Lab3

Table of contents

# 1. Grupo:

Eduardo Amorim Vilela de Salis - 11226805

Débora Mauricio Kono - 9896754

Douglas Silva Cardoso - 11766990

Loading required package: foreach

Loading required package: iterators

Package 'GA' version 3.2.3  
Type 'citation("GA")' for citing this R package in publications.

Attaching package: 'GA'

The following object is masked from 'package:utils':  
  
 de

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.0 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ ggplot2 3.4.1 ✔ tibble 3.2.0  
✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
✔ purrr 1.0.1   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ purrr::accumulate() masks foreach::accumulate()  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
✖ purrr::when() masks foreach::when()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors  
The legacy packages maptools, rgdal, and rgeos, underpinning this package  
will retire shortly. Please refer to R-spatial evolution reports on  
https://r-spatial.org/r/2023/05/15/evolution4.html for details.  
This package is now running under evolution status 0   
  
ℹ Google's Terms of Service: <https://mapsplatform.google.com>  
ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.  
  
Attaching package: 'maps'  
  
  
The following object is masked from 'package:purrr':  
  
 map

# 2. Laboratório GA – O Problema da Mochila

## 2.1 Dados:

* **Comentário**: Nessa etapa temos o seguinte problema: temos várias possibilidades de items (cada um com um peso específico e pontos de sobrevivência) para levar em uma mochila, porém esta possui uma capacidade limitada de peso que pode levar. Ou seja , temos que decidir perante recursos (*capacidade*) limitados (peso da mochila) para que tomemos a m elhor decisão possível (maximizar pontos de sobrevivência)

item peso sobrevivencia  
1 casaco de chuva 2 5  
2 canivete 1 3  
3 água\nmineral 6 15  
4 luvas 1 5  
5 saco de dormir 4 6  
6 barraca 9 18  
7 fogão\nportátil 5 8  
8 comida enlatada 8 20  
9 lanches 3 8

## 2.2 Solução simplista

* **Comentário**: em um exemplo hipotético feito para atender o problema, não atingiríamos uma capacidade ótima. A capacidade máxima da mochila (25) não seria aproveitada (soma dos pesos resulta em 10) e apenas 26 pontos de sobrevivência foram obtidos. Em seguida veremos como técnincas matemáticas podem trazer ganhos perante exemplos mais simples

# Conteúdo da Mochila: 1 significa que colocamos o item  
# Conteúdo da Mochila: 0 significa que não colocamos o item  
cromossomas <- c(0,1,1,0,0,0,0,0,1)  
dados[cromossomas==1,]

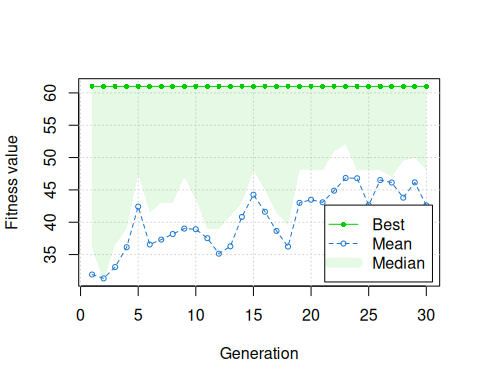
item peso sobrevivencia  
2 canivete 1 3  
3 água\nmineral 6 15  
9 lanches 3 8

### 2.2.1 Solução pelo do Algorítimo genético (G.A)

**Comentário**:

* Foram testados três implantações do Algorítimo Genético, uma com uma população de 30 , outra com 50 , e outra com 100.
* Os ganhos obtidos ao se aumentar a população não foram muito impactantes. Além disso, próximo à oitentésima geração, o encaixe médio começa a cair. Portanto não faz sentido ter uma população muito grande para esse problema. Com uma população de 100 aumentamos apenas 1 ponto de sobrevivência perante os demais . Todos os algorítmos trouxeram ganhos (alocação de recursos e pontos de sobrevivência)
* Com uma população de 30 obtivemos:
  + “Items: casaco de chuva, água,luvas , fogão, comida enlatada , lanches
  + “Peso total: 25
  + “Pontos de sobrevivência: 61
* Com uma população de 50 obtivemos:
  + “Items:”casaco de chuva” “água” “luvas” “fogão” “comida enlatada” “lanches”
  + “Peso total: 25
  + “Pontos de sobrevivência: 61
* Com uma população de 100 obtivemos:
  + “Items:”casaco de chuva” “canivete” “água” “luvas” “saco de dormir” “comida enlatada” “lanches”
  + “Peso total: 25
  + “Pontos de sobrevivência: 62

