

MAP 2112 – Introdução à Lógica de Programação e Modelagem Computacional

1º Semestre - 2019

Prof. Dr. Luis Carlos de Castro Santos

lsantos@ime.usp.br/lccs13@yahoo.com

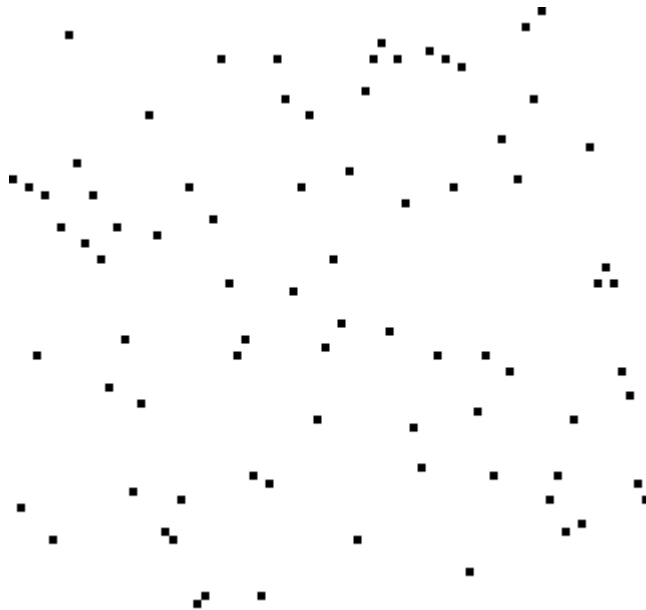
Objetivos

- Introduzir noções de programação e apresentar exercícios de modelagem computacional. Explorar o uso de Excel/Calc, e de Python ou R.

- **Algoritmo** é uma sequência finita de instruções bem definidas e não ambíguas, cada uma das quais devendo ser executadas mecânica ou eletronicamente em um intervalo de tempo finito e com uma quantidade de esforço finita.
- O conceito de algoritmo é frequentemente ilustrado pelo exemplo de uma receita culinária, embora muitos algoritmos sejam mais complexos. Eles podem repetir passos (fazer [iterações](#)) ou necessitar de decisões (tais como comparações ou [lógica](#)) até que a tarefa seja completada. Um algoritmo corretamente executado não irá resolver um problema se estiver implementado incorretamente ou se não for apropriado ao problema.
- Um algoritmo não representa, necessariamente, um [programa de computador](#), e sim os passos necessários para realizar uma tarefa. Sua implementação pode ser feita por um [computador](#), por outro tipo de [autômato](#) ou mesmo por um ser humano.

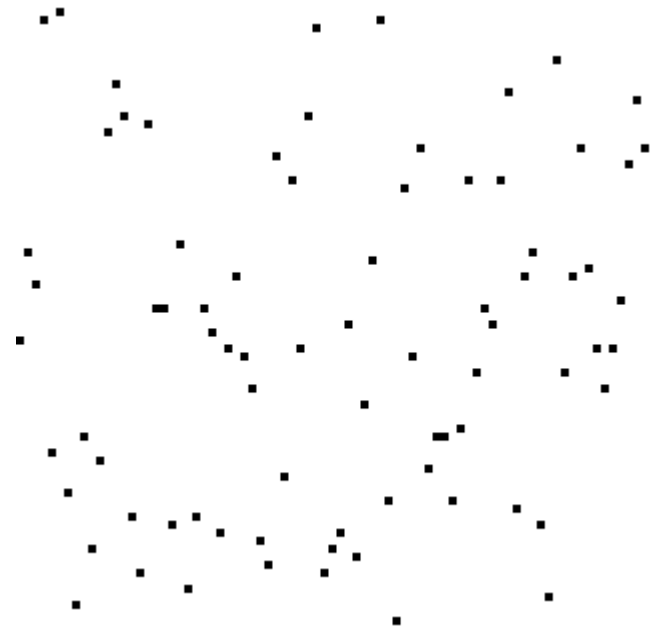
Problema: Dado um vetor de valores (aleatórios) ordene-os em ordem crescente.

