

MAP 2112 – Introdução à Lógica de Programação e Modelagem Computacional

1º Semestre - 2020

Prof. Dr. Luis Carlos de Castro Santos

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

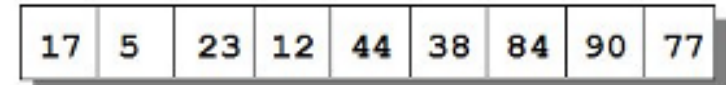
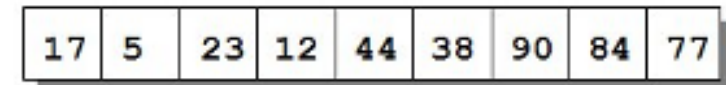
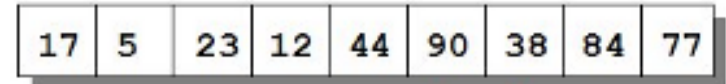
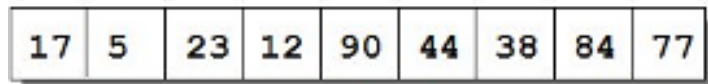
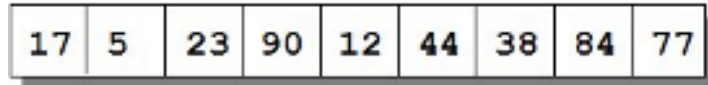
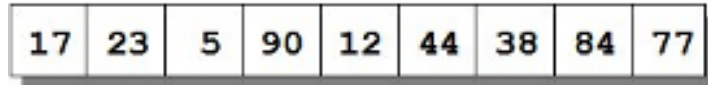
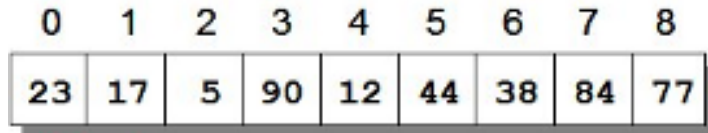
Objetivos

- 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.

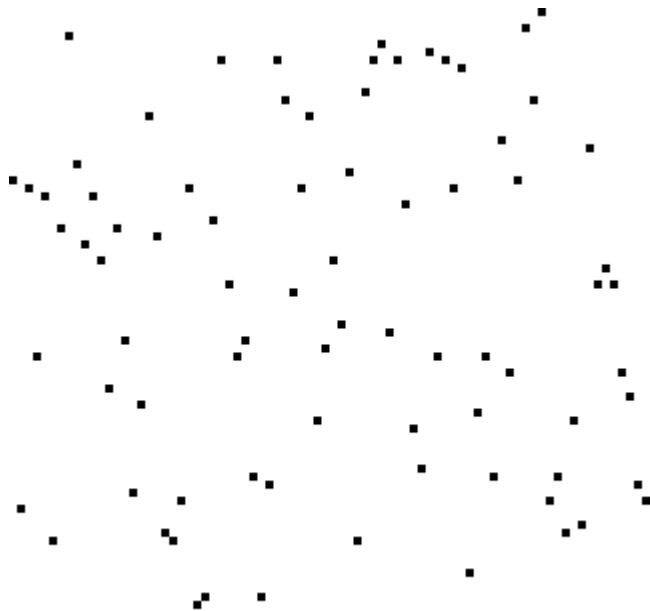
Problema: Dado um vetor de valores (aleatórios) ordene-os em ordem crescente.



