

Mestrado Profissional em Avaliação e
Monitoramento de Políticas Públicas

Métodos Quantitativos I

Aula 1: Apresentação do curso e conceitos básicos

Professores: Daniel Grimaldi e Arthur Bragança

3º Trimestre - 2025

Apresentação do curso

Os Instrutores

Daniel Grimaldi: Economista graduado pela UFRJ, com mestrado na USP e Ph.D. pela George Mason. Trabalha com monitoramento e avaliação de política pública desde 2011, com experiências no Ipea, BNDES, BID e agora na Secretaria de Avaliação de Políticas Públicas e Assuntos Econômicos (SMA/MPO).

Arthur Bragança: Economista sênior na Prática Global de Meio Ambiente, Recursos Naturais e Economia Azul do Banco Mundial. É bacharel em Economia pela Universidade Federal de Minas Gerais, doutor em Economia na PUC-Rio e acadêmico visitante na Universidade de Harvard. Antes de ingressar no Banco Mundial, trabalhou como chefe de avaliação de políticas na Iniciativa de Política Climática (CPI).

Escopo do curso

Esse é um curso introdutório de **econometria**.

*The Econometric Society is an international society for the advancement of economic theory in its relation to **statistics** and **mathematics**. (...) Its main object shall be to promote studies that aim at a unification of the theoretical-quantitative and the empirical-quantitative approach to **economic problems** and that are penetrated by constructive and rigorous thinking similar to that which has come to **dominate in the natural sciences**. (Frisch 1933)*

Escopo do curso

Esse é um curso **introdutório de econometria aplicada**.

- Objetivo é que todos terminem o curso com capacidade de usar ferramental quantitativo para estudar problemas socioeconômicos.
 - Compreender, contratar e implementar análises econométricas.
- Não vamos nos aprofundar na teoria...
 - Corolário 1: não vamos cobrar provas formais de teoremas, estimadores etc

Escopo do curso

Esse é um curso **introdutório de econometria aplicada**.

- ❖ Objetivo é que todos terminem o curso com capacidade de usar ferramental quantitativo para estudar problemas socioeconômicos.
 - ❖ Compreender, contratar e implementar análises econométricas.
- ❖ ... mas vamos cobrar consolidação dos conceitos por meio de **aplicação direta do ferramental quantitativo**...
 - ❖ Corolário 2: vocês precisarão aprender e usar linguagem de programação ao longo do curso.

Visão Geral

1 slide para falar sobre as listas, ementa email dos instrutores, site do curso etc

Conceitos básicos

Espaço amostral e evento

- ❖ Um **espaço amostral** (Ω) é o conjunto de todos os resultados possíveis para um experimento aleatório.
- ❖ Um **evento** é qualquer conjunto de resultados definidos dentro do espaço amostral.
 - ❖ $A = \{A_1, A_2, \dots, A_n\}; A_i \in \Omega \forall i$
- ❖ Se o resultado A_i foi observado e $A_i \in A$, então dizemos que o evento A ocorreu.
 - ❖ Um evento B está contido em $A \Leftrightarrow B_i \in A \forall i$
 - ❖ A e B serão **eventos disjuntos** $\Leftrightarrow A \cap B = \emptyset$
 - ❖ O **complementar de** (A^c) é formado por todos os resultados que fazem parte do conjunto amostral, mas não estão contidos em A , de tal forma que $A \cup A^c = \Omega$

Probabilidade

- ❖ Se Ω é enumerável, então $P(A) = \frac{\text{Qtd. de elementos de } A}{\text{Qtd. de elementos em } \Omega}$
- ❖ Se Ω não for enumerável, então $P(A) = \frac{\text{Comprimento de } A}{\text{Comprimento de } \Omega}$
- ❖ Uma função $\varphi(A, \Omega)$ é uma probabilidade \Leftrightarrow satisfaz os Axiomas de Kolmogorov:
 - ❖ (i) $P(\Omega) = 1$;
 - ❖ (ii) $\forall A \in \Omega, P(A) \geq 0$;
 - ❖ (iii) Para toda sequência A_1, A_2, \dots, A_n de eventos disjuntos, temos que $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$

Probabilidade condicional e independência

- Sendo $P(B) > 0$, a **probabilidade condicional** de A dado que ocorreu B ($P(A|B)$) é dada por $\frac{P(A \cap B)}{P(B)}$. Caso ($P(B) = 0 \Rightarrow P(A|B) = P(A)$).
- Eventos A e B são **independentes**
 $\Leftrightarrow P(A \cap B) = P(A)P(B)$
- Intuição: quando eventos são independentes, a ocorrência de um não informa nada sobre a ocorrência do outro.
 - $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A)$

Introdução ao R

Por quê usar o R?

❖ Gratuito

- ❖ Tempo e orçamento direcionados para o trabalho

❖ Comunidade ativa

- ❖ Avanços metodológicos chegam primeiro no R;
- ❖ Farto material (gratuito) para treinamento;
- ❖ Diversos fóruns para troca de experiências.

❖ É uma linguagem de programação, não um software

- ❖ Amplitude maior de tarefas (data munging, data scrapping, recursos gráficos, automação...)
- ❖ Permite integração com outras linguagens (essa apresentação foi feita com **R + Latex**).

If statistics programs/languages were cars...



Instalando o R

- ❖ Primeiro, você precisa instalar o *R*
- ❖ Mas ter também o *RStudio* faz toda a diferença!
 - ❖ Interface gráfica mais agradável e recursos ‘point-and-click’;
 - ❖ Permite fazer a gestão, instalação e atualização de pacotes (**Sim, você vai precisar instalar/atualizar pacotes todo o tempo!**);
 - ❖ Permite integrar facilmente, via *RMarkdown*, programação em R com *LaTeX*, *SQL*, Python, Julia, C, C++ etc.