Algoritmo Elementar de Ordenação



https://www.ime.usp.br/~pf/algoritmos/aulas/img/Insertion_Sort_Animation.gif

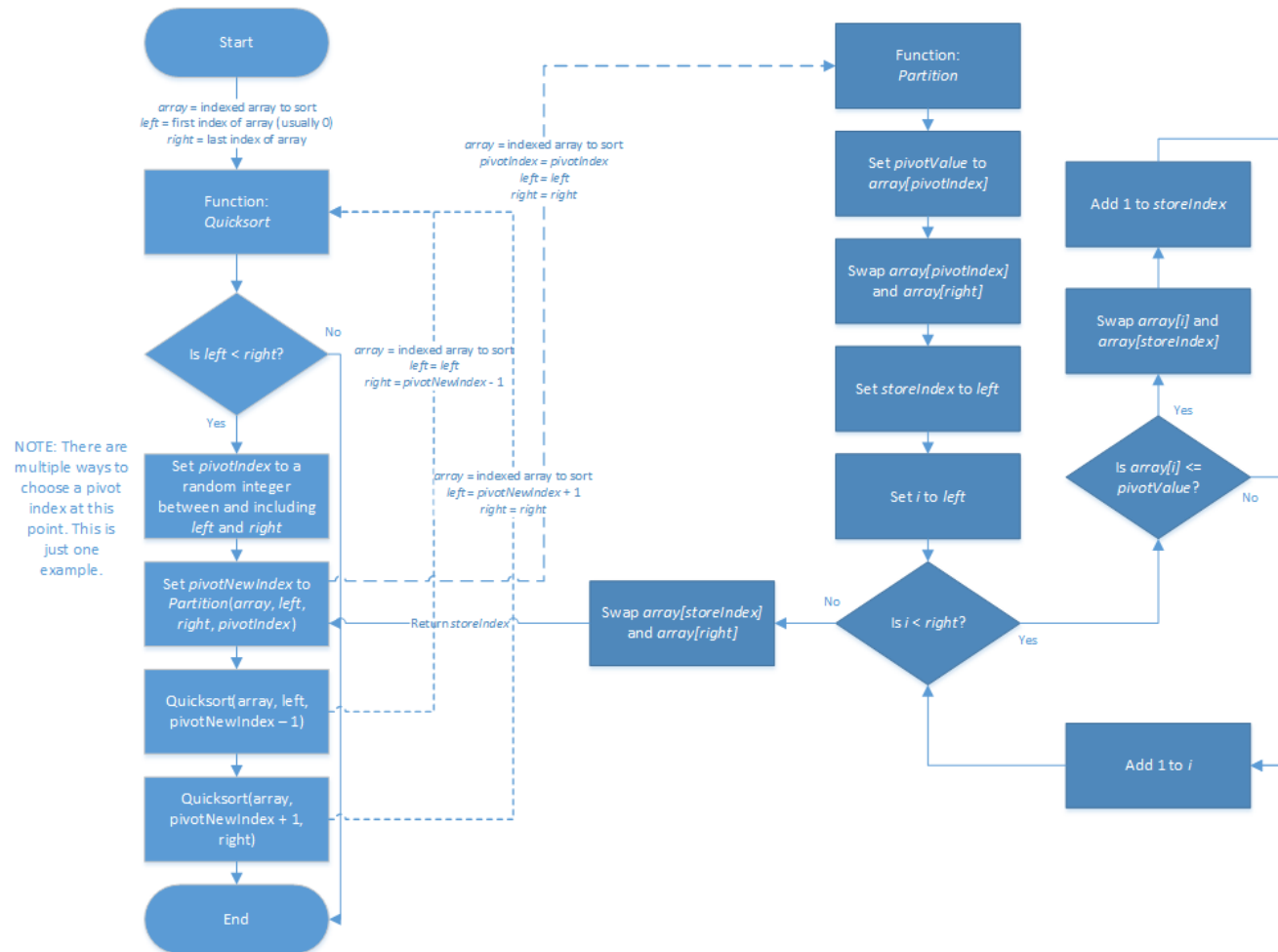
Algoritmo Quicksort



<https://www.ime.usp.br/~pf/algoritmos/aulas/img/Quicksort-github.gif>

Diferentes algoritmos podem produzir o mesmo resultado com desempenhos computacionais diferentes (complexidade)

Fluxograma



© 2013 Jeff Allen

Video instrutivo:

<https://www.youtube.com/watch?v=WP7KDIjG6IM>

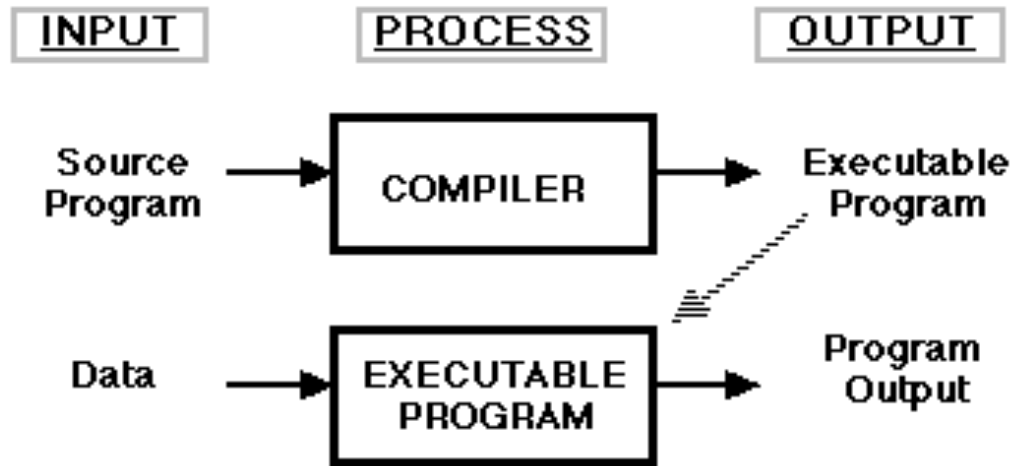
Code sample: Quicksort

Quicksort in C

```
qsort( a, lo, hi ) int a[], hi, lo;
{
    int h, l, p, t;
    if (lo < hi) {
        l = lo; h = hi; p = a[hi];
        do {
            while ((l < h) && (a[l] <= p))
                l = l+1;
            while ((h > l) && (a[h] >= p))
                h = h-1;
            if (l < h) {
                t = a[l]; a[l] = a[h]; a[h] = t;
            }
        } while (l < h);

        t = a[l]; a[l] = a[hi]; a[hi] = t;
        qsort( a, lo, l-1 );
        qsort( a, l+1, hi );
    }
}
```

