

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

- ❖ Todas as aulas terão uma parte conceitual e uma parte dedicada a programação aplicada
- ❖ A avaliação será feita por meio de participação em sala (20%) e 8 listas de exercícios (10% cada).
- ❖ Todo o material do curso será postado numa página do *Github*

# Conceitos básicos



# Espaço amostral e evento

- ❖ Um **espaço amostral** ( $\Omega$ ) 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  $\Omega$  é enumerável, então  $P(A) = \frac{\text{Qtd. de elementos de } A}{\text{Qtd. de elementos em } \Omega}$
- ❖ Se  $\Omega$  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  $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$

# Proriedades da Probabilidade

- ❖ (i)  $P(A) = 1 - P(A^C)$
- ❖ (ii) Sendo A e B dois eventos quaisquer, vale que  $P(B) = P(B \cap A) + P(B \cap A^C)$
- ❖ (iii) Se  $A \subset B$ , então  $P(A) \leq P(B)$
- ❖ (iv) Se  $A \subset B$ , então  $P(A) \leq P(B)$
- ❖ (v) Para quaisquer eventos  $A_1, A_2, \dots$ , vale que:  
$$P\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} P(A_i)$$

# 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.
  - $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A)$

# O Básico de 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 shows the RStudio IDE interface. The main editor window contains R code for creating and manipulating data structures. The Environment pane on the right shows the current workspace with variables like `lista_1`, `matrix_1`, `vetor_1`, and `vetor_2`. The R Documentation pane at the bottom right shows the documentation for the `Matrix Transpose` function.

```

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 - # criando
136 - matriz_1 <- matrix(1:20, nrow=4, byrow=TRUE)
137 - # editando
138 - matriz_2 <- matriz_1[1:2, ]
139 - # operações com matrizes
140 - matriz_2[2, ]
141 -
142 - [1,] 1 3 5 7 9
143 - [2,] 2 4 6 8 10
144 -
145 - [1,] 1 2 3 4 5
146 - [2,] 2 4 6 8 10
147 -
148 - [1,] 1 2 3 4 5
149 - [2,] 2 4 6 8 10
150 -
151 - [1,] 1 2 3 4 5
152 - [2,] 2 4 6 8 10
153 -
154 - [1,] 1 2 3 4 5
155 - [2,] 2 4 6 8 10
156 -
157 - [1,] 1 2 3 4 5
158 - [2,] 2 4 6 8 10
159 -
160 - [1,] 1 2 3 4 5
161 - [2,] 2 4 6 8 10
162 -
163 - [1,] 1 2 3 4 5
164 - [2,] 2 4 6 8 10
165 -
166 - [1,] 1 2 3 4 5
167 - [2,] 2 4 6 8 10
168 -
169 - [1,] 1 2 3 4 5
170 - [2,] 2 4 6 8 10
171 -
172 - [1,] 1 2 3 4 5
173 - [2,] 2 4 6 8 10
174 -
175 - [1,] 1 2 3 4 5
176 - [2,] 2 4 6 8 10
177 -
178 - [1,] 1 2 3 4 5
179 - [2,] 2 4 6 8 10
180 -
181 - [1,] 1 2 3 4 5
182 - [2,] 2 4 6 8 10
183 -
184 - [1,] 1 2 3 4 5
185 - [2,] 2 4 6 8 10
186 -
187 - [1,] 1 2 3 4 5
188 - [2,] 2 4 6 8 10
189 -
190 - [1,] 1 2 3 4 5
191 - [2,] 2 4 6 8 10
192 -
193 - [1,] 1 2 3 4 5
194 - [2,] 2 4 6 8 10
195 -
196 - [1,] 1 2 3 4 5