Essa apresentação foi gerada com um arquivo RMarkdown

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for data structures and matrices.


```

102- require(tidyverse)
103- ...
104-
105- ## Data structures
106-
107- - vetor (unidimensional e homogêneo);
108- - matriz (bidimensional e homogêneo);
109- - Array (n-dimensional e homogêneo);
110- - Lista (unidimensional e heterogêneo);
111- - data frame (bidimensional e heterogêneo)
112-
113- ## Vetores
114-
115- ##Code chunk 2: Manipulando vetores##
116- \vsml1
117- ```{r, echo=TRUE, eval=TRUE, tidy=TRUE, tidy.opts=list(arrow=TRUE, indent=2)}
118- # criando
119- vetor_1 <- c(1, 2, 3, 4, 5)
120- # editando
121- vetor_2 <- c(vetor_1, 6:10)
122- vetor_2
123- # operações com vetores
124- c(vetor_2[2])[2]
125-
126- [1] 1 2 3 4 5 6 7 8 9 10
127- [1] 4
128-
129- ## Matrices
130-
131- \vspace{0.6cm}
132- ##Code chunk 3: Manipulando matrizes##
133- \st15
134-
135- [1] 1 2 3 4 5
136- [2] 2 4 6 8 100
137-
138- [1] 1 2 3 4 5
139- [2] 2 4 6 8 100
140-
141- > lista_1[[1]]
142- [1] [1] [2] [3] [4] [5]
143- [1] 1 3 5 7 9
144- [2] 2 4 6 8 100
145- > lista_1[[2]]
146- [1] 1 2 3 4 5
147- > # criando
148- > lista_2 <- list(matrix_1, vetor_1)
149- > lista_2[[1]]
150- [1] [1] [2] [3] [4] [5]
151- [1] 1 3 5 7 9
152- [2] 2 4 6 8 100
153- > # criando
154- > lista_3 <- list(matrix_1, vetor_1)
155- > lista_3[[1]]
156- [1] [1] [2] [3] [4] [5]
157- [1] 1 3 5 7 9
158- [2] 2 4 6 8 100
159- > # criando
160- > lista_4 <- list(matrix_1, vetor_1)
161- > lista_4[[1]]
162- [1] [1] [2] [3] [4] [5]
163- [1] 1 3 5 7 9
164- [2] 2 4 6 8 100
165- > # criando
166- > lista_5 <- list(matrix_1, vetor_1)
167- > lista_5[[1]]
168- [1] [1] [2] [3] [4] [5]
169- [1] 1 3 5 7 9
170- [2] 2 4 6 8 100
171- > # criando
172- > lista_6 <- list(matrix_1, vetor_1)
173- > lista_6[[1]]
174- [1] [1] [2] [3] [4] [5]
175- [1] 1 3 5 7 9
176- [2] 2 4 6 8 100
177- > # criando
178- > lista_7 <- list(matrix_1, vetor_1)
179- > lista_7[[1]]
180- [1] [1] [2] [3] [4] [5]
181- [1] 1 3 5 7 9
182- [2] 2 4 6 8 100
183- > # criando
184- > lista_8 <- list(matrix_1, vetor_1)
185- > lista_8[[1]]
186- [1] [1] [2] [3] [4] [5]
187- [1] 1 3 5 7 9
188- [2] 2 4 6 8 100
189- > # criando
190- > lista_9 <- list(matrix_1, vetor_1)
191- > lista_9[[1]]
192- [1] [1] [2] [3] [4] [5]
193- [1] 1 3 5 7 9
194- [2] 2 4 6 8 100
195- > # criando
196- > lista_10 <- list(matrix_1, vetor_1)
197- > lista_10[[1]]
198- [1] [1] [2] [3] [4] [5]
199- [1] 1 3 5 7 9
200- [2] 2 4 6 8 100
201- > # criando
202- > lista_11 <- list(matrix_1, vetor_1)
203- > lista_11[[1]]
204- [1] [1] [2] [3] [4] [5]
205- [1] 1 3 5 7 9
206- [2] 2 4 6 8 100
207- > # criando
208- > lista_12 <- list(matrix_1, vetor_1)
209- > lista_12[[1]]
210- [1] [1] [2] [3] [4] [5]
211- [1] 1 3 5 7 9
212- [2] 2 4 6 8 100
213- > # criando
214- > lista_13 <- list(matrix_1, vetor_1)
215- > lista_13[[1]]
216- [1] [1] [2] [3] [4] [5]
217- [1] 1 3 5 7 9
218- [2] 2 4 6 8 100
219- > # criando
220- > lista_14 <- list(matrix_1, vetor_1)
221- > lista_14[[1]]
222- [1] [1] [2] [3] [4] [5]
223- [1] 1 3 5 7 9
224- [2] 2 4 6 8 100
225- > # criando
226- > lista_15 <- list(matrix_1, vetor_1)
227- > lista_15[[1]]
228- [1] [1] [2] [3] [4] [5]
229- [1] 1 3 5 7 9
230- [2] 2 4 6 8 100
231- > # criando
232- > lista_16 <- list(matrix_1, vetor_1)
233- > lista_16[[1]]
234- [1] [1] [2] [3] [4] [5]
235- [1] 1 3 5 7 9
236- [2] 2 4 6 8 100
237- > # criando
238- > lista_17 <- list(matrix_1, vetor_1)