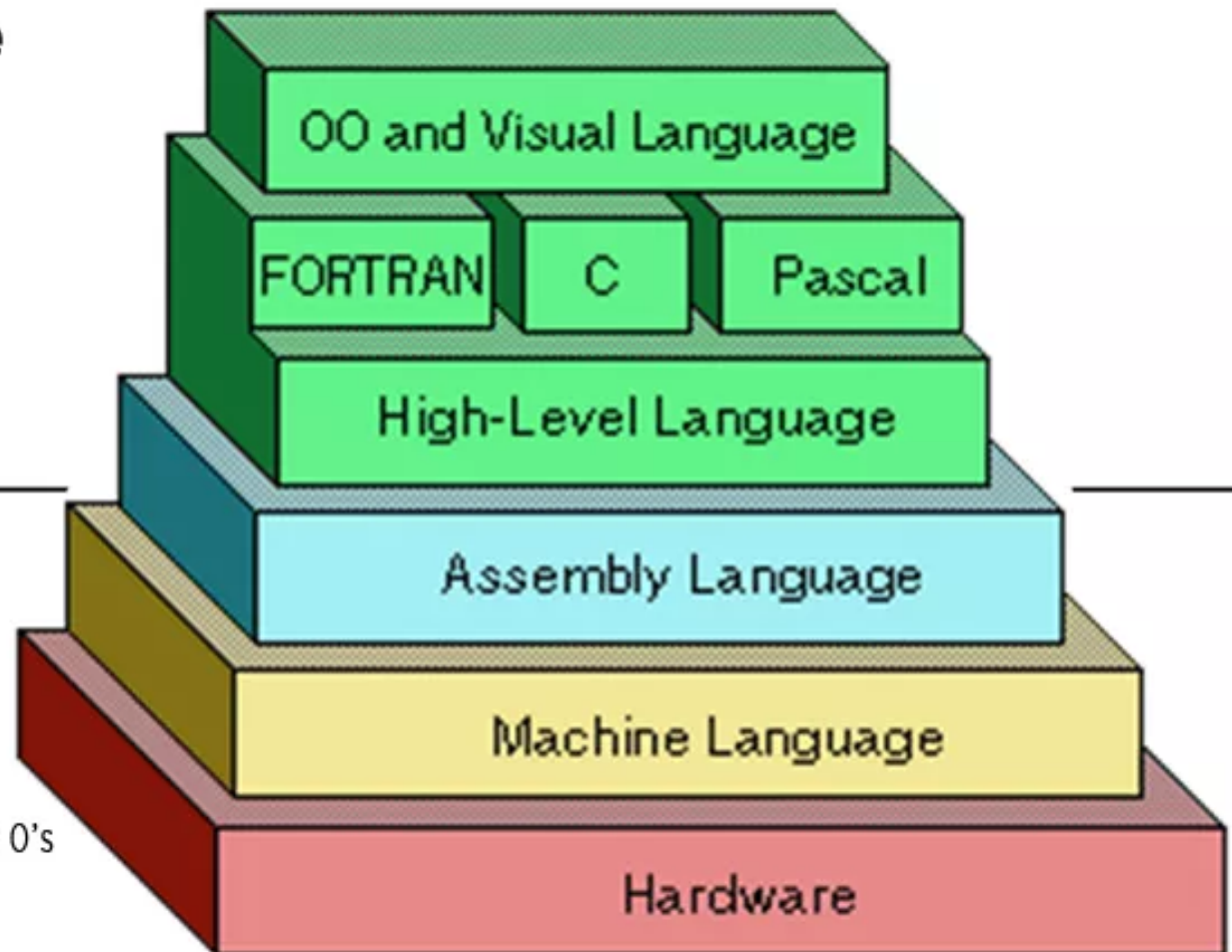
```
~/Desktop/helloworld.c - Sublime Text
helloworld.c
1 #include <stdio.h>
2
3 int main (void) {
4     printf ("Hello, World!\n");
5
6     return 0;
7 }
```



```
>HELLO WORLD!
>
```