197 - [2,] 2 4 6 8 10
198 -
199 - [1,] 1 2 3 4 5
200 - [2,] 2 4 6 8 10
201 -
202 - [1,] 1 2 3 4 5
203 - [2,] 2 4 6 8 10
204 -
205 - [1,] 1 2 3 4 5
206 - [2,] 2 4 6 8 10
207 -
208 - [1,] 1 2 3 4 5
209 - [2,] 2 4 6 8 10
210 -
211 - [1,] 1 2 3 4 5
212 - [2,] 2 4 6 8 10
213 -
214 - [1,] 1 2 3 4 5
215 - [2,] 2 4 6 8 10
216 -
217 - [1,] 1 2 3 4 5
218 - [2,] 2 4 6 8 10
219 -
220 - [1,] 1 2 3 4 5
221 - [2,] 2 4 6 8 10
222 -
223 - [1,] 1 2 3 4 5
224 - [2,] 2 4 6 8 10
225 -
226 - [1,] 1 2 3 4 5
227 - [2,] 2 4 6 8 10
228 -
229 - [1,] 1 2 3 4 5
230 - [2,] 2 4 6 8 10
231 -
232 - [1,] 1 2 3 4 5
233 - [2,] 2 4 6 8 10
234 -
235 - [1,] 1 2 3 4 5
236 - [2,] 2 4 6 8 10
237 -
238 - [1,] 1 2 3 4 5
239 - [2,] 2 4 6 8 10
240 -
241 - [1,] 1 2 3 4 5
242 - [2,] 2 4 6 8 10
243 -
244 - [1,] 1 2 3 4 5
245 - [2,] 2 4 6 8 10
246 -
247 - [1,] 1 2 3 4 5
248 - [2,] 2 4 6 8 10
249 -
250 - [1,] 1 2 3 4 5
251 - [2,] 2 4 6 8 10
252 -
253 - [1,] 1 2 3 4 5
254 - [2,] 2 4 6 8 10
255 -
256 - [1,] 1 2 3 4 5
257 - [2,] 2 4 6 8 10
258 -
259 - [1,] 1 2 3 4 5
260 - [2,] 2 4 6 8 10
261 -
262 - [1,] 1 2 3 4 5
263 - [2,] 2 4 6 8 10
264 -
265 - [1,] 1 2 3 4 5
266 - [2,] 2 4 6 8 10
267 -
268 - [1,] 1 2 3 4 5
269 - [2,] 2 4 6 8 10
270 -
271 - [1,] 1 2 3 4 5
272 - [2,] 2 4 6 8 10
273 -
274 - [1,] 1 2 3 4 5
275 - [2,] 2 4 6 8 10
276 -
277 - [1,] 1 2 3 4 5
278 - [2,] 2 4 6 8 10
279 -
280 - [1,] 1 2 3 4 5
281 - [2,] 2 4 6 8 10
282 -
283 - [1,] 1 2 3 4 5
284 - [2,] 2 4 6 8 10
285 -
286 - [1,] 1 2 3 4 5
287 - [2,] 2 4 6 8 10
288 -
289 - [1,] 1 2 3 4 5
290 - [2,] 2 4 6 8 10
291 -
292 - [1,] 1 2 3 4 5
293 - [2,] 2 4 6 8 10
294 -
295 - [1,] 1 2 3 4 5
296 - [2,] 2 4 6 8 10
297 -
298 - [1,] 1 2 3 4 5
299 - [2,] 2 4 6 8 10
300 -
301 - [1,] 1 2 3 4 5
302 - [2,] 2 4 6 8 10
303 -
304 - [1,] 1 2 3 4 5
305 - [2,] 2 4 6 8 10
306 -
307 - [1,] 1 2 3 4 5
308 - [2,] 2 4 6 8 10
309 -
310 - [1,] 1 2 3 4 5
311 - [2,] 2 4 6 8 10
312 -
313 - [1,] 1 2 3 4 5
314 - [2,] 2 4 6 8 10
315 -
316 - [1,] 1 2 3 4 5
317 - [2,] 2 4 6 8 10
318 -
319 - [1,] 1 2 3 4 5
320 - [2,] 2 4 6 8 10
321 -
322 - [1,] 1 2 3 4 5
323 - [2,] 2 4 6 8 10
324 -
325 - [1,] 1 2 3 4 5
326 - [2,] 2 4 6 8 10
327 -
328 - [1,] 1 2 3 4 5
329 - [2,] 2 4 6 8 10
330 -
331 - [1,] 1 2 3 4 5
332 - [2,] 2 4 6 8 10
333 -
334 - [1,] 1 2 3 4 5
335 - [2,] 2 4 6 8 10
336 -
337 - [1,] 1 2 3 4 5
338 - [2,] 2 4 6 8 10
339 -
340 - [1,] 1 2 3 4 5
341 - [2,] 2 4 6 8 10
342 -
343 - [1,] 1 2 3 4 5
344 - [2,] 2 4 6 8 10
345 -
346 - [1,] 1 2 3 4 5
347 - [2,] 2 4 6 8 10
348 -
349 - [1,] 1 2 3 4 5
350 - [2,] 2 4 6 8 10
351 -
352 - [1,] 1 2 3 4 5
353 - [2,] 2 4 6 8 10
354 -
355 - [1,] 1 2 3 4 5
356 - [2,] 2 4 6 8 10
357 -
358 - [1,] 1 2 3 4 5
359 - [2,] 2 4 6 8 10
360 -
361 - [1,] 1 2 3 4 5
362 - [2,] 2 4 6 8 10
363 -
364 - [1,] 1 2 3 4 5
365 - [2,] 2 4 6 8 10
366 -
367 - [1,] 1 2 3 4 5
368 - [2,] 2 4 6 8 10
369 -
370 - [1,] 1 2 3 4 5
371 - [2,] 2 4 6 8 10
372 -
373 - [1,] 1 2 3 4 5
374 - [2,] 2 4 6 8 10
375 -
376 - [1,] 1 2 3 4 5
377 - [2,] 2 4 6 8 10
378 -
379 - [1,] 1 2 3 4 5
380 - [2,] 2 4 6 8 10
381 -
382 - [1,] 1 2 3 4 5
383 - [2,] 2 4 6 8 10
384 -
385 - [1,] 1 2 3 4 5
386 - [2,] 2 4 6 8 10
387 -
388 - [1,] 1 2 3 4 5
389 - [2,] 2 4 6 8 10
390 -
391 - [1,] 1 2 3 4 5
392 - [2,] 2 4 6 8 10
393 -
394 - [1,] 1 2 3 4 5
395 - [2,] 2 4 6 8 10
396 -
397 - [1,] 1 2 3 4 5
398 - [2,] 2 4 6 8 10
399 -
400 - [1,] 1 2 3 4 5
401 - [2,] 2 4 6 8 10
402 -
403 - [1,] 1 2 3 4 5
404 - [2,] 2 4 6 8 10
405 -
406 - [1,] 1 2 3 4 5
407 - [2,] 2 4 6 8 10
408 -
409 - [1,] 1 2 3 4 5
410 - [2,] 2 4 6 8 10
411 -
412 - [1,] 1 2 3 4 5
413 - [2,] 2 4 6 8 10
414 -
415 - [1,] 1 2 3 4 5
416 - [2,] 2 4 6 8 10
417 -
418 - [1,] 1 2 3 4 5
419 - [2,] 2 4 6 8 10
420 -
421 - [1,] 1 2 3 4 5
422 - [2,] 2 4 6 8 10
423 -
424 - [1,] 1 2 3 4 5
425 - [2,] 2 4 6 8 10
426 -
427 - [1,] 1 2 3 4 5
428 - [2,] 2 4 6 8 10
429 -
430 - [1,] 1 2 3 4 5
431 - [2,] 2 4 6 8 10
432 -
433 - [1,] 1 2 3 4 5
434 - [2,] 2 4 6 8 10
435 -
436 - [1,] 1 2 3 4 5
437 - [2,] 2 4 6 8 10
438 -
439 - [1,] 1 2 3 4 5
440 - [2,] 2 4 6 8 10
441 -
442 - [1,] 1 2 3 4 5
443 - [2,] 2 4 6 8 10
444 -
445 - [1,] 1 2 3 4 5
446 - [2,] 2 4 6 8 10
447 -
448 - [1,] 1 2 3 4 5
449 - [2,] 2 4 6 8 10
450 -
451 - [1,] 1 2 3 4 5
452 - [2,] 2 4 6 8 10
453 -
454 - [1,] 1 2 3 4 5
455 - [2,] 2 4 6 8 10
456 -
457 - [1,] 1 2 3 4 5
458 - [2,] 2 4 6 8 10
459 -
460 - [1,] 1 2 3 4 5
461 - [2,] 2 4 6 8 10
462 -
463 - [1,] 1 2 3 4 5
464 - [2,] 2 4 6 8 10
465 -
466 - [1,] 1 2 3 4 5
467 - [2,] 2 4 6 8 10
468 -
469 - [1,] 1 2 3 4 5
470 - [2,] 2 4 6 8 10
471 -
472 - [1,] 1 2 3 4 5
473 - [2,] 2 4 6 8 10
474 -
475 - [1,] 1 2 3 4 5
476 - [2,] 2 4 6 8 10
477 -
478 - [1,] 1 2 3 4 5
479 - [2,] 2 4 6 8 10
480 -
481 - [1,] 1 2 3 4 5
482 - [2,] 2 4 6 8 10
483 -
484 - [1,] 1 2 3 4 5
485 - [2,] 2 4 6 8 10
486 -
487 - [1,] 1 2 3 4 5
488 - [2,] 2 4 6 8 10
489 -
490 - [1,] 1 2 3 4 5
491 - [2,] 2 4 6 8 10
492 -
493 - [1,] 1 2 3 4 5
494 - [2,] 2 4 6 8 10
495 -
496 - [1,] 1 2 3 4 5
497 - [2,] 2 4 6 8 10
498 -
499 - [1,] 1 2 3 4 5
500 - [2,] 2 4 6 8 10
501 -
502 - [1,] 1 2 3 4 5
503 - [2,] 2 4 6 8 10
504 -
505 - [1,] 1 2 3 4 5
506 - [2,] 2 4 6 8 10
507 -
508 - [1,] 1 2 3 4 5
509 - [2,] 2 4 6 8 10
510 -
511 - [1,] 1 2 3 4 5
512 - [2,] 2 4 6 8 10
513 -
514 - [1,] 1 2 3 4 5
515 - [2,] 2 4 6 8 10
516 -
517 - [1,] 1 2 3 4 5
518 - [2,] 2 4 6 8 10
519 -
520 - [1,] 1 2 3 4 5
521 - [2,] 2 4 6 8 10
522 -
523 - [1,] 1 2 3 4 5
524 - [2,] 2 4 6 8 10
525 -
526 - [1,] 1 2 3 4 5
527 - [2,] 2 4 6 8 10
528 -
529 - [1,] 1 2 3 4 5
530 - [2,] 2 4 6 8 10
531 -
532 - [1,] 1 2 3 4 5
533 - [2,] 2 4 6 8 10
534 -
535 - [1,] 1 2 3 4 5
536 - [2,] 2 4