} \sim 1$ and the predictions obtained by the model are as good as kicking the average for all cases.

```
>lm(hale ~ log_docs, data=uni_df) %>% summary
Call:
lm(formula = hale ~ log_docs, data = uni_df)

Residuals:
     Min       1Q   Median       3Q      Max
-12.0964  -2.3988   0.3233   2.8229   8.6708

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  64.4613     0.3162  203.84   <2e-16 ***
log_docs      3.7303     0.2009   18.57   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.779 on 143 degrees of freedom
  (119 observations deleted due to missingness)
Multiple R-squared:  0.7069,    Adjusted R-squared:  0.7049
F-statistic: 344.9 on 1 and 143 DF,  p-value: < 2.2e-16
```

To obtain the predicted values, we use the *predict* method:

```
>head(predict(doc_lmfit))

       2        3        4        7        8        9
59.90747 57.23226 65.39962 66.11533 69.54483 68.30608
```

It is also possible to obtain predictions for new values by specifying the *newdata* argument. For a country with 1.5 doctors / 1,000 inhabitants:

```
>predict(doc_lmfit,newdata = data.frame(log_docs=log(1.5)))
       1
65.97381
```

Assumptions There are some auxiliary procedures to check for possible flaws and points in the model that need attention. For example, residues can be asymmetrical. This indicates that performance changes at different intervals (heteroscedacity). Different violations require different attitudes, such as treating outliers or changing the model type. A complete list of premises, along with the R codes to test them, is available in the auxiliary material (*lm-asssumptions.R*)

Correlations and nonparametric tests

We thoroughly verified analyzes involving normal distribution, t distribution and linear relationships. However, measures often do not follow a defined distribution. Thus, making inferences using the ** parameters ** described $(\mu, \sigma, t...)$ nos levaria a direitos erradas. Para lidar com distribuições arbitrárias, vamos abrir mão deles e conhecer ferramentas *não-paramétricas* : the rank correlation coefficient ρ Spearman's test and Mann Whitney's U test.

Ranks and Spearman's ρ

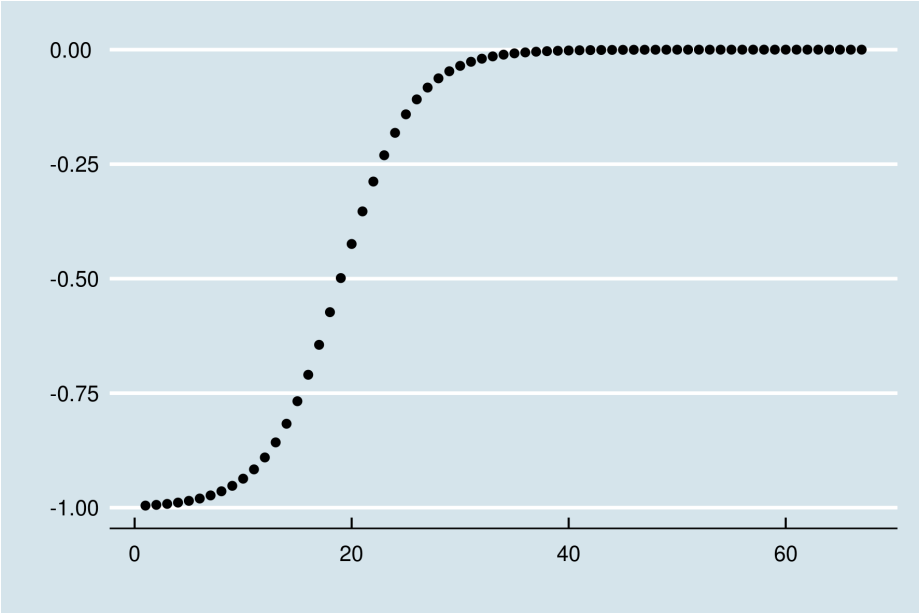
Linear relationships maintain constant proportions and we learn how to quantify them. On the other hand, two variables can have relations of other types, non-linear. In particular, if the measures have very extreme values * (outliers) * a calculation like the previous one suffers a lot with biases. A simple solution to this problem is to rank the values. Thus, the items in the set are treated by their position in relation to other items, regardless of the associated values. Example:

$$S = (1, 3, 89, 89, 39, 209) \rightarrow S_{ranked} = (1, 2, 4, 4, 3, 5)$$

Spearman's ρ is that Pearson's product-moment coefficient applied to ranks. Thus, we measure the degree to which two variables increase (or decrease) in magnitude by observing only the order of observations. That is: **greater than** , **equal** or **less than** . Specifically, we investigate whether there is a * monotonicity * relationship between them.

For the (sigmoid) relationship, between x and y below:

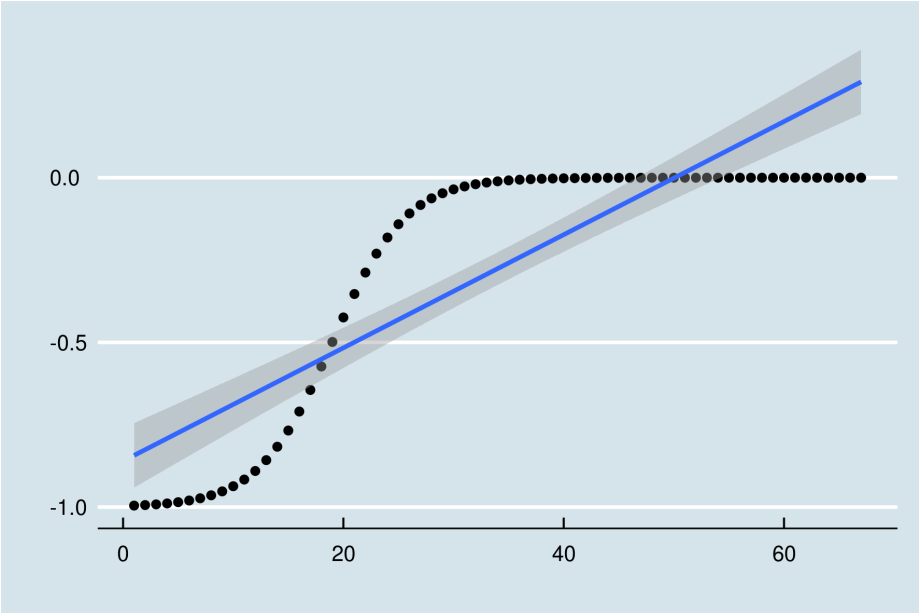
```
>set.seed(2600)
>sig_data <- data.frame(y_vals = -(1 / (1 + exp(seq(-10,10,by =0.3) )*100 ) ),
                        x_vals = 1:67)
>ggplot(sig_data,aes(x=x_vals,y=y_vals))+
geom_point()+theme_economist()+xlab("")+ylab("")
```



Pearson's coefficient is $\rho \sim 0.850$[15] :

```
>cor.test(sig_data$y_vals,
+          sig_data$x_vals)

    Pearson's product-moment
    correlation

data:  sig_data$y_vals and +sig_data$x_vals
t = 12.993, df = 65, p-value <
2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7658181 0.9051711
sample estimates:
      cor
0.8497162

>ggplot(sig_data,aes(x=x_vals,y=y_vals))+
  geom_point()+ geom_smooth(method="lm")+
  theme_economist()+xlab("")+ylab("")
```

[15] As noted in the graph, the linear correlation is not that high. The coefficient approaches 1 $\rho \sim 0.850$ because the upper deviations symmetrically compensate for the lower ones. The example reinforces the importance of plotting the data for better understanding (see Anscombe Quartet).



Since the relationship is perfectly monotonic, the ordered pairs (x_i, y_i) always have the same rank. The fifth highest value in x is also the fifth highest value in y. Therefore, Spearman's coefficient is 1:

```
>cor.test(sig_data$y_vals,
+          sig_data$x_vals,method = "spearman")

    Spearman's rank correlation rho

data:  sig_data$y_vals and sig_data$x_vals
S = 0, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
  1
```

The coefficient ρ Spearman's is preferable when the measurements appear to differ greatly in terms of the family of the distribution of origin. Especially, when the average does not seem to correspond well to the center of the distributions. Remember that Pearson's coefficient is based on deviations from the mean in both samples.

Mann-Whitney U test

The Mann-Whitney U test uses U statistics to make inferences. The rationale is identical to the Student's t test. We establish null hypothesis $H_0$and alternative hypothesis H_1.
Then, we calculate the probability that our observations will happen if the null hypothesis is true. This time, we will use the U statistic. Remember that the t statistic was calculated based on parameters extracted from the sample:

$$t = Z/s = (\mu' - \mu)/\frac{\sigma}{\sqrt{n}}$$

The U statistic does not depend on parameters (e.g. μ, σ), being calculated based on each observation.

First, we calculate the ranks for each measure r_i joining the observations of samples A and B, of sample sizes n_a and n_b in just one set ($N_{tot} = n_a + n_b$).

Then, we separate the samples again and calculate the sum of the ranks in each group, called R_a and R_b.
The U statistic is given by the following expression:

$$U_a = R_a - \frac{n_a(n_a+1)}{2}$$
$$U_b = R_b - \frac{n_b(n_b+1)}{2}$$

We use the smallest value of U to query the corresponding probability (p-value) for the null hypothesis.

The term $\frac{n(n+1)}{2}$ corresponds to the minimum sum of ranks for the sample. Ranks are a regular sequence $(1, 2, 3, ...)$, so that the sum of all values is identical to the sum of an arithmetic progression of N terms.

$$\Sigma_{ranks} = \frac{N(N+1)}{2}$$

While R_i corresponds to the sum of the ranks calculated with the two samples, the term above would correspond to the minimum sum of the ranks for a sample, if the ranks occupied the initial sequence $A = (1, 2, 3, 4, ..., n_a)$in the joint sample.

The definition for the test is not unanimous in the literature, so that some authors and software (e.g. R) implement the calculation with the above subtraction and others (e.g. S-PLUS) do not. In R, the functions **dwilcox (x, m, n)** and **pwilcox (q, m, n)** return the cumulative distribution and density for the U statistic corresponding to samples with sizes m and n. **wilcox.test (x, y, . . .)** is the basic implementation of the Mann Whitney test. The Mann Whitney test is the Wilcoxon test for two samples.

Exercises

1. Pearson's product-moment coefficient describes which types of relationship?
   - Is it useful for modeling quadratic relationships between variables?
   - We cite non-linear relationships, such as $E = mc^2$.Cite another example of a natural phenomenon with a non-linear profile where the ρ Pearson's does not work.
2. Create a function that calculates the nth moment for a sample:

- `n_moment <- function(x,n) {sum((x- mean(x))^n)/length(x)}`
- Calculate the skewness value. As mentioned in the chapter, it is the 3rd moment normalized [by the 2nd moment to the exponent 3/2].

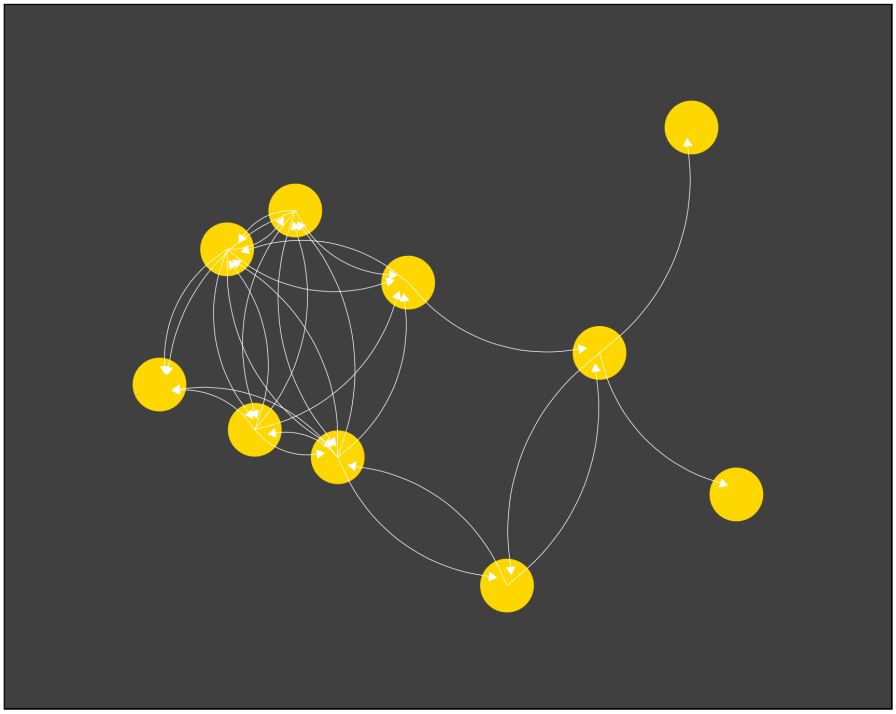
$$\frac{\mu_3}{\mu_2^{3/2}}$$

- Calculate the value of kurtosis. As mentioned, it is the 4th standardized moment [by the square of the 2nd moment minus 3].

$$\frac{\mu_4}{\mu_2^2 - 3}$$

- Values can be checked with implementations `e1071::skewness` and `e1071::kurtosis`

3. Using the * iris * dataset, compare the 4 numerical variables (*Sepal / Petal Lenght / Width*) between species (*Species*) using Student's t test and U Mann Whitney test. In any case, do the methods differ regarding the rejection of the null hypothesis?
   - Get the effect size (Cohen's D) for the differences.
4. Using the * iris * dataset:
   - Make a scatterplot between two measurements. The pairs function can help.
     - Check for significant linear correlation between variables.
     - If present, adjust a linear regression model.
     - Adjust a regression model for each species.
     - Note the values of R^2 for each model. What is your impression of the performance changes?



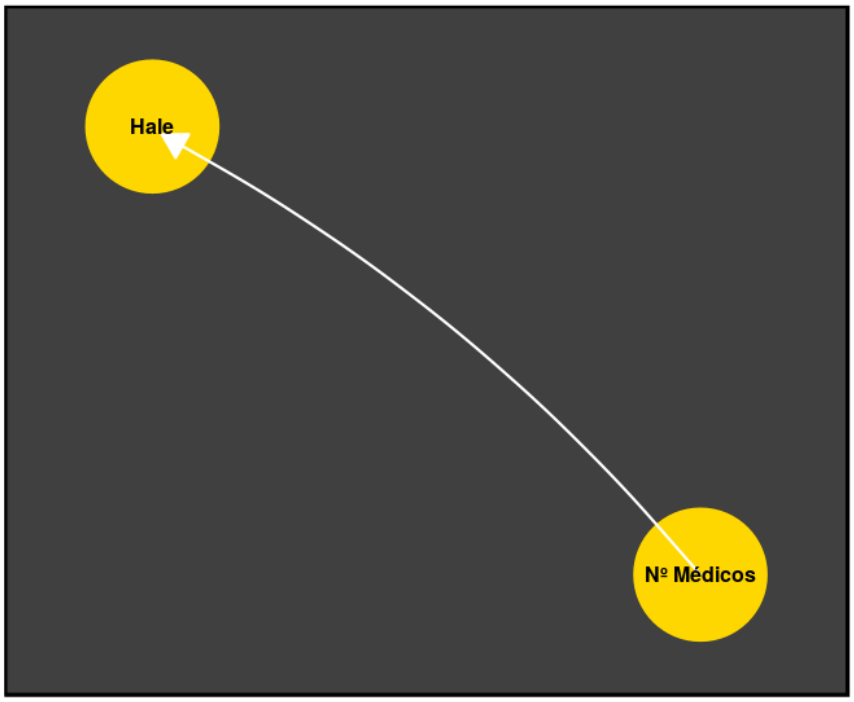
Chapter 4: Multivariate analysis, graphs and causal inference

Introduction

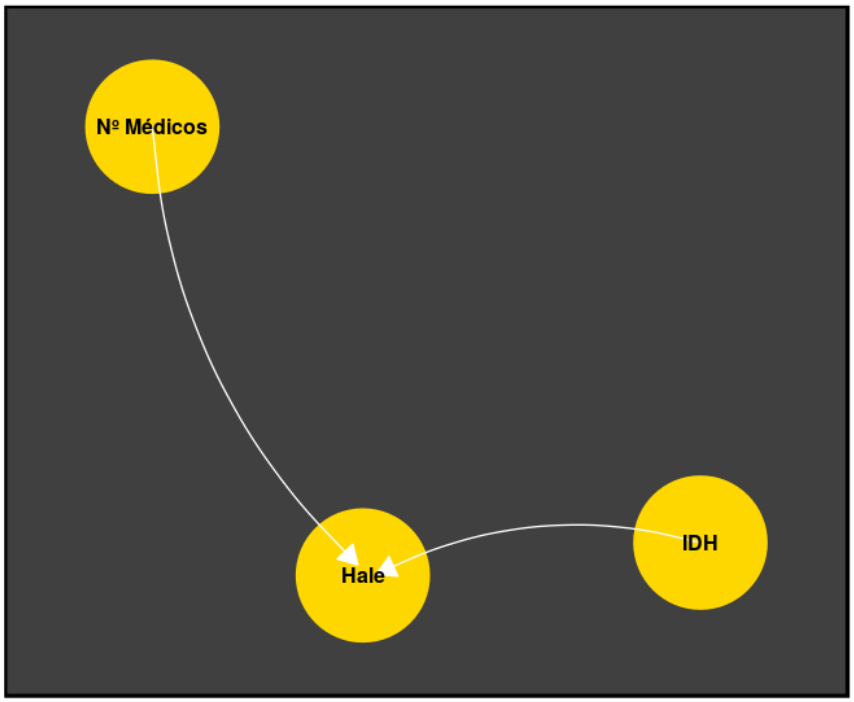
In this chapter, we will incorporate constructs as a basis for studying a concept from the cradle of Western philosophy: *causality*. Aristotelian philosophy investigates material, formal, efficient and final causes. Causes express the idea of isolating relationships between factors. Most definitions involve *effects* that depend, even partially, on previous *causes*. Causal relationships *explain* the evolution of systems under certain conditions. Up to this point, we have applied mathematical modeling for one or two random variables. Different procedures were used for correlation, comparison and regression. In this chapter, we will deal with multivariate analysis. Causal diagrams and bias control, mediation, moderation, multiple regression, principal component analysis and factor analysis.

Graphs and causal trajectories

We can use the following diagrams to illustrate a simple linear regression:



Or multiple with two predictors:



It is easy to relate *nodes to variables* and *connections to relations* described by the estimated equations. Formally, we treat these abstractions as **graphs**. The field started to be treated by Euler in 1736. We call the points nodes, or vertices, and the connections of edges (*edges*). Each edge connects two nodes. The concept was used to solve the problem of Königsberg bridges. Given a series of bridges connecting different parts of the city, how to make a route that crosses each one only once?

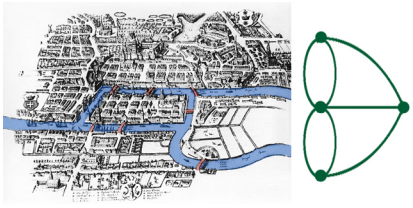


Figure 12: .

Euler showed that it was impossible. Note that we don't use distances. We only describe how elements are connected. We can tie several structures together. The graphs above, for example, are directed and have linked equations.

The equations and procedures we used previously are solutions equivalent to graphical representations. It is possible to generalize the idea, using diagrams to treat mathematically formulations of scientific theories.

Graphs and causal trajectories

"The ideal method of science is the study of the direct influence of one condition on another in experiments in which all other possible causes of variation are eliminated.", Sewall Wright, Correlation and Causation, 1921

The little-known origin of this field lies in the work of a geneticist, Sewall Wright. He assumed that the correlation between variables is the result of the influence of many causal trajectories. Then, he proposed a way to measure the influence of each trajectory on a target variable.

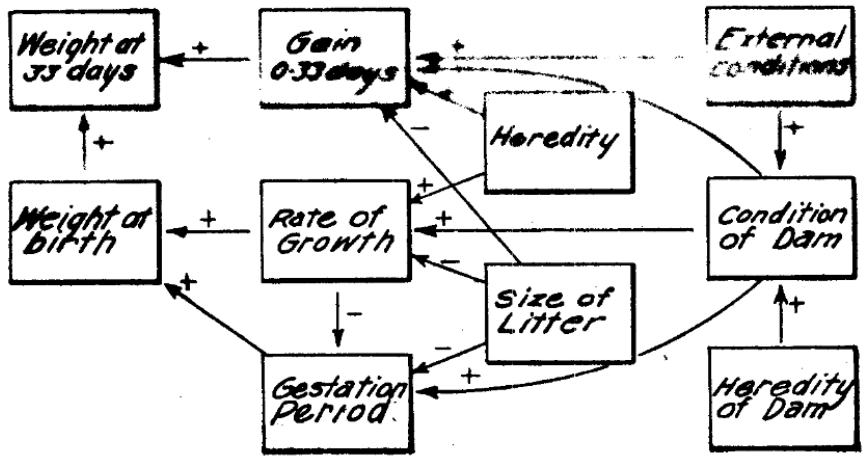


FIG. 1.—Diagram illustrating the interrelations among the factors which determine the weight of guinea pigs at birth and at weaning (33 days).

Figure 13: Diagram showing the relationship between factors influencing the weight of a guinea pig. Wright, 1921

Using directed graphs (connections have a source and a destination), it is possible to link the notions of correlation and regression in order to illustrate causal paths between linear relationships. Sewall started using only acyclic graphs (without trajectories returning to the same point of origin) directed, DAGs, under restricted conditions.

Decades later, the field was extrapolated to other, more general scenarios. In particular, the boom in the availability of computing power in the 1960s and 1970s drove the emergence of different estimators for parameters in these models. It is expected that the number of parameters will increase according to the complexity.

Valuable work was done by Judea Pearl to unify the approaches. Pearl showed that many *frameworks* are special situations of structural equation models. He wrote comprehensive texts aligning applied mathematics to an epistemological basis. Especially noteworthy is the concept of *counterfactual*. To estimate a causal effect, we imagine what the conditions would be in a scenario with no action by the causal agent. Pearl conducts a careful logical-semantic study of definitions in an attempt to build a coherent system of empirical research.

Examining covariates with causal models Causal models based on graphs assume unidirectional effects. This precludes the accurate description of many cases. On the other hand, parsimonious use is a valuable tool for making causal inferences. The following DAG analyzes the quality of a beer. It depends on water, hops (hops) and malt. We want to understand how the composition of solid ingredients (hops & malt) interferes with the final purity, evaluated by the absence of pesticides. We have data from some local factories. The water concentration in each city also varies, which directly interferes with the final purity of the beer. In addition, water is used to water the soil with hops and malt, also indirectly interfering in the outcome.

Note William Gosset developed the t test while working on beer production.

```
library(dagitty)
library(ggdag)

# Example in basic syntax
dag_o <- dagitty("dag{A <- B}")
# dagify used for plots
dagified <- dagify(Quality ~ Water + HopsMalt,
                   HopsMalt ~ Soil,Soil ~ Water,
                   exposure = "HopsMalt",
                   outcome = "Quality")
p1 <- ggdag(dagified) + theme_dag_blank()
p1
```

Tracing a DAG allows you to examine the possible paths through which information flows and, thus, make inferences. In a practical way, we want to:

1 . Test whether the proposed causal model is compatible with the observations.
2 . Estimate the effect by conditioning it to the appropriate covariates.

The graph implies some *conditional independence*. This means that, if it is correct, some variables will be independent.

```
impliedConditionalIndependencies(dagified)
#HopsMalt _||_ Water | Soil
#Quality _||_ Soil | HopsMalt, Water
```

The notation `A _||_ B | C, D, E, F...` indicates that A must be independent of B, if we condition the effect estimate to the covariables C, D, E, F ... "Conditioning a" means to include the covariate in the descriptive model. The simplest form is through multiple regression.

Multiple regression

In simple linear models, we calculate parameters for an intercept β_0, straight slope β_1 and variance of errors σ^2_ϵ. For example, we examine the purity of beer as a function of the purity of solid ingredients.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon$$

$$\text{beer}_i = \beta_0 + \beta_1 \text{HopsMalt}_i + \epsilon$$

In *multiple linear regression*, we introduce one more predictor variable, such as water and soil purity:

$$\text{beer}_i = \beta_0 + \beta_1 \text{HopsMalt}_i + \beta_2 \text{Water}_i + \epsilon$$

In general, we have two goals: **(1)** improve the model's performance by adding pertinent information; **(2)** examine the relationships considering multiple variables.