── Genetic Algorithm ───────────────────   
  
GA settings:   
Type = binary   
Population size = 50   
Number of generations = 30   
Elitism = 2   
Crossover probability = 0.8   
Mutation probability = 0.1   
  
GA results:   
Iterations = 30   
Fitness function value = 61   
Solution =   
 x1 x2 x3 x4 x5 x6 x7 x8 x9  
[1,] 1 0 1 1 0 0 1 1 1

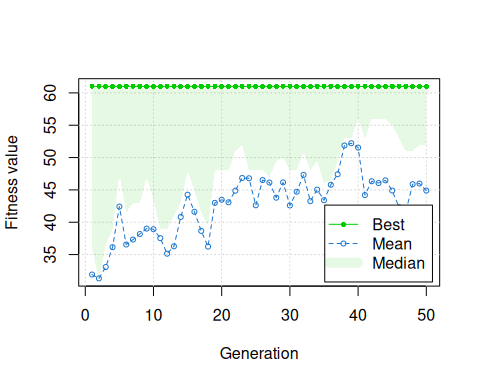


item peso sobrevivencia  
1 casaco de chuva 2 5  
3 água\nmineral 6 15  
4 luvas 1 5  
7 fogão\nportátil 5 8  
8 comida enlatada 8 20  
9 lanches 3 8

[1] "Items: casaco de chuva" "Items: água\nmineral"   
[3] "Items: luvas" "Items: fogão\nportátil"  
[5] "Items: comida enlatada" "Items: lanches"

[1] "Peso total: 25"

[1] "Pontos de sobrevivência: 61"



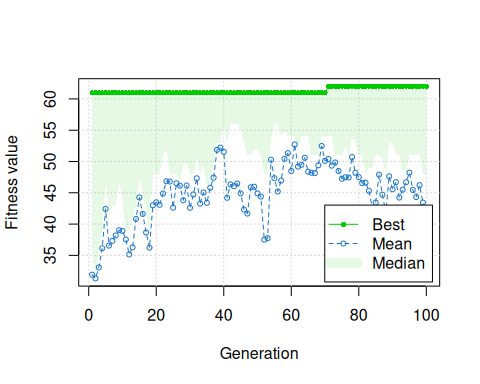
── Genetic Algorithm ───────────────────   
  
GA settings:   
Type = binary   
Population size = 50   
Number of generations = 50   
Elitism = 2   
Crossover probability = 0.8   
Mutation probability = 0.1   
  
GA results:   
Iterations = 50   
Fitness function value = 61   
Solution =   
 x1 x2 x3 x4 x5 x6 x7 x8 x9  
[1,] 1 0 1 1 0 0 1 1 1

item peso sobrevivencia  
1 casaco de chuva 2 5  
3 água\nmineral 6 15  
4 luvas 1 5  
7 fogão\nportátil 5 8  
8 comida enlatada 8 20  
9 lanches 3 8

[1] "Items: casaco de chuva" "Items: água\nmineral"   
[3] "Items: luvas" "Items: fogão\nportátil"  
[5] "Items: comida enlatada" "Items: lanches"

[1] "Peso total: 25"

[1] "Pontos de sobrevivência: 61"



── Genetic Algorithm ───────────────────   
  
GA settings:   
Type = binary   
Population size = 50   
Number of generations = 100   
Elitism = 2   
Crossover probability = 0.8   
Mutation probability = 0.1   
  