Bubble sort

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

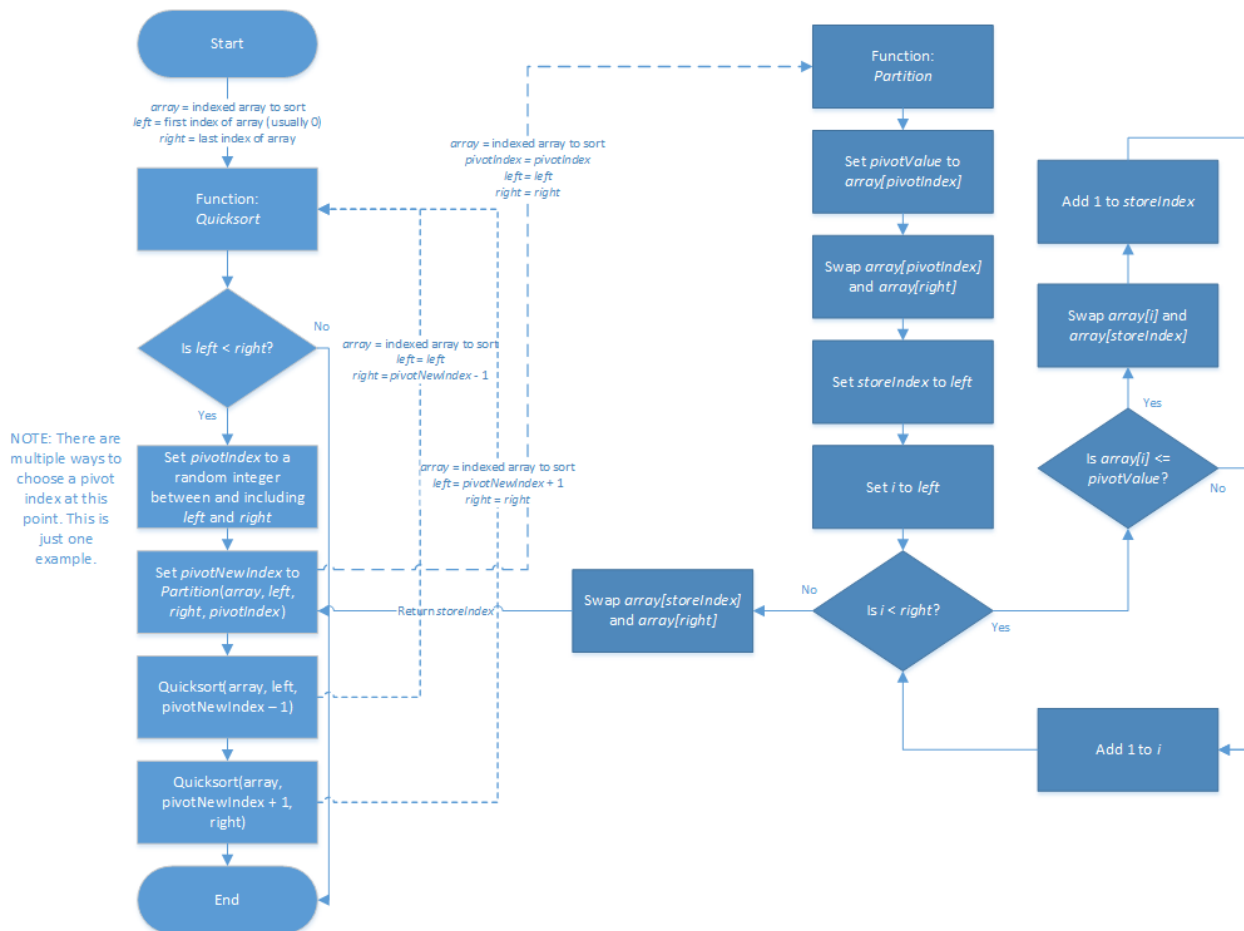
Algoritmo Quicksort



<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 – representação das operações elementares (independente da linguagem)



© 2013 Jeff Allen

Video instrutivo:

<https://www.youtube.com/watch?v=GqVRiqNI1UA>

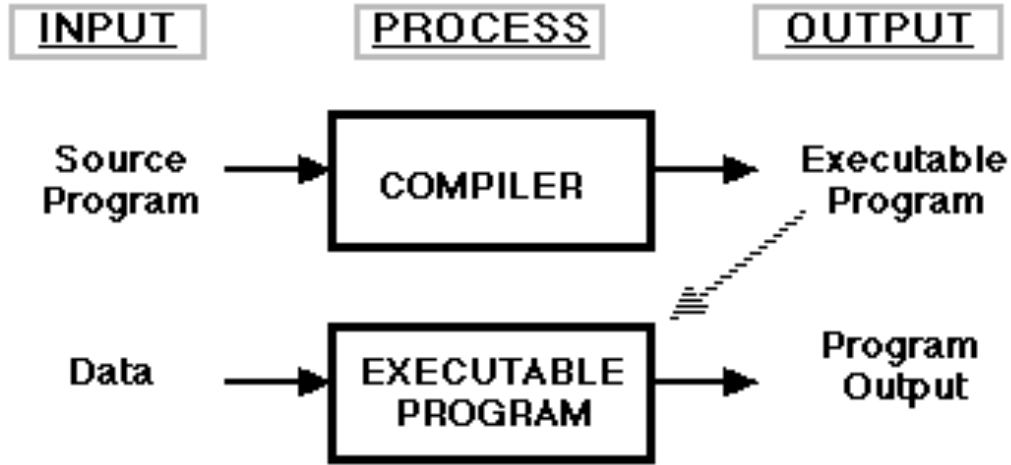
Code sample: Quicksort

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 );
    }
}
```

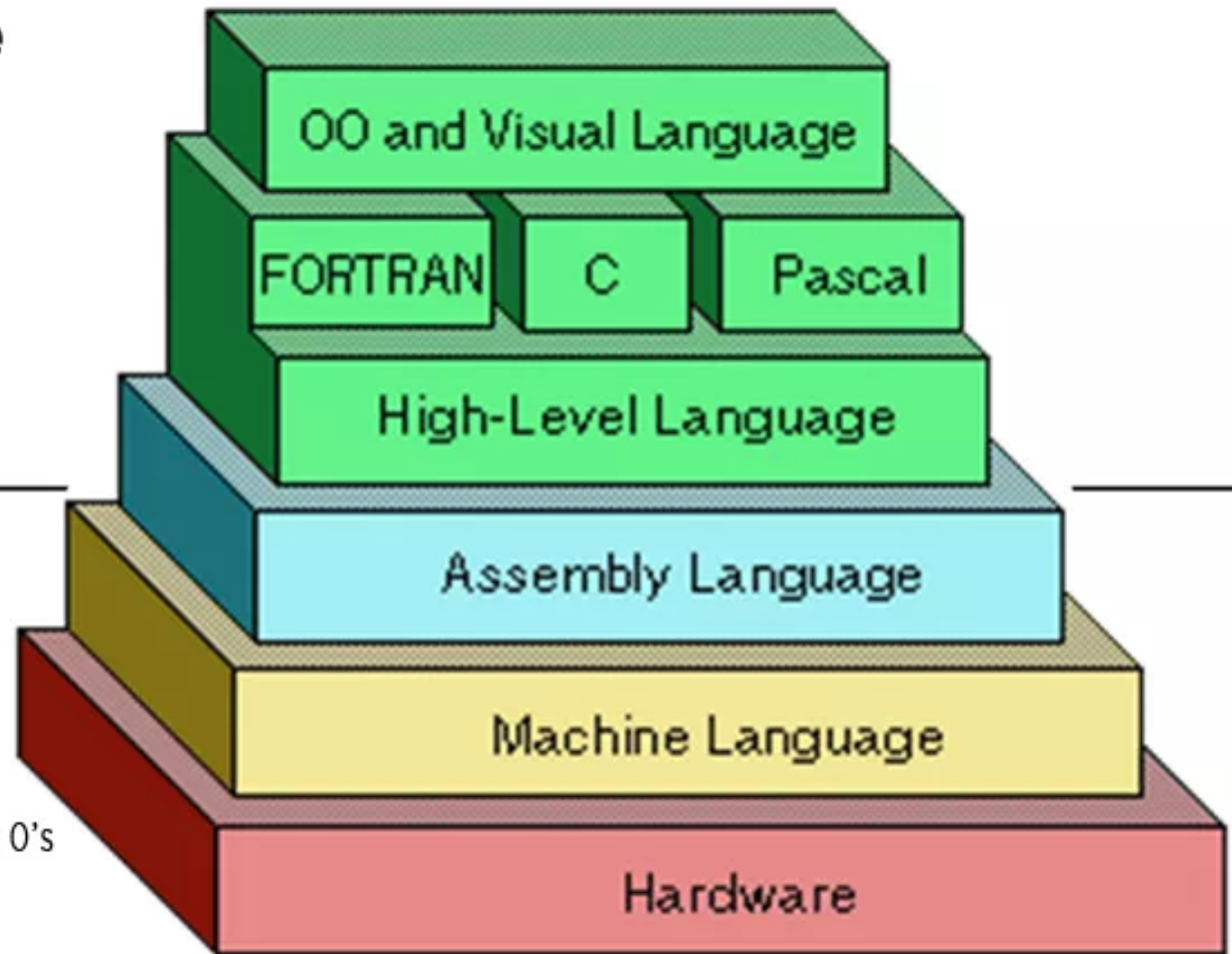
```
~/Desktop/helloworld.c - Sublime Text
helloworld.c
1 #include <stdio.h>
2
3 int main (void) {
4     printf ("Hello, World!\n");
5
6     return 0;
7 }
```