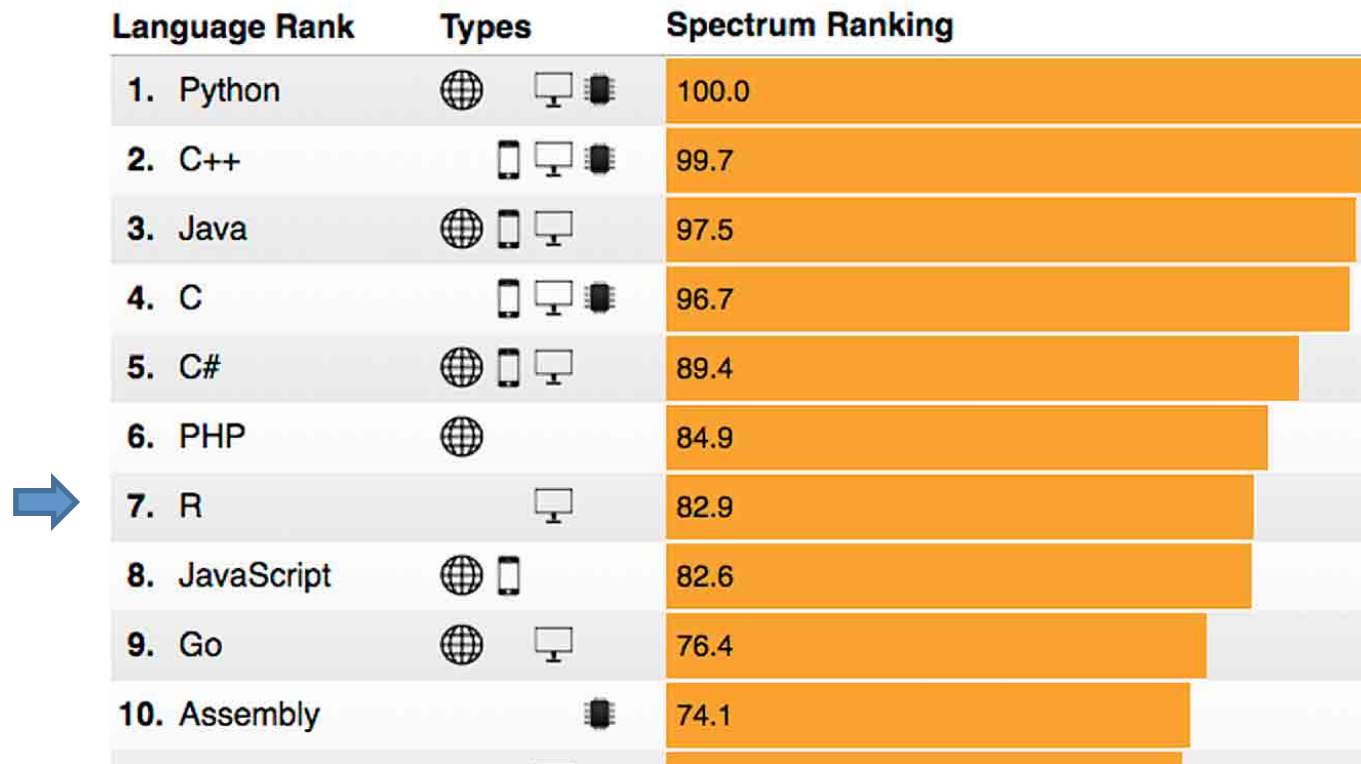
High Level Language

- Easy for Programmers to understand
- Contains English Words



Low Level Language

- The computer's own Language
- Binary numbers, in 1's and 0's

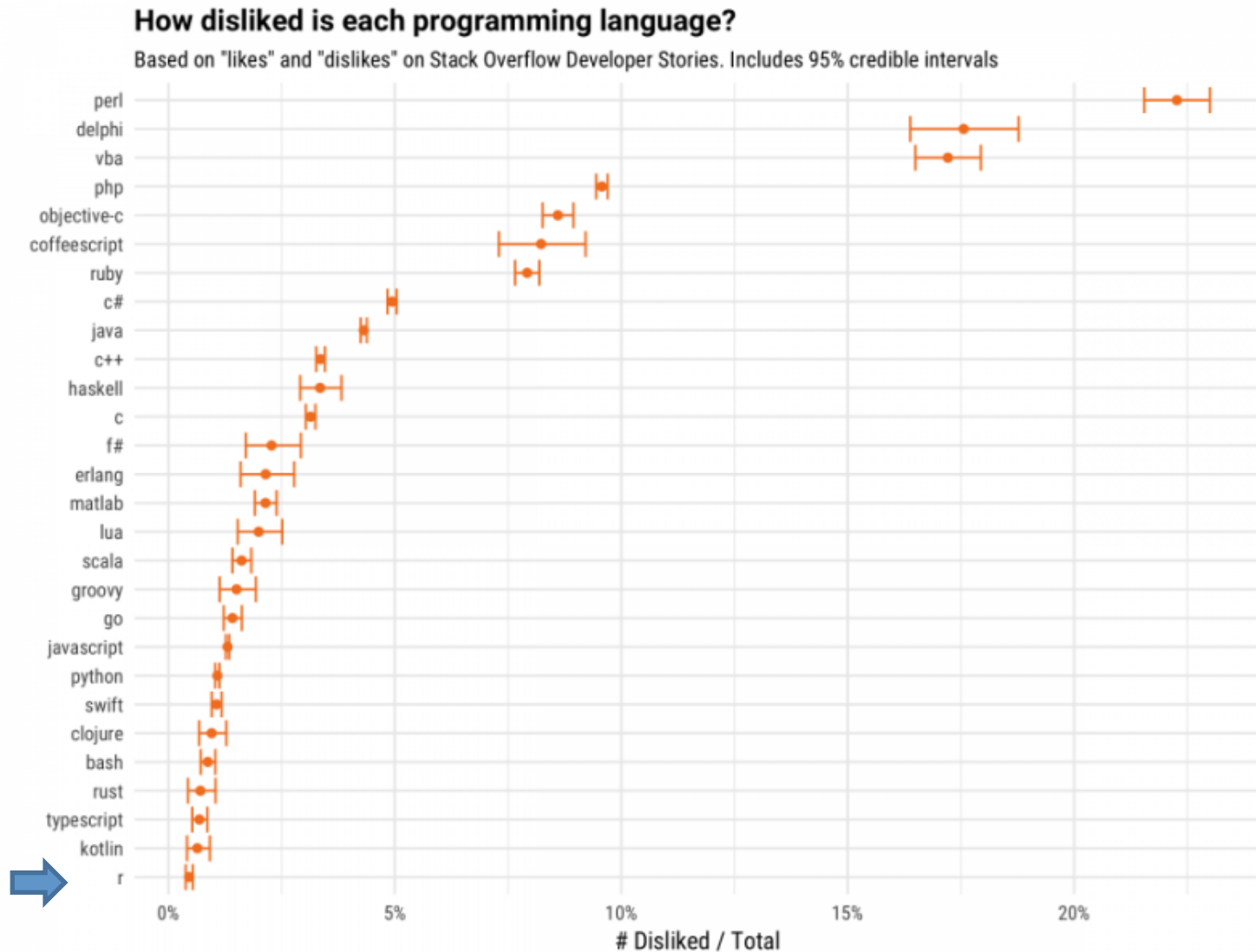


<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>

The PYPL Popularity of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

Worldwide, Feb 2019 compared to a year ago:			
Rank	Language	Share	Trend
1	Python	26.42 %	+5.2 %
2	Java	21.2 %	-1.3 %
3	Javascript	8.21 %	-0.3 %
4	C#	7.57 %	-0.5 %
5	PHP	7.34 %	-1.2 %
6	C/C++	6.23 %	-0.3 %
7	R	4.13 %	-0.1 %
8	Objective-C	3.04 %	-0.8 %
9	Swift	2.56 %	-0.6 %
10	Matlab	1.98 %	-0.4 %