GA results:   
Iterations = 100   
Fitness function value = 62   
Solution =   
 x1 x2 x3 x4 x5 x6 x7 x8 x9  
[1,] 1 1 1 1 1 0 0 1 1

item peso sobrevivencia  
1 casaco de chuva 2 5  
2 canivete 1 3  
3 água\nmineral 6 15  
4 luvas 1 5  
5 saco de dormir 4 6  
8 comida enlatada 8 20  
9 lanches 3 8

[1] "Items: casaco de chuva" "Items: canivete"   
[3] "Items: água\nmineral" "Items: luvas"   
[5] "Items: saco de dormir" "Items: comida enlatada"  
[7] "Items: lanches"

[1] "Peso total: 25"

[1] "Pontos de sobrevivência: 62"

# 3. Resolvendo TSP usando Algoritmo Genético em R

## 3.1 Dados

**Comentários**: nossos dados correspondem às distâncias entre as principais cidades europeis , disponiblizado pelo pacote datasets. A proposta deste laboratório consiste em solucionar o problema do caixeiro viajante por meio do algoritmo genético. A tarefa envolve descobrir o trajeto de menor extensão a ser percorrido, ao passar por todas as cidades da matriz uma única vez, antes de retornar à cidade inicial.

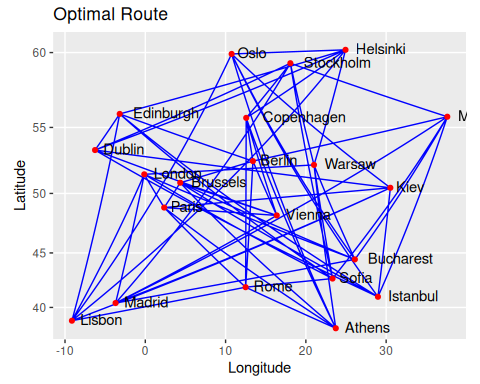
Athens Barcelona Brussels Calais Cherbourg Cologne Copenhagen  
Barcelona 3313   
Brussels 2963 1318   
Calais 3175 1326 204   
Cherbourg 3339 1294 583 460   
Cologne 2762 1498 206 409 785   
Copenhagen 3276 2218 966 1136 1545 760   
Geneva 2610 803 677 747 853 1662 1418  
Gibraltar 4485 1172 2256 2224 2047 2436 3196  
Hamburg 2977 2018 597 714 1115 460 460  
Hook of Holland 3030 1490 172 330 731 269 269  
Lisbon 4532 1305 2084 2052 1827 2290 2971  
Lyons 2753 645 690 739 789 714 1458  
Madrid 3949 636 1558 1550 1347 1764 2498  
Marseilles 2865 521 1011 1059 1101 1035 1778  
Milan 2282 1014 925 1077 1209 911 1537  
Munich 2179 1365 747 977 1160 583 1104  
Paris 3000 1033 285 280 340 465 1176  
Rome 817 1460 1511 1662 1794 1497 2050  
Stockholm 3927 2868 1616 1786 2196 1403 650  
Vienna 1991 1802 1175 1381 1588 937 1455  
 Geneva Gibraltar Hamburg Hook of Holland Lisbon Lyons Madrid  
Barcelona   
Brussels   
Calais   
Cherbourg   
Cologne   
Copenhagen   
Geneva   
Gibraltar 1975   
Hamburg 1118 2897   
Hook of Holland 895 2428 550   
Lisbon 1936 676 2671 2280   
Lyons 158 1817 1159 863 1178   
Madrid 1439 698 2198 1730 668 1281   
Marseilles 425 1693 1479 1183 1762 320 1157  
Milan 328 2185 1238 1098 2250 328 1724  
Munich 591 2565 805 851 2507 724 2010  
Paris 513 1971 877 457 1799 471 1273  
Rome 995 2631 1751 1683 2700 1048 2097  
Stockholm 2068 3886 949 1500 3231 2108 3188  
Vienna 1019 2974 1155 1205 2937 1157 2409  
 Marseilles Milan Munich Paris Rome Stockholm  
Barcelona   
Brussels   
Calais   
Cherbourg   
Cologne   
Copenhagen   
Geneva   
Gibraltar   
Hamburg   
Hook of Holland   
Lisbon   
Lyons   
Madrid   
Marseilles   
Milan 618   
Munich 1109 331   
Paris 792 856 821   
Rome 1011 586 946 1476   
Stockholm 2428 2187 1754 1827 2707   
Vienna 1363 898 428 1249 1209 2105