```
> HELLO WORLD!
>
```

High Level Language

- Easy for Programmers to understand
- Contains English Words



Low Level Languages

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

Language Ranking: IEEE Spectrum

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0



<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2019>

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

Worldwide, Feb 2020 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.88 %	+4.1 %
2		Java	19.05 %	-1.8 %
3		Javascript	8.17 %	+0.1 %
4		C#	7.3 %	-0.1 %
5		PHP	6.15 %	-1.0 %
6		C/C++	5.92 %	-0.2 %
7		R	3.74 %	-0.2 %
8		Objective-C	2.42 %	-0.6 %
9		Swift	2.28 %	-0.2 %
10	↑	TypeScript	1.84 %	+0.3 %



<http://pypl.github.io/PYPL.html>





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

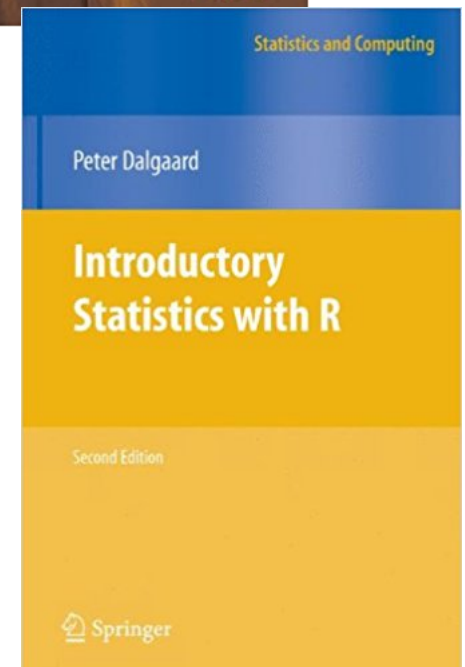
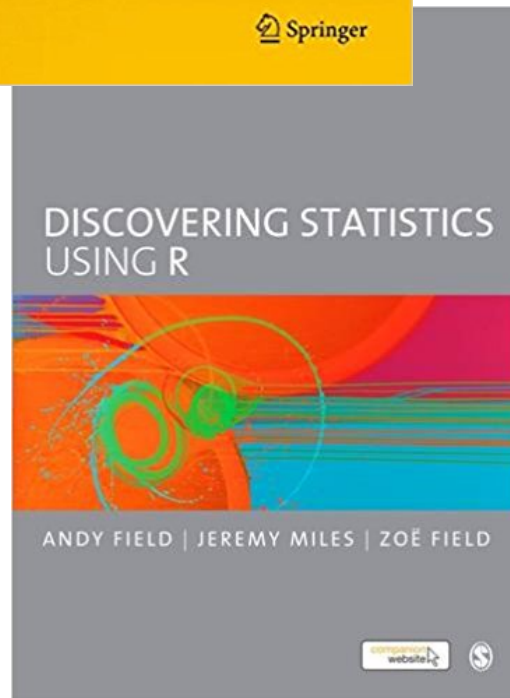
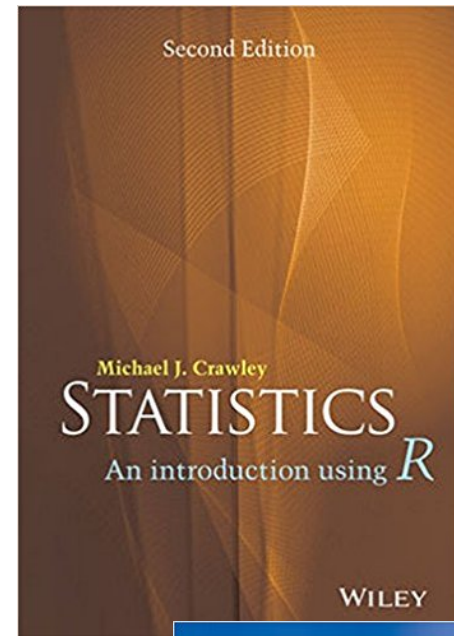
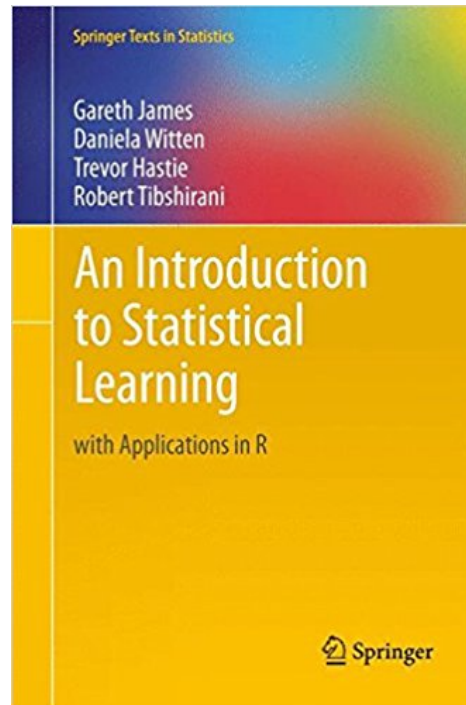
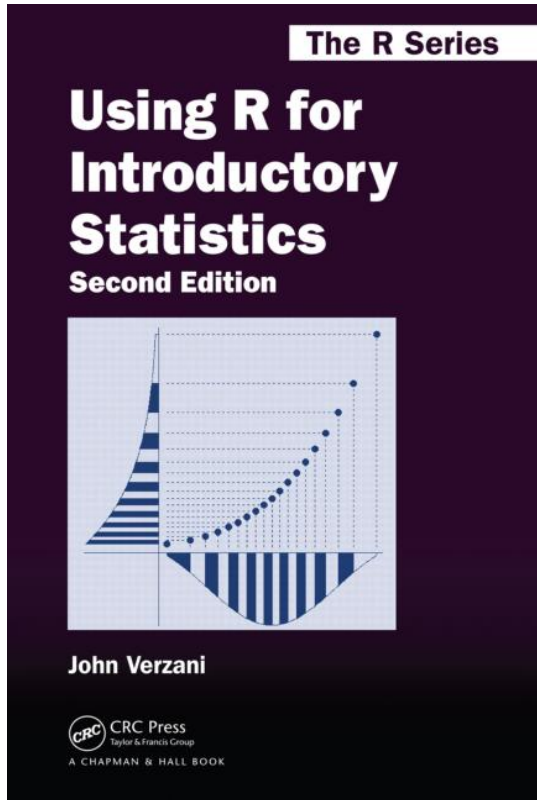
<https://www.r-project.org/>

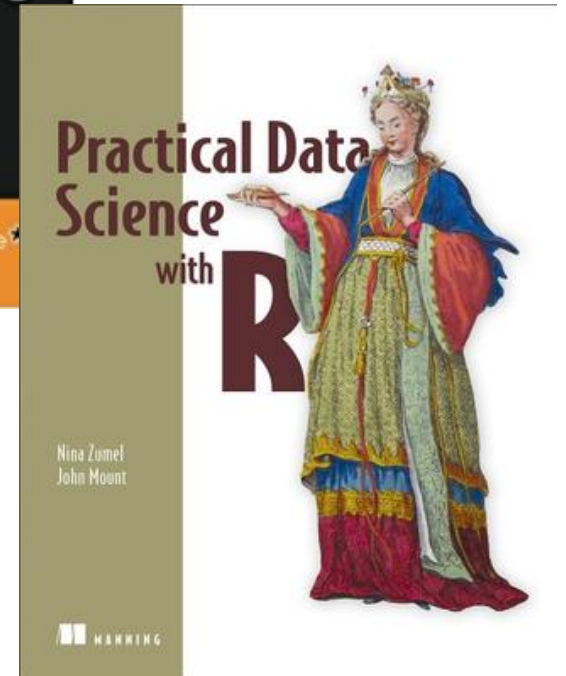
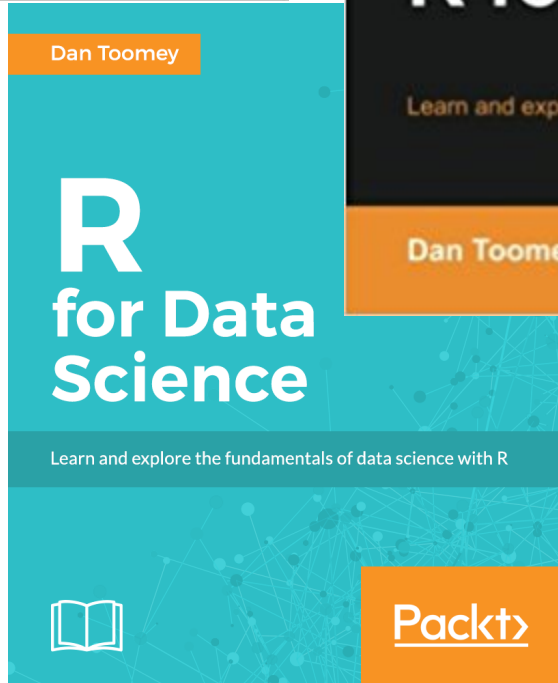
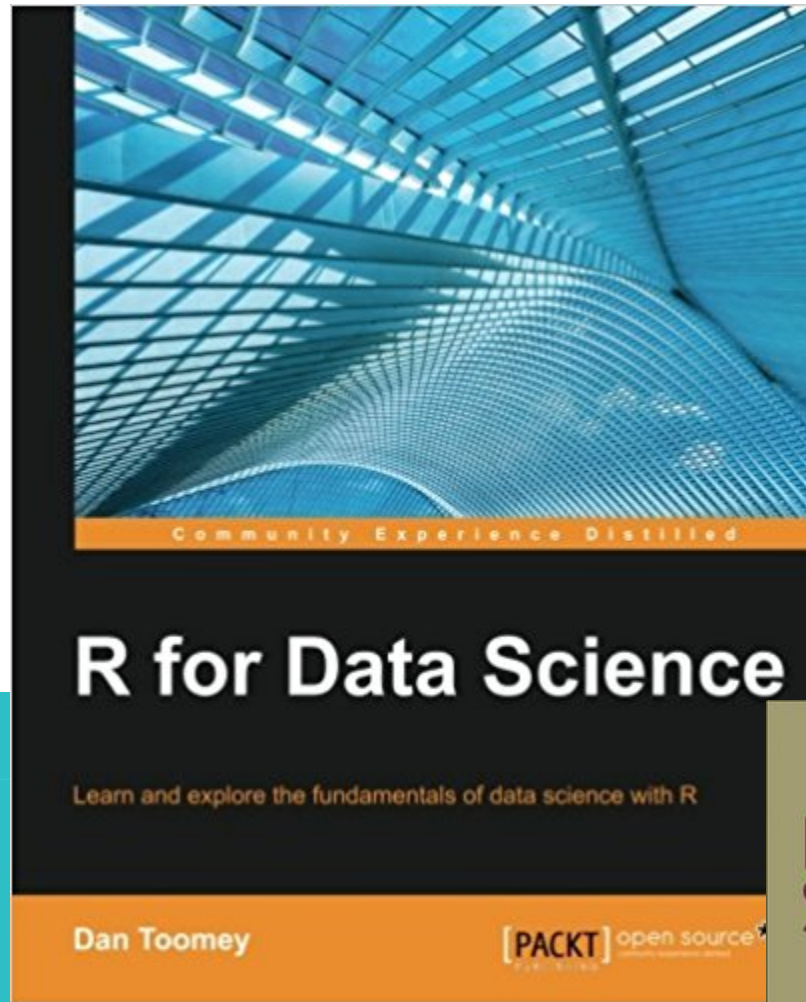
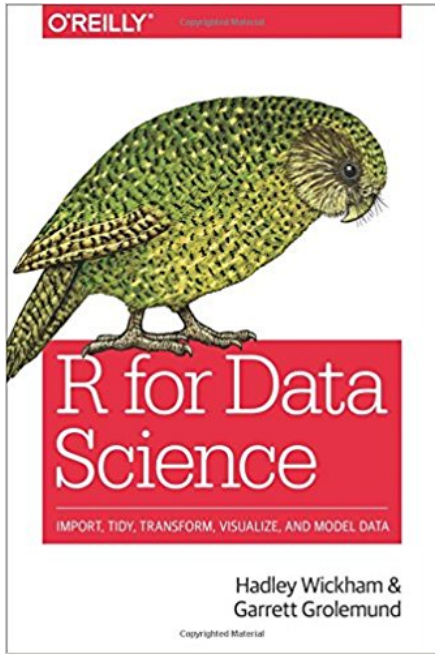
<https://www.rstudio.com/>



The image shows the RStudio interface with four key components highlighted by red boxes:

- 1- Code Editor:** The top-left pane contains R code for loading ggplot2, viewing and summarizing the diamonds dataset, calculating average carat size, and creating a faceted plot of carat vs price by clarity.