<https://www.technotification.com/2018/05/programming-languages-disliked.html>



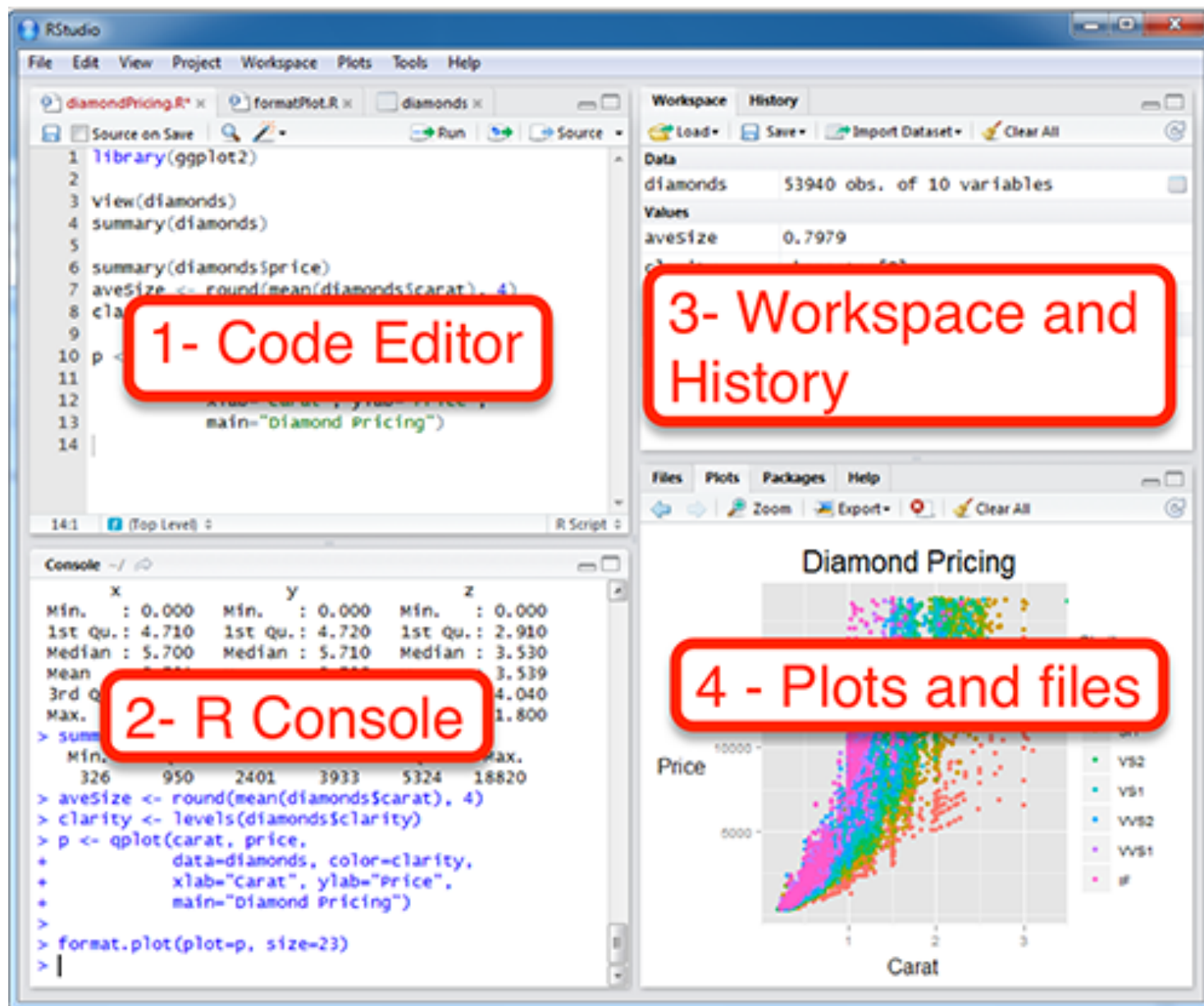


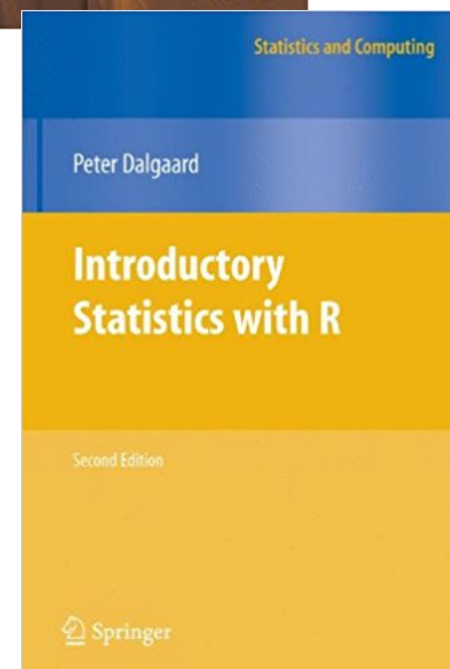
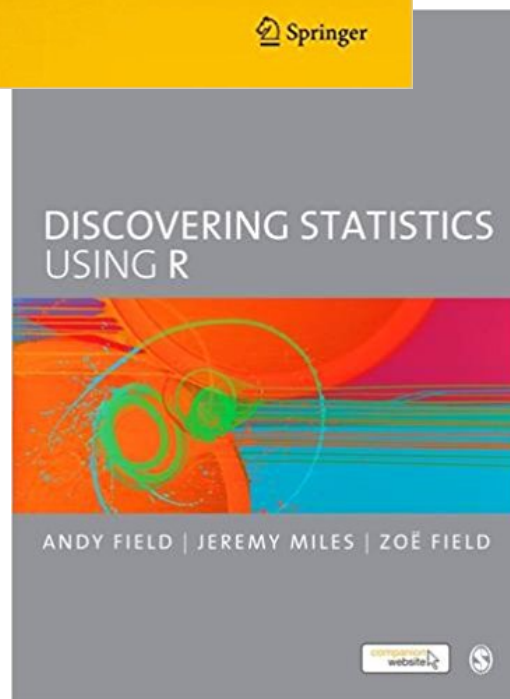
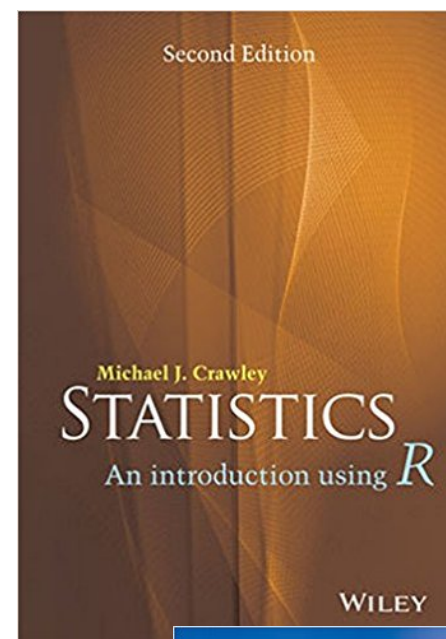
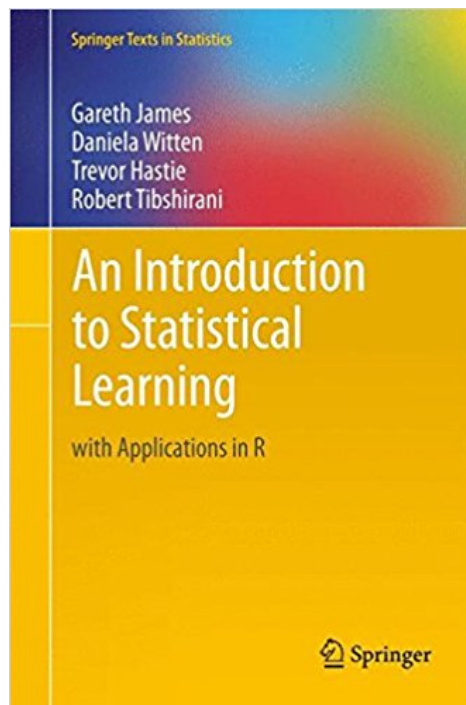
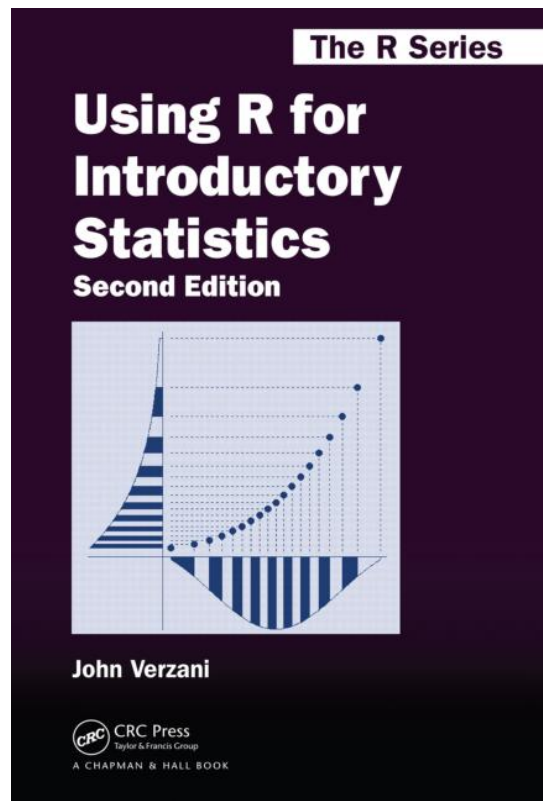
<https://cran.r-project.org/>