239- > lista_17[[1]]
240- [1] [1] [2] [3] [4] [5]
241- [1] 1 3 5 7 9
242- [2] 2 4 6 8 100
243- > # criando
244- > lista_18 <- list(matrix_1, vetor_1)
245- > lista_18[[1]]
246- [1] [1] [2] [3] [4] [5]
247- [1] 1 3 5 7 9
248- [2] 2 4 6 8 100
249- > # criando
250- > lista_19 <- list(matrix_1, vetor_1)
251- > lista_19[[1]]
252- [1] [1] [2] [3] [4] [5]
253- [1] 1 3 5 7 9
254- [2] 2 4 6 8 100
255- > # criando
256- > lista_20 <- list(matrix_1, vetor_1)
257- > lista_20[[1]]
258- [1] [1] [2] [3] [4] [5]
259- [1] 1 3 5 7 9
260- [2] 2 4 6 8 100
261- > # criando
262- > lista_21 <- list(matrix_1, vetor_1)
263- > lista_21[[1]]
264- [1] [1] [2] [3] [4] [5]
265- [1] 1 3 5 7 9
266- [2] 2 4 6 8 100
267- > # criando
268- > lista_22 <- list(matrix_1, vetor_1)
269- > lista_22[[1]]
270- [1] [1] [2] [3] [4] [5]
271- [1] 1 3 5 7 9
272- [2] 2 4 6 8 100
273- > # criando
274- > lista_23 <- list(matrix_1, vetor_1)
275- > lista_23[[1]]
276- [1] [1] [2] [3] [4] [5]
277- [1] 1 3 5 7 9
278- [2] 2 4 6 8 100
279- > # criando
280- > lista_24 <- list(matrix_1, vetor_1)
281- > lista_24[[1]]
282- [1] [1] [2] [3] [4] [5]
283- [1] 1 3 5 7 9
284- [2] 2 4 6 8 100
285- > # criando
286- > lista_25 <- list(matrix_1, vetor_1)
287- > lista_25[[1]]
288- [1] [1] [2] [3] [4] [5]
289- [1] 1 3 5 7 9
290- [2] 2 4 6 8 100
291- > # criando
292- > lista_26 <- list(matrix_1, vetor_1)
293- > lista_26[[1]]
294- [1] [1] [2] [3] [4] [5]
295- [1] 1 3 5 7 9
296- [2] 2 4 6 8 100
297- > # criando
298- > lista_27 <- list(matrix_1, vetor_1)
299- > lista_27[[1]]
300- [1] [1] [2] [3] [4] [5]
301- [1] 1 3 5 7 9
302- [2] 2 4 6 8 100
303- > # criando
304- > lista_28 <- list(matrix_1, vetor_1)
305- > lista_28[[1]]
306- [1] [1] [2] [3] [4] [5]
307- [1] 1 3 5 7 9
308- [2] 2 4 6 8 100
309- > # criando
310- > lista_29 <- list(matrix_1, vetor_1)
311- > lista_29[[1]]
312- [1] [1] [2] [3] [4] [5]
313- [1] 1 3 5 7 9
314- [2] 2 4 6 8 100
315- > # criando
316- > lista_30 <- list(matrix_1, vetor_1)
317- > lista_30[[1]]
318- [1] [1] [2] [3] [4] [5]
319- [1] 1 3 5 7 9
320- [2] 2 4 6 8 100
321- > # criando
322- > lista_31 <- list(matrix_1, vetor_1)
323- > lista_31[[1]]
324- [1] [1] [2] [3] [4] [5]
325- [1] 1 3 5 7 9
326- [2] 2 4 6 8 100
327- > # criando
328- > lista_32 <- list(matrix_1, vetor_1)
329- > lista_32[[1]]
330- [1] [1] [2] [3] [4] [5]
331- [1] 1 3 5 7 9
332- [2] 2 4 6 8 100
333- > # criando
334- > lista_33 <- list(matrix_1, vetor_1)
335- > lista_33[[1]]
336- [1] [1] [2] [3] [4] [5]
337- [1] 1 3 5 7 9
338- [2] 2 4 6 8 100
339- > # criando
340- > lista_34 <- list(matrix_1, vetor_1)
341- > lista_34[[1]]
342- [1] [1] [2] [3] [4] [5]
343- [1] 1 3 5 7 9
344- [2] 2 4 6 8 100
345- > # criando
346- > lista_35 <- list(matrix_1, vetor_1)
347- > lista_35[[1]]
348- [1] [1] [2] [3] [4] [5]
349- [1] 1 3 5 7 9
350- [2] 2 4 6 8 100
351- > # criando
352- > lista_36 <- list(matrix_1, vetor_1)
353- > lista_36[[1]]
354- [1] [1] [2] [3] [4] [5]
355- [1] 1 3 5 7 9
356- [2] 2 4 6 8 100
357- > # criando
358- > lista_37 <- list(matrix_1, vetor_1)
359- > lista_37[[1]]
360- [1] [1] [2] [3] [4] [5]
361- [1] 1 3 5 7 9
362- [2] 2 4 6 8 100
363- > # criando
364- > lista_38 <- list(matrix_1, vetor_1)
365- > lista_38[[1]]
366- [1] [1] [2] [3] [4] [5]
367- [1] 1 3 5 7 9
368- [2] 2 4 6 8 100
369- > # criando
370- > lista_39 <- list(matrix_1, vetor_1)
371- > lista_39[[1]]
372- [1] [1] [2] [3] [4] [5]
373- [1] 1 3 5 7 9
374- [2] 2 4 6 8 100
375- > # criando