## 3.2 Aplicação do GA

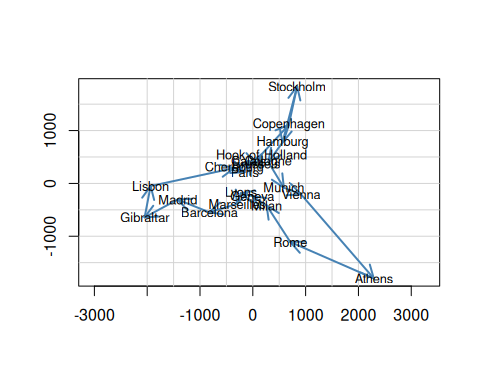
**Comentários**: testamos com uma população maior (100), fizemos também várias iterações aplicando o algorítmo genético para obter diversas rotas possíveis. Criamos um gráfico com as rotas dado a latitude e longitude das cidades. Adicionamos o gráfico contendo o treinamento do algorítmo.

── Genetic Algorithm ───────────────────   
  
GA settings:   
Type = permutation   
Population size = 100   
Number of generations = 5000   
Elitism = 5   
Crossover probability = 0.8   
Mutation probability = 0.2   
  
GA results:   
Iterations = 1338   
Fitness function value = 7.756147e-05   
Solutions =   
 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 ... x20 x21  
[1,] 6 17 21 1 19 16 8 13 15 2 20 10  
[2,] 2 14 9 12 5 18 4 3 11 7 13 15  
[3,] 8 13 15 2 14 9 12 5 18 4 19 16  
[4,] 1 19 16 8 13 15 2 14 9 12 17 21  
[5,] 19 16 8 13 15 2 14 9 12 5 21 1

* **Rota ótima usando a latitude e longitude das cidades**



[1] "Cologne > Munich > Vienna > Athens > Rome > Milan > Geneva > Lyons > Marseilles > Barcelona > Madrid > Gibraltar > Lisbon > Cherbourg > Paris > Calais > Brussels > Hook of Holland > Copenhagen > Stockholm > Hamburg"  
[2] "Barcelona > Madrid > Gibraltar > Lisbon > Cherbourg > Paris > Calais > Brussels > Hook of Holland > Copenhagen > Stockholm > Hamburg > Cologne > Munich > Vienna > Athens > Rome > Milan > Geneva > Lyons > Marseilles"  
[3] "Geneva > Lyons > Marseilles > Barcelona > Madrid > Gibraltar > Lisbon > Cherbourg > Paris > Calais > Brussels > Hook of Holland > Copenhagen > Stockholm > Hamburg > Cologne > Munich > Vienna > Athens > Rome > Milan"  
[4] "Athens > Rome > Milan > Geneva > Lyons > Marseilles > Barcelona > Madrid > Gibraltar > Lisbon > Cherbourg > Paris > Calais > Brussels > Hook of Holland > Copenhagen > Stockholm > Hamburg > Cologne > Munich > Vienna"  
[5] "Rome > Milan > Geneva > Lyons > Marseilles > Barcelona > Madrid > Gibraltar > Lisbon > Cherbourg > Paris > Calais > Brussels > Hook of Holland > Copenhagen > Stockholm > Hamburg > Cologne > Munich > Vienna > Athens"



Attaching package: 'igraph'

The following objects are masked from 'package:lubridate':  
  