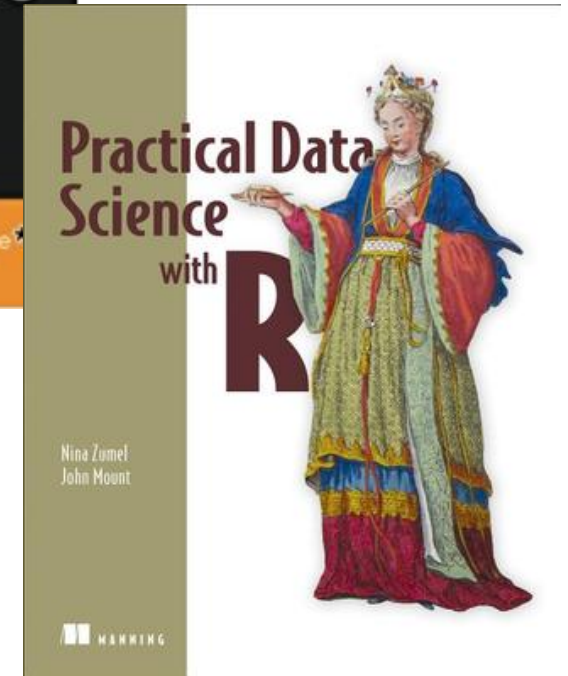
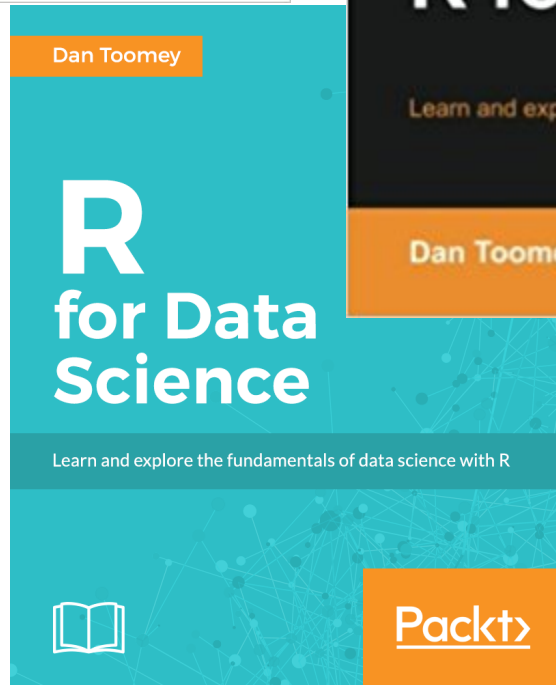
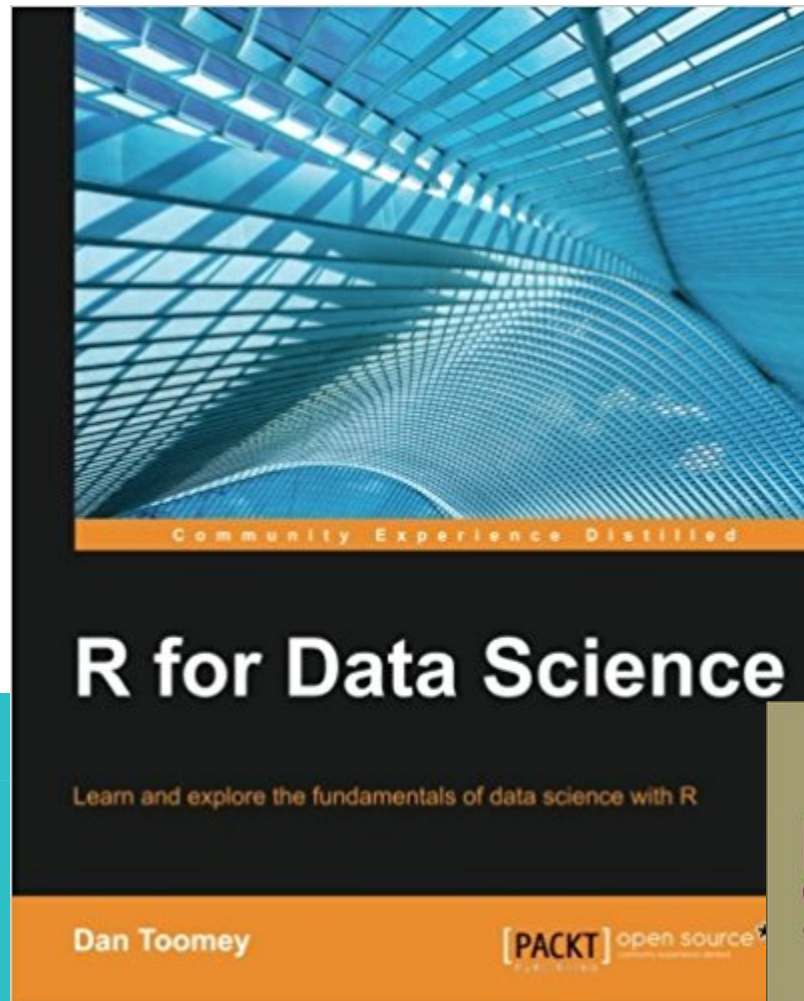
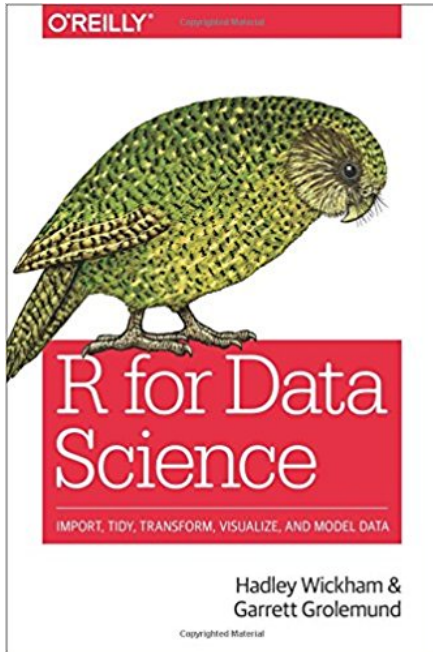
<https://www.r-project.org/>