- 2- R Console:** The bottom-left pane shows the output of the code, including summary statistics for carat (x, y, z) and the execution of the plot command.
- 3- Workspace and History:** The top-right pane shows the current workspace containing the 'diamonds' data frame (53940 observations) and the 'aveSize' variable (mean value 0.7979).
- 4 - Plots and files:** The bottom-right pane displays a faceted scatter plot titled 'Diamond Pricing' showing Price vs Carat, faceted by clarity (VS2, VS1, VS2, VS1, IF).

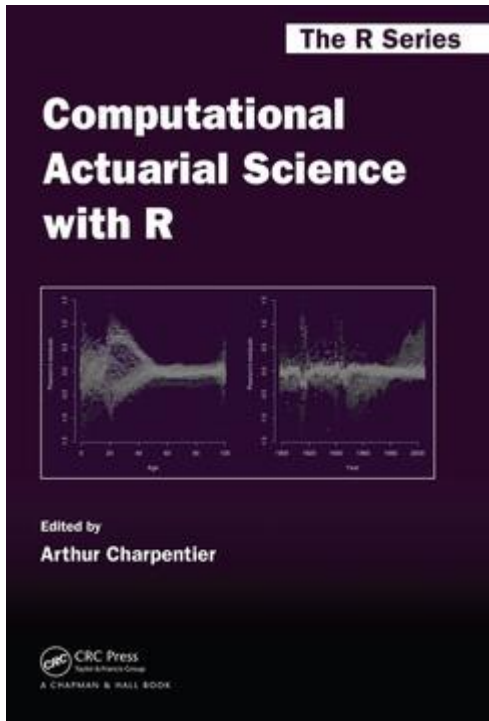




Computational actuarial science with R

Workshop of the 21st International Congress on Insurance:
Mathematics and Economics – IME 2017

July 6-7, 2017



Machine Learning & Data Science for Actuaries, with R

Arthur Charpentier (Université de Rennes 1 & UQàM)

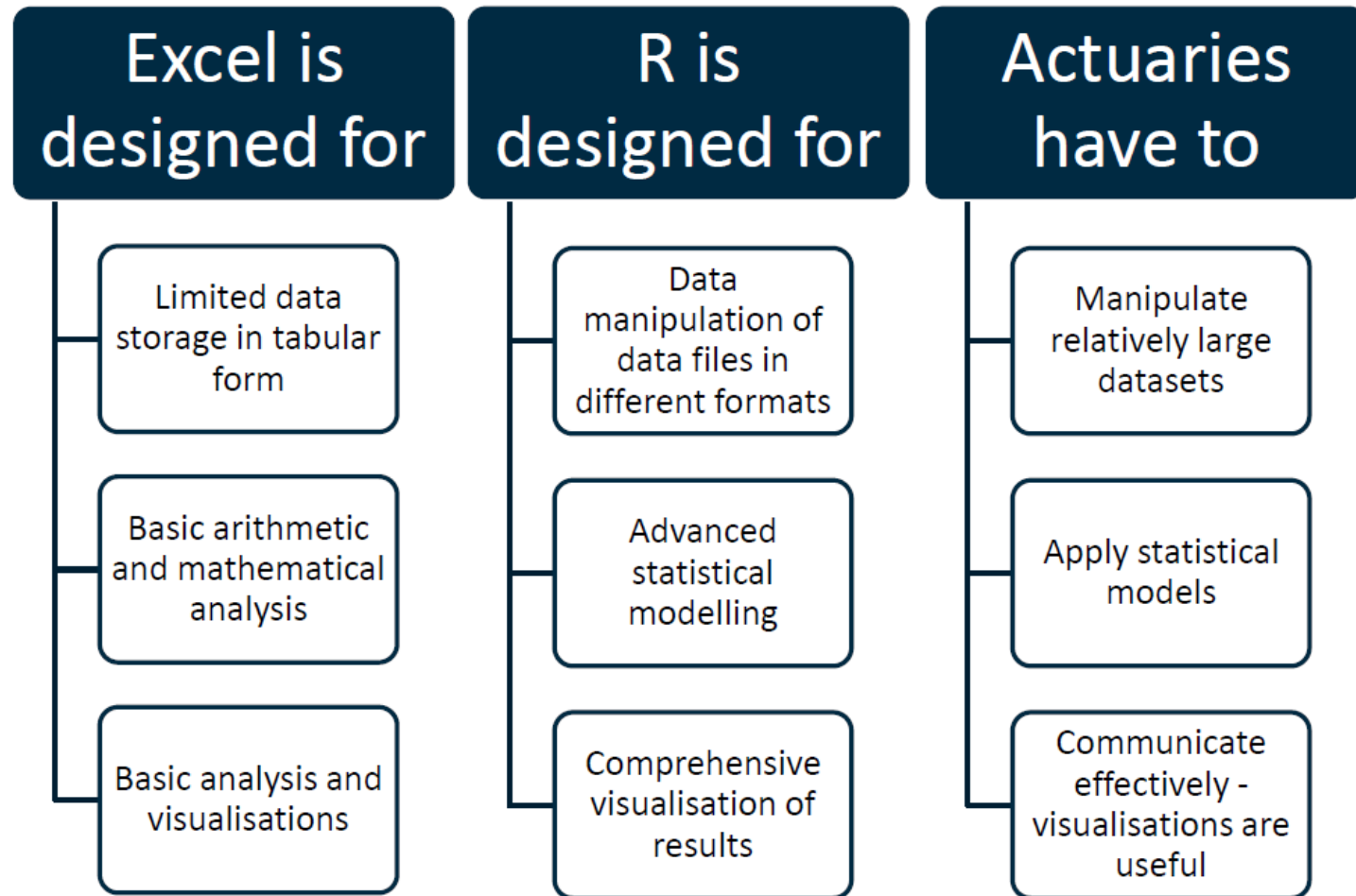


Universitat de Barcelona, April 2016.

<http://freakonometrics.hypotheses.org>



R For Actuaries: What, Why and Where?



MAP 2112 – Introdução à Lógica de Programação e Modelagem Computacional

1º Semestre - 2020

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		Ingresso	
12033 - Bacharelado em Ciências Contábeis	2	2017/1	2
12042 - Bacharelado em Ciências Atuariais	51	2018/1	1
14010 - Bacharelado em Meteorologia	1	2019/1	9
		2020/1	42

Cálculo ?

Programação ? Excel ?

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 (IAS) do DLR na Alemanha (1994-1995). Atualmente é professor doutor do Departamento de Matemática Aplicada da Universidade de São Paulo e Gerente de Desenvolvimento Tecnológico de Sistemas e Software na Aviação Comercial 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 21:00. Sala FEA prédio 1 sala G-6.

Provas: Nos dias 31 de março (P1); 26 de maio (P2) e 30 de junho (P3). A confirmar

Entrega de Trabalho Computacional em grupo em 5 de julho (T1).

Não haverá sub.

Avaliação

Os alunos serão avaliados de acordo com:

$MF = 0.7*(MP)+0.3(MT)$, 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.



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

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

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



Instalem 1º o R e depois o R Studio se tiverem computador disponível, ou encontrem um laboratório onde eles estejam disponíveis.

Planejem na sua semana o dia e horário que irão se dedicar a essa disciplina.