376- > lista_40 <- list(matrix_1, vetor_1)
377- > lista_40[[1]]
378- [1] [1] [2] [3] [4] [5]
379- [1] 1 3 5 7 9
380- [2] 2 4 6 8 100
381- > # criando
382- > lista_41 <- list(matrix_1, vetor_1)
383- > lista_41[[1]]
384- [1] [1] [2] [3] [4] [5]
385- [1] 1 3 5 7 9
386- [2] 2 4 6 8 100
387- > # criando
388- > lista_42 <- list(matrix_1, vetor_1)
389- > lista_42[[1]]
390- [1] [1] [2] [3] [4] [5]
391- [1] 1 3 5 7 9
392- [2] 2 4 6 8 100
393- > # criando
394- > lista_43 <- list(matrix_1, vetor_1)
395- > lista_43[[1]]
396- [1] [1] [2] [3] [4] [5]
397- [1] 1 3 5 7 9
398- [2] 2 4 6 8 100
399- > # criando
400- > lista_44 <- list(matrix_1, vetor_1)
401- > lista_44[[1]]
402- [1] [1] [2] [3] [4] [5]
403- [1] 1 3 5 7 9
404- [2] 2 4 6 8 100
405- > # criando
406- > lista_45 <- list(matrix_1, vetor_1)
407- > lista_45[[1]]
408- [1] [1] [2] [3] [4] [5]
409- [1] 1 3 5 7 9
410- [2] 2 4 6 8 100
411- > # criando
412- > lista_46 <- list(matrix_1, vetor_1)
413- > lista_46[[1]]
414- [1] [1] [2] [3] [4] [5]
415- [1] 1 3 5 7 9
416- [2] 2 4 6 8 100
417- > # criando
418- > lista_47 <- list(matrix_1, vetor_1)
419- > lista_47[[1]]
420- [1] [1] [2] [3] [4] [5]
421- [1] 1 3 5 7 9
422- [2] 2 4 6 8 100
423- > # criando
424- > lista_48 <- list(matrix_1, vetor_1)
425- > lista_48[[1]]
426- [1] [1] [2] [3] [4] [5]
427- [1] 1 3 5 7 9
428- [2] 2 4 6 8 100
429- > # criando
430- > lista_49 <- list(matrix_1, vetor_1)
431- > lista_49[[1]]
432- [1] [1] [2] [3] [4] [5]
433- [1] 1 3 5 7 9
434- [2] 2 4 6 8 100
435- > # criando
436- > lista_50 <- list(matrix_1, vetor_1)
437- > lista_50[[1]]
438- [1] [1] [2] [3] [4] [5]
439- [1] 1 3 5 7 9
440- [2] 2 4 6 8 100
441- > # criando
442- > lista_51 <- list(matrix_1, vetor_1)
443- > lista_51[[1]]
444- [1] [1] [2] [3] [4] [5]
445- [1] 1 3 5 7 9
446- [2] 2 4 6 8 100
447- > # criando
448- > lista_52 <- list(matrix_1, vetor_1)
449- > lista_52[[1]]
450- [1] [1] [2] [3] [4] [5]
451- [1] 1 3 5 7 9
452- [2] 2 4 6 8 100
453- > # criando
454- > lista_53 <- list(matrix_1, vetor_1)
455- > lista_53[[1]]
456- [1] [1] [2] [3] [4] [5]
457- [1] 1 3 5 7 9
458- [2] 2 4 6 8 100
459- > # criando
460- > lista_54 <- list(matrix_1, vetor_1)
461- > lista_54[[1]]
462- [1] [1] [2] [3] [4] [5]
463- [1] 1 3 5 7 9
464- [2] 2 4 6 8 100
465- > # criando
466- > lista_55 <- list(matrix_1, vetor_1)
467- > lista_55[[1]]
468- [1] [1] [2] [3] [4] [5]
469- [1] 1 3 5 7 9
470- [2] 2 4 6 8 100
471- > # criando
472- > lista_56 <- list(matrix_1, vetor_1)
473- > lista_56[[1]]
474- [1] [1] [2] [3] [4] [5]
475- [1] 1 3 5 7 9
476- [2] 2 4 6 8 100
477- > # criando
478- > lista_57 <- list(matrix_1, vetor_1)
479- > lista_57[[1]]
480- [1] [1] [2] [3] [4] [5]
481- [1] 1 3 5 7 9
482- [2] 2 4 6 8 100
483- > # criando
484- > lista_58 <- list(matrix_1, vetor_1)
485- > lista_58[[1]]
486- [1] [1] [2] [3] [4] [5]
487- [1] 1 3 5 7 9
488- [2] 2 4 6 8 100
489- > # criando
490- > lista_59 <- list(matrix_1, vetor_1)
491- > lista_59[[1]]
492- [1] [1] [2] [3] [4] [5]
493- [1] 1 3 5 7 9
494- [2] 2 4 6 8 100
495- > # criando
496- > lista_60 <- list(matrix_1, vetor_1)
497- > lista_60[[1]]
498- [1] [1] [2] [3] [4] [5]
499- [1] 1 3 5 7 9
500- [2] 2 4 6 8 100
501- > # criando
502- > lista_61 <- list(matrix_1, vetor_1)
503- > lista_61[[1]]
504- [1] [1] [2] [3] [4] [5]
505- [1] 1 3 5 7 9
506- [2] 2 4 6 8 100
507- > # criando
508- > lista_62 <- list(matrix_1, vetor_1)
509- > lista_62[[1]]
510- [1] [1] [2] [3] [4] [5]
511- [1] 1 3 5 7 9
512- [2] 2 4 6 8 100