The first objective is intuitively obvious, however we need to be careful with information redundancy. Specifically, there is an almost inevitable trade-off between complexity and robustness of the model. Adding variables or using more flexible relationship classes means giving freedom to overfitting the data (**overfitting**). That is, our model will learn idiosyncrasies about the available data (WHO and World Bank datasets) and not about the relationship between abstractions (e.g. healthy life expectancy). We will see in another chapter how to mitigate this problem.

Another objective for multiple regression is to examine the modifying effect of the added variables. In particular, it is common to include auxiliary variables to correct estimates.

Example: we want to estimate a parameter β_1 for the relationship between height and weight. We adjust a model: $Altura = \beta_0 + \beta_1 * Peso + \epsilon$. However, we know that the height and average weight of men is greater than that of women. When examining the relationship between height and weight, we can include the variable *gender* in the model, $Altura = \beta_0 + \beta_1 * Peso + \beta_2 * Sexo + \epsilon$.
Our estimate of β_1 is modified to take into account the effects of sex.[^21]

However, including all covariates in a single model would result in inadequate effect estimates. The coefficients can even show relations with the opposite direction in the presence of wrong conditionals: negative, when the real effect is positive. Strict use of confounding control in multivariate analysis avoids misinterpretation of coefficients. This is done by selecting the set of adjustments respecting the adopted diagram.

We will see a formalization of this concept below, with a general platform to examine many specific variables and procedures for calculating effects and cases such as mediation and moderation.

[^ 21]: Sex is a dichotomous variable (male / female). We usually encode them in binary form (0/1; e.g.: male = 1 / female = 0). Thus, a male subject will have the height estimate increased in $\beta_2 * 1$, while females will have this term zeroed $\beta_2 * 0$. We call this trick *dummy coding*.

Conditional independence Returning to the example, the following conditional independence is implied by the DAG:

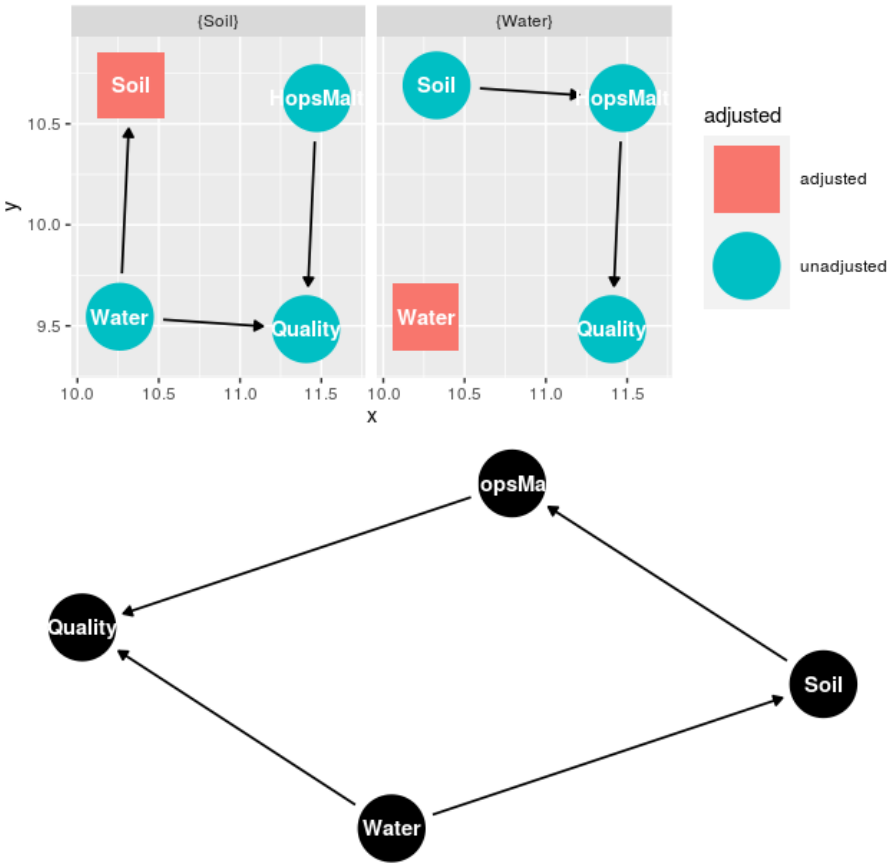
```
impliedConditionalIndependencies(dagified)
#HopsMalt _||_ Water | Soil
#Quality _||_ Soil | HopsMalt, Water
```

We can simulate the data and test the independence:

```
beer_data <- simulateSEM(dagified,b.lower = 0.20,b.upper=0.25)
# HopsMalt _||_ Water | Soil
lm(HopsMalt ~ Water + Soil,beer_data)
# Quality _||_ Soil | HopsMalt, Water
lm(Quality ~ Soil + HopsMalt + Water,beer_data)
```

It is expected that the estimate of the effect (coefficient) will be close to zero (no association) since we condition it to the indicated covariates. We see that this happens for the example:

```
#(...)
#Coefficients:
#(Intercept)        Water         Soil
#    0.03709      0.02105      0.28069

#(...)
#Coefficients:
#(Intercept)         Soil     HopsMalt
#    0.05652      0.08347      0.15118
```



Adjusted estimate (effect) of effect Once we accept the DAG as appropriate, we can use it as a reference for calculating unbiased estimates. This means that we are adjusting the final value according to the ways in which the information can flow in the examined covariates.

The `adjustmentSets` function returns which sets of covariates we can include to obtain unbiased estimates. The function `ggdag_adjustment_set` visually tells us which paths we are closing when conditioning a group of covariates. Sometimes (as in the example), we have alternative sets:

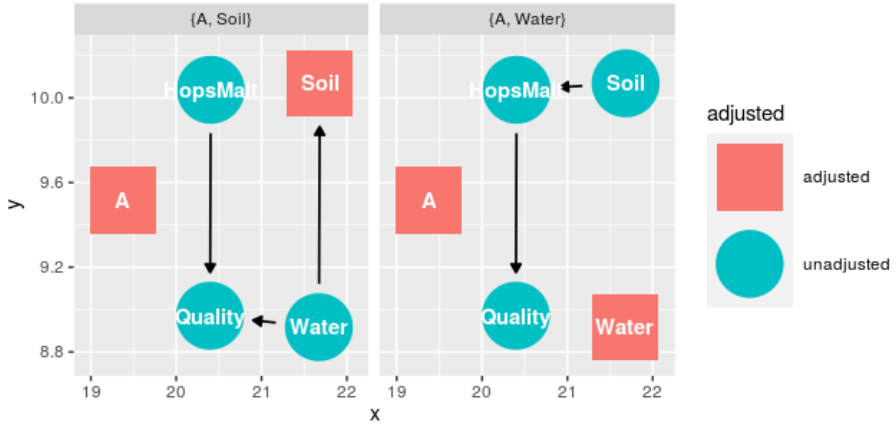
```
p2 <- ggdag_adjustment_set(dagified,exposure="HopsMalt",outcome="Quality")
multiplot(p1,p2)
adjustmentSets(dagified)
# { Water }
# { Soil }
```

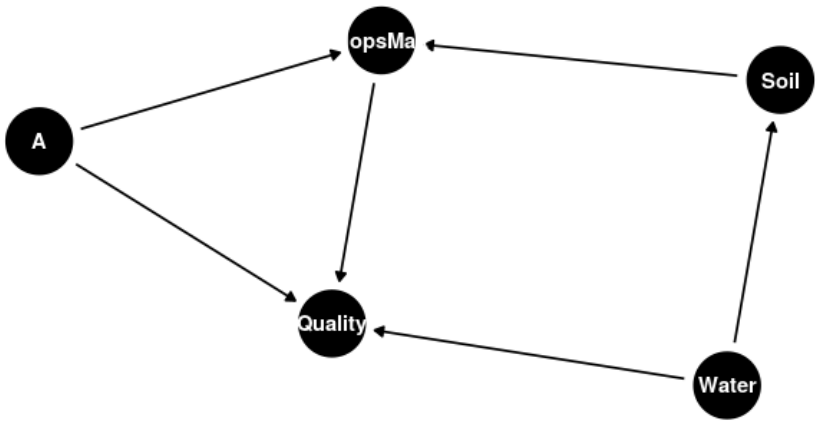
One requires conditioning to the water and the other requires conditioning to

the soil. The plotted graphs indicate the flow of information in each set of adjustments. We can report the values for one of the paths or for both paths.

We can imagine a new factor, which interferes with the final purity and also that of the solid ingredients.

```
dagified2 <- dagify(Quality ~ Water + HopsMalt + A,
                    HopsMalt ~ Soil + A , Soil ~ Water,
                    exposure = "HopsMalt",
                    outcome = "Quality")

p3 <- ggdag(dagified2)  + theme_dag_blank()
p4 <- ggdag_adjustment_set(dagified2,exposure="HopsMalt",outcome="Quality")
multiplot(p4,p3)
```





Now, we should test the following conditions:

```
beer_data2 <- simulateSEM(dagified2,b.lower = 0.20,b.upper=0.25)
#A _||_ Soil
```

```r
lm(A ~ Soil,beer_data2)
#A _||_ Water
lm(A ~ Water,beer_data2)
#HopsMalt _||_ Water | Soil
lm(HopsMalt ~ Water + Soil,beer_data2)
#Quality _||_ Soil | A, HopsMalt, Water
lm(Quality ~ Soil + A,beer_data2)
```

The adjustment possibilities for unbiased estimation are:

```r
adjustmentSets(dagified2)
# { A, Water }
lm(Quality ~ HopsMalt + Water + A,beer_data2)
# { A, Soil }
lm(Quality ~ HopsMalt + Soil + A,beer_data2)
```

We usually translate the procedures above stating that the estimate for *"the relationship between X and Y is controlled for confounders [A, B and C]"*. At this point, it is obvious that linguistic simplification is dangerous. **It is recommended that confounders be mitigated experimentally (e.g. randomization).** The lack of caution in translating mathematical abstractions into natural language is responsible for the unfair reputation of statistics as a tool for mistakes. Just as the p-value is misinterpreted many times, "control for confounders" "is nothing more than the adjustment of estimates considering a causal model. Although all premises and procedures are strictly adhered to, the use of targeted acyclic graphs (DAGs) is limited to describe certain phenomena and is vulnerable. Dawid AP. Beware of the DAG !. InCausality: objectives and assessment 2010 Feb 18 (pp. 59-86)

Collinearity If the predictor variables are highly correlated, it is possible that we are providing redundant information to the model. This may be necessary to estimate effects correctly, but it can also be harmful. In particular, it can prevent the calculation of coefficients (non-identifiable model) or increase complexity without adding information.

The VIF *Variance inflation factor* is an indicator that helps to understand the influence of collinearity in the estimates.

VIF

The intuition here is that if the variables are closely related $X_1 \sim X_2$, the values of β estimated in $Y = \beta_1 X_1 + \beta_2 X_2 + ...$ they will not be unique. For example, we could exchange β_1 per β_2 and the solution would remain largely unchanged. VIF estimates collinearity in relation to the combination of other predictors used.

To calculate the VIF for a $ X '$ predictor, we adjust a new regression, in which

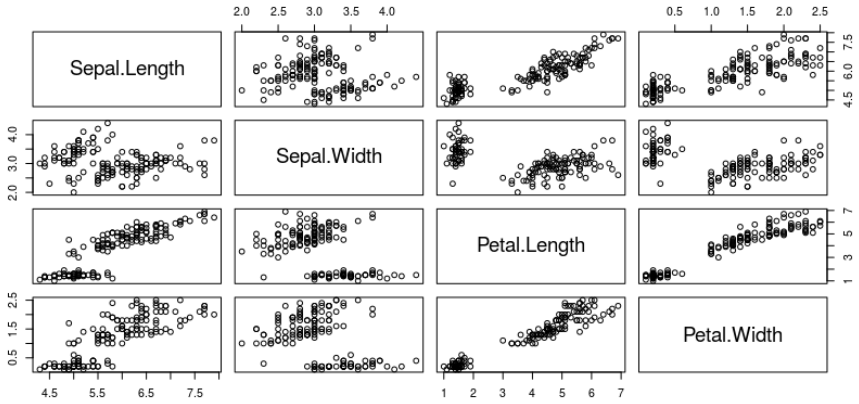
the response variable is X' and the predictors are the other predictor variables. The VIF is given by: $\frac{1}{1-R^2}$, being R^2 the regression determination coefficient, as we calculated before. High VIF values reflect values of R^2 high, that is: the linear combination of other variables would explain the predictor variable in question very well. There is no canonical rule, but VIF$>$ 10 ($R^2 = 0.9$) and VIF$>$ 5($R^2 = 0.8$)

The **vif** function of the *car* package implements the procedure. We adjusted a multiple linear regression for the length of the sepals in the *iris* dataset from 3 other variables. We can verify that there is collinearity ($VIF_{pet.leng.} \sim 15.1$, $VIF_{pet.wid.} \sim 14.2$) between width and length of the petal. On the other hand, collinearity with the sepal length is low ($VIF_{pet.wid.} \sim 1.3$).

```
>car::vif(lm(Sepal.Length ~ Petal.Length + Petal.Width + Sepal.Width,
    data=iris))
Petal.Length  Petal.Width  Sepal.Width
   15.097572    14.234335     1.270815
```

If there is high collinearity and we are only looking for predictive power, removing one of the predictors to eliminate redundancy can be beneficial. As always, visual inspection helps.

```
>pairs(iris[,1:4])
```



As we can see, using two non-collinear predictor variables (multiple regression) increases the model's performance in relation to simple regression ($R^2 \sim 0.84$ *vs* $R^2 = 0.76$).

```
>lm(Sepal.Length ~ Petal.Length,
+         data=iris) %>% summary

(...)
Multiple R-squared:   0.76, Adjusted R-squared:  0.7583
```

```
F-statistic: 468.6 on 1 and 148 DF,  p-value: < 2.2e-16

>lm(Sepal.Length ~ Petal.Length + Sepal.Width,
+         data=iris) %>% summary
(...)
Multiple R-squared:  0.8402,    Adjusted R-squared:  0.838
F-statistic: 386.4 on 2 and 147 DF,  p-value: < 2.2e-16
```

Mediation and Moderation Mediation

A curious idea is that one variable may be mediating the action of another on an outcome. A classic example is the relationship between smoking and cancer. We know that there is a harmful action due to the temperature of the inhaled air, as well as the chemical components absorbed. In mediation models, we try to quantify the portion that is explained by intermediate variables. For this, we use the following procedure:

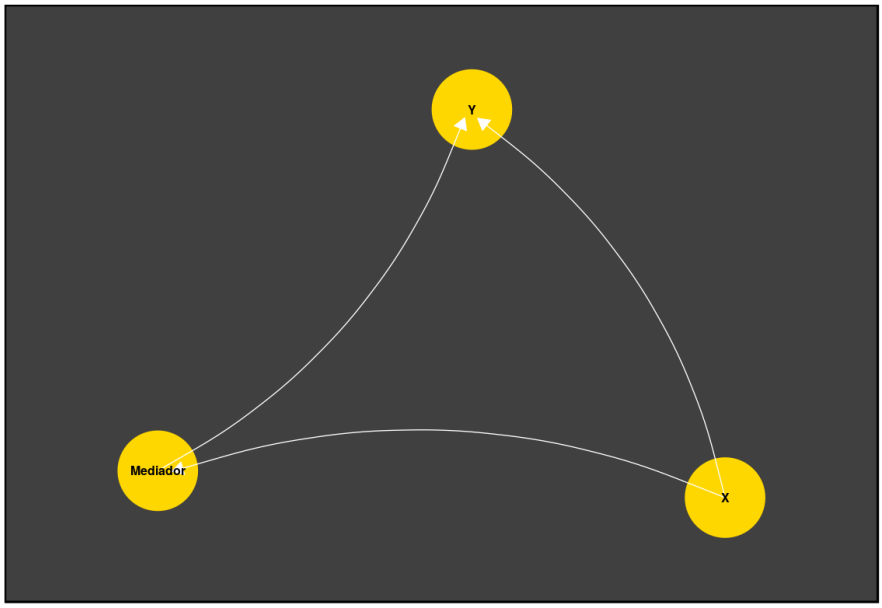
1. Verify plausibility of relationships individually through regression models between variables of interest. We adjusted 3 models: *(1)* independent variable and target variable ($Y \sim X_1\beta_1$),
(2) mediator variable and target variable ($Y \sim X_2\beta_2$),
(3) independent variable and mediating variable ($X_2 \sim X_1\beta_3$).

The direct effect of the independent variable on the target variable is quantified β_1.

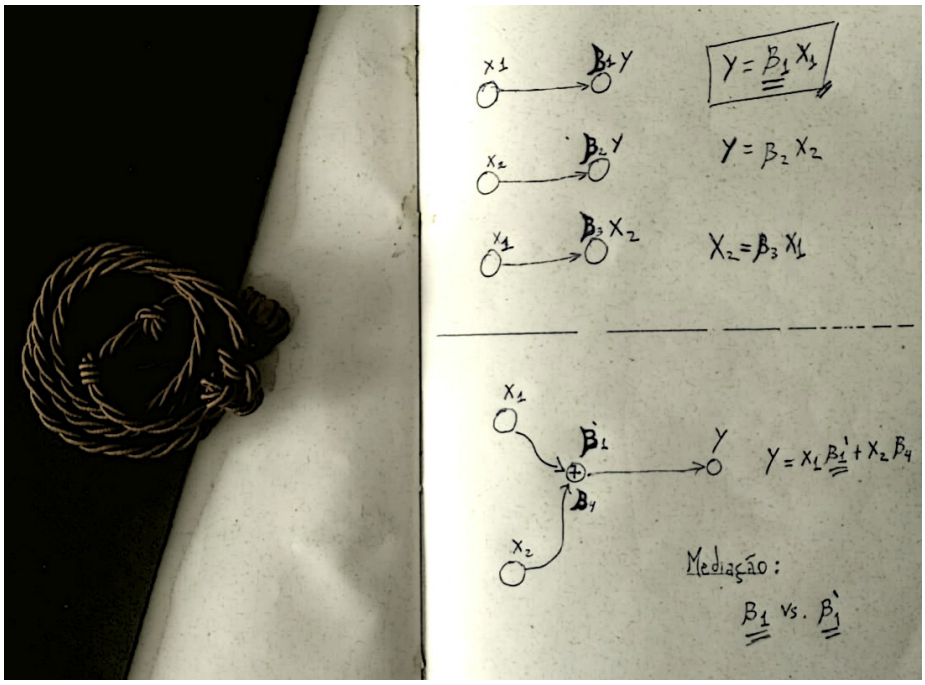
2. Check for changes obtained by introducing the mediator variable. We fit a fourth model *(4)*, with the linear combination of independent variable and mediating variable. We then absorb the difference between the new (β_1') coefficient of X_1 and the old one (β_1) $Y = X_1\beta_1' + X_2\beta_4$.

If mediation exists, it is expected that the coefficient β_1' is not significant or has a very small magnitude in relation to the coefficient of the direct effect β_1.

Following the suggested example, it is expected that there is a relationship between smoking and cancer. Furthermore, it is expected that the inclusion of a mediator (e.g. nicotine concentration) explains part of the effect, reducing the coefficient of X_1. The following diagram expresses the idea contained in the desired process.



The diagram below illustrates steps strictly. The 3 regressions for checking assumptions are in the upper section and the multiple regression in the lower sector. Error terms have been suppressed. Estimates for the relationship between X_1 and Y are $\hat{\beta}_1$ and $\hat{\beta}_1{}'$ highlighted in the equations. The behavior of these parameters defines the conclusions about the mediation model.



There is no guarantee that real systems will behave according to the estimated parameters. We used multiple regression to estimate the partial effect attributed to the gauges, but the removal of these factors in the real phenomenon can result in changes in the system not predicted by the model. Certainty would depend on a fairly accurate description of the phenomenon by regressions ($R^2 \sim 1$), which is rarely verified outside of simpler physical phenomena.

Therefore, it is recommended that adjustments are made in the experimental phase. In our example, this would involve controlling the concentration of absorbed nicotine *in vivo*. Obviously, ethical reasons and limited resources often preclude direct manipulation of the object of study. Methods such as the one described, although fragile, allow the study of interactions and causal relationships. However, increased attention is needed when making conclusions and, especially, when translating them into natural language.

In R:

```
>fit_yx1 <- lm(y ~ x1, data)
>fit_yx2 <- lm(y ~ x2, data)
# Mediation
>fit_yx1x2 <- lm(y ~ y1 + y2)
>summary(fit_yx1)
```

```
(...)
>summary(fit_yx2)
(...)
>summary(fit_yx1x2)
(...)
```

The numerical difference between β_{x_1} is the magnitude of the indirect effect (* Ind. Effect *). We can use an estimate of standard error to derive an associated t-statistic and p-value (Sobel test). Using CRAN libs: Using the dataset `bh1996`, with measures on leadership, well-being and hours of work.The question is: does the leadership climate mediate the relationship between working hours and well-being?

```
>library(bda)
>library(multilevel) # dataset bh1996
>data(bh1996)

# LEAD : Leadership climate
# WBEING : Welfare
# HRS : Work hours


>sobel(pred=bh1996$HRS,med=bh1996$LEAD,out=bh1996$WBEING)
$`Mod1: Y~X`
               Estimate  Std. Error   t value     Pr(>|t|)
(Intercept)  3.51693620 0.052902697  66.47934 0.000000e+00
pred        -0.06523285 0.004590274 -14.21110 3.078129e-45
$`Mod2: Y~X+M`
               Estimate Std. Error   t value      Pr(>|t|)
(Intercept)  1.86832973 0.06413083  29.13310 1.024201e-176
pred        -0.04311316 0.00421918 -10.21837  2.382257e-24
med          0.48386196 0.01242129  38.95426 4.967825e-302
$`Mod3: M~X`
               Estimate  Std. Error   t value     Pr(>|t|)
(Intercept)  3.40718349 0.045154735  75.45573 0.000000e+00
pred        -0.04571488 0.003917997 -11.66792 3.488366e-31

$Indirect.Effect
[1] -0.02211969
$SE
[1] 0.001978985
(...)

>mediation.test(iv = bh1996$HRS,mv = bh1996$LEAD,dv = bh1996$WBEING)
                Sobel        Aroian       Goodman
z.value -1.117729e+01 -1.117391e+01 -1.118067e+01
```

```
    p.value  5.267356e-29  5.471647e-29  5.070460e-29
    # Aroian e Goodman são outros testes para o parâmetro de efeito indireto
```

Moderation and Interactions

Models including moderation terms are those that include **interaction** between variables. Using the jargon of causal inference, it is the same as an effect modifier. As we discussed earlier, the relationship between smoking and cancer can be explained by intermediate factors, such as the concentration of nicotine and the presence of genetic variants of risk. We can assume that the concentration of nicotine inhaled daily has an independent effect. Likewise, a genetic configuration has a causal effect in itself.

$$Risk = Nicotina * \beta_1 + Genes_{(+)}\beta_2$$

n moderation, we add a term to our linear combination. It is a coefficient for multiplication between independent variables.

$$Risk = Nicotina * \beta_1 + Genes_{(+)}\beta_2 + Nicotina * Genes_{(+)}\beta_3$$

Is *smoking* **and* *having* risk genes * different from combining the effect of both separately?

This is one of the few cases where it is easier to observe the algebraic aspect beforehand. We are multiplying the predictor values X_1 and X_2. If both make the same sense (+ or −), the interaction will have a positive effect. Otherwise, negative. Still, we see that the magnitudes are multiplied. The coefficient β_3 quantifies this multiplication in relation to the effect in y,either by changing the direction (β_3 negative) or scaling the absolute value.

$$y = X_1 * \beta_1 + X_2 * \beta 2 + X_1 X_2 \beta_3$$

The relationship of y in relation to each predictor, it ceases to be linear. As we can verify by analyzing the partial derivatives. For $\frac{d}{dx_1}$:

$$\frac{d}{dx_1}(y) = \frac{d}{dx_1}(x_1\beta_1 + x_2\beta_2 + x_1x_2\beta_3)$$

The second term does not depend on X_1, so:

$$\frac{d}{dx_1}(y) = \frac{d}{dx_1}(\beta_1 + x_2\beta_3)$$

The slope (* slope *), which was previously a constant (straight line) β_1 starts to have an added term, which is the multiplication of the estimated constant β_3 by the value of x_2. So we have a different inclination for each moderator value!

These details make the coefficients interpretable difficult. Typically, heuristics, such as centering data around the mean, are used to simplify the context.