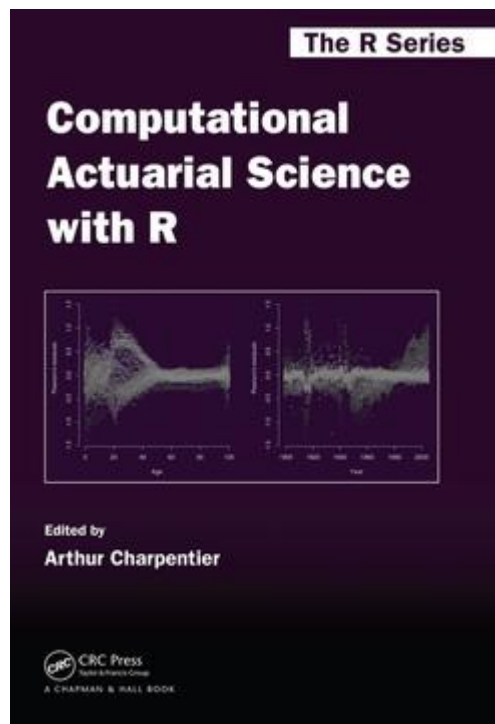
<https://www.rstudio.com/>











Machine Learning & Data Science for Actuaries, with R

Arthur Charpentier (Université de Rennes 1 & UQÀM)



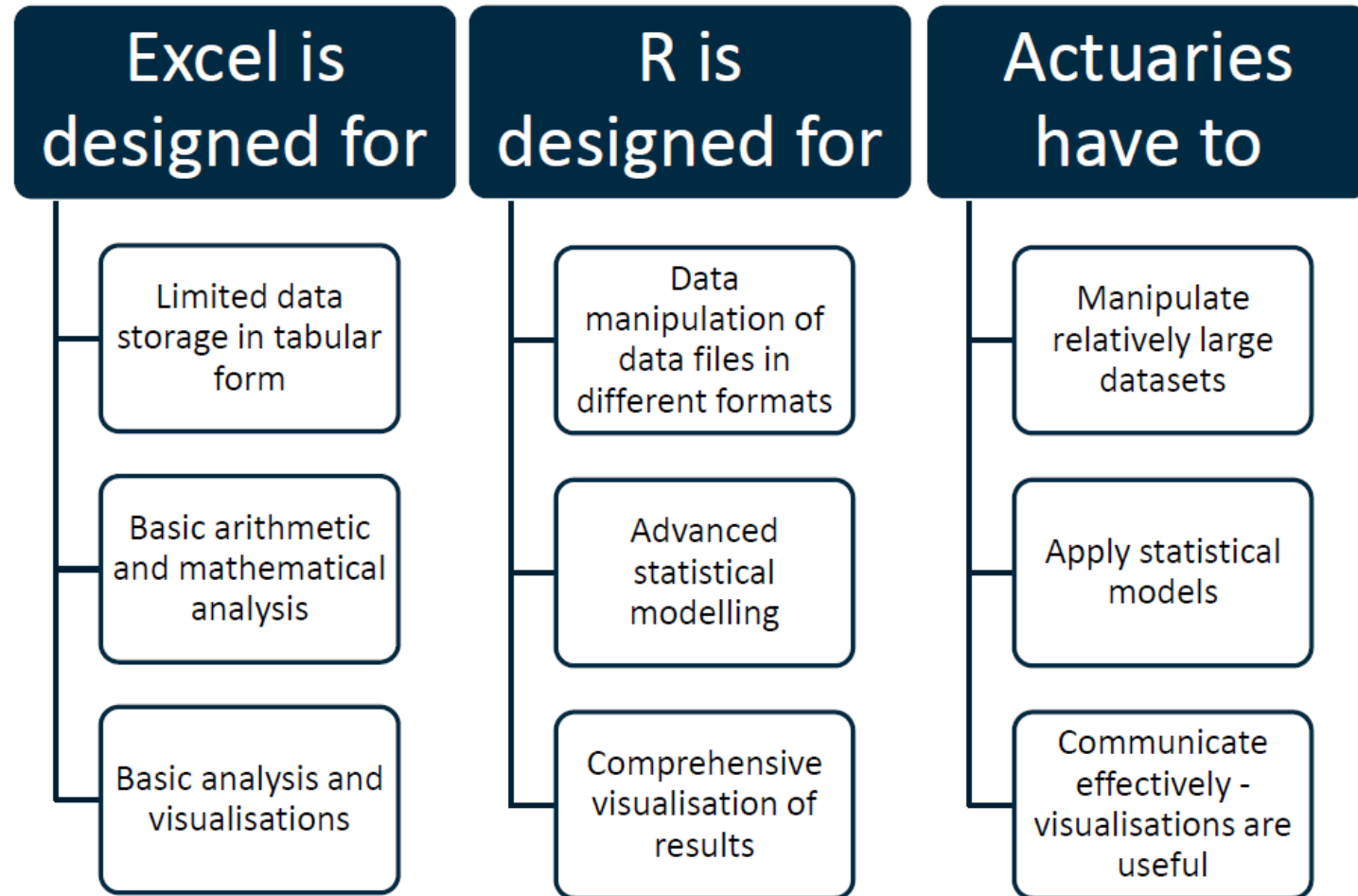
Universitat de Barcelona, April 2016.

<http://freakonometrics.hypotheses.org>



Society of Actuaries in Ireland

R For Actuaries: What, Why and Where?