513- > # criando
514- > lista_63 <- list(matrix_1, vetor_1)
515- > lista_63[[1]]
516- [1] [1] [2] [3] [4] [5]
517- [1] 1 3 5 7 9
518- [2] 2 4 6 8 100
519- > # criando
520- > lista_64 <- list(matrix_1, vetor_1)
521- > lista_64[[1]]
522- [1] [1] [2] [3] [4] [5]
523- [1] 1 3 5 7 9
524- [2] 2 4 6 8 100
525- > # criando
526- > lista_65 <- list(matrix_1, vetor_1)
527- > lista_65[[1]]
528- [1] [1] [2] [3] [4] [5]
529- [1] 1 3 5 7 9
530- [2] 2 4 6 8 100
531- > # criando
532- > lista_66 <- list(matrix_1, vetor_1)
533- > lista_66[[1]]
534- [1] [1] [2] [3] [4] [5]
535- [1] 1 3 5 7 9
536- [2] 2 4 6 8 100
537- > # criando
538- > lista_67 <- list(matrix_1, vetor_1)
539- > lista_67[[1]]
540- [1] [1] [2] [3] [4] [5]
541- [1] 1 3 5 7 9
542- [2] 2 4 6 8 100
543- > # criando
544- > lista_68 <- list(matrix_1, vetor_1)
545- > lista_68[[1]]
546- [1] [1] [2] [3] [4] [5]
547- [1] 1 3 5 7 9
548- [2] 2 4 6 8 100
549- > # criando
550- > lista_69 <- list(matrix_1, vetor_1)
551- > lista_69[[1]]
552- [1] [1] [2] [3] [4] [5]
553- [1] 1 3 5 7 9
554- [2] 2 4 6 8 100
555- > # criando
556- > lista_70 <- list(matrix_1, vetor_1)
557- > lista_70[[1]]
558- [1] [1] [2] [3] [4] [5]
559- [1] 1 3 5 7 9
560- [2] 2 4 6 8 100
561- > # criando
562- > lista_71 <- list(matrix_1, vetor_1)
563- > lista_71[[1]]
564- [1] [1] [2] [3] [4] [5]
565- [1] 1 3 5 7 9
566- [2] 2 4 6 8 100
567- > # criando
568- > lista_72 <- list(matrix_1, vetor_1)
569- > lista_72[[1]]
570- [1] [1] [2] [3] [4] [5]
571- [1] 1 3 5 7 9
572- [2] 2 4 6 8 100
573- > # criando
574- > lista_73 <- list(matrix_1, vetor_1)
575- > lista_73[[1]]
576- [1] [1] [2] [3] [4] [5]
577- [1] 1 3 5 7 9
578- [2] 2 4 6 8 100
579- > # criando
580- > lista_74 <- list(matrix_1, vetor_1)
581- > lista_74[[1]]
582- [1] [1] [2] [3] [4] [5]
583- [1] 1 3 5 7 9
584- [2] 2 4 6 8 100
585- > # criando
586- > lista_75 <- list(matrix_1, vetor_1)
587- > lista_75[[1]]
588- [1] [1] [2] [3] [4] [5]
589- [1] 1 3 5 7 9
590- [2] 2 4 6 8 100
591- > # criando
592- > lista_76 <- list(matrix_1, vetor_1)
593- > lista_76[[1]]
594- [1] [1] [2] [3] [4] [5]
595- [1] 1 3 5 7 9
596- [2] 2 4 6 8 100
597- > # criando
598- > lista_77 <- list(matrix_1, vetor_1)
599- > lista_77[[1]]
600- [1] [1] [2] [3] [4] [5]
601- [1] 1 3 5 7 9
602- [2] 2 4 6 8 100
603- > # criando
604- > lista_78 <- list(matrix_1, vetor_1)
605- > lista_78[[1]]
606- [1] [1] [2] [3] [4] [5]
607- [1] 1 3 5 7 9
608- [2] 2 4 6 8 100
609- > # criando
610- > lista_79 <- list(matrix_1, vetor_1)
611- > lista_79[[1]]
612- [1] [1] [2] [3] [4] [5]
613- [1] 1 3 5 7 9
614- [2] 2 4 6 8 100
615- > # criando
616- > lista_80 <- list(matrix_1, vetor_1)
617- > lista_80[[1]]
618- [1] [1] [2] [3] [4] [5]
619- [1] 1 3 5 7 9
620- [2] 2 4 6 8 100
621- > # criando
622- > lista_81 <- list(matrix_1, vetor_1)
623- > lista_81[[1]]
624- [1] [1] [2] [3] [4] [5]
625- [1] 1 3 5 7 9
626- [2] 2 4 6 8 100
627- > # criando
628- > lista_82 <- list(matrix_1, vetor_1)
629- > lista_82[[1]]
630- [1] [1] [2] [3] [4] [5]
631- [1] 1 3 5 7 9
632- [2] 2 4 6 8 100
633- > # criando
634- > lista_83 <- list(matrix_1, vetor_1)
635- > lista_83[[1]]
636- [1] [1] [2] [3] [4] [5]
637- [1] 1 3 5 7 9
638- [2] 2 4 6 8 100
639- > # criando
640- > lista_84 <- list(matrix_1, vetor_1)
641- > lista_84[[1]]
642- [1] [1] [2] [3] [4] [5]
643- [1] 1 3 5 7 9
644- [2] 2 4 6 8 100
645- > # criando
646- > lista_85 <- list(matrix_1, vetor_1)
647- > lista_85[[1]]
648- [1] [1] [2] [3] [4] [5]