Latent measures and factor analysis Consider the problem of measuring something inaccessible through secondary means.For example, the concept of *quality of life* is easily conceivable, although it is not linked to a tangible measure, such as *height* or *femur size*. A number of methods have been developed to deal with the task of estimating *latent variables*. In particular, these models are very popular with psychometrists. We can apply latent variable models for many contexts.

This is done when we use correct answers in a test formulated by experts to quantify a skill. * Item Response Theory * is used in tests such as ENEM (Brazil), SAT and GRE (USA). We relate the skill estimate (θ) with the probability of hitting (1) or wrong (0).

Personality traits can also be studied in this way. We can assign a person's F of a person through his score on a battery of tests $X_1, X_2, X_3, ...$ related to this attribute.

Be the items:

1.I like to be with other people (1 to 7) 2.I usually talk to strangers (1 to 7) 3.I usually express my opinions (1 to 7) 4.I am considered a communicative person (1 to 7)

An individual's score will be a sequence of 4 numbers. A very extroverted individual can score (7,7,6,7) and an introvert (2,3,2,1). We can think that the series of measures is influenced by a construct (extroversion).

Factor analysis starts from the premise that **covariance** in direct measures is the result of **latent influences shared** by the items. So, we can estimate a parameter λ for the relationship between each item and the latent trait F. For this, we will use the covariance matrix.

The values of λ quantify the relationship between items and latent factors and serve, for example, to select items more related to the target traits in a psychometric instrument.

As in linear regression, the model describes each measure as a combination of individual scores for latent factors F multiplied by the weight for the item λ_{Item1} and errors.
The measure of the item 1 to i -th subject considering n latent factors F_n is:

$$x_{1,i} = \sum_{1} nF_i\lambda_n + \epsilon$$

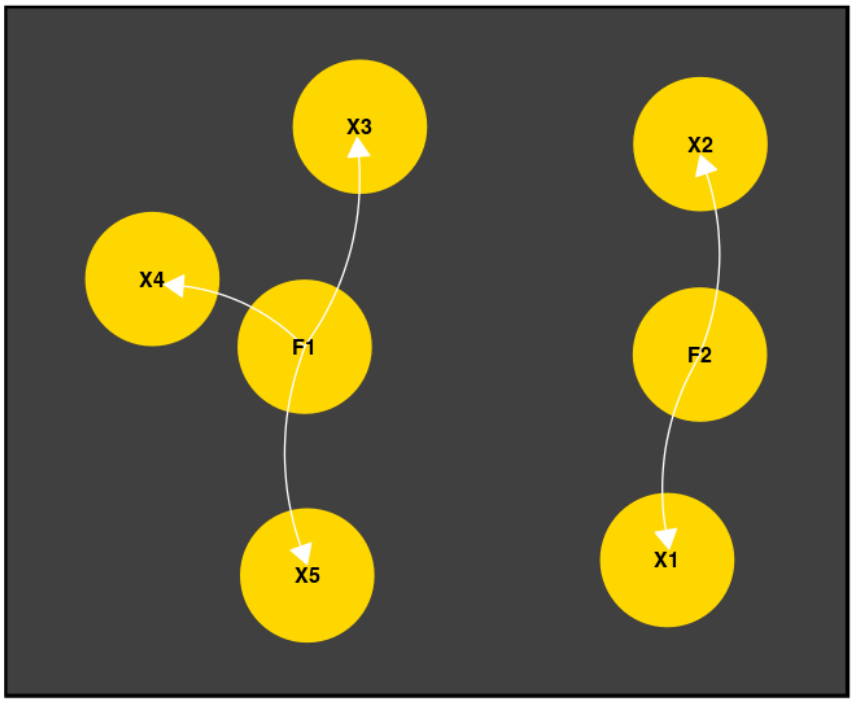
.

Thus, we speak of more than one latent construct. Instead of working with a big latent factor (extroversion), we can link the four items above to two less specific concepts: "sociability" and "expressiveness".

The value of the 4 items for the n-th subject, considering two latent factors, with weights λ_i, λ_i is:

$$x_{1,n} = F_{1,n}\lambda_1 + F_{2,n}\lambda_1' + \epsilon$$
$$x_{2,n} = F_{1,n}\lambda_2 + F_{2,n}\lambda_2' + \epsilon$$
$$x_{3,n} = F_{1,n}\lambda_3 + F_{2,n}\lambda_3' + \epsilon$$
$$x_{4,n} = F_{1,n}\lambda_4 + F_{2,n}\lambda_4' + \epsilon$$

We can see that the matrix Λ will have 8 elements, with 4 pesos for the factor F_2. Knowing the two latent scores of each subject, it would be possible to reconstruct the observations with some degree of loss. Note that we express any item with only two parameters (F_1 e F_2). The information in our dataset could then be condensed into $[nx4]$ dimensions for $[nx2]$.



To estimate the above parameters, we assume that the variance of **each item has an intrinsic variance and a shared variance, which is determined by the latent factors**. We used a covariance matrix between the items to estimate the weights of the latent factors. In addition, we estimated parameters related to the matrix diagonal (variances). In our example, we would have a dimension matrix $[4x4]$.

$$CovMat_x = \begin{pmatrix} . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \end{pmatrix}$$

As we saw in chapter 2, each value is given by:

$$Cov(X, X') = \sum_{i=1}^{N}(x_i - \mu_x)(x'_i - \mu_{x'})$$

The diagonal reflects the covariance of a variable with itself, the variance:

$$\begin{aligned} Cov(X, X) &= \sum_{i=1}^{N}(x_i - \mu_x)(x_i - \mu_x) \\ &= \sum_{i=1}^{N}(x_i - \mu_x)^2 \\ &= Var(X) \end{aligned}$$

For example, the covariance matrix for *iris*:

```
> cov(iris[,1:4])
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
> var(iris[,1])
[1] 0.6856935
```

Using matrix notation, either X a matrix with $m = 4$ columns of $n = 150$ observations, the covariance matrix Cov_{4x4} is:

$$Cov(X') = X'^T X' \frac{1}{n} = X'^T X' \frac{1}{150}$$

X' is the matrix whose values were centralized by the average $x' = x - \mu$. So the product of X by the transpose returns in each element x_{ij} the value $\sum_i^n (x_i - \mu_i)(x_j - \mu_j)$. Easy to implement manually:

```
> iris2$Sepal.Length <- iris$Sepal.Length - mean(iris$Sepal.Length)
> iris2$Sepal.Width <- iris$Sepal.Width - mean(iris$Sepal.Width)
> iris2$Petal.Length <- iris$Petal.Length - mean(iris$Petal.Length)
> iris2$Petal.Width <- iris$Petal.Width - mean(iris$Petal.Width
```

```
> (t(as.matrix(iris2[,1:4])) %*% as.matrix(iris2[,1:4]))*1/150
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.68112222 -0.04215111    1.2658200   0.5128289
Sepal.Width   -0.04215111  0.18871289   -0.3274587  -0.1208284
Petal.Length   1.26582000 -0.32745867    3.0955027   1.2869720
Petal.Width    0.51282889 -0.12082844    1.2869720   0.5771329
```

Based on the principles outlined, the solution desired by us is such that:

1. The covariance between measures is explained by combinations of shared latent variables.

2.The data will be explained by a lower rank matrix. In our case: $\Lambda_{[n\times m]}, m < 4$.

3. For each observation, we will have a latent score value of F_i for each factor. The final value of an item is given by the individual contribution of each factor plus an individual variance. As we saw:

$$x_{1,i} = \sum_{1} nF_i\lambda_n + \epsilon$$

.

4. Each factor has an intrinsic variance, which we will estimate by adding a diagonal matrix ψ to our weight matrix.

We estimate the parameters to maximize the probabilities (* Max. Likelihood *) of the values observed in X given the equations.

$$L(X^T X\frac{1}{n}|\Lambda, \psi)$$

We determine the cost function by knowing Λ and ψ:

$$C \sim \Lambda\Lambda^T + \psi$$

On what ψ is a diagonal matrix with the same rank as Λ. As we saw earlier, the diagonal contains the variances, so the parameters in ψ regulate the variance portion of items governed by factors λ.We say that the diagonal in $\Lambda\Lambda^T$ contains **communalities** (intrinsic variance).

The optimization process to minimize errors is more complex than that of linear regression. The possible estimators here are many, none of them with a simple analytical solution or guarantee of convergence.

Similarities between dimension reduction techniques: EFA, probabilistic PCA, PCA, Autoencoder. We can take into account the previous solution without a ψ linked diagonal matrix:

$$Cov \sim \Lambda\Lambda^T$$

This formulation is equivalent to Principal Component Analysis (PCA). Here, our weights will also estimate intrinsic variance. It is a computationally inexpensive method to reduce dimensions while preserving information. Mathematically, the difference between PCA and EFA lies in the fact that the latter estimates separately parameters for shared covariance and individual variance. A little-known 'intermediate' technique is the probabilistic PCA (PPCA), in which we take into account a simpler diagonal matrix.

$$Cov \sim \Lambda\Lambda^T + \sigma^2 I$$

That is: an identity matrix with noise introduced through only one parameter (σ^2).

A curiosity is that the diagonal ends up influencing less with the increase in the rank of the matrices. So, the result of the above techniques converges in situations with high dimensionality ($n \to \infty$). A more complete discussion can be found elsewhere (see references).

In summary:

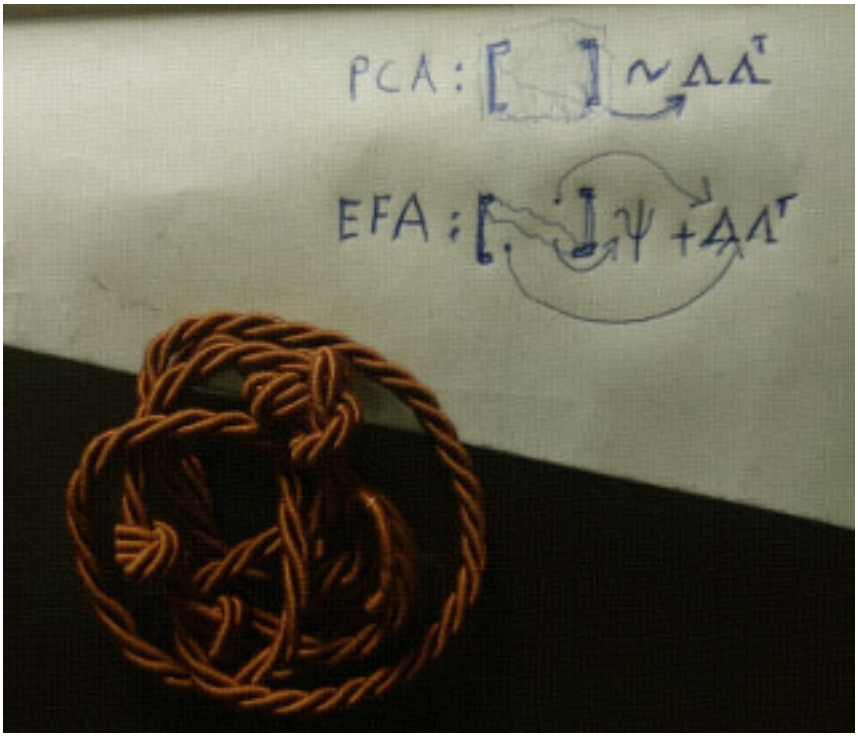


Figure 14: .

.

$$PCA : Cov \sim \Lambda\Lambda^T$$

$$PPCA : Cov \sim \Lambda\Lambda^T + \sigma^2 I$$

$$EFA : Cov \sim \Lambda\Lambda^T + \psi$$

(Here, we use $\sim$ not to call resemblance, but rather that we will maximize the likelihood of Cov with an expression depending on the terms on the right)

Also, neural networks of the *autoencoder* type have a similar formulation. Specifically, an auto encoder with an inner layer and certain restrictions on the activation function is identical to the PCA. However, we can use **more** dimensions than the input, in addition to multiple layers and nonlinear functions. In this way, we increase the power of the generative model, as well as being more vulnerable to overfitting.

We will return to the subject when the focus is on environment models, information compression, generative models and reduction of dimensions.

Number of factors

We haven't touched on a crucial point: what is the optimal number of factors? Is it better to use a model that takes into account *extroversion* or one that uses *sociability* and *expressiveness*?

We can explain covariance using an arbitrary number of latent factors. The tendency is to observe improvement in performance indicators under the penalty of saturation (e.g. overfitting, difficult interpretability). There are established procedures to balance the explanatory power with simplicity of the model.

n general, a minimum number of factors are sought that maximize the explanatory power. Considering degrees of freedom(df) and model errors (statistical X^2), two popular indices are RMSEA and CFI. As in the calculation of R^2,the rationale is to dimension errors, but here we penalize the number of parameters.

Another widely used metric is to observe the influence of each factor on the covariance matrix.

By multiplying a vector by a matrix, we change its magnitude and direction.

Vectors aligned with the matrix (e.g. those diagonal from the transformation above) only change in size after transformation.

They are the eigenvectors of the matrix.

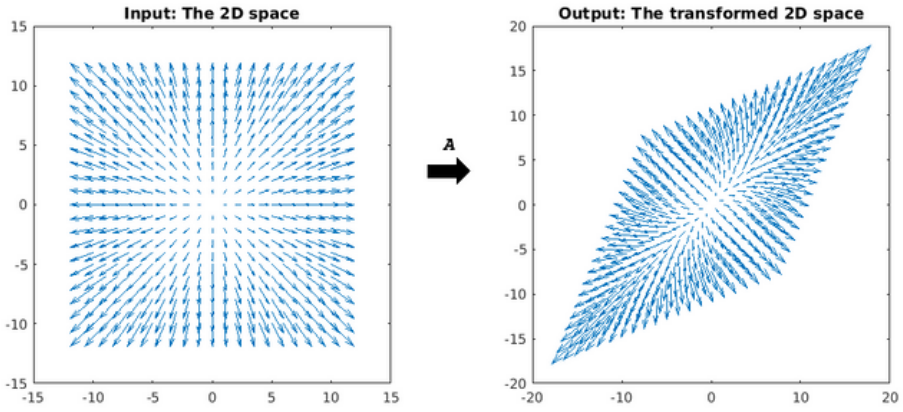


Figure 15: Multiplication effect between vectors and an A matrix

One way of extracting factors is through the main axes. In this method, we decompose the original matrix into orthogonal vectors multiplied by scalars (auto-decomposition, *eigen / spectral decompositon*): eigenvalues and eigenvectors (axes).

In general, the first axes have greater eigenvalues. There are several heuristics recommending methods for choosing numbers of factors by the size of the eigenvalues. One is to consider only eigenvalues greater than 1. Another is to consider the point on the curve where there is an apparent point of discontinuation ("knee").

It is reasonable to think that eigenvectors associated with high eigenvalues capture a lot of information about the variance (individual and shared) of the items.

Confirmatory factor analysis

The processes described above are exploratory in nature. We seek the best fit for latent factors without first determining a structure. It is a good procedure for reducing dimensions and compressing information, however, if we want interpretability and scientific validity, there are some sensitive points.

Thinking about the elaboration of a scale to measure a personality trait, we return to Popper's argument (chapter 2) against inductivism. It is desirable that we have a previous model and testable hypotheses beforehand. Otherwise, it is easy to find a model offering a good fit in almost any case.

In confirmatory factor analysis, we make a direct restriction to the model. The parameters are predetermined based on a diagram (graph) expressed by the person who conducts the analysis. Thus, we can specify a relationship. In the diagram above, the first latent factor has loads in relation to items X_3, X_4, X_5

and the second factor with X_1, X_2.

In this case, the estimators will be a little more complex.

Structural equations Structural equations are the *framework* covering any of the previous models, including graph topolologies and arbitrary relationships (e.g. non-parametric / probabilistic).

Thus, we can draw a diagram of relationships between entities, declare relationships between measures and test the adequacy of the model. As we have seen, Judea Pearl sewed these quantitative methods on a coherent philosophical basis, making use of the concepts of counterfactual and hypothesis testing. These models are useful in many fields to describe statistically multiple element relationships in a complex system. As always, we must be careful with the flexibility of the model. In particular, some recommended procedures are difficult to reconcile with a hypothetical-deductive basis (e.g. ad-hoc change of the model after observing modification indices).

Applications The Big Five personality traits are constructs consistently found in the search for latent factors. They are: pleasantness, neuroticism, openness to experiences, conscientiousness, extraversion

We will use data from https://openpsychometrics.org/. The BIG5 dataset has demographic data (age, gender, country) and 50 measurements on items from the International Personality Item Pool. The sample size is 19.719. We will do exploratory and confirmatory factor analysis through the **psych**, **sem** and **lavaan** packages.

```
>system("wget http://openpsychometrics.org/_rawdata/BIG5.zip")
(...)
Resolving openpsychometrics.org (openpsychometrics.org)... 69.164.197.103
Connecting to openpsychometrics.org (openpsychometrics.org)|69.164.197.103|:80... connect
(...)
Saving to: 'BIG5.zip'
(...)
2019-02-04 09:09:39 (624 KB/s) - 'BIG5.zip' saved [523351/523351]
> system("unzip BIG5.zip")
Archive:  BIG5.zip
inflating: BIG5/codebook.txt
inflating: BIG5/data.csv

>library(psych)
>library(lavaan)
>library(sem)
>bigf_data <- read.csv("BIG5/data.csv",sep = "\t")
>names(bigf_data)
```

```
 [1] "race"    "age"     "engnat"  "gender"  "hand"
 [6] "source"  "country" "E1"      "E2"      "E3"
[11] "E4"      "E5"      "E6"      "E7"      "E8"
[16] "E9"      "E10"     "N1"      "N2"      "N3"
[21] "N4"      "N5"      "N6"      "N7"      "N8"
[26] "N9"      "N10"     "A1"      "A2"      "A3"
[31] "A4"      "A5"      "A6"      "A7"      "A8"
[36] "A9"      "A10"     "C1"      "C2"      "C3"
[41] "C4"      "C5"      "C6"      "C7"      "C8"
[46] "C9"      "C10"     "O1"      "O2"      "O3"
[51] "O4"      "O5"      "O6"      "O7"      "O8"
[56] "O9"      "O10"
```

Let's see what happens if we adjust a model with 5 latent factors:

```
>library(lavaan)
>library(psych)
>efa_big <- fa(bigf_data[,8:57],nfactors = 5)
>efa_big
(..)
RMSEA index =  0.055  and the 90 % confidence intervals are  0.054 0.055
```

We observed a low value of RMSEA, which indicates a low magnitude of errors by degree of freedom. It is interesting to note that we do not indicate which items assess which factors (e.g. Items O1 and O2 are linked to opening up to experience). If the premises are correct, for each item, the solution found must indicate a high load in one factor and a low one in others.

That's what happens. Selecting estimates for three items in three groups. The factor with the highest load is marked with an asterisk.

```
(...)
Factor Analysis using method =  minres
Call: fa(r = bigf_data[, 8:57], nfactors = 5)
Standardized loadings (pattern matrix) based upon correlation matrix
      MR1   MR2   MR3   MR5   MR4   h2   u2 com
E1   0.69*  0.04 -0.03 -0.01  0.00 0.46 0.54 1.0
E2  -0.70* -0.08 -0.04  0.04  0.00 0.48 0.52 1.0
E3   0.63* -0.17  0.16  0.09 -0.06 0.57 0.43 1.3
(...)
N1  -0.06  0.69*  0.10  0.05 -0.05 0.49 0.51 1.1
N2   0.07 -0.50* -0.01 -0.09  0.05 0.26 0.74 1.1
N3  -0.12  0.61*  0.20  0.10  0.01 0.43 0.57 1.3
(...)
A1   0.05  0.09 -0.44*  0.02 -0.07 0.20 0.80 1.2
A2   0.28 -0.04  0.50* -0.05  0.06 0.41 0.59 1.6
A3   0.17  0.27 -0.41* -0.15  0.10 0.27 0.73 2.6
```

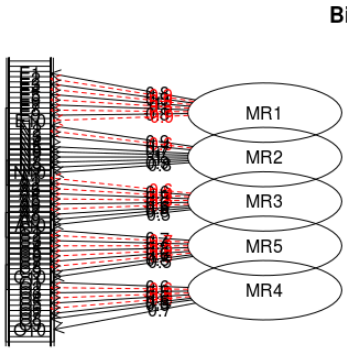
```
(...)
```

Extracting the solution:

```
>efa_bigst <- structure.sem(efa_big)
>efa_bigst
       Path        Parameter Value
[1,] "MR1->E1"   "F1E1"    NA
[2,] "MR1->E2"   "F1E2"    NA
[3,] "MR1->E3"   "F1E3"    NA
[4,] "MR1->E4"   "F1E4"    NA
```

We have the factors (MR) and the nodes to which they are connected, discarding those of lesser magnitude / significance. We can adjust a confirmatory model from these specifications:

```
>big_sem <- sem(efa_bigst,S = cov(bigf_data[,8:57]),N = 19719)
>summary(big_sem)
(...)
>sem.diagram(big_sem,main = "Big Five",e.size=0.05)
```



The lavaan package allows you to specify a larger family of models and is very popular for SEM in R. The syntax is:

```
>model <- c('
F1 =~ X1 + X2 + X3
F2 =~ Y1 + Y2 + Y3')
>lavaan (model,data,...)
```

References Wright, S. (1921). "Correlation and causation". J. Agricultural Research. 20: 557–585.

https://stats.stackexchange.com/questions/123063/is-there-any-good-reason-to-use-pca-instead-of-efa-also-can-pca-be-a-substitut https://steemit.com/steemstem/@dexterdev/linear-transformations-a-20-sbd-coding-contest-announcement

Exercises

1. Examine the VIF of the multiple regression used in the mediation process with database *bh1996*. *Is there a collinearity between mediator and main predictor?

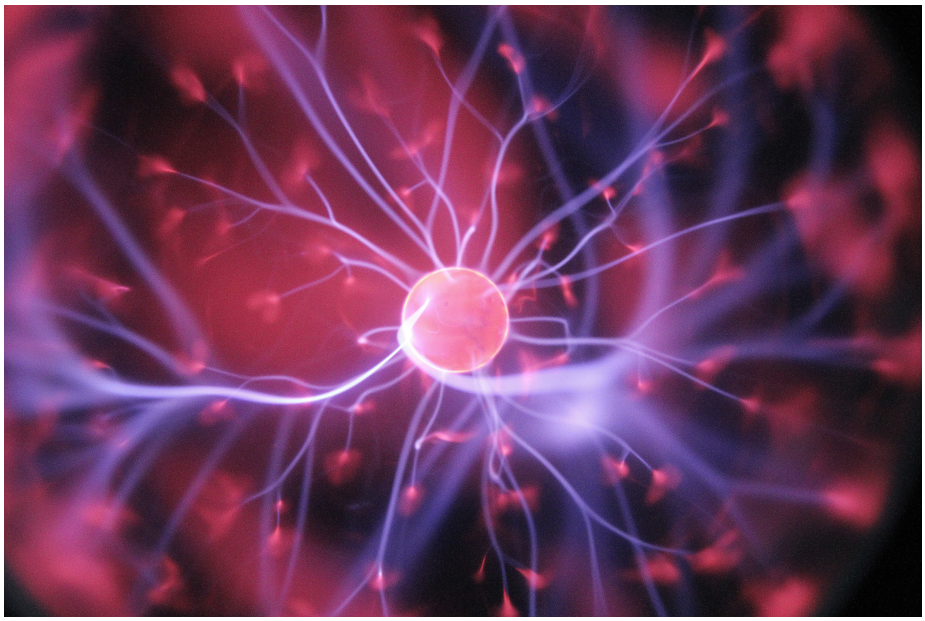
2.Examine the performance change (e.g. R^2) after inclusion of the mediator in the model. * If the mediator variable M explain the same causal pathways as the predictor variable X_1, is this change expected to be big? Argue.

3. Using data of your choice:
    - Adjust a simple linear regression
    - Add another predictor (multiple linear regression)
    - Check for collinearity
    - Check other assumptions by looking at auxiliary material **/ aux** (e.g. independence from errors with Durbin-Watson)
    - Test a mediation relationship using 3 variables
4. Using the data *iris*:

- Choose two correlated measures and check if the species *moderates* the relationship between them. Remember: you must add an interaction term `var1 * var` to the regression formula.

- Perform **(1)** principal component analysis (PCA) and **(2)** exploratory factor analysis (EFA) for numerical variables.

- Extract **(1)** the projection of each observation in the first two components, PC_1, PC_2, and **(2)** the score generated from each factor. The `princomp` function returns an accessible object `$scores`.
- Check the correlation between both.