MAP 2112 – Introdução à Lógica de Programação e Modelagem Computacional

1º Semestre - 2019

Programa

- Apresentação de linguagens existentes atuais (e.g. Excel/Calc, VBA, SAS, **R**, Python, C/C++). Introdução ao conceito de algoritmo. Operações com laços e funções. Introdução à estruturação de dados em tabelas. Técnicas de busca e ordenação de dados. Modelagem de problemas aplicados.

QUEM SÃO VOCÊS ?

Curso

12033 - Bacharelado em Ciências Contábeis – 2
12042 - Bacharelado em Ciências Atuariais - 59
(vazio) – (?) – 1

Ingresso

2012/1 – 1
2013/1 – 1
2014/1 – 1
2015/1 – 1
2017/1 – 4
2018/1 – 6
2019/1 – 48

Cálculo ?

Programação ? Excel ?

QUEM SOU EU ?

Luis Carlos de Castro Santos

É graduado em Engenharia Mecânica pela Universidade de Brasília (1986), mestre em Engenharia Aeronáutica pelo Instituto Tecnológico de Aeronáutica (1988) e doutor em Aerospace Engineering pelo Georgia Institute Of Technology (1993). Fez pós-doutorado no Institut für Aerodynamik und Strömungstechnik do DLR na Alemanha (1994-1995). Atualmente é professor doutor do Departamento de Matemática Aplicada da Universidade de São Paulo e supervisor de engenharia da área de CFD e Simulação de Sistemas da EMBRAER. Tem experiência profissional nas áreas de Mecânica dos Fluidos Computacional, Simulação de Sistemas Dinâmicos, Análise Numérica e Otimização.

<http://buscatextual.cnpq.br/buscatextual/busca.do>

Atividades Programadas

Aulas teóricas: terças-feiras de 19:30 às 21:10. Sala FEA A7.

Provas: Nos dias 9 de abril (P1); 21 de maio (P2) e 25 de junho (P3).

Entrega de Trabalho Computacional em grupo em 30 de junho (T1).

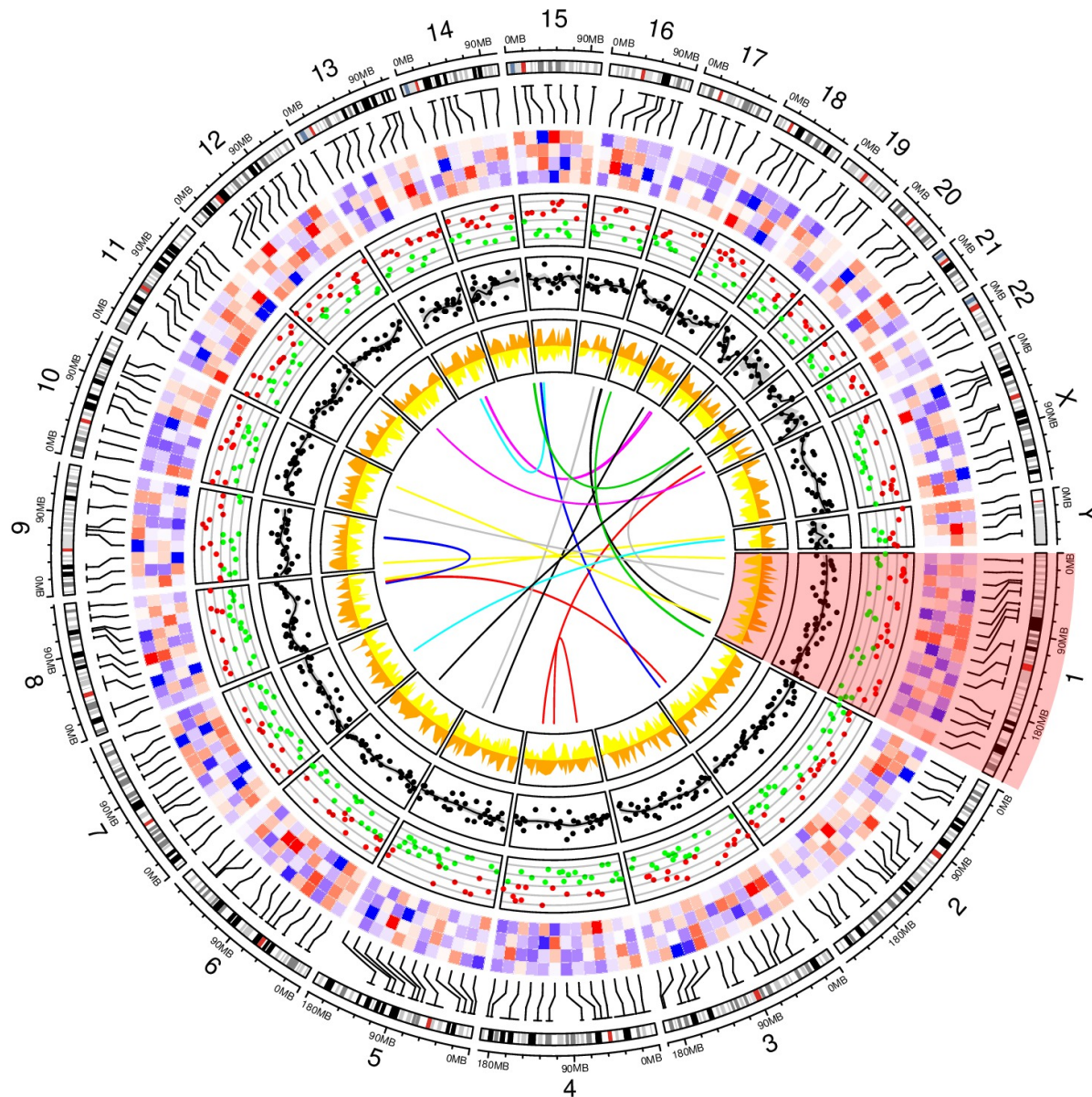
Não haverá sub.

Avaliação

Os alunos serão avaliados de acordo com:

$MF = 0.7 \cdot (MP) + 0.3 \cdot (MT)$, onde MP = media das duas melhores notas de prova e MT = notas de trabalho em grupo. A aprovação se dá com média 5.

As regras para o trabalho em grupo serão divulgadas oportunamente.



http://zuguang.de/circlize/example/genomic_general.html