649- [1] 1 3 5 7 9
650- [2] 2 4 6 8 100
651- > # criando
652- > lista_86 <- list(matrix_1, vetor_1)
653- > lista_86[[1]]
654- [1] [1] [2] [3] [4] [5]
655- [1] 1 3 5 7 9
656- [2] 2 4 6 8 100
657- > # criando
658- > lista_87 <- list(matrix_1, vetor_1)
659- > lista_87[[1]]
660- [1] [1] [2] [3] [4] [5]
661- [1] 1 3 5 7 9
662- [2] 2 4 6 8 100
663- > # criando
664- > lista_88 <- list(matrix_1, vetor_1)
665- > lista_88[[1]]
666- [1] [1] [2] [3] [4] [5]
667- [1] 1 3 5 7 9
668- [2] 2 4 6 8 100
669- > # criando
670- > lista_89 <- list(matrix_1, vetor_1)
671- > lista_89[[1]]
672- [1] [1] [2] [3] [4] [5]
673- [1] 1 3 5 7 9
674- [2] 2 4 6 8 100
675- > # criando
676- > lista_90 <- list(matrix_1, vetor_1)
677- > lista_90[[1]]
678- [1] [1] [2] [3] [4] [5]
679- [1] 1 3 5 7 9
680- [2] 2 4 6 8 100
681- > # criando
682- > lista_91 <- list(matrix_1, vetor_1)
683- > lista_91[[1]]
684- [1] [1] [2] [3] [4] [5]
685- [1] 1 3 5 7 9
686- [2] 2 4 6 8 100
687- > # criando
688- > lista_92 <- list(matrix_1, vetor_1)
689- > lista_92[[1]]
690- [1] [1] [2] [3] [4] [5]
691- [1] 1 3 5 7 9
692- [2] 2 4 6 8 100
693- > # criando
694- > lista_93 <- list(matrix_1, vetor_1)
695- > lista_93[[1]]
696- [1] [1] [2] [3] [4] [5]
697- [1] 1 3 5 7 9
698- [2] 2 4 6 8 100
699- > # criando
700- > lista_94 <- list(matrix_1, vetor_1)
701- > lista_94[[1]]
702- [1] [1] [2] [3] [4] [5]
703- [1] 1 3 5 7 9
704- [2] 2 4 6 8 100
705- > # criando
706- > lista_95 <- list(matrix_1, vetor_1)
707- > lista_95[[1]]
708- [1] [1] [2] [3] [4] [5]
709- [1] 1 3 5 7 9
710- [2] 2 4 6 8 100
711- > # criando
712- > lista_96 <- list(matrix_1, vetor_1)
713- > lista_96[[1]]
714- [1] [1] [2] [3] [4] [5]
715- [1] 1 3 5 7 9
716- [2] 2 4 6 8 100
717- > # criando
718- > lista_97 <- list(matrix_1, vetor_1)
719- > lista_97[[1]]
720- [1] [1] [2] [3] [4] [5]
721- [1] 1 3 5 7 9
722- [2] 2 4 6 8 100
723- > # criando
724- > lista_98 <- list(matrix_1, vetor_1)
725- > lista_98[[1]]
726- [1] [1] [2] [3] [4] [5]
727- [1] 1 3 5 7 9
728- [2] 2 4 6 8 100
729- > # criando
730- > lista_99 <- list(matrix_1, vetor_1)
731- > lista_99[[1]]
732- [1] [1] [2] [3] [4] [5]
733- [1] 1 3 5 7 9
734- [2] 2 4 6 8 100
735- > # criando
736- > lista_100 <- list(matrix_1, vetor_1)
737- > lista_100[[1]]
738- [1] [1] [2] [3] [4] [5]
739- [1] 1 3 5 7 9
740- [2] 2 4 6 8 100
741- > # criando
742- > lista_101 <- list(matrix_1, vetor_1)
743- > lista_101[[1]]
744- [1] [1] [2] [3] [4] [5]
745- [1] 1 3 5 7 9
746- [2] 2 4 6 8 100
747- > # criando
748- > lista_102 <- list(matrix_1, vetor_1)
749- > lista_102[[1]]
750- [1] [1] [2] [3] [4] [5]
751- [1] 1 3 5 7 9
752- [2] 2 4 6 8 100
753- > # criando
754- > lista_103 <- list(matrix_1, vetor_1)
755- > lista_103[[1]]
756- [1] [1] [2] [3] [4] [5]
757- [1] 1 3 5 7 9
758- [2] 2 4 6 8 100
759- > # criando
760- > lista_104 <- list(matrix_1, vetor_1)
761- > lista_104[[1]]
762- [1] [1] [2] [3] [4] [5]
763- [1] 1 3 5 7 9
764- [2] 2 4 6 8 100
765- > # criando
766- > lista_105 <- list(matrix_1, vetor_1)
767- > lista_105[[1]]
768- [1] [1] [2] [3] [4] [5]
769- [1] 1 3 5 7 9
770- [2] 2 4 6 8 100
771- > # criando
772- > lista_106 <- list(matrix_1, vetor_1)
773- > lista_106[[1]]
774- [1] [1] [2] [3] [4] [5]
775- [1] 1 3 5 7 9
776- [2] 2 4 6 8 100
777- > # criando
778- > lista_107 <- list(matrix_1, vetor_1)
779- > lista_107[[1]]
780- [1] [1] [2] [3] [4] [5]
781- [1] 1 3 5 7 9
782- [2] 2 4 6 8 100
783- > # criando
784- > lista_108 <- list(matrix_1, vetor_1)
7
```


Instalando pacotes

Code Chunk 1: Instalação de pacotes no R

```
# 'Simples'  
install.packages("tidyverse")  
require(tidyverse)  
  
# 'Sofisticado'  
if (!require(tidyverse)) {  
  install.packages("tidyverse")  
  require(tidyverse)  
}
```


Estruturas de dados em R

- ❏ Vetor (unidimensional e homogêneo);
- ❏ Matrix (bidimensional e homogêneo);
- ❏ Array (n-dimensional e homogêneo);
- ❏ Lista (unidimensional e heterogêneo);
- ❏ Data frame (bidimensional e heterogêneo)

Vetores

Code Chunk 2: Manipulando vetores

```
# criando
vetor_1 <- c(1, 2, 3, 4, 5)
# editando
vetor_2 <- c(vetor_1, 6:10)
vetor_2

## [1] 1 2 3 4 5 6 7 8 9 10

# operações com vetores
c(vetor_2 * 2)[2]

## [1] 4
```

Matrizes

Code Chunk 3: Manipulando matrizes

```
# criando
```

```
matrix_1 <- matrix(vetor_2, nrow = 2, ncol = 5)  
matrix_1[2, 5] <- 100  
matrix_1
```

```
##      [,1] [,2] [,3] [,4] [,5]  
## [1,]    1    3    5    7    9  
## [2,]    2    4    6    8  100
```

```
# operações com matrizes
```

```
matrix_1 * 2
```

```
##      [,1] [,2] [,3] [,4] [,5]  
## [1,]    2    6   10   14   18  
## [2,]    4    8   12   16  200
```

```
matrix_1 %*% t(matrix_1)
```

```
##      [,1] [,2]  
## [1,]  165 1000  
## [2,] 1000 10120
```

Listas

Code Chunk 4: Manipulando listas

```
# criando
```

```
lista_1 <- list(matrix_1, vetor_1)
```

```
lista_1[[1]]
```

```
##      [,1] [,2] [,3] [,4] [,5]
```

```
## [1,]    1    3    5    7    9
```

```
## [2,]    2    4    6    8   100
```

```
lista_1[[2]]
```

```
## [1] 1 2 3 4 5
```

```
lista_1[[1]][2, 5]
```

```
## [1] 100
```

```
lista_1[[2]][2]
```

```
## [1] 2
```

Criando bases no R

Code Chunk 5: Criando bases de dados no R

```
sample_n=1000
data <- data.frame("id"=paste("pf", 1:sample_n, sep="_"),
                  "A"=sample(c(0.1, 0.3, 1.2), size=sample_n,
                             prob=c(0.45, 0.45, 0.1)), replace=TRUE),
                  "k"=rnorm(n=sample_n, mean=15, sd=3),
                  "l"=rnorm(n=sample_n, mean=5, sd=1),
                  "tipo_local"=sample(c("resex", "assent", "outros_rural",
                                       "outros_urbano"),
                                       size=sample_n,
                                       prob=c(0.6, 0.3, 0.05, 0.05),
                                       replace=TRUE))

head(data$tipo_local, 5)
```

```
## [1] "assent" "assent" "resex"  "assent" "resex"
```

```
head(data[,5], 5)
```

```
## [1] "assent" "assent" "resex"  "assent" "resex"
```

Exportando bases no R

Code Chunk 6: Exportando bases

```
# gerando directorio
if(dir.exists("./01 - Bases")){
  unlink("./01 - Bases", recursive = TRUE)
}
dir.create("./01 - Bases")

## Warning in dir.create("./01 - Bases"): './01 - Bases' already exists

# em formato nativo do R
save(data, file="./01 - Bases/teste_data.Rdata")
# em arquivo de texto (txt, csv etc)
write.table(data, file = "./01 - Bases/teste_data.txt", sep = " ",
  row.names = FALSE)
write.table(data, file = "./01 - Bases/teste_data.csv", sep = ";",
  row.names = FALSE)
write_csv2(data, file="./01 - Bases/teste_data.csv", append = TRUE)
# write_delim é outra opção

# em arquivo formato excel (requer package openxlsx)
write.xlsx(data, file="./01 - Bases/teste_data.xlsx")

# em arquivo STATA (requer package: haven)
write_dta(data, path = "./01 - Bases/teste_data.dta")
```


Importando dados

Code Chunk 7: Importando dados no R

```
# R native file
load(file="./01 - Bases/teste_data.Rdata")

# arquivo em formato Stata
data <- read_dta("./01 - Bases/teste_data.dta")

# arquivo de texto (txt, csv etc)
data <- read_delim(file="./01 - Bases/teste_data.txt", delim = " ")
data <- read_csv2(file="./01 - Bases/teste_data.txt")

# arquivo em formato excel
data <- read_xlsx("./01 - Bases/teste_data.xlsx", sheet = 1)
str(data)

## 'data.frame':    1000 obs. of  5 variables:
## $ id          : chr  "pf_1" "pf_2" "pf_3" "pf_4" ...
## $ A           : num  0.3 0.1 0.1 0.3 0.1 0.3 0.1 0.1 0.3 0.3 ...
## $ k           : num  12.2 14.8 12.6 22.2 17.3 ...
## $ l           : num  6.81 5.91 5.88 4.85 4.53 ...
## $ tipo_local : chr  "assent" "assent" "resex" "assent" ...
```

Code Chunk 8: Criação de variáveis

```
data <- data %>%
  mutate(alpha=0.5,
         y=A*l^alpha*k^(1-alpha)*1000,
         across(matches("[a-z]{1}$"), log, .names = "log_{.col}"),
         true_status=case_when(y/12<218/2 ~ "extrem.pobreza",
                               y/12<218 ~ "pobreza",
                               TRUE ~ "regular")) %>%
  relocate(k, log_l, log_k, .after=1)
head(data[,c("id", "y", "true_status")])
```

```
##      id      y      true_status
## 1 pf_1 2737.8516      regular
## 2 pf_2  936.3718 extrem.pobreza
## 3 pf_3  860.7327 extrem.pobreza
## 4 pf_4 3116.2447      regular
## 5 pf_5  884.1046 extrem.pobreza
## 6 pf_6 1900.1876      pobreza
```

Code Chunk 9: bases de dados relacionadas

```
# criando informação de elegibilidade
data.elegib <- data %>%
  mutate(observavel=rbinom(sample_n, 1, 0.9),
         obs_status=case_when(observavel==1 ~ true_status,
                              TRUE ~ NA)) %>%

  select(id, obs_status)

data %<>%
  left_join(data.elegib, by="id") %>%
  mutate(elegivel=case_when(tipo_local %in% c("resex", "assent")
                           & obs_status %in%
                           c("extrem.pobreza", "pobreza") ~ 1,
                           TRUE ~ 0))

# existem outras opções de join: (inner_join, right_join, full_join)
head(select(data, id, tipo_local, elegivel), 4)
```

```
##      id tipo_local elegivel
## 1 pf_1      assent         0
## 2 pf_2      assent         1
## 3 pf_3      resex         1
## 4 pf_4      assent         0
```

Code Chunk 10: colapsando a base de dados

```
data.temp <- data %>%  
  filter(!is.na(obs_status)) %>%  
  group_by(elegivel) %>%  
  summarise(across(matches("^[a-z]{1}$"), mean, .names="mean_{.col}"),  
            across(matches("^[a-z]{1}$"), sd, .names="sd_{.col}"),  
            n=n_distinct(id))  
data.temp
```

```
## # A tibble: 2 x 10  
##   elegivel mean_A mean_l mean_k mean_y sd_A sd_l sd_k sd_y n  
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>  
## 1      0 0.487  5.30  16.0 4285. 0.403 0.968 3.04 3378. 342  
## 2      1 0.168  4.81  14.5 1350. 0.0950 0.965 3.02 704. 561
```

Code Chunk 11: incluindo variáveis associadas a grupos

```
data.temp <- data %>%  
  select(l, k, tipo_local) %>%  
  group_by(tipo_local) %>%  
  mutate(across(matches("[a-z]{1}$"), mean, .names="mean_{.col}"),  
         across(matches("[a-z]{1}$"), sd, .names="sd_{.col}")) %>%  
  ungroup()  
head(data.temp, 5)
```

```
## # A tibble: 5 x 7  
##       l      k tipo_local mean_l mean_k sd_l sd_k  
##   <dbl> <dbl> <chr>      <dbl> <dbl> <dbl> <dbl>  
## 1  6.81  12.2 assent      5.00  15.0 0.991  2.98  
## 2  5.91  14.8 assent      5.00  15.0 0.991  2.98  
## 3  5.88  12.6 resex       4.99  15.1 0.993  3.14  
## 4  4.85  22.2 assent      5.00  15.0 0.991  2.98  
## 5  4.53  17.3 resex       4.99  15.1 0.993  3.14
```

Code Chunk 12: incluindo observações

```
data.temp <- data.frame("l"=rnorm(n=100, mean=5, sd=1),  
                        "k"=rnorm(n=100, mean=10, sd=2),  
                        "tipo_local"=rep(c("assent", "outro_rural"), 50)) %>%  
  rbind(select(data, l, k, tipo_local))  
# rbindlist funciona para uma lista com múltiplas bases  
# (requer package data.table)  
str(data.temp)
```

```
## 'data.frame':    1100 obs. of  3 variables:  
## $ l            : num  4.81 5.54 5.05 3.87 6.23 ...  
## $ k            : num  11.83 10.15 11.55 11.11 6.38 ...  
## $ tipo_local: chr  "assent" "outro_rural" "assent" "outro_rural" ...
```

Estrutura para looping

Code Chunk 13: Usando for para implementar looping

Criando status de tratamento

```
data$tratado <- NA
```

```
for (i in 1:nrow(data)){
```

```
  if(data$elegivel[i]==1){
```

```
    data$tratado[i] <- rbinom(1, 1, 0.9)
```

```
  }else{
```

```
    data$tratado[i] <- rbinom(1, 1, 0.1)
```

```
  }
```

```
}
```

```
table(data$tratado, data$elegivel, dnn = c("tratado", "elegivel"))
```

```
##          elegivel
```

```
## tratado    0    1
```

```
##          0 400  68
```

```
##          1  39 493
```

Criando functions

Code Chunk 14: Criando functions

```
# Criando function
get_id_info <- function(temp.id, source=data, s=FALSE){
  temp <- filter(source,id==temp.id) %>%
    select(id, elegivel) %>%
    mutate(tratado=case_when(elegivel==1 ~ rbinom(1, 1, 0.9),
                             TRUE ~ rbinom(1, 1, 0.1)))

  if(s){output <- temp$tratado}else{
    output <- list("id"=temp$id, "status"=temp$elegivel,
                  "tratamento"=temp$tratado)}
  output
}
dump("get_id_info", file = "get_id_info.R")
rm(get_id_info)
source("./get_id_info.R")
unlist(get_id_info("pf_10"))
```

```
##           id      status tratamento
##    "pf_10"      "1"      "1"
```


lapply, sapply, mapply...

Code Chunk 15: Looping sem usar um *for*

```
# Criando function
```

```
data %<>%
```

```
  select(-tratado)
```

```
names(data)
```

```
## [1] "id"          "A"           "l"           "k"           "log_l"
```

```
## [6] "log_k"       "tipo_local"  "alpha"       "y"           "log_A"
```

```
## [11] "log_y"       "true_status" "obs_status"  "elegivel"
```

```
data$tratado <- sapply(data$id, get_id_info, s=TRUE)
```

```
table(data$tratado, data$elegivel, dnn = c("tratado", "elegivel"))
```

```
##          elegivel
```

```
## tratado    0    1
```

```
##          0 384  51
```

```
##          1  55 510
```

Dúvidas com R

- ❖ Google:
 - ❖ 'how to [o que você quiser] R'
 - ❖ 'how to [o que você quiser] R Cran' (se estiver procurando por um package específico)
 - ❖ Fóruns importantes para dúvidas de programação são [stackoverflow](#) e [R-Bloggers](#)
- ❖ Programadores de R adoram montar Cheat sheets, que são pequenos resumos de comandos. Elas são muito úteis (principalmente no início). Salvamos diversas delas no grupo de treinamento do DAE.

Referências estruturadas

- ❖ R Basics: *R programming for data science*, *R for data science* e *R Cookbook*
- ❖ Manipulação de dados: as referências de *dplyr* e *tidyr*
- ❖ Gráficos: as referências de *ggplot2* e *R Graph Gallery*
- ❖ Tabelas: as referências de *Kable* e *Stargazer*
- ❖ RMarkdown: *The Definitive guide*
- ❖ Econometrics: *Causal inference: the mixtape* e *The Effect*

Hands on!

Simulação...