Chapter 5: Neurons

In March 2016, the AlphaGo software won over a Go master. Invented over 2,500 years ago, the game motivated advances in mathematics. Exist $2,08 * 10^{170}$ valid ways to arrange the pieces on the board. The Chinese polymath Shen Kuo (1031–1095) came to a close result 10^{172} centuries ago. It is worth remembering that the number of atoms in the observable universe is 10^{80}.

In the previous chapter, we learned basic formulations of a predictive model with regression. Here, we will know the first intelligent machine in history implementing a *perceptron*. It is capable of handling more dimensions (e.g. image processing). Closed solution estimators do not exist as in linear regression, so we use local information to 'walk' (*gradient descent*) towards a minimum.

We will extend our toolbox to cover more complex, non-linear relationships. By chaining together simple neurons, we can learn complex signals without appealing to complex, intractable, or overly flexible functions.

Rosenblatt's perceptron

Frank Rosenblatt (1928 - 1971) was born and died on July 11, but this is not the most curious fact of this psychologist's biography. He was responsible for the development of the first artificial neuron. In his words, the first non-biological object to recreate an organization of the external environment with meaning.

It can tell the difference between a cat and a dog, although it wouldn't be able to tell whether the dog was to the left or right of the cat. Right now it is of no practical use, Dr. Rosenblatt conceded, but he said that one day it might be useful to send one into outer space to take in impressions for us. - New Yorker, December, 1958[ˆ22]

The apparatus reproduced the understanding of the time about the functioning of a neuron. The body receives signals from dendrites and, after hidden processing, produces an output in the form of an electrical signal through the axon. The first mathematization would come from the McCulloch & Pitts model ("A Logical Calculus of the Ideas Immanent in Nervous Activity", 1943).

In 1949, Donald Hebb described in his classic *The Organization of Behavior* a plausible mechanism for learning. Commonly expressed in the maxim "Cells that fire together wire together" (cells that fire together, connect to each other).

In order to create a machine that could process inputs directly from the physical environment (eg light and sound), Rosenblatt conceived an elegant extension of the model in 1957 ("The Perceptron [*from Latin, percipio, to understand"*] - a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory"). Composed of three parts: the S (sensory) system; the system A (association) and the system R (response). The McChulloch & Pitts crude "logical" neuron was modified in order to process inputs by weights prior to output. Learning takes place by changing these weights.

Initially, the perceptron was simulated on an IBM 704 (also the cradle of the FORTRAN and LISP languages). Then, implemented as a physical device, named Mark I Perceptron. [ˆ 23] A more in-depth study was published by him in 1962 (Principles of neurodynamics).

[ˆ 22]: He can tell a cat from a dog, even though he can't tell if the dog was on the cat's left or right. At the moment, it has no practical use, Dr. Rosenblatt admitted, but said that one day it could be useful to send an [apparatus] to space to capture impressions for us. [ˆ 23]: Mark I is a title commonly used for the first version of a machine.

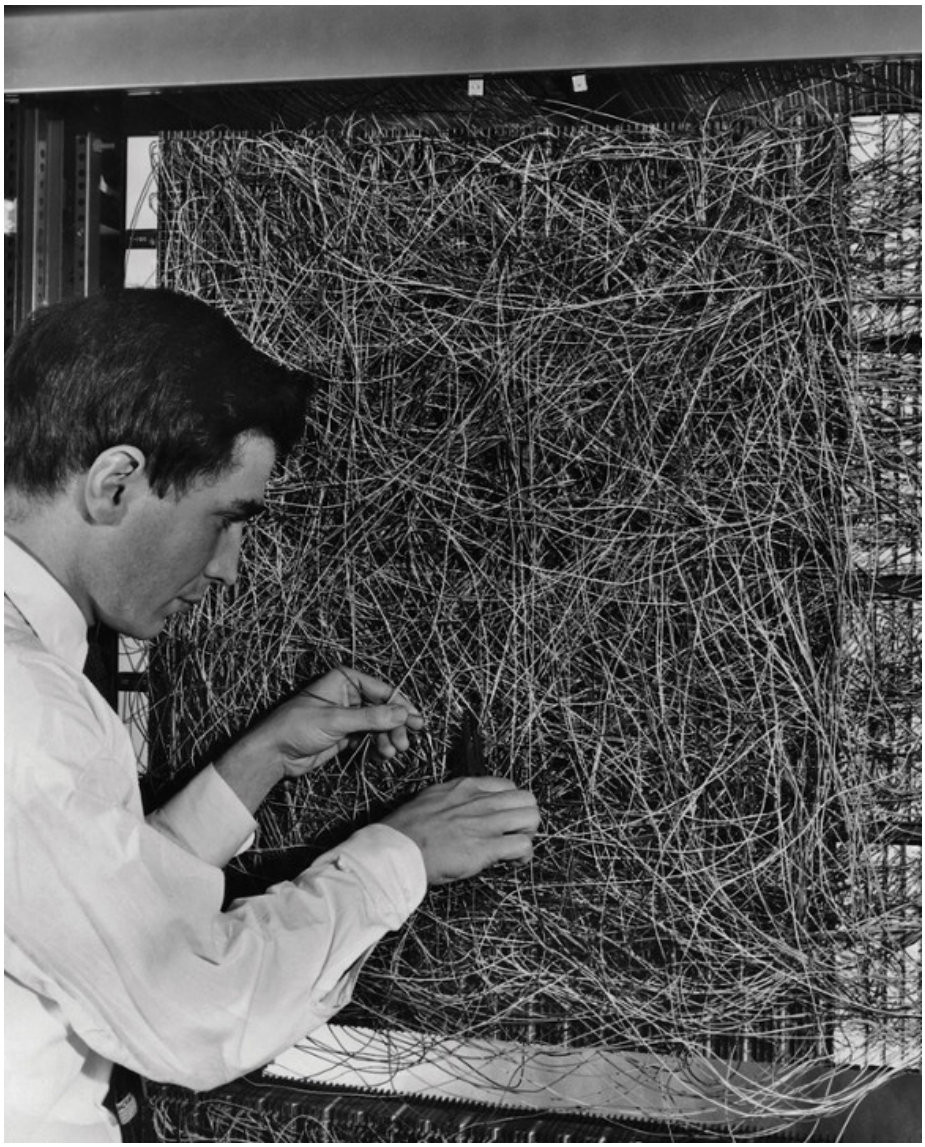


Figure 16: Frank Rosenblatt and Mark I.

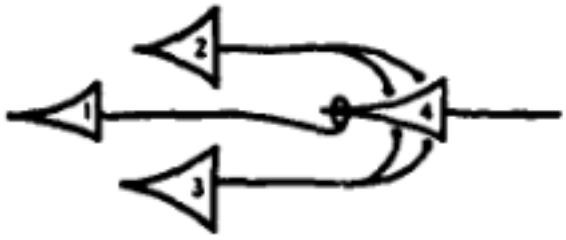


Figure 17: Logical cell diagram in McCulloch & Pitts

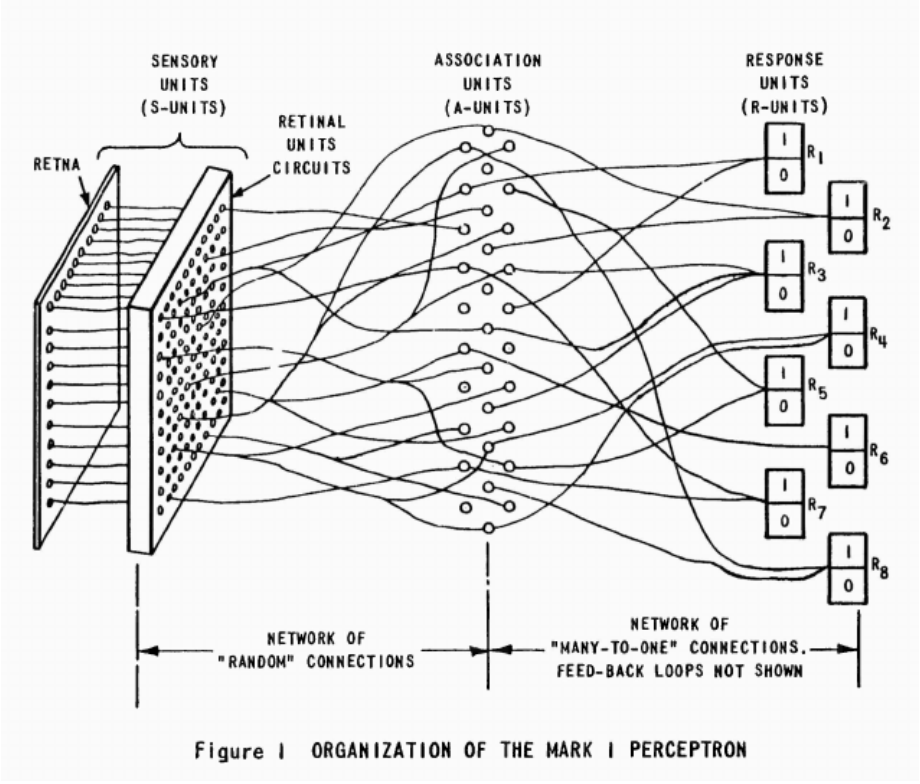


Figure 18: Organization of the Mark I, taken from its original use manual

Rosenblatt starred in heated debates about artificial intelligence in the scientific community with Marvin Minsky, a friend of his adolescence. In 1969, Minsky and a mathematician (Seymour Papert) published a book centered on Perceptron (Perceptrons: An Introduction to Computational Geometry). In it, they proved that the artificial neuron was unable to solve non-linear XOR-type problems. For an eXclusive OR (OR eXclusive) problem, the neuron must fire on stimulus A or stimulus B, but not on both.

The impact was devastating on the current optimism and a period of 10 years of very low production passed, known as the 'dark age' of connectionism. The resumption of artificial neurons took place only in the 1980s. Unfortunately, Rosenblatt died prematurely in 1972 in a boat accident, not seeing the revival of perceptrons.

Knowing the origins of the model, it is curious that most courses introduce perceptrons from a purely mathematical point of view, pointing out the similarity with neurons as mere curiosity. On the contrary, inspiration in biological neurons and subsequent success in the assigned tasks speaks in favor of a fantastic case of success via reverse engineering.

Creating neurons

Mark I was created for visual recognition and can be considered a grandfather of computer vision. It had a photosensitive input field of 20x20 (400) Cadmium Sulphide cells, the S units. When reacting with light, CdS emits an electron:

$$CdS + \gamma \rightarrow e^{-} + CdS^{+}$$

If the cell is activated, it sends the electronic signal to an intermediate unit A. The intermediate unit, in turn, transmits an electronic signal to the output. **The signal strength is regulated by previous successes** in order to adjust the signal for the correct classification. The physical apparatus mimics the mathematical model of the **classifier**.

A luminous signal excites each field differently, activating cells according to the amount of light captured. Mathematically, we represent each light-sensitive neuron as a cell in the input matrix.

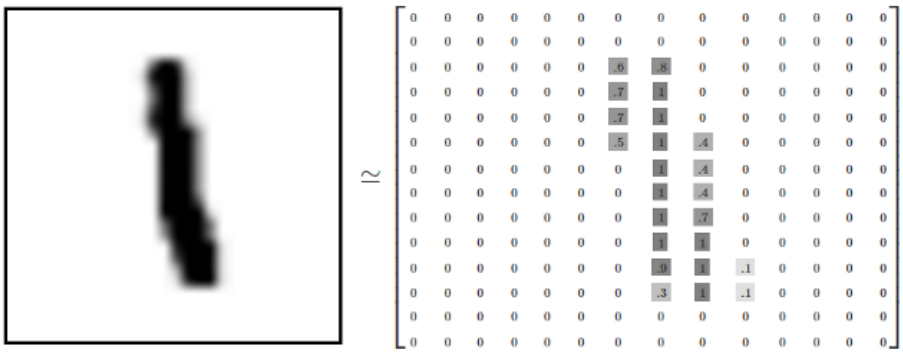
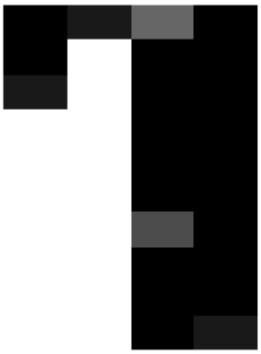


Figure 19: Example of "1" in cursive and its representation in a 2x2 matrix. http://colah.github.io/posts/2014-10-Visualizing-MNIST/

The digit above ('1') is in an image with 14 x 14 pixels (196 values between: 1, black; and 0, white). These pixels can be stretched and viewed as a matrix X of dimension $[196x1]$ with values between 0 and 1 in each element. Let's simulate a similar image:

```
>library(magrittr)
>set.seed(2600)
>my.image.data <- c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                   0,0,0,0,1,.9,.6,1,0,0,0,0,0,0,
                   0,0,0,0,1,0,1,1,0,0,0,0,0,0,
                   0,0,0,0,0.9,0,1,1,0,0,0,0,0,0,
                   0,0,0,0,0,0,1,1,0,0,0,0,0,0,
                   0,0,0,0,0,0,1,1,0,0,0,0,0,0,
```

```
                  0,0,0,0,0,0,1,1,0,0,0,0,0,0,
                  0,0,0,0,0,0,.7,1,0,0,0,0,0,0,
                  0,0,0,0,0,0,1,1,0,0,0,0,0,0,
                  0,0,0,0,0,0,1,1,0,0,0,0,0,0,
                  0,0,0,0,0,0,1,.9,0,0,0,0,0,0,
                  0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                  0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                  0,0,0,0,0,0,0,0,0,0,0,0,0,0) %>%
matrix(.,14,14,byrow=T)
 > image(t(my.image.data[14:1,]), axes = FALSE, col = grey(seq(1, 0, length = 256)))
```



Here is our image [14x14]. The computer reads the values between 0 (white) and 1 (white), providing us with the corresponding visual signal in a color palette. Here we use 256 gray tones.

In multiple linear regression, we calculate a weight β_i for each variable. The rationale is similar: we weight each pixel by its respective weights w_i. In analogy, each image is an observation of 196 variables.

Classification

In the linear regression task, the output should be a real number $Y \sim \beta * X$ with $X, Y \in \mathbb{R}$, such as the average number of professionals or life expectancy. We will use perceptron for another task, classification, in which the possibilities of exit are **categories**. That is, the output is *discretized*, usually in a binary set (e.g. $\{-1, 1\}$ or $\{0, 1\}$) which signals belonging to the class. In our notation, the neuron must fire (output $y = 1$) if you recognize an object or remain at rest ($y = -1$) if not.

Algebraically, it is a multiplication of the matrices between image x_j, of dimension

[196x1] by an array $W_{[196X1]}$ which brings i weights (**w** *eights*) estimated for each pixel for each class. This formulation is identical to that made in linear regression. For a discrete output, we force the result to +1 or -1 with an activation function (ϕ). O output linear $W^T X$ é transformado:

$$y = \phi(W^T X)$$

Thus, the product $W^T X$ it must have a value proportional to the probability of activation: if the input belongs to the class, the result must be high.

We will use the function *Heaviside step*:

$$\phi(x) = \begin{cases} +1 & se \quad x \geq 0 \\ -1 & se \quad x < 0 \end{cases}$$

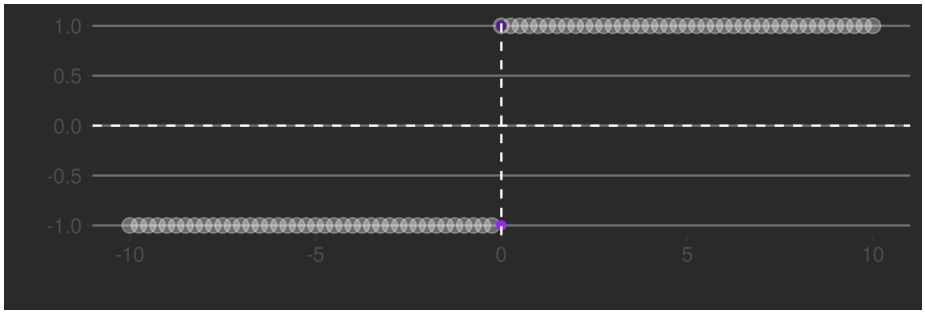


Figure 20: Heaviside step function

In R:

```
# Heaviside
>phi_heavi <- function(x){ifelse(x >=0,1,-1)}
# Starting weights based on normal distribution
>my_weights <- rnorm(196)
>w <- matrix(my_weights,196,1)
# Multiplication using the operator %*%
>as.vector(my.image.data) %*% w
# Score
         [,1]
[1,] -0.3794718
# Activation function
>as.vector(my.image.data) %*% w %>% phi_heavi
     [,1]
[1,]   1
```

For the example above, our neuron with random weights was activated for the stimulus containing the '7'. Initially, we established random weights from a

normal distribution (`my_weights <- rnorm(...)`). The process of training the classifier is to observe the responses many examples of images x_i, changing the values of W so that the highest scores are those of the correct classes. So, neuron just fires $y = 1$ when faced with the proper stimulus.

The training process is quite simple: Be x_{i_j} or i-th pixel of observation j. And w_0 the initial corresponding weight, the updated weight, w' is:

$$w' = w_0 + \Delta w$$

On what Δw indicates the magnitude and direction of the change in weight. Let us accept, for now, the formula:

$$\Delta w_i = \eta(scorej - output_j)x_i$$

On what x_{i_j} is the value of i-th pixel, w_i is the i-th weight and η a constant called *learning rate*, which determines the size of the increments made by the algorithm. We will show the derivation of this equation below.

Auto MaRk I Using the abstractions above, we coded our R perceptron, Auto MaRk I. **Arguments:** Examples (x, real numbers vector) and expected states (y, shoot = 1 vs. do not fire = -1) must be the same size. **Eta:** Number specifying learning constant.

Auto MaRK I initializes a random weight for each entry and, also in a random order, cycles through the examples updating the weights.

```
>mark_i <- function(x, y, eta) {
  # initializes random weights of normal distribution
  w <- rnorm(dim(x)[2]) # number of weights = number of columns in x
  ypreds <- rep(0,dim(x)[1]) # initializes predictions at 0
     # Processes the observations in x at random
    for (i in sample(1:length(y),replace=F)) {
      # prediction
      ypred <- sum(w * as.numeric(x[i, ])) %>% phi_heavi
      # update em w
      delta_w <- eta * (y[i] - ypred) * as.numeric(x[i, ])
      #note: x[i,] will be multiplied as a matrix (dot product)
      w <- w + delta_w
      ypreds[i] <- ypred # save current prediction
    }
  print(paste("Weights: ",w))
  return(ypreds)
}
```

We will test it for the proposed problem, separating *setosa* from *versicolor* flowers. Data preparation:

```
>train_df <- iris[1:100, c(1, 2, 5)]
>train_df[, 4] <- -1
>train_df[train_df[, 3] == "setosa", 4] <- 1
>names(train_df) <- c("s.len", "s.wid", "species","target")
>head(train_df)
      s.len s.wid species target
1   5.1   3.5  setosa      1
2   4.9   3.0  setosa      1
3   4.7   3.2  setosa      1
4   4.6   3.1  setosa      1
5   5.0   3.6  setosa      1
6   5.4   3.9  setosa      1
> train_df[60:65,]
   s.len s.wid    species target
60   5.2   2.7 versicolor     -1
61   5.0   2.0 versicolor     -1
62   5.9   3.0 versicolor     -1
63   6.0   2.2 versicolor     -1
```

```
64   6.1   2.9 versicolor     -1
65   5.6   2.9 versicolor     -1
>x_features <- train_df[, c(1, 2)]
>y_target <- train_df[, 4]
```

And then, we can activate it:

```
>y_preds <- mark_i(x_features, y_target, 0.002)
[1] "Weights:  -0.117938333229087" "Weights:  0.212055910242074"
> table(y_preds,train_df$target) # matriz de confusao
y_preds -1  1
      -1 27  5
       1  23 45
> y_preds
  [1]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [25]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1  1  1
 [49]  1  1 -1  1  1 -1  1 -1 -1  1  1  1 -1  1 -1  1  1  1  1  1  1  1 -1 -1
 [73]  1 -1 -1  1 -1 -1  1  1 -1  1 -1  1  1  1  1  1  1 -1  1  1 -1  1 -1  1
 [97] -1  1  1 -1
```

Using $\eta = 0.002$, we got 72% accuracy (correct classifications, diagonal in the confusion matrix). We can modify the learning rate. With $\eta = 0.05$, stay with 51%. With $\eta = 0.1$, we have 60%. Considerable accuracy compared to expected with guesswork. However, these solutions are not stable and repeated passages generate very different predictions.

```
> y_preds <- mark_i(x_features, y_target, 0.05)
  [1] "Weights:  -1.26323926081935" "Weights:  1.85983709987067"
> table(y_preds,train_df$target)
y_preds -1  1
  -1 35 16
  1  15 34

> y_preds <- mark_i(x_features, y_target, 0.1)
  [1] "Weights:  -1.83248546552824" "Weights:  3.19075461158561"
> table(y_preds,train_df$target)
y_preds -1  1
  -1 31 21
  1  19 29

> y_preds <- mark_i(x_features, y_target, 0.01)
[1] "Weights:  -0.250410476080629"
[2] "Weights:  0.447470183281492"
> table(y_preds,train_df$target)
 y_preds -1  1
-1 25 27
1   25 23
```

What's "wrong" with our estimator?

During the exhibition, the following rule helped us, but it was not explained.

$$\Delta w_i = \eta(scorej - output_j)x_i$$

Before, we verified (Chap. 2) a closed solution to the regression problem, in which the best estimate for the slope of the line, β, could be calculated directly. Perceptron updates its weights recursively, learning a little (Δw_i) with each example. A new stimulus determines how much (magnitude in Δw) and in what direction ($+$ or $-$) a weight must change to decrease errors.

Gradient Descent for Perceptron

When optimizing estimates, we focus on finding maximums or minimums for defined spaces. In general, these are surfaces describing the size of the errors as a function of the weights adopted by the model. Our goal is to find the *lowest* location. For very uneven surfaces, we accept a sufficiently *low* point.

In linear regression, the space is known, it is possible to go to the lowest point directly. For other models, this is not so simple.
Δw_i can be obtained using the concept of *Gradient Descent*. The process is like going down a hill *blindfolded*. We can only know the local inclination (difference between left foot and right foot). We can go down taking steps towards the lowest foot. What we need then is the slope of the surface related to errors as a function of weights.

Taking into account each j-th observation, we first define a loss function L expressing the sum of errors in n examples.

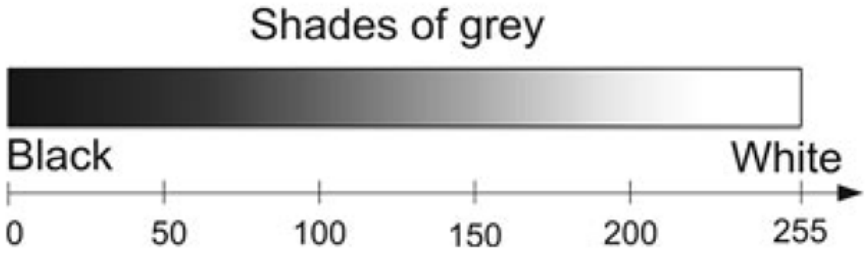
$$L = \sum_{j}^{n} E(score_j, output_j)$$

We will use the Euclidean distance between the desired score and the output for our error function. The desired score is the optimal response and the output is a product between weights and input:

$$E = d_{eucl.}(score_j, output_j) = (score_j - output_j)^2$$

This function describes the surface in terms of errors using a quadratic relationship: erring upwards has the same weight as erring downwards and extreme errors are magnified (x^2) polynomially.

The process involves implementing an error function between network results and a virtual space for optimal scores. The success of the training depends on a correspondence between the chosen distance function and the actual distance in the space in which the data was generated. We don't know if that reflects reality. In the example, each pixel reflects a signal from 0 to 255. The figure below shows the correspondence between measurement values and visual scale.

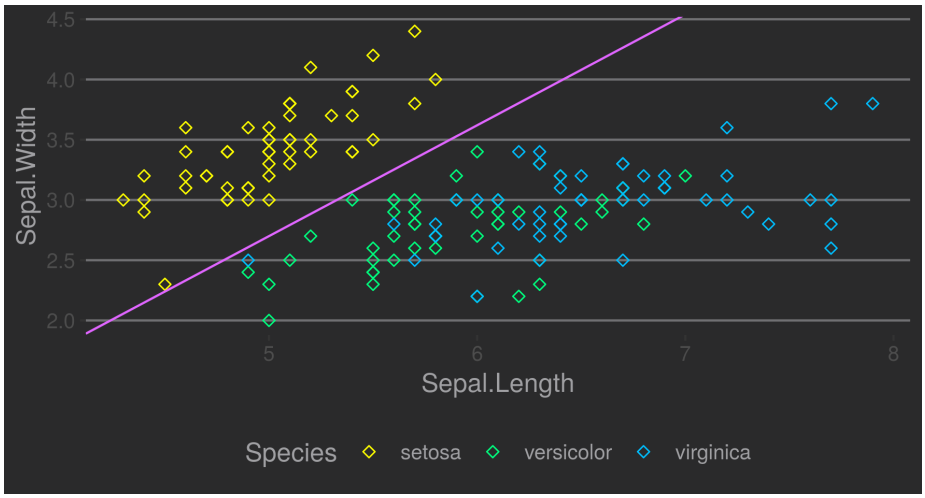


Intuition for sensitivity to light can be perceived in a continuous interval between total incidence of light (extreme values of white, measure: 255) and total absence (extreme values of black, measure: 0). Assuming that we can assign a label to each shade of gray and that this set is sortable by * clarity *, we say that there is isomorphism of order between the sets. This implies that the eulidian distance must work reasonably in our measurements as in real numbers $\mathbb{R}$.

It remains to be seen whether the projection of the observations is linearly separable. It is intuitive for human beings to know which problems will be separable: just imagine the task of differentiating types of images with a ruler on a black and white screen.

If the data are linearly separable, the algorithm converges with a sufficient number of examples. Using *iris*, it would work to separate *setosa* flowers from another class, but we would not have good results separating virginica from versicolor.

```
>ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,color=Species))+
geom_point(shape=5)+ geom_abline(slope = 0.92,intercept = -1.9,color="mediumorchid1")+
scale_colour_manual(values = c("yellow", "springgreen", "deepskyblue"))+
theme_hc(style = "darkunica")
```



To find out the minimum L, we will find poles using partial derivatives. Or, its equivalent for functions of multiple variables (multidimensional spaces), the gradient (∇).

For each observation x_j, the partial derivative of the loss function in relation to a weight w_i expresses the rate of change in the global error as a function of that weight. $\frac{d}{dw_i}L(w_i) = \frac{d}{dw_i}\frac{1}{n}\sum_j nE(score_j, output_j)$

We then know whether to adjust the weight up or down, as well as the magnitude of the step. Algebraically, we will modify w following the inverse of the gradient. The learning rate is a hyperparameter that artificially regulates the size of this step:

$$\Delta w_i = -\eta \frac{dL}{dw_i}$$

$$= -\eta \frac{d}{dw_i}\frac{1}{n}\sum_j^n E(score_j, output_j)$$

Remembering that the error is given by the Euclidean distance:

$$= -\eta \frac{d}{dw_i} \frac{1}{n} \sum_{j}^{n} (score_j - output_j)^2$$

.

We do $f(x) = (score_j - output_j)$ e $g(x) = x^2$, so that

$$L = \frac{1}{n} \sum_{j}^{n} E(score_j, output_j) = (g \circ f)$$

$$= \frac{1}{n} \sum_{j}^{n} (score_j - output_j)^2$$

We can solve $\frac{d}{dw_i} L$ applying the chain rule

$$(g \circ f)' = (g' \circ f) f'$$

and the 'tumble rule' for derivatives of polynomials ($\frac{d}{dx}(x^n) = nx^{n-1}$).

So,

$$f' = \frac{d}{dw_i}(score_j - output_j)$$

The output is given by the scalar product between weights w_j and entries x_j:

$$f' = \frac{d}{dw_i}(score_j - w_j \cdot x_j)$$

The desired score does not depend on the weights, so the first derivative is 0.

$$f' = 0 - \frac{d}{dw_i} w_j \cdot x_j$$

$$= -\frac{d}{dw_i} \sum_{i,j}^{n} w_{i,j} * x_{i,j}$$

$$== -\frac{d}{dw_i}(w_0 * x_0 + ... + w_i * x_i + w_n * x_n)$$

Terms not dependent on w_i are also zeroed and we are left with the first term of the sum:

$$f' = -\frac{d}{dw_i} w_i x_i$$

The function to be derived now describes a linear relationship (polynomial of degree 1) in w_i and have:

$$f' = (-x_{i,j})$$

Knowing f', we look for the other term in $(g \circ f)'$:

$$(g \circ f) = (score_j - output_j)^{2-1}$$

$$(g' \circ f) = 2(score_j - output_j)^{2-1}$$

$$= 2(score_j - output_j)$$

Finally, the partial derivative of the loss function for the i-th weight w_i is:

$$\frac{dL}{dw_i} = \sum_j^n \frac{d}{dw_i}(score_j - output_j)^2$$

.

$$= \sum_{i,j}^n 2(score_j - w_j \cdot x_j)(-x_{i,j})$$

To simplify the expression and establish the size of the increments on the weights, we scale by a constant, given by $-\frac{1}{2}\eta_0$:

$$-\frac{1}{2} * \eta_0 \frac{dL}{dw_i} = -\frac{1}{2}\eta_0 * 2(score_j - output_j)(-x_j)$$

$$\Delta w_i = \eta_0 \sum_j^n (score_j - w \cdot x)(x_j)$$

And η_0 is a [hyper] parameter that simplifies the equation and defines the size of the increments used.

As we implemented in Auto MaRK I.

```
(...)
ypred <- sum(w * as.numeric(x[i, ])) %>% phi_heavi
delta_w <- eta * (y[i] - ypred) * as.numeric(x[i, ]) #<--------------------
w <- w + delta_w
(...)
```

We call η hyperparameter. The choice of values for hyperparameters is one of the challenges in statistical learning. Repeating the learning with *iris*, let's test:

```
> y_preds <- mark_i(x_features, y_target, 0.01)
[1] "Weights:  -0.0153861618736636" "Weights:  0.0812191914731158"
> table(y_preds,train_df$target)
  y_preds -1  1
      -1 25 27
       1  25 23
>  y_preds <- mark_i(x_features, y_target, 0.01)
[1] "Weights:  -0.685141728446126" "Weights:  1.03174770234754"
> table(y_preds,train_df$target)
  y_preds -1  1
       -1 47 10
        1   3 40
> y_preds <- mark_i(x_features, y_target, 0.01)
[1] "Weights:  -0.193515893657872" "Weights:  0.180589056542887"
> table(y_preds,train_df$target)
  y_preds -1  1
       -1 19 37
        1  31 13
> y_preds <- mark_i(x_features, y_target, 0.01)
[1] "Weights:  -0.0672147799277951" "Weights:  0.115145797950982"
> table(y_preds,train_df$target)
  y_preds -1  1
       -1 45 12
        1   5 38
```

Using $\eta = 0.01$, we have 48%. However, running repeatedly returns very good ratings (Acc. > 0.8) or very bad. What is up?

In general, large steps make fine adjustments impossible and may not converge, just as it is impossible for a large animal to explore a narrow valley. Small fees take longer (n observations) to reach a minimum. If the space is irregular, there is also a greater chance of reaching a secondary minimum instead of the bottom of the space. Just as a small animal travels the path more slowly and may have the illusion that it has reached lows quickly.

A trivial way is to test many possible values and observe performance, however this is not feasible for large volumes of data and / or many parameters. There are several heuristic processes and algorithms for finding optimal values. We can also adjust parameters throughout the learning process or test different starting points.

A popular way to optimize training is to partition the dataset into pieces and present the partitions (epochs) repeatedly to the classifier or to accumulate epoch errors instead of individual examples. Thus, we calculate aggregate errors and avoid local minimums. To avoid a lot of changes and to go in circles, we move for longer in only one direction before recalculating the route. Epochs can be recombined and / or resubmitted, artificially increasing the n to calculate

gradients.

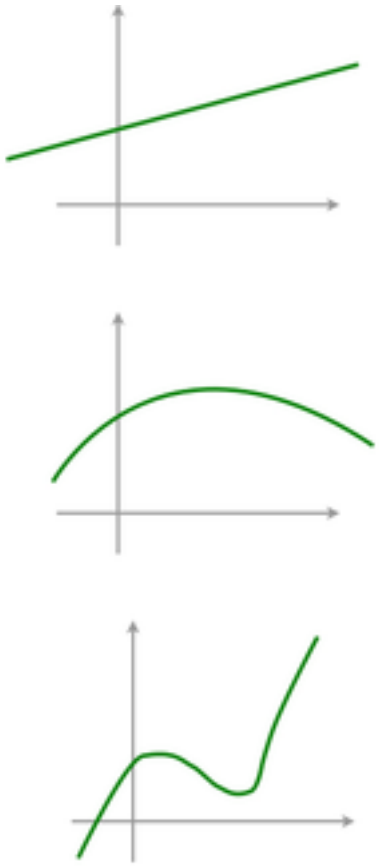
Deep learning

Intuition

With learning through examples, we optimize our classifier (changing weights W) to minimize the loss gradually. One of the conditions for the * perceptron * above to work was the linear separability of classes in the examined space. Some problems are more difficult, being separated by curves. Others are even more difficult, requiring many transformations and specific functions. An alternative is to use higher order polynomials. Rather than $Y \sim \beta_0 + \beta_1 X$, we can introduce terms with greater exponents in X:

$$Y \sim \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + ...$$

Inclusion makes the function more flexible, which can be better suited to the data. In linear regression, we adjust the angle and height of a fixed bar to reduce the distance to the points. With quadratic terms, it is possible to bend this bar in relation to the center, but the ends must go in the same direction. With cubic terms, this is made more flexible.



The introduction of polynomial terms of a higher order makes it considerably more difficult to optimize the estimates.

A neuron *linearly sensitive* to input and equipped with a barrier (*thrsehold*) for firing is able to solve simpler classification problems. For more difficult problems, instead of implementing radically different and / or more complex processing cells, nature uses an ingenious device. Common neurons are chained together: simple calculations and local communication of the units make learning possible.

The data is presented to the perceptrons on the front lines. The output of the first cells is used as input for neurons in the next layer. Thus, we were able to implement suitable transformations (rotations, twists, scalings, folds) in sequence, so that complex abstractions can be captured.

Going Deep

The actual versions of most concepts created by humans are not identical to each other. In other words, there is no rigid set of rules for classifying most

entities around us. Many entities are different, but similar enough to belong to the same category.

All are naturally recognized as felines, but vary in size, color and proportion throughout the body. This is an interesting and ancient problem, best known in the idea of Platonic entities, which capture the essence of a concept. Some contemporary philosophers take human abstractions as instances of a more generic concept: biological maps contained in neural networks. A brilliant exhibition is made by Paul Churchland in *Plato's Camera*. These maps are associated in a hierarchical way. Numerous patterns at lower levels and less of them at higher layers. In the case of vision, superficial neurons pick up luminous points. The pattern of sensory activation captured on the strip and sent to the primary visual cortex is the first map, which is twisted and filtered up the path. At higher levels, individual signals from sensitive cones of energy make up the color palette we perceive.

Intermediate neurons have configurations that identify simple characteristics: eyes and subcomponents of the face. Finally, we have deeper layers, linked to complex abstractions and higher functions (e.g. language).

Deducing surfaces

A classifier must capture this abstract structure from treatable mathematical models. To examine this aspect, let us use an example. The graph below represents thousands of samples with: (1) the natural daily testosterone curve (white) and the curves for measurements using anabolic steroids (yellow).

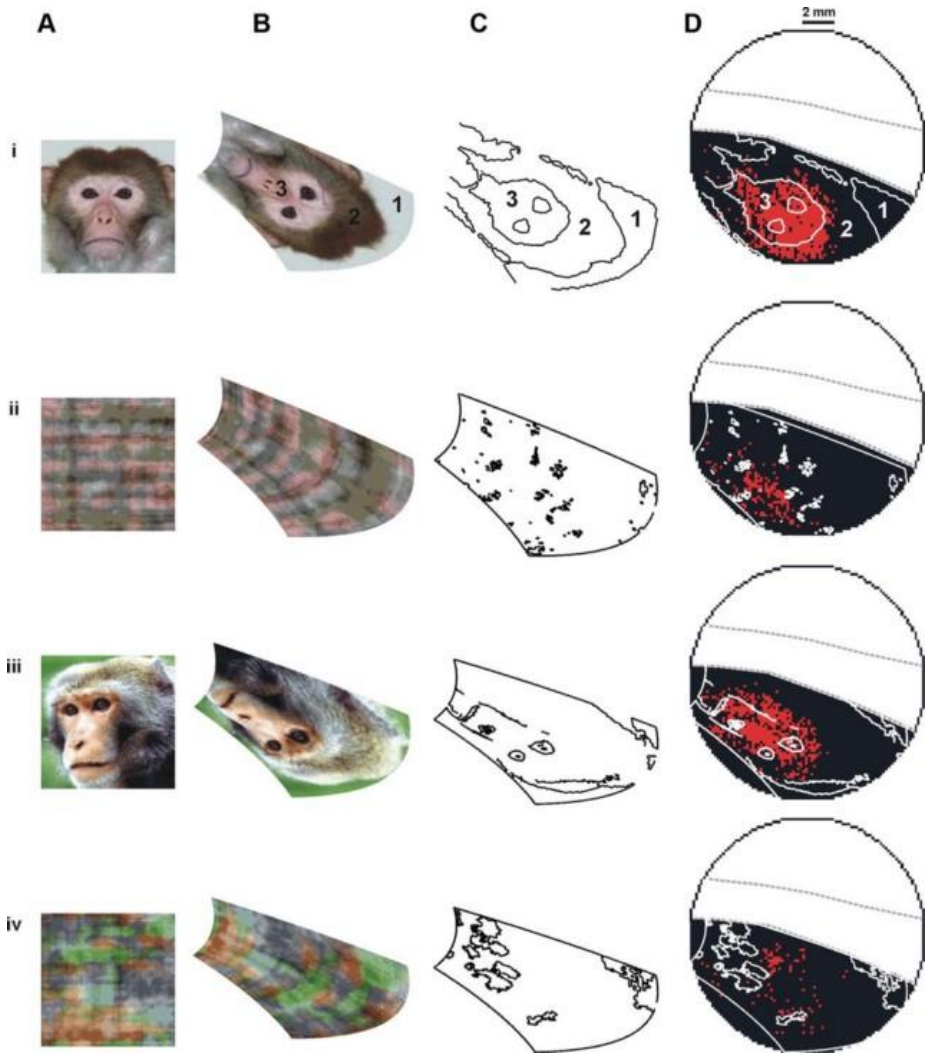


Figure 21: Response to visual stimuli in Macaca V1 fascicularis http://www.jneurosci.org/content/32/40/13971

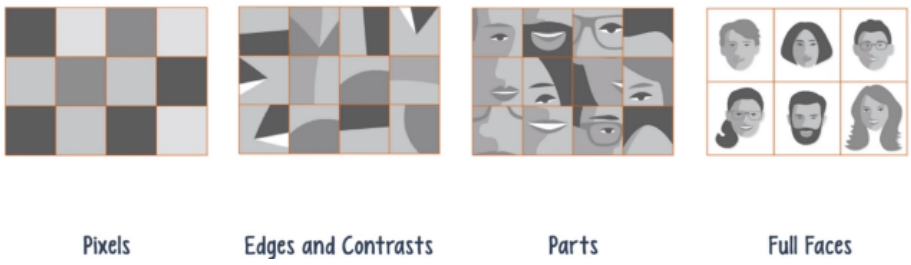
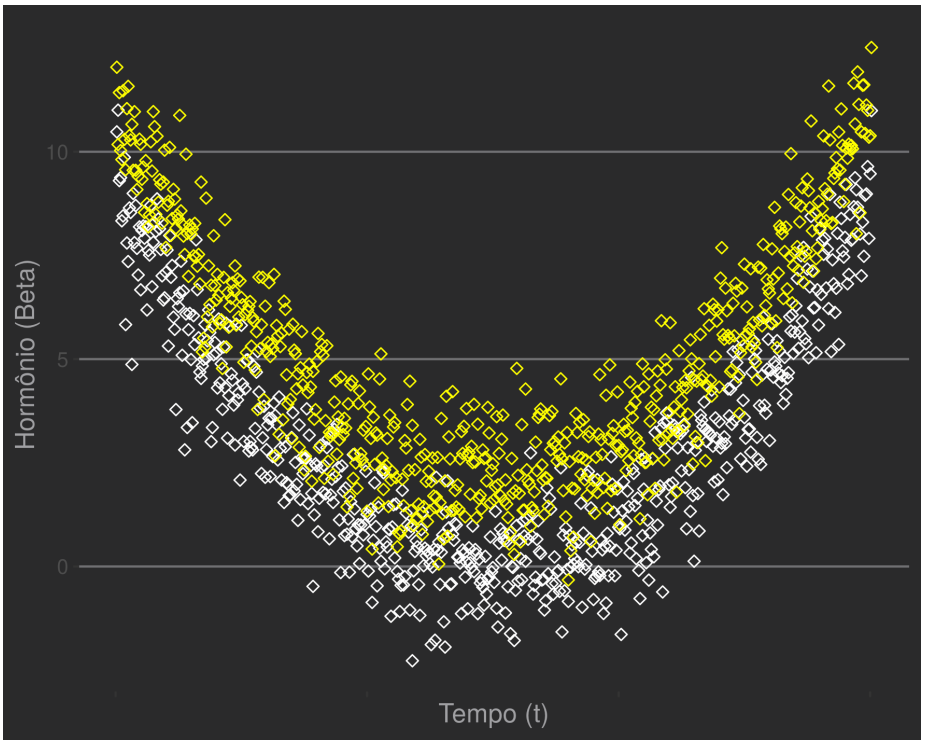
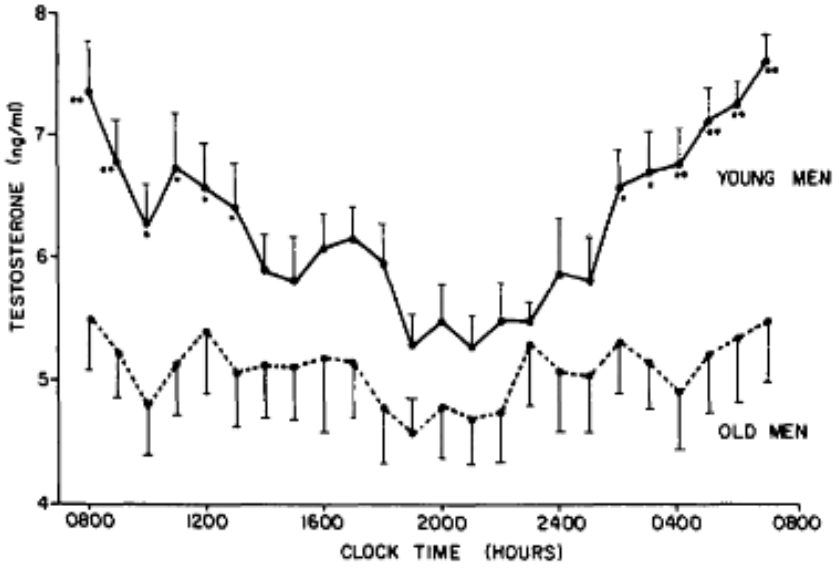


Figure 22: Taken from: https://www.youtube.com/watch?v=SeyIg6ArS4Y

```
>normal <- (purrr::map(seq(-3,3,0.01), .f =function(x) x^2) %>%
as.numeric)+ rnorm(601)
>over <-  (purrr::map(seq(-3,3,0.01), .f =function(x) x^2+2) %>%
as.numeric)+ rnorm(601)
>horm_df <- data.frame(norm = normal, ov = over,time=1:601)
>ggplot(data=horm_df,aes(y=norm,x=time))+
>  geom_point(color="white",shape=5)+
>  geom_point(data=horm_df,aes(y=over,x=time),color="yellow",shape=5)+
>  ylab("Hormone (Beta)")+xlab("Time (t)")+
>  scale_x_continuous(labels=NULL)+
>  theme_hc(style="darkunica")
```





As hypothetical members of an athletic committee, our goal here is, given a sample, to find out if the athlete is on steroids. When we experiment, there will normally be noise (errors) in the measurement and we will receive inaccurate

measurements of the curve. Variations in that day's diet, urination, sweating, stress and other factors. We know that testosterone fluctuates daily following a curve.

For each measure, we have the time (t, horizontal axis) and the hormonal level (β, vertical axis).

A very popular model for classifications is that of logistic regression. In it, we estimate probability for an event based on the probabilities of a sigmoid function. We have a probability (value between 0 and 1) defined by:

$$P(h, \beta) = \frac{1}{1 + e^{-(i+t*h+\beta*y+\epsilon)}}$$

ϵ represents the error and i is a constant.

The equation seems strange, but it appears when we try to calculate probabilities from a linear combination of our parameters:

$$P(x) \sim i + t * x + \beta_i * y + \epsilon$$

This allows the estimation process to be almost identical to that of linear regression, which is easily treatable.

In an R line:

```
>class_df <- class_df <- data.frame(measures=c(horm_df$norm,horm_df$ov),
time=c(horm_df$time,horm_df$time),
target=c(rep(0,601),rep(1,601)))
>logist.fit <- glm(target ~ measures + time, family=binomial,data=class_df)
```

Another consequence is that a linear relationship makes the magnitude and meaning of these relationships interpretable. For example, a positive parameter (e.g. $\beta = 0.241$) indicates that increases in $\beta = -0.9517$) have the opposite effect. Many health risk assessments or finance credit assessments estimate probabilities based on the parameters of a logistic regression.

We use a *decision boundary* dependent on linear relationships. Technically, a hyperplane. A hyperplane divides the space into two parts. It is the plane generalization (zero curvature) for any dimensions. The hyperplane is a $n - 1$ space in a n dimensions space. The line is a two-dimensional hyperplane (our case), the traditional plane is a three-dimensional hyperplane. For higher dimensions, viewing is less simple.

For our nonlinear example, it would be difficult to capture the differences between doped athletes using just this equation.

Above, a sigmoid neuron, which is equivalent to logistic regression. It is like the previous plan, but seen from above, we divided it into two regions for classification. Why? The linear classifier optimizes your responses taking into

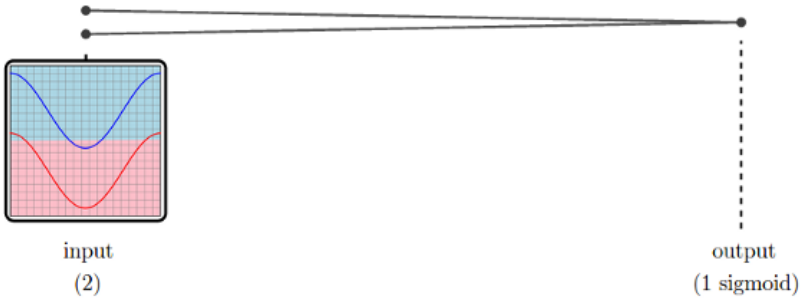


Figure 23: http://colah.github.io/posts/2015-01-Visualizing-Representations/

account only the absolute value of the hormonal measure. That is, values above a limit will be considered dopping, not considering time.

The coefficient for the estimated time tends to be close to 0. When trying to divide the groups with a ruler, it is best to try a straight line parallel to the axis x.
We can verify this directly through the parameters estimated in our regression model. Changing this would make the dividing line inclined with respect to the x-axis, worsening the classification for low or high values.

```
> summary(logist.fit)
    Call:
glm(formula = target ~ measures + time, family = binomial, data = class_df)
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.93641  -1.02791  -0.07236   1.12396   1.63490

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.439e-01  1.504e-01  -6.276 3.48e-10 ***
measures     2.411e-01  2.186e-02  11.027  < 2e-16 ***
time        -2.597e-05  3.621e-04  -0.072    0.943
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1666.3  on 1201  degrees of freedom
Residual deviance: 1526.0  on 1199  degrees of freedom
AIC: 1532

Number of Fisher Scoring iterations: 4
```

```
> prob <- predict(logist.fit,type=c("response"))
> class_df$prob <- prob
> curve <- roc(target ~ prob, data = class_df)
> curve
 Call:
 roc.formula(formula = target ~ prob, data = class_df)

 Data: prob in 601 controls (target 0) < 601 cases (target 1).
 Area under the curve: 0.6964
```

Who can help us?

We went back to neural networks to solve the problem. When we process the signal in stages, each layer modifies the data for the subsequent layers, transforming and filtering / shaping.

The intermediate layers allow for the gradual transformation of the signal, and the system gets it right using only two simple classifiers (sigmoids). In the example above, we have a layer of 2 neurons between the input and the output.

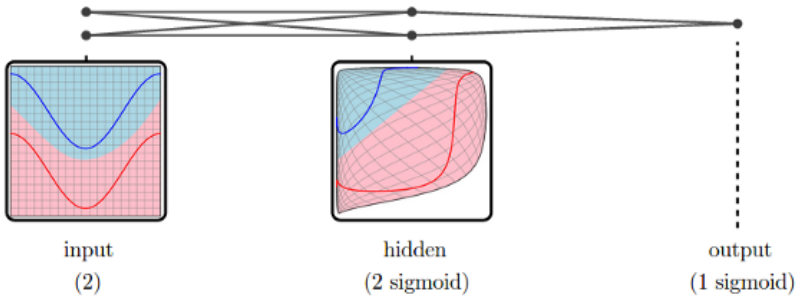


Figure 24: Visualization of signal processing, making it linearly separable. Fonte: http://colah.github.io/posts/2015-01-Visualizing-Representations/

Now, the first layer (hidden) modifies the entry with two sigmoid units and the second layer can classify correctly using only one line, something that was impossible before. In theory, this model can better capture the characteristics that generated the data (hormonal fluctuation throughout the day).

Neurons

Note that the diagram above resembles a neural network. This type of classifier was inspired by the microscopic organization of real neurons and its functioning

is believed to be somewhat analogous. The convolutional neural networks architecture, state of the art in image recognition, was inspired by the visual cortex of mammals [^ 25]. Other bio inspired models (Spiking neural networks, LTSMs . . .) present unprecedented performances for complex and poorly structured tasks, such as speech recognition and text translation. The most accepted theory is that the neural machinery of animals was designed by evolutionary processes, such as natural selection. Thus, it presents colorful forms of complexity depending on the task performed.

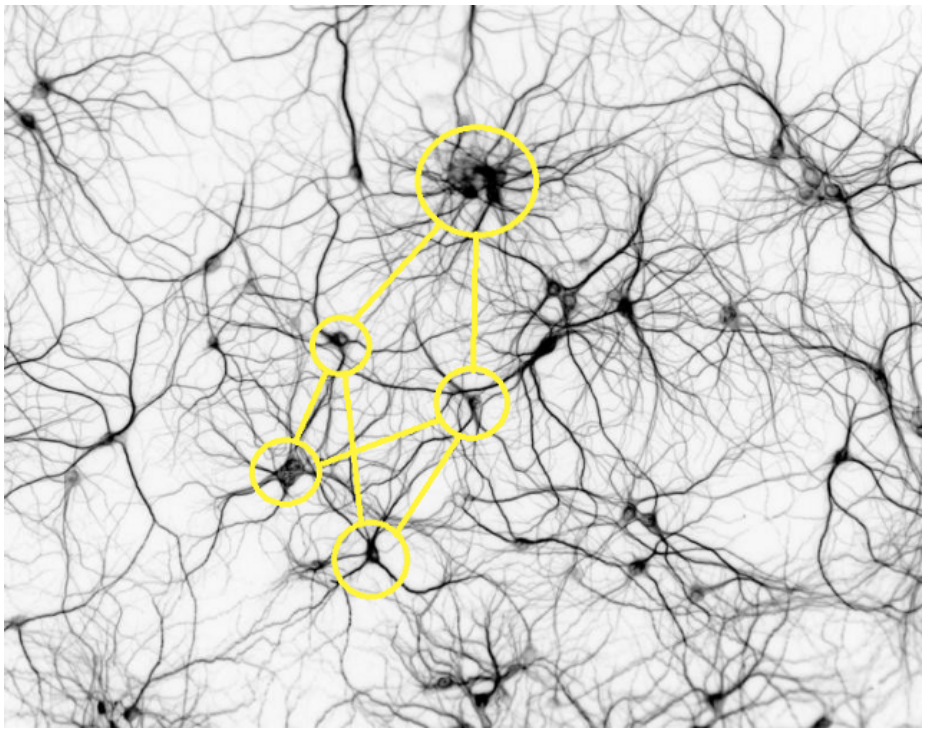


Figure 25: Modified from http://www.rzagabe.com/2014/11/03/an-introduction-to-artificial-neural-networks.html

As we can see, biological networks are complex, with up to tens of billions of parallel processing units connected. The highlighted zone has an isomorphic graph to that described in the text.

In deep (deep) face recognition models, surface layer neurons capture edges, angles and vertices, intermediate layers detect the presence of eyes, mouth, nose. Finally, layers at the end of the architecture decide whether it is a face or not and who it belongs to.

Efficiency and applications We can formally demonstrate that a neural network with only one inner layer is capable of approximating any function. The proof is not there, since, in the end, what we do is create a lookup table for the input and output values using neurons. In practice, it is difficult to obtain good performances. So difficult that neural networks have also been forgotten decades. If you run the model below, based on our example, you will see that accuracy is close to logistic regression. It takes some knowledge and time to fine tune the details. It usually depends on the quality and quantity of the data. The boom came with the discovery of network topologies specifically good for certain tasks (e.g. LSTM for natural language, *Conv Nets* for computer vision). In other words, modeling a neural network for unprecedented problems can be challenging.

The following code shows how to implement a network with a similar topology using lib **caret**. We achieved 81% accuracy using 5 neurons.

```
# Neural Net para o exemplo
>library(caret)
> class_df$time_sc <- scale(class_df$time)
> nn_horm <- caret::train(x = class_df[,c(1,5)], y=factor(class_df$target),method="mlp")
Multi-Layer Perceptron

1202 samples
   2 predictors
   2 classes: '0', '1'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 1202, 1202, 1202, 1202, 1202, 1202, ...
Resampling results across tuning parameters:

  size  Accuracy   Kappa
  1     0.6488305  0.2948640
  3     0.8181583  0.6355261
  5     0.8198874  0.6393824

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was size = 5.
```

Neural networks spent some time forgotten, until some twists [^ 26] allowed the effective training of these networks. Algorithms to improve training, as well as economical or especially good architectures in certain tasks. In addition, the use of graphics processors (GPU), designed for the linear algebra operations that we discussed (with matrices) allowed to train on a larger volume of data.

Backpropagation

Backpropagation is a key process to enable the training of classifiers in deep learning. It is the concept of propagating gradients of the loss function along the network in order to update each node. Historically, it appeared in studies on control theory.

As we have seen, we can see the neural network as a sequence of plugged functions. Algebraically, if the first node is $q(x, y)$, the neuron f that receives its output as input has value $f(q(x, y))$ or $f \circ q$.

Example
Input neuron: $q(x, y) = 3x + 2y$
Second neuron: $f(z) = z^2$
Final output: $f(q(x, y)) = q^2 = (3x + 2y)^2$

At first glance complex functions will have gradients that are difficult to calculate. In addition, we have to calculate values for each neuron in different layers. *Backpropagation* uses the *chain rule* to calculate derivatives by layer. By linking sequences of elementary functions with a known derivative, we can achieve complex mappings and still calculate the gradient without much effort.

We can obtain the gradient of the loss function at the highest hierarchy node (f), with respect to one of the input variables (x) in the lowest hierarchy. The operation is computationally cheap, simply multiplying the partial derivatives of the errors in each part.

$$\frac{df}{dx} = \frac{df}{dq}\frac{dq}{dx}$$

It is possible to recursively calculate, therefore local and parallel, along the layers. Doing the same above to df/dy, we will have the values of df/dx and df/dy which is precisely our gradient in f.

```
# Double value (x, y) for inputs
 > x <- 1
 > y <- 3
 q <- 3 * x + 2 * y # first layer
 f <- q ^ 2 # second layer
 # Backprop - Changes in higher hierarchy
 # given by lower layer inputs
 dfdq <- 2 * q # derived from x ^ 2; variation of f as a function of q
 dqdx <- 3 # Derivative of 3x; variation of q as a function of x
 dqdy <- 2 # Derivative of 2x; variation of q as a function of y
 # Get gradient of f (x, y) by multiplying the partials
 dfdx = dfdq * dqdx
 dfdy = dfdq * dqdy
 grad = c (dfdx, dfdy)
```

```
> grad
[1] 24 16
```

Using this logic, we calculate the gradients for the error function and train the model.

We can then implement our neural network, Mark II.

Mark II Our network will have an input perceptron with the same size as the input. However, we added an extra weight, which will correspond to an intercept.

Note that

$$y = w_0 + w_1 x_1 + w_2 x_2$$

is the same that

$$y = 1 * w_0 + w_1 x_1 + w_2 x_2$$

So, we add a weight w_0 and we also force an extra dimension in the input, which will always have value 1. We call this trick of adding an intercept (*bias*) to *bias trick*. It helps to establish a baseline value for the output, facilitating convergence.

```
library(magrittr)
library(ggplot2)
set.seed(2600)

mark_ii <- function(x, y, eta, reps=1) {

  # initializes random weights of normal distribution
  w1 <- rnorm(n = (dim(x)[2]+1)) %>% as.matrix # number of weights = number of columns in x
```

Then, neurons of the middle layer, two, each with two weights.

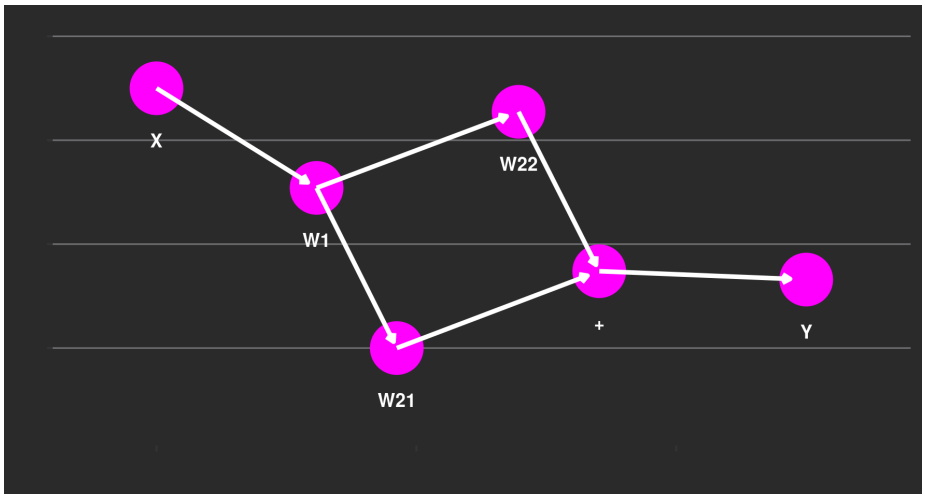
```
  w21 <- rnorm(2) %>% as.matrix
  w22 <- rnorm(2) %>% as.matrix
```

We reset the predictions and start the training loops. For the neural network, we need many examples of exposure, so we embedded in Mark II a parameter (reps) responsible for repeating the training process with the dataset. Strictly speaking, it would be best to partition the dataset into smaller fragments for each epoch, but let's keep things simple.

```
  ypreds <- rep (0, dim (x) [1]) # initializes predictions at 0
   yerrors <- rep (0, dim (x) [1]) # initializes predictions at 0
  for (j in 1:reps){
    print(paste("This is training epoch:",j))
    print(paste("Current weights:",w1,w21,w21))
```

Predictions: the first layer adds the product of its weights through the entrance and the unit (*bias trick*). The neurons in the second layer add the product of their weights to the output. The final output is the sum of the outputs in the middle layer.

```
# Processes x observations at random
for (i in sample(1:length(y),replace=F)) {
  # predicao
  ypred1 <- sum(w1 %*% c(as.numeric(x[i, ]),1))

  ypred21 <- sum(w21 %*% as.numeric(ypred1))
  ypred22 <- sum(w22 %*% as.numeric(ypred1))

  out <- sum(ypred21,ypred22)
```



Now, the rules for updating the weights following derivations with chain rule. For intermediate neurons, we have: $\frac{d}{dw_{21}}$ and $\frac{d}{dw_{21}}$ of $(target - (pred22 + pred21))^2$.

$$\frac{d}{dw_{21}}(target - (pred22 + pred21))^2$$

Applying the chain rule and knowing that the prediction of the second neuron W_{22} does not depend on the weights in W_{21}:

$$= 2(target - (pred22 + pred21)) * \frac{d}{dw_{21}}(-1)(pred22 + pred21)$$

$$= 2(target - (pred22 + pred21)) * \frac{d}{dw_{21}}(-1)(w_{21} * ypred1)$$

Which is the derivative for the perceptron weights:

$$= 2(target - (pred22 + pred21)) * (ypred1)(-1)$$

However, calculating the weights of w_1 as a function of the output requires a little more:

$$\frac{d}{dw_1}(target - (pred22 + pred21))^2$$

$$= 2(target - (pred22 + pred21)) * \frac{d}{dw_1}(-1)(pred22 + pred21)$$

$$= 2(target - (pred22 + pred21)) * \frac{d}{dw_1}(-1)(\sum w_{22} \sum w_1 x + \sum w_{21} \sum w_1 x)$$

Using the derivative of sums and verifying that terms not related to w_1 rated only:

$$2(target - (pred22 + pred21)) * (-1)(\sum w_{22} x + \sum w_{21} x)$$

```
      # update em w . Eta already set to 1/2 * eta
      delta_w22 <- eta * (-1) * (y[i] - (ypred21 + ypred22)) * ypred1
      delta_w21 <- eta * (-1) * (y[i] - (ypred21 + ypred22)) * ypred1
      delta_w1 <- eta * (y[i] - (ypred21 + ypred22)) * -1 *
        (sum(w21 %*% c(as.numeric(x[i,]),1)) + sum(w22 %*% c(as.numeric(x[i,]),1)))

      w1 <- w1 - delta_w1
      w21 <- w21 - delta_w21
      w22 <- w22 - delta_w22
      ypreds[i] <- out # current prediction save21
      yerrors[i] <- ypreds[i] - y[i]
    }
        print(paste("Mean squared error:", mean((yerrors)^2)))
  }
  return(ypreds)
}
```

So, we can test it in a dataset.

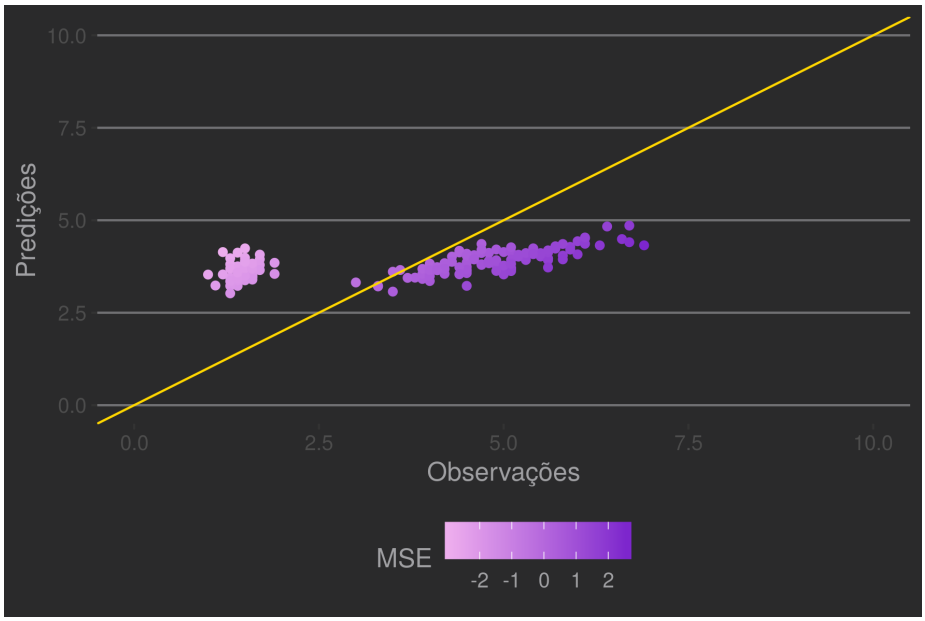
```
    >train_df <- iris[, c(1, 2, 3)]
    >names(train_df) <- c("s.len", "s.wid", "p.len")
    >head(train_df)
    >train_df[60:65,]

    >x_features <- train_df[, c(1, 2)]
    >y_target <- train_df[, 3]
```

```
# Good convergence
>mark_ii_preds <- mark_ii(x = x_features,y = y_target,
                         eta=0.000001,reps = 40)
[1] "This is training epoch: 1"
[1] "Current weights: -0.45050790019773 -0.0197893400687895 -0.0197893400687895"
[2] "Current weights: 0.150011803623929 2.13458518518008 2.13458518518008"
[3] "Current weights: 1.48235899015804 -0.0197893400687895 -0.0197893400687895"
[1] "Mean squared error: 1133.22204821886"
(...)
[1] "This is training epoch: 2"
[1] "Current weights: -0.67126807499406 -0.0609395311239563 -0.0609395311239563"
[2] "Current weights: -0.0707483711724013 2.09343499412492 2.09343499412492"
[3] "Current weights: 1.26159881536171 -0.0609395311239563 -0.0609395311239563"
[1] "Mean squared error: 176.747586724131"
(...)
[1] "This is training epoch: 4"
[1] "Current weights: -0.791488817323548 -0.0700721883119202 -0.0700721883119202"
[2] "Current weights: -0.19096911350189 2.08430233693696 2.08430233693696"
[3] "Current weights: 1.14137807303222 -0.0700721883119202 -0.0700721883119202"
[1] "Mean squared error: 7.32496712284895"
[1] "This is training epoch: 5"
[1] "Current weights: -0.805708526415977 -0.0705118739404967 -0.0705118739404967"
[2] "Current weights: -0.205188822594319 2.08386265130838 2.08386265130838"
[3] "Current weights: 1.12715836393979 -0.0705118739404967 -0.0705118739404967"
[1] "Mean squared error: 3.31246116798174"
(...)
[1] "Mean squared error: 2.50706426321967"
(...)
[1] "Mean squared error: 2.50638029884829"
(...)
[1] "Mean squared error: 2.50640582517322"
```

We can see the model converging as the weights stabilize and our measure of error falls. Using the η above, the network tends to converge with correlation $\rho \sim 0.60$ between predictions and original data.

```
>acc_data <- data.frame(y_preds=mark_ii_preds,
                        y_targs=y_target)

>acc_data$errors <- y_target - mark_ii_preds
>cor.test(acc_data$y_preds,acc_data$y_targs)

    Pearson's product-moment correlation

data:  acc_data$y_preds and acc_data$y_targs
```

```
t = 8.9717, df = 148, p-value = 1.203e-15
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4788098 0.6883163
sample estimates:
      cor
0.5935271

>ggplot(acc_data,aes(y=y_preds,x=y_targs,color=errors))+
  geom_point()+xlim(0,10)+ylim(0,10)+
  geom_abline(slope = 1,intercept = 0)
```



In a practical way, we do not need to calculate the gradients or the topology of the network (number of neurons, layers and how they are connected). Machine learning libraries automate parts of the process, offering rapid usability for many efficient classifiers. Using the lib `caret`:

```
> library(caret)
# https://topepo.github.io/caret/train-models-by-tag.html
> train(x=x_features,y = y_target,method = "mlpWeightDecay")
    Multi-Layer Perceptron

150 samples
  2 predictors

No pre-processing
```

```
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 150, 150, 150, 150, 150, 150, ...
Resampling results across tuning parameters:

  size  decay  RMSE      Rsquared   MAE
  1     0e+00  1.830946  0.3132915  1.5795672
  1     1e-04  1.831956  0.4041400  1.5641681
  1     1e-01  2.203828  0.5889224  1.9507964
  3     0e+00  1.035326  0.6731265  0.8242900
  3     1e-04  1.129702  0.6322950  0.8921468
  3     1e-01  2.230236  0.6531256  1.9114274
  5     0e+00  1.094755  0.6558700  0.8567348
  5     1e-04  1.121093  0.6523228  0.9007250
  5     1e-01  2.143342  0.6639255  1.7652741

RMSE was used to select the optimal model using the
 smallest value.
The final values used for the model were size = 3 and decay = 0.
```

We have $R^2 \sim 0.673$ with 3 hidden units. Other architectures (e.g. define `method = "brnn"`) nclude nodes with different activation functions, as well as variations for the operation of other points.

References For a complete story about neural networks: J. Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks, 61, p 85–117, 2015. (Based on 2014 TR with 88 pages and 888 references, with PDF & LATEX source & complete public BIBTEX file).

http://web.csulb.edu/~cwallis/artificialn/History.htm https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html https://rpubs.com/FaiHas/197581

Exercises

1. A camera is placed on the ceiling and we need to create an algorithm that determines whether the ball is on the left or right. Would a perceptron like the one presented be able to correctly learn how to indicate possession of the ball?
2. In biological neurons, we model activation as a function of the voltage in the neuronal membrane. What role models are there? See advanced free software for network simulation in Neural Ensemble (https://neuralensemble.org/projects/)
3. Redesign the learning algorithm (loop `for`) so that the learning rate η be reduced to $\frac{\eta}{10}$ in the last examples.
4. Implement Mark I adapted to learn with epochs and gradient testing η small.
5. Explore other neural network architectures using *caret*.

Chapter 6: Context and Bayesian Inference

Odds

"The probable is what happens most of the time", Aristotle, Rhetoric.

A popularized probabilistic approach to applied mathematics is that of *Bayesian Inference*. The procedures presented above are usually called *frequencists*. Often, the information obtained is almost identical, but the perspective changes considerably.

In principle, we use different paths.

Frequentists and Bayesians

Frequentist approaches place probabilities as approximations for scenarios with an infinite number of events. The examples visited in the first chapters often made this analogy.

To return to a trivial example: if we throw an honest coin endlessly, the proportion of * heads * tends to what value? For many drawings, the proportion tends to 0.5. Simulation:

```
> set.seed(2600)
> coin_t <- function(x) {
sample(size=x,x=c(0,1), prob = c(0.5,0.5), replace = T) %>%
(function(y) sum(y)/length(y))}
> coin_t(3)
[1] 0.6666667
> coin_t(10)
[1] 0.4
> coin_t(30)
[1] 0.5666667
> coin_t(100)
[1] 0.51
> coin_t(1000)
[1] 0.498
> coin_t(100000)
[1] 0.50098
> coin_t(10000000)
[1] 0.4999367
```

The idea of infinite hypothetical populations or procedures is common. The hypothetical-deductive method links theories to observations through falsifiable hypotheses. The most accepted conception, recently compiled by K. Popper, deals directly with probabilities as important entities for the natural sciences. More than that, it illustrates the concept of calculating the plausibility of experimental results in the presence of a hypothesis under study.

We calculate a probability associated with the occurrence of an observation. In the two sample t test (chapter 1), we define the null hypothesis as a function of the nozzle averages(μ) and other parameters (σ,*df*). $H_0 : \mu_{amostra_1} = \mu_{amostra_2}$. The procedure of imagining the observed events as instances of a family of similar events is perfectly suited to Popperian precepts. It remains the common science bean and rice to test predictions of a given paradigm. The gradual refinement of a theory involves the accumulation of knowledge and testing of *auxiliary hypotheses* resulting from basic assumptions (*hard core* in the terminology of Imre Lakatos).

Bayesian prisms instrumentalize probabilities as primitive beings, more basic notions related to *plausibility*, *degree of belief*, *expectation* for a given situation. The key point is that we fail to guide the procedures aiming at a probability for the events. The probabilities themselves become central entities. Specifically, how our beliefs about something change after observations.

In the case of birds:

- Frequency inference *: Assuming the average difference between nozzle sizes is 0, what is the probability for my observations? Being H_0 defined by $H_0 : \mu_{amostra_1} = \mu_{amostra_2}$, we want to know:
  $P(H_0) < 0,05$?

Bayesian inference: What are the probabilities associated with the possible values for the difference between $\mu_{amostra_1}$ e $\mu_{amostra_2}$? Considering a model and data, what is the probabilistic distribution of $\mu_{diff_{1-2}}$
$P(\mu_{diff_{1-2}}) =$?

In addition to intuitive constructs, a Bayesian platform offers two powerful features: sensitivity to prior information about a phenomenon (*priors*) and stochastic estimators (e.g. *Markov Chain Monte Carlo*). Thus, we can (1) make use of arbitrary information (e.g. an expert's intuition) and (2) reduce the dependence on analytical (closed) solutions for equations that describe the models.

Bayesian epistemology? Rather, we associate scenarios with hypotheses and estimate parameters (probabilities) to test them. The *parameters* now have a more central conceptual role.

A parameter is a symbol, an approximation to an idea (*to*, "near", *metron*, "measure"). In the initial chapters, we use parameters for constructs that behave like numbers (e.g., there are elements that can be ordered by some notion of size and operations, such as sum and multiplication).

We estimate parameters(μ_{diff} and p value) to test a hypothesis about the average difference between nozzle sizes in species A and B. In chapter 2, a parameter (β and a p value) to test a hypothesis about the correlation between healthy

life expectancy and number of doctors in a country. More than that, we use statistics to test hypotheses and calculate confidence intervals.

It is very difficult to understand the usefulness of the previous procedures without knowing the hypothetical-deductive north guiding them. The following excerpt is in *Data Analysis, A Bayesian Tutorial* (Sivia & Skilling, 2006), by Oxford professors: *"The masters, such as Fisher, Neyman and Pearson, provided a variety of different principles, which has merely resulted in a plethora of tests and procedures* ***without any clear underlying rationale*** *. This* ***lack of unifying principles*** *is, perhaps, at the heart of the shortcomings of the cook-book approach to statistics that students are often taught even today."*

We can even use probabilities obtained via Bayesian inference to continue testing hypotheses. However, it is convenient to introduce Bayesian tools to the thinking of philosophers who offered other alternatives[16].

Many scientific methods: Feyerabend, Carnap and Quine

In the first chapter, we come into contact with the hypothetical-deductive method and falsifiability as a scientific demarcation criterion. Despite being dominant, this rationale has interesting vulnerabilities. We will better understand contrary arguments and alternative proposals through three 20th century philosophers. This is a convenient time, as we take the spotlight out of hypotheses.

Paul Feyerabend (1924 - 1994)

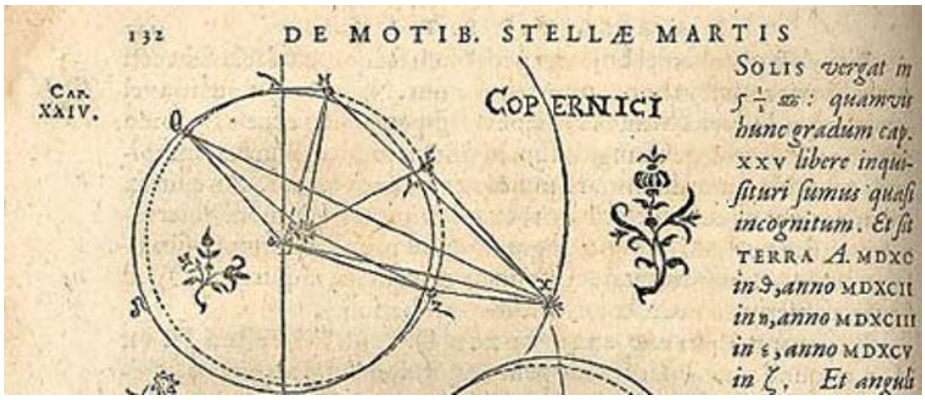
Known for his unique personality and radical ideas, Paul Feyerabend, in *Against the Method* (1975), argues that much of the significant progress has taken place outside the scientific method.

Personal beliefs and biographical details are responsible for changes in our knowledge. More than that, using falsifiability and the hypothetical-deductive method would have made us reject heliocentrism and other key ideas for progress. In fact, Ptolemy's geocentric system (Earth in the center of the system) was more accurate (!) Than Copernicus (Sun in the center) using the same number of parameters for calculating the orbits. The Copernican model was closer to reality as understood today, but the intermediate stage of theoretical conception was 'worse' [17].

In addition to being less accurate, it was more complex in some ways, including more epicycles: auxiliary orbits used as a device for calculations. The Copernican Revolution only consolidated the paradigm shift with subsequent contributions from Tycho Brahe, Kepler, Galileo and Newton, about 1 century later.

[16] There is a more comprehensive research program in philosophy on Bayesian epistemology, but this is not our focus. See The Open Handbook of Formal Epistemology

[17] Stanley E. Babb, "Accuracy of Planetary Theories, Particularly for Mars", Isis, Sep. 1977, pp. 426



Faced with the inconsistencies between a method and the inevitable unpredictability of the human endeavor to discover the Universe, Feyerabend proposes *epistemic anarchism* under the motto *"Anything goes"*('Anything goes'). That is, any resources are valid in an attempt to attack a problem or conceive a model of reality.

It is tempting to think that, given the depth of the work, the defense of such a forceful posture is obviously an application of the precepts defended in the book as necessary to spread an idea. Other philosophers help us to conceive of a science not based on a hypothetical-deductive method in a less radical way.

Rudolph Carnap (1891 - 1970)

Carnap, from the Vienna Circle, also opposed Popper. In "Testability and Meaning" (1936-7), he argues that falsifiability is no different from verificationism. It involves testing each statement itself, a problem that [others] (https://en.wikipedia.org/wiki/Ludwig_Wittgenstein) also addressed.

In the face of unexpected results in an experiment, the automatic procedure for a scientist involves checking the integrity of the designed conditions. Check the sample composition, collection methods, loss mechanisms, exclusion and inclusion criteria, analysis premises. This is not intellectual dishonesty: they are minor, real and easily accessible factors that may have invalidated the basic theory. The same is true for techniques of analysis and conceptualization of constructs.

Taking care of these points is desirable and exposes the inevitable Achilles' heel of falsifiability. It is impossible to refute a hypothesis / assertion in isolation. Each experimental or logical procedure involves the interdependence between the symbols used.

Willard van Orman Quine (1908 – 2000)

A philosophical school starts from the above problem. Duhem-Quine's thesis postulates that it is impossible to test any scientific hypothesis, since there are always premises accepted as truth.

In *'The two dogmas of empiricism'*, Quine considers the propositions and logical relationships between them to be only one system, which can only be studied together. The exercises illustrated in the previous volume tests the suitability of the data for the t distribution family. It also assumes that nozzle sizes are measurable using numbers and that these can be compared to values from other samples.

A princípio, essas declarações parecem triviais. Entretanto, considerando os fatores humanos da ciência, a mudança de lentes é significativa. Discutivelmente, abordar um problema dessa maneira é historicamente mais frutífero. As contribuições mais contundentes são advindas de cientistas dedicados a estudar um contexto ou problema como um todo. É raro, talvez inédito, que um grupo testando hipóteses sem um eixo consistente tenha obtido avanços admiráveis.

Freely estimating the parameters we speak of naturally is much more intuitive than adapting an idea to hypothetical-deductive procedures.

Bayesian inference

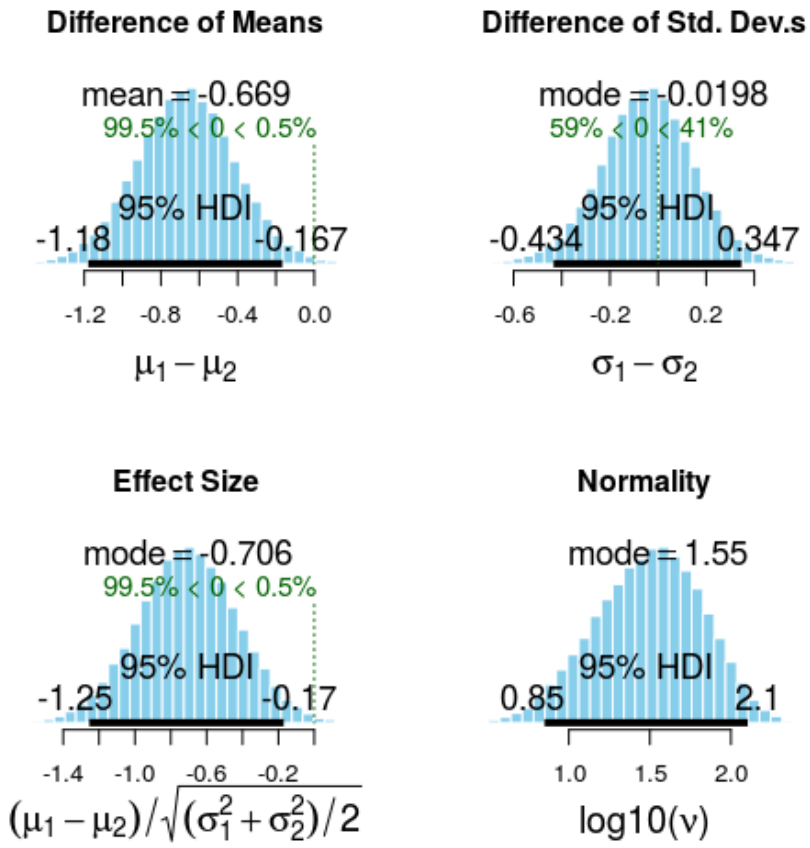
In chapter 1, when doing a t test, we calculate the t statistic corresponding to the differences found and then the probability of obtaining equal or more extreme values. It is possible to use Bayesian inference to analyze an identical

situation. As mentioned before, we are not very interested in the p-value. The question is *"What are the likely values for the difference between A and B?"*.

The probabilistic distribution obtained represents our beliefs in the plausibility of each value.

Using the BEST library and 30 observations taken from samples of normal distribution ($\mu_a = 0; \mu_b = 0.6; \sigma_a = \sigma_b = 1$) normal.

```
> library(ggthemes)
> library(rstan)
> library(reshape2)
> library(BEST)
> library(ggplot2)
> options(mc.cores = parallel::detectCores() - 1)
> set.seed(2600)
> a <- rnorm(n = 30, sd = 1, mean = 0)
> b <- rnorm(n = 30, sd = 1, mean = 0.6)

# BEST
> BESTout <- BESTmcmc(a, b)

### BEST plots
> par(mfrow=c(2,2))
> sapply(c("mean", "sd", "effect", "nu"), function(p) plot(BESTout, which=p))
> layout(1)
```



The distribution in the upper left corner corresponds to our estimates for possible values of the difference between A and B. We can use the average as a point estimate:($diff_{\mu_a \mu_b} = -0.669$). The range indicated as 95% HDI (High density interval) contains 95% of the distribution. Its meaning is closer to the intuition of a probable region for the values than the classic confidence interval.

Behind the curtains

Obviously, we will understand the art involved here. The flexibility and power of Bayesian models allows us to deal with a series of problems that are difficult to deal with otherwise. However, it is easy to fall into traps or run into difficulties during the process.

It is extremely important to understand the components involved in order not to make any major mistakes.

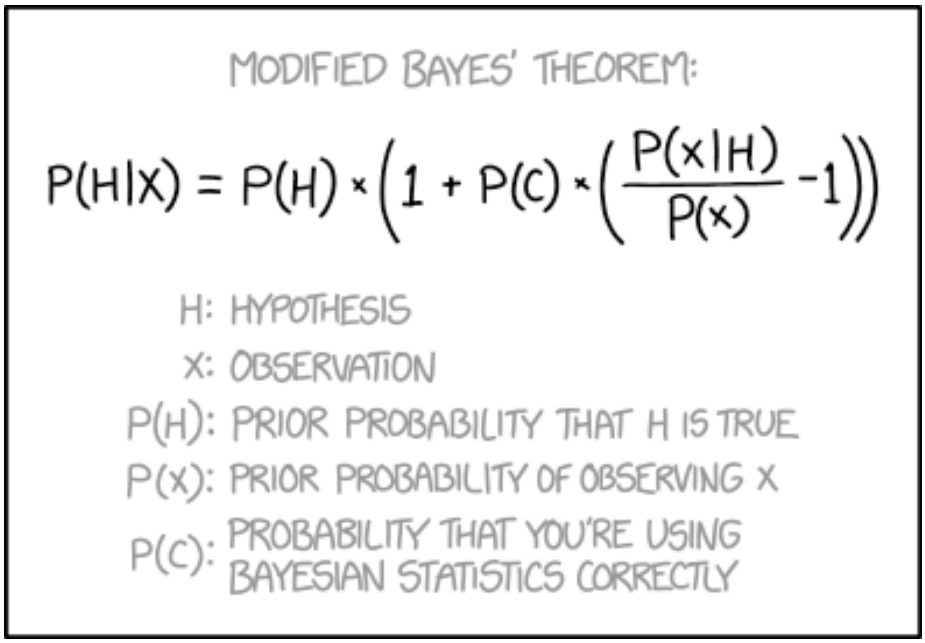


Figure 26: https://xkcd.com/2059/ Modified Bayes' theorem, including the probability that you are using Bayesian statistics correctly

The Bayes Theorem

$$P(B \mid A) = \frac{P(A) \mid B)P(B)}{P(A)}, P(A) \neq 0$$

It is the famous form of the theorem and tells us about probabilities of subsequent / concurrent events.

It is usually presented to treat simple problems: *knowing the result of a positive medical test, how likely is the patient to have the disease?*. Bayes' theorem links the baseline probability of the disease to the probability of a subsequent positive test. Some pitfalls of intuition are broken: even if the test has good sensitivity (high probability of a positive result in the face of the disease), the probability will be low if the baseline chances are also.

The theorem was conceived in a greater effort by the reverend (Thomas Bayes, 1701-1761) for a problem of inference. Interestingly, it is quite similar to what we will undertake.
Suppose we assign a p probability($0 \leqslant p \leqslant 1$) for the launch of a coin with a *crown* result. By observing some results, we can calibrate our estimate. We can start by assuming an honest currency 0.5. With a high frequency of *crowns*, it is rational to increase our estimate of the value of p ($p \sim 1$). Bayes demonstrated how to make these updates in the face of evidence.

Intuition The text of **An essay towards solving a Problem in the Doctrine of Chances (1973)** presents a series of demonstrations until reaching the statement:

Proposition 4 : *If there be two subesequent events be determined every day, and each day the probability of the 2nd [event] is* $\frac{b}{N}$ *and the probability of both* $\frac{P}{N}$*, and I am to receive N if both of the events happen the 1st day on which the 2nd does; I say, according to these conditions, the probability of my obtaining N is* $\frac{P}{b}$*. (. . .)*

The style is a little complicated. With current notation: Considering two subsequent events, *(1)* the probability of the second happening is $\frac{b}{N}$ ($P(A)$), *(2)* the likelihood of both of them happening is $\frac{P}{N}$ ($P(A \cap B)$. *(3)* Knowing that the second happened, the probability that the first also happened is $\frac{P}{b}$. N is canceled and * (3) * is the ratio between *(2)* and *(1)*:

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}, P(A) \neq 0$$

Considering two events, **A** and **B**, the probability of B happening knowing that A happened ($P(B \mid A)$) is identical to the probability of A and B ($P(A \cap B)$) happen, normalized by the probability of A happening individually.

By the definition of conditional probability, $P(A \cap B) = P(A \mid B)P(B)$, so:

$$P(B \mid A) = \frac{(A \mid B)P(B)}{P(A)}, P(A) \neq 0$$

Thus, we can estimate probabilities of events. In Bayesian inference, we use the theorem to estimate the probable values (probabilistic distribution) of a parameter (θ) in the face of observations (X).

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)}, P(X) \neq 0$$

Later We call the first term, the parameter estimate after calibration by the observations $P(\theta \mid X)$, ** **posterior distribution** (*posterior distribution* translates well into Portuguese). All procedures are designed to calculate it and represent the distribution used in the final inferences. For example, we want the subsequent distribution of values for the difference between A and B.

Marginal probability The denominator of the term on the right is the independent probability for the occurrence of the data ($P(X)$). It is used to normalize quantities and is called marginal probability / likelihood, **marginal likelihood**, or even model evidence, **model evidence**.

Likelihood The first term on the right, $P(X \mid \theta)$, called likelihood (**likelihood**) and determines the probability of occurrence of observations $P(X)$ given a parameter θ.
It is probably the most sensitive point in modeling, as it describes how the relationship between theoretical model and observations takes place. As discussed before, equations correspond to precise laws involving more than one construct. The mapping between $ P (X) $ observations and a parameter is given by the *likelihood function* (**likelihood function**) chosen, $f(\theta)$.

Example: the number of immune fighting cells circulating in the blood is associated with an inflammatory response. The higher, the more likely an infection is to the doctor. But which law associates the number of cells (between 0 and 10^5) with the likelihood of infection?
If the outcomes studied are binary ($y_i \in \{0, 1\}$, e.g. positive or negative diagnosis), we can use a logistic relationship (see Chapter 4) to estimate probabilities as a function of observed variables (X) and parameter(s) θ.

$$P(X \mid \theta) \sim f(X, \theta) : y_i = \frac{1}{1 + e^{-(\theta * x_i + c)}}$$

Other functions could be chosen (e.g. Heaviside step from the previous chapter). This depends on the phenomenon, the theory and the measures analyzed.

Priors How do we estimate the chances of infection before seeing the test results? Before the exam, we have some notion of how the parameter behaves. It can be very precise or bring a lot of uncertainty. We call the baseline estimate $P(\theta)$ **prior** and appears in the expression multiplying the likelihood value. In the language of probabilities, it is a distribution. Our previous beliefs may be uninformative (e.g. we do not examine the patient; uniform distribution over possible values) or quite defined (e.g. the patient is asymptomatic; distribution concentrated in the vicinity of 0).

```
> a <- runif(10000)
> b <- runif(10000, min = 0, max=0.2)
> priors <- data.frame(uniform=a, low=b)
> ggplot(priors)+
geom_density(aes(x=uniform),color="#F0E442")+
geom_jitter(aes(y=uniform*4.5,x=seq(0,0.2,length.out = 10000)),
color="#009E73",alpha=0.015)+
geom_density(aes(x=low),color="#009E73")+
geom_jitter(aes(y=low*3,x=seq(0,1,length.out = 10000)),
color="#F0E442",alpha=0.01)+ylab("Density")+
xlab("Priors: informative (green) or uncertain (yellow)")+
theme_hc(style="darkunica")+theme(axis.text.y=element_blank())
```

Knowing our constructs, we can then rewrite the procedures:

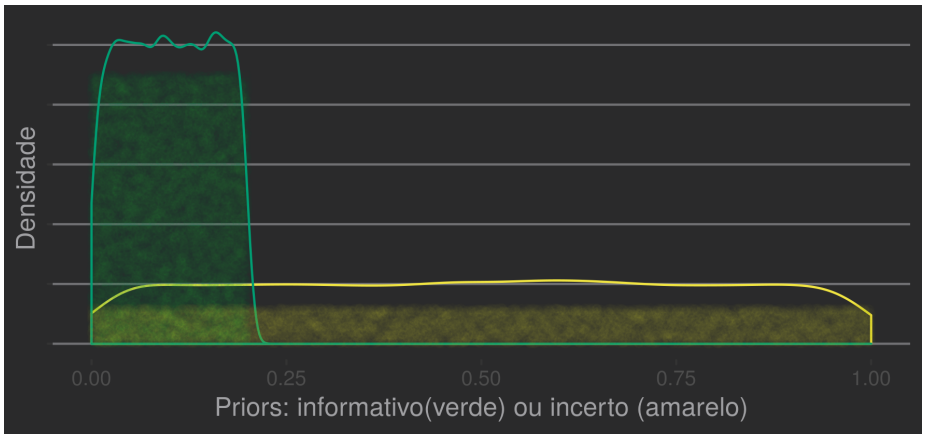


Figure 27: The green prior assumes greater probabilities for low values. The yellow prior is not very informative, assigning similar probabilities throughout the range

$$\text{Posterior} = \frac{\text{Prob. of observations given by} f(X, \theta) * \text{Prior}}{\text{Prob. marginal for observations}}$$

To obtain the *posterior*, we multiply the probability given by the *likelihood function* by our previous estimates (*prior*) and normalize by the *marginal probability* of the observations.

Later narratives are constructed according to the distribution of *posterior*.

Master Foo and the Recruiter[18]

A technical recruiter, upon discovering that the paths of Unix hackers were strange, sought to talk to Master Foo to learn about the Path. Master Foo met the recruit in the human resources offices of a large corporation.

The recruit said, "I have noticed that Unix hackers scorn or get nervous when I ask them how many years of experience they have in a new programming language. Why does it happen?"

Master Foo got up and started walking through the office. The recruiter was intrigued, and asked "What are you doing"?

"I'm learning to walk", replied Mestre Foo.

"I saw you walking through the door," the recruiter exclaimed, "and you're not tripping over your feet. Obviously, you know how to walk."

[18] http://www.catb.org/~esr/writings/unix-koans/recruiter.html

"Yes, but this floor is new to me" replied Master Foo.

Upon hearing this, the recruiter was enlightened.

Dear Stan

The implementations of the Bayesian models are made in Stan, a C ++ package specialized in Bayesian inference. The models are written in their own dialect, but the syntax is very similar to that of mathematical notation, so the translation of the chapter analyzes is straightforward. We specify the model in an auxiliary file of extension *.stan*, which is manipulated by R packages for visualization and other utilities.

There and back again

We will reproduce in the Bayesian way two known examples: difference between means (analogous to the t test) and correlation.

Here, it is clear that the rationale is more direct than the previous one.

Comparing samples of normal distribution Let us remember (chap. 1) that, to compare samples using the t test: (1) we assume normality in the data source; (2) we imagine the distribution of means normalized by standard error in similar hypothetical samples, taken from the same population; (3) we calculate the p-value by knowing the distribution (Student's t).

We can now obtain a later distribution for the difference between samples. (1) We assume normality in the data source (likelihood function); (2) we provide our prior estimates (prior); (3) we update the values against the data and to obtain the latter.

We adopted the following parameterization:

Values observed in samples 1 and 2, vectors N dimensions: y_1, y_2
Unknown target parameters, the means in each sample and a common standard deviation: μ_1, μ_2, σ
Priors assuming an average of 0 in both groups and a standard deviation of 1: $\mu_1 \sim N(0,1), \mu_2 \sim N(0,1), \sigma \sim N(1,1)$ Likelihood function, indicating that each observation is from a population with normal distribution: $y \sim N(\mu, \sigma)$

We also specify for Stan that it generates (1) values for the difference between the subsequent distributions of μ_1 and μ_2, μ_{diff} and (2) effect size with Cohen's D, dividing the value by the standard deviation.

The code must be saved in a ".stan" file.

```
data {
  int<lower=0> N;
```

```
  vector[N] y_1;
  vector[N] y_2;
}
parameters {
  real mu_1;
  real mu_2;
  real sigma;
}
model {
  //priors
  mu_1 ~ normal(0, 1);
  mu_2 ~ normal(0, 1);
  sigma ~ normal(1, 1);

  //likelihood - Likelihood
  for (n in 1:N){
    y_1[n] ~ normal(mu_1, sigma);
    y_2[n] ~ normal(mu_2, sigma);
  }
}
generated quantities{
  real mudiff;
  real cohenD;

  mudiff = mu_1 - mu_2;
  cohenD = mudiff/sigma;
}
```

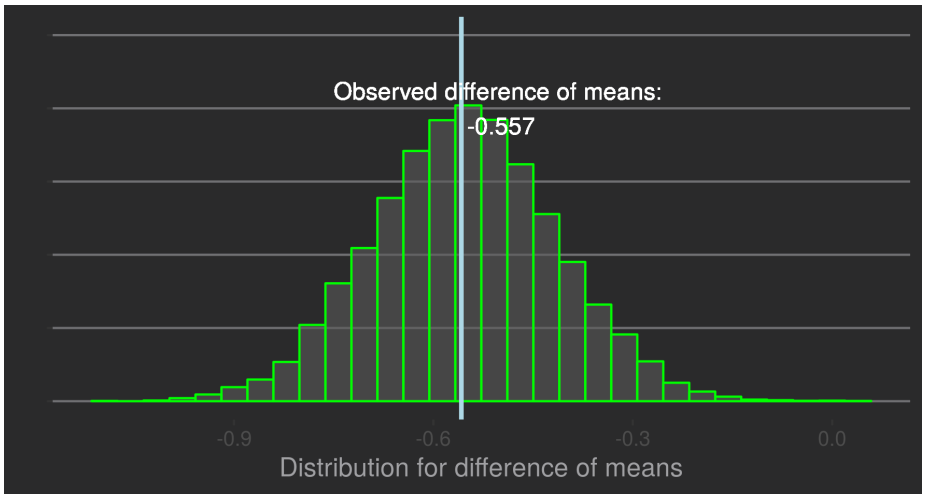
So, let's start the analysis through the interface in R. We created a list with components homonymous to the variables of the Stan file (y_1: sample 1, y_2: sample 2, N: sample size).

```
> a <- rnorm(n = 100, sd = 1, mean = 0)
> b <- rnorm(n = 100, sd = 1, mean = 0.6)
> sample_data <- list(y_1=a,y_2=b,N=length(a))
> fit <- rstan::stan(file="aux/bayes-t.stan",
        data=sample_data,
        iter=3000, warmup=100, chains = 6)
 SAMPLING FOR MODEL 'bayes-t' NOW (CHAIN 1).
 (...)
```

The above command will start the calculations. We will plot the later distributions of μ_1, $\mu_2$and the difference between these (μ_{diff})

```
> obs_diff <- mean(a) - mean(b)
> obs_diff
[1] -0.5579295
```

```
> posteriors <- extract(fit,par = c("mu_1","mu_2","mudiff"))
> lapply(posteriors,mean)
$mu_1
[1] 0.07303457

$mu_2
[1] 0.6261336

$mudiff
[1] -0.553099
> ggplot(data.frame(muDiff=posteriors$mudiff), aes(x=muDiff))+
   geom_histogram(alpha=0.6,color="green")+
   geom_vline(xintercept=obs_diff,
              color="light blue",size=1)+ # line for observed difference
   xlab("Distribuição para diferença de médias")+ylab("")+ ylim(0,2500)+
   geom_text(label="Diferença observada:\n -0.557",
              color="white",x=mean(muDiff)+0.05,y=2000)+
   theme_hc(style="darkunica")+
   theme(axis.text.y=element_blank())
```



The above distribution contains other information. We lost Student's elegant analytical estimate to test the hypothesis about a parameter (e.g.$H_0 : \mu_{\text{diff}} = 0$). On the other hand, we have a global view of the entire estimated distribution for μ_{diff}!

Linear correlation We will reproduce the correlation analysis in Chapter 2, when we talk about health indicators. The important variables are the logarithm

of the number of doctors and the healthy life expectancy (Health Adjusted Life Expectancy). The bank was created with name `uni_df`, containing the variables `log_docs` and `hale`.

Systematizing our approach, we will choose **Priors:** *Correlation* ρ: Let's assume that it is positive between the number of doctors and healthy life expectancy. We will indicate a low value (0.1) for this correlation.

$$N(0.1, 1)$$

Averages and deviations μ and σ: We don't have much of an average idea for the logarithm of the number of doctors. A slight inspection shows that the values are low in magnitude. We will indicate uninformative priors for $\mu_{\text{medicos}}, \sigma_{\text{medicos}}$ in the form of Gaussians with a mean of 0 and high deviations.

$$\mu_{\text{medicos}} \sim N(0, 2), \sigma_{\text{medicos}} \sim N(0, 10)$$

A brief search on search engines suggests that an average μ_{hale} * 60 * years is a reasonable guess. Let's estimate the prior of the standard deviation σ_{hale} in 5.

$$\mu_{\text{hale}} \sim N(60, 3), \sigma_{\text{hale}} \sim N(5, 2)$$

Likelihood function: Our model for the data is that it is given through a normal bivariate distribution, with averages μ_1, μ_2 and deviations σ_1, σ_2. As we saw earlier, the definition for Pearson's coefficient between samples X and X' is

$$\rho_{XX'} = \frac{cov(X, X')}{\sigma_X \sigma_{X'}}$$

So,

$$cov(X, X') = \sigma_X \sigma_{X'} * \rho_{XX'}$$

We can then define the covariance matrix of our bivariate distribution:

$$\text{Cov. Matrix} = \begin{pmatrix} \sigma_1^2 & \sigma_1 \sigma_{2'} * \rho \\ \sigma_1 \sigma_{2'} * \rho & \sigma_2^2 \end{pmatrix}$$

Our code in Stan:

```
data {
    int<lower=1> N;
    vector[2] x[N];
}

parameters {
```

```
    vector[2] mu;
    real<lower=0> sigma[2];
    real<lower=-1, upper=1> rho;
}

transformed parameters {
    // Matriz de covariancias
    cov_matrix[2] cov = [[      sigma[1] ^ 2    , sigma[1] * sigma[2] * rho],
                         [sigma[1] * sigma[2] * rho,     sigma[2] ^ 2    ]];
}

model {
  // Priors
  sigma ~ normal(0,1);
  mu ~ normal(0.2, 1);

  // Likelihood - Bivariate normal
  x ~ multi_normal_lpdf(mu, cov);

}

generated quantities {
  // Samples with ordered pairs
  vector[2] x_rand;
  x_rand = multi_normal_rng(mu, cov);
}
```

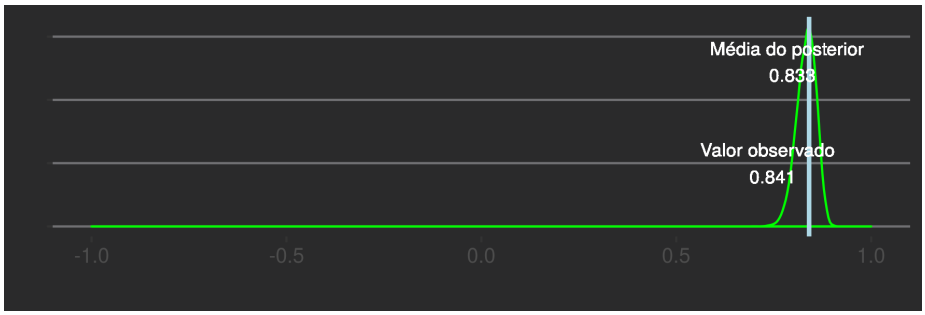
And then we can start the estimates.

```
    # Stan doesn't accept missing values
    > c_cases <- uni_df[complete.cases(uni_df[,c(3,4)]),]
    > vec_2 <- matrix(data = c(c_cases$hale,c_cases$log_docs),ncol = 2,nrow = 145)
    > health_data  <- list(N=nrow(c_cases),x = vec_2)
    > fit <- rstan::stan(file="aux/corr-docs.stan",
            data=health_data,
            iter=3000, warmup=120, chains = 6)
    SAMPLING FOR MODEL 'corr-docs' NOW (CHAIN 1).
    (...)
```

And then, let's look at our later estimate for the value of ρ:

```
    > obs_rho <- cor.test(vec_2[,1],vec_2[,2])$estimate
    > posterior <- rstan::extract(fit,par = c("rho"))
    > ggplot(data.frame(rho=posterior$rho), aes(x=rho))+
       geom_density(alpha=0.6,color="green")+
       geom_vline(xintercept=obs_rho,
                  color="light blue",size=1)+ # line for observed difference
```

```
    xlab("")+ylab("")+ xlim(-1,1)+
    geom_text(label="Valor observado \n 0.841",
              color="white",x=obs_rho-0.1, y = 5,
              size=3)+
    geom_text(label="Média do posterior \n 0.833",
              color="white",x=obs_rho-0.05, y = 13,
              size=3)+
    theme_hc(style="darkunic")+
    theme(axis.text.y=element_blank())
```



We note that the later estimates for ρ were reasonably distributed around the empirically calculated value in the sample. We can also observe in the distribution intervals with high probability density (HDI, High density intervals) or other purposes.

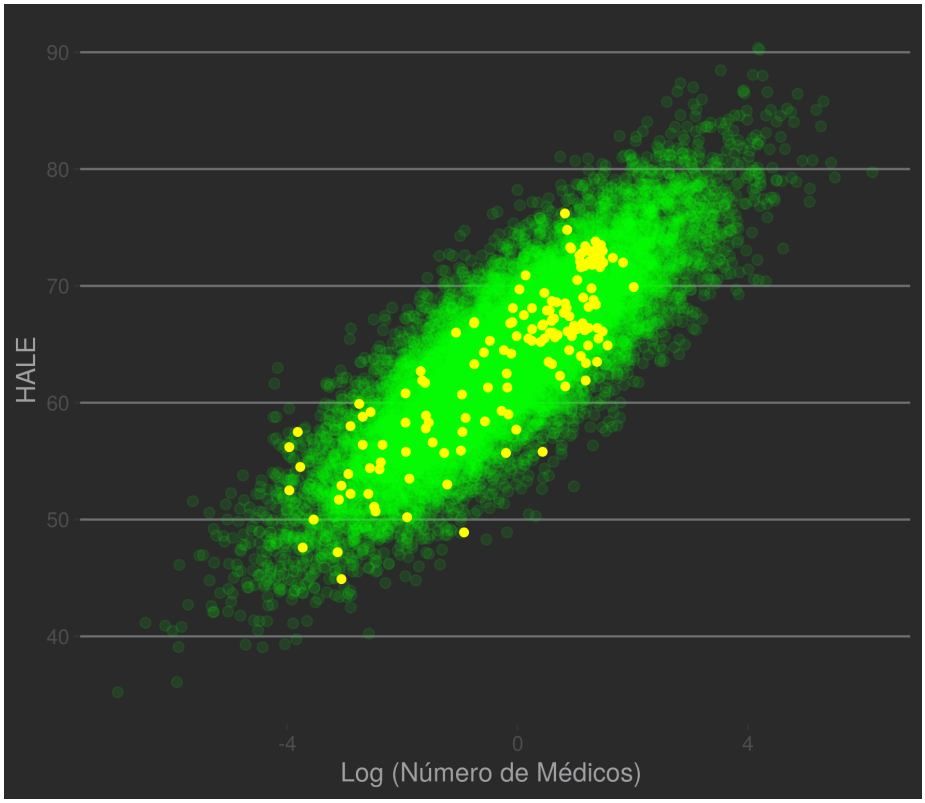
```
> quantile(posterior$rho,probs = c(0.025,0.5,0.975))
     2.5%       50%     97.5%
0.7790645 0.8353651 0.8777544
> cor.test(vec_2[,1],vec_2[,2])$conf.int
[1] 0.7854248 0.8828027
```

HDI is often close to the confidence interval as traditionally calculated, but this is not guaranteed.

We can plot our random sample generated from the later and visually inspect how the sample values would be within the estimated probability.

```
>x.rand = extract(fit, c("x_rand"))[[1]]
>plot(uni_df[,c("log_docs","hale")],
    xlim=c(-5,5), ylim=c(20, 100), pch=16)
>dataEllipse(x.rand, levels = c(0.75,0.95,0.99),
            fill=T, plot.points = FALSE)
> sample_data <- data.frame(x.rand)
> names(sample_data) <- c("HALE","Logdocs")
> ggplot(sample_data,aes(x=Logdocs,y=HALE))+
     geom_point(alpha=0.1,color="green",size=2)+
     xlab("Log (Number of doctors) ") + ylab("HALE")+
```

```
        geom_point(data=uni_df,aes(x=log_docs,y=hale),color="yellow")+
              theme_hc(style="darkunica")
```



You can experiment with different priors (families and parameters) watching how the final value changes.

Markov Chain Monte Carlo Estimators and Methods

In the above implementations, we start from the equation involving priors, likelihood and marginal probabilities.

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)}, P(X) \neq 0$$

Using Stan, we inform priors, the likelihood function, observations and all the dirty work is done without further effort. The estimate of $P(\theta \mid X)$ can be done in different ways. One involves starting from a distribution $P(\kappa)$ and gradually minimize a measure of the difference (in general, the * Kullback-Leibler divergence *) between it and* $P(\theta \mid X)$. *Esses métodos (cálculo variacional,*

Variational Bayesian methods*) involve analytical solutions for each model. We will address another method: **Markov Chain Monte Carlo**.

Not everyone who walks aimlessly is lost [19]

Closed solutions

When we talk about regression (Chap. 2), we estimate the straight slopes β_i. We used a *likelihood function* *, with the same meaning used here, defining the probability of the observations given a theoretical model.

We obtained solutions that maximized this function *(maximum likelihood)*. For the case of linear regression, we point out closed solutions

$$\text{Max log likelihood}(\beta_0, \beta_1, \sigma^2)$$

$$= \text{Max log} \prod_{i=1}^{n} P(y_i|x_i; \beta_0, \beta_1, \sigma^2)$$

For example, the slope (β_1) is

$$\hat{\beta_1} = \frac{cov(XY)}{\sigma_x^2}$$

Gradient Descent

In chapter 4, we show another way of estimating parameters, analyzing a loss function. Using partial derivatives, we calculate the gradient, analogous to the *slope* of a surface in 3 dimensions. This was possible because we knew the derivatives in each node (neuron). The network consists of sequencing units in layers, so the chain rule works perfectly (*backpropagation*).

$$(g \circ f)' = (g' \circ f)f'$$

Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) estimators work to treat problems with no closed solution and in which we don't know gradients exactly. Other forms of treatment exist. Here we cover an MCMC strategy called Metropolis-Hastings. To estimate our later, $P(\theta \mid X)$, we use an algorithm that allows us to obtain representative samples of $P(\theta \mid X)$. For this, the condition is that there is a function $f(x)$ proportional to the density of $P(\theta \mid X)$ and that we can calculate it.

[19] All that is gold does not glitter,/*Not all those who wander are lost*; The old that is strong does not wither,/ Deep roots are not reached by the frost./From the ashes, a fire shall be woken,/A light from the shadows shall spring;/Renewed shall be blade that was broken,/The crownless again shall be king. **J.R.R Tolkien. The Fellowship of the ring 1954,**

1 - We start with parameters in a state (e.g. $s_0 : \beta_0 = 0.1, \beta_1 = 0.2$) and analyze the function (e.g. f : log likelihood function) in that state ($f(s_0)$) considering the parameters in s_0. 2 - Next, we take a step in a random direction, changing the values of β_i. A widely used option is that of a Gaussian with a center in the previous state (* random walk *). We reassessed the state ($f(s_1)$).
2.1 - If it is more likely, $f(s_1) > f(s_0)$, so s_1 it is accepted as a new starting point. 2.2 - If it is less likely, but close enough to the previous state, $f(s_1) - f(s_0) < \epsilon$, we also take s_1 as a starting point for the next random step. 2.3 - If he is less likely with a large margin, $f(s_1) - f(s_0) > \epsilon$, we reject s_1 and raffled off a new random state.

The process moves towards more likely states, with some likelihood of visiting less likely states. If the function chosen is proportional to the density of the posterior, $f(x) \sim \text{dens}(P(\theta \mid X))$, the frequency of parameters in the sample of states visited, s_i, correspond to the latter. It is a common practice to discard the first iterations (*warm up*), as the values can be very representative of places with low density.

Equations For practical purposes, we will work with an unknown parameter $\sigma^2 = 1$.

The function f proportional must be proportional to the density of the posterior.

$$\text{Posterior} \propto \frac{\text{Prior} \times \text{Likelihood}}{\text{Prob. Marginal}}$$

Marginal odds It is the probability of the observations $P(X)$. They are constant in the process, serving only to normalize estimates, so:

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood}$$

Priors
Our prior is normal, with a mean of 0 and standard deviation 1, $P(\mu) \sim N(0, 1)$.

Likelihood If the observations are independent, we only need to multiply the probability of each one. We assume that the distribution of measures is normal, with a mean μ and deviation σ^2. and deviation s_i, the likelihood of observations X considering the μ_i is:

$$P(X|\mu_i) =$$

$$\prod_{j=1}^{n} P(x_j|N(\mu_i, 1)) =$$

$$\prod_{j=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_j-\mu_i)}{2}}$$

Function proportional to posterior density Usaremos o log likelihood for the advantages described above: product becomes a sum and we pass the interval $[0;1]$ for $[-\infty, 0)$ (or $(0, +\infty]$ multiplying by -1).

$$\log(\text{Posterior}) \propto log(\text{Prior} \times \text{Likelihood})$$

$$f : L(s_i) = \log(P(X|\mu_i, 1) \times N(0,1))$$
$$\log(\prod_{j=1}^{n} P(x_j|N(\mu_i, 1)) \times N(0,1)) =$$
$$\log(\prod_{j=1}^{n} P(x_j|N(\mu_i, 1))) + \log(N(0,1)) =$$

The second term is a normal distribution with known mean and variance. We only need to use values transformed by logarithm. The first term is [20]:

$$\sum_{j=1}^{n} \log(P(x_j|N(\mu_i, 1))) =$$
$$= -\frac{n}{2}\log(2\pi\sigma_i^2) - \frac{1}{2\sigma_i^2}\sum_{j=1}^{n}(x_j - \mu_i)^2$$

Finally, we can calculate for each state a value for the parameters μ_i, σ_i, accept or reject them.

Implementation We will implement MCMC as a proof of concept to illustrate the convergence mechanism. For a real application with robust results, a few more efforts would be necessary. For example, the steps in our program will always be identical, the normalization of the values was done by hand for the sample and we use only one string to estimate the latter.

Stan uses a highly sophisticated version of MCMC, in which the evolution of the system is guided by a (Hamiltonian) function of the total energy. It is possible to observe a gradient and, as in physical phenomena, states with lower energy levels are more likely to be occupied (e.g. Boltzmann distribution in statistical mechanics).

[20] Deduction in https://www.statlect.com/fundamentals-of-statistics/normal-distribution-maximum-likelihood

Using the algorithm described above for the difference between means, we generate the samples `a` and `b`, $n = 400$, of populations with averages $\mu_a = 0, \mu_b = 0.6$, and normal distribution.

```
>set.seed(2600)

>n_obs <- 400
>a <- rnorm(n=n_obs, sd =1, mean=0)
>b <- rnorm(n=n_obs, sd=1, mean=0.6)
```

Let's define our likelihood function (using * -log * transformation):

```
>likel <- function(n,x,mu,sigma){
  l_val <- (-n/2)*log(2*pi*sigma^2) - (1/2*sigma^2)*sum((x - mu)^2)
  return(-l_val) # multiplica(-1)
}
```

Defining the role to provide $\log(N(0,1))$. We will obtain the probabilities and their logarithm for a n large, representative. This number will be normalized by the size of our sample to allow steps on a reasonable scale in the chain calculations.

```
>log_norm <- function(n,mu,sigma){
  require(magrittr) # para o operador %>%
  # Trick to get ~ uniform distribution in [-Inf,+Inf]
  unif_dist <- 1/runif(n = n, min = -1,max = 1)
  l_val <- dnorm(x=unif_dist,mean = 0,sd = 1, log=T)
  l_val <- car::recode(l_val,"-Inf:-1000=-1000") %>% sum # recod. extreme values
  return(-l_val)
}
```

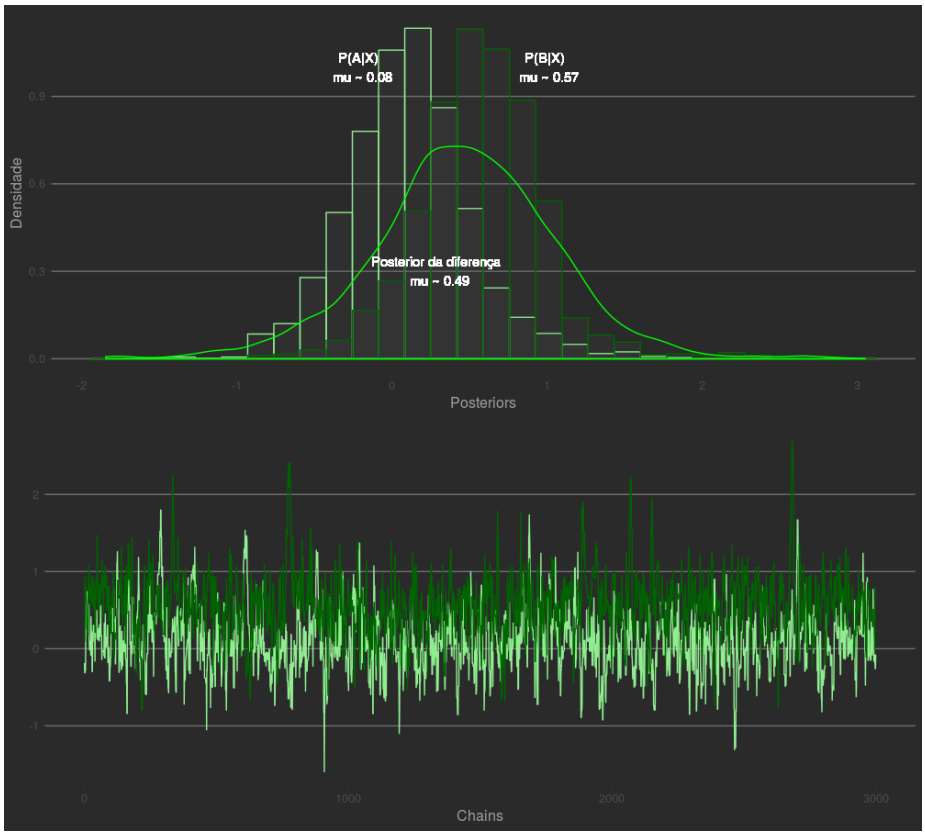
And a loop to run the MCMC simulation:

```
  # MCMC chain
  >mc_chain <- function(obs,iter=4000,n_obs=length(obs)){
    # seeds and objects
    sample <- matrix(nrow = iter, ncol = 2)
    s1_mu <- rnorm(n=1,mean=0) # initial mean
    s_sigma <- 1 # variancia = 1
    s1_lik <- 2000
    for (i in 1:iter){
      # Salva estado
      s0_mu <- s1_mu
      s0_lik <- s1_lik

      # Take a step (random walk)
      s1_mu <- s1_mu + rnorm(n=1,mean = 0, sd=0.5)
      s1_lik <- likel(n=n_obs,x=obs,mu=s1_mu,sigma=s_sigma) +
```

```
        # log do prior se baseian numa densidade de n=10000 e é normalizado por 1000
        log_norm(n=10000,mu=0,sigma=1)/1000

      # Rejects differences greater than 5, assuming the value in the previous state
      if(s1_lik - s0_lik > 5)
        s1_mu <- s0_mu
      sample[i,] <- c(s1_mu,s_sigma) # Save
    }
    return(sample[1001:iter,1]) # Discard the first 1000 samples (warm-up)
  }
```

We can then obtain our subsequent distributions for μ_A, μ_B and for the difference.
We will also visualize the evolution of states over time.

```
>posterior_a <- mc_chain(obs = a,iter = 4000)
>posterior_b <- mc_chain(obs = b,iter = 4000)
>posteriors_data <- data.frame(post_a=posterior_a, post_b=posterior_b)
>posts_plot <- ggplot(data = posteriors_data, aes(x=posterior_a)) +
   geom_histogram(aes(y=..density..),color = "light green", alpha=0.1) +
   geom_histogram(aes(x=posterior_b, y=..density..), alpha=0.1, color="dark green") +
   geom_density(aes(x=(posterior_b - posterior_a)), color="green") +
   xlab("Posteriors") + ylab("Densidade") +
   geom_text(label="P(A|X) \n mu ~ 0.08",color="white",x=-0.2,y=1)+
   geom_text(label="P(B|X) \n mu ~ 0.57",color="white",x=1,y=1)+
   geom_text(label="Difference later \n mu ~ 0.49",color="white",x=0.3,y=0.3)+
   theme_hc(style = "darkunica")
>traces_plot <- ggplot(data=posteriors_data,
  aes(y=posterior_a,x=1:nrow(posteriors_data)))+
  geom_line(color="light green")+xlab("Chains")+ylab("")+
  geom_line(aes(y=posterior_b,x=1:nrow(posteriors_data)),
  color="dark green")+
  theme_hc(style="darkunica")
> multiplot(posts_plot,traces_plot,cols = 1)
```



The top panel of the view highlights later distributions of A (light green) and B (dark green), as well as the difference. They reflect reasonably well the distributions of origin $(N(0,1), N(0.6,1)$ inferred from the data. In the lower panel, we have the chains for A (lower average, with signal oscillating at a lower level) and B (higher average, with signal oscillating above). Although it is an illustrative model, the result looks good, with representative distributions.

Exercises

1. Using Stan, implement linear regression for data of your choice. The *likelihood function* for observations can be a Gaussian whose mean is by the regression equation. The user guide should help. https://mc-stan.org/docs/2_18/stan-users-guide/linear-regression.html
   - Implement linear regression with more than one predictor.
   - Compare the mean of the posterior ones for the coefficients β with the classic point estimate using `glm`.
2. With the library `BEST` conduct the comparison of means of the final example, invoking the function `BESTmcmc` and specify the argument `numSavedSteps = 3000`.

- Extract the subsequent distributions, `mu1` and`mu2`, from the resulting object.
- Get the difference between `mu1 - mu2` distributions and compare visually (density or histogram) with the posterior one that we generate through the handmade MCMC.

3. Improve the MCMC simulation by modifying the `mc_chain` function.
   - Obtain the final sample for the later one by drawing values generated by 4 independent chains.
   - Make the size of the steps decrease linearly with the number of simulations elapsed.