 %--%, union

The following objects are masked from 'package:dplyr':  
  
 as\_data\_frame, groups, union

The following objects are masked from 'package:purrr':  
  
 compose, simplify

The following object is masked from 'package:tidyr':  
  
 crossing

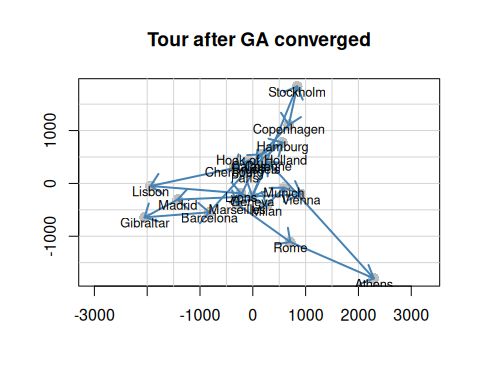
The following object is masked from 'package:tibble':  
  
 as\_data\_frame

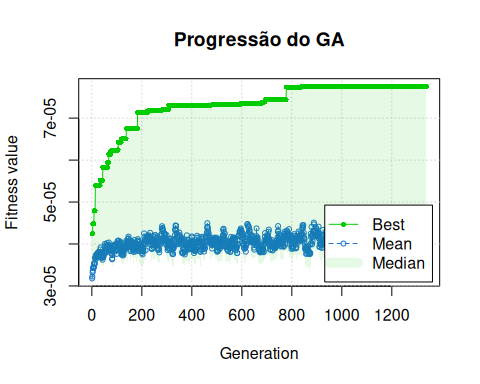
The following objects are masked from 'package:stats':  
  
 decompose, spectrum

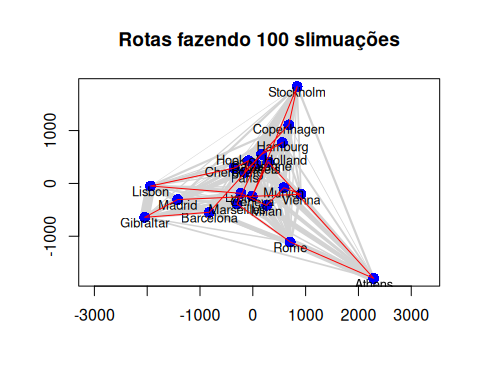
The following object is masked from 'package:base':  
  
 union

Warning in ga(type = "permutation", fitness = tpsFitness, distMatrix = D, :  
'min' arg is deprecated. Use 'lower' instead.

Warning in ga(type = "permutation", fitness = tpsFitness, distMatrix = D, :  
'max' arg is deprecated. Use 'upper' instead.







## 3.3 Conclusão

O ecommerce representa um grande salto tecnológico no ambiente de negócios. Comprar coisas do conforto de casa, em um serivço disponível 24 horas por dia e 7 dias por semana , realizar trocas, encontrar várias opções , e receber os produtos em um prazo antes inimaginável são alguns das ganhos que esse serviço nos trouxe. Por meio dele foram eliminadas barreiras geográficas e o acesso a diferentes mercados foi democratizado. Sobretudo , vale resaltar o seu papel fundamental durante a pandemia da COVID-19, na qual permitiu com que as pessoas pudessem comprar sem sair de casa , e dessa forma atenuando a proliferação da doença .

Por meio dele foram impulsionadas uma série de inovações nas tecnologias logísticas,a quais tornaram entrega mais rápida, eficiente e confiável. As empresas têm investido em soluções como centros de distribuição automatizados, veículos de entrega autônomos e rastreamento em tempo real para otimizar a cadeia de suprimentos. Os avanços tecnológicos permitiram a redução de custos, a melhoria da precisão no gerenciamento de estoques e a personalização das experiências de entrega, aumentando a satisfação do cliente. Dentro dessas inovações se encontram os algoritmos de otimização, que possuem um papel fundamental na logística do e-commerce, garantindo que as operações sejam executadas de forma eficiente e econômica. O problema clássico do caixeiro viajante, por exemplo, envolve encontrar a rota mais curta para visitar um conjunto de cidades e retornar ao ponto de partida. Esse problema tem aplicação direta nas entregas, onde o objetivo é encontrar a melhor rota para otimizar o tempo e minimizar os custos.

Neste laboratório lidamos com um tipo de modelo que é essencial na tomada de decisão, pois permite , dado a limitações de recursos (restrições matemáticas do problema), ele nos permite encontrar uma solução ótima (maximizar ou mnimizar uma função). Dessa maneira , conseguimos superar limitações cognitivas humanas e retirar viéses da tomada de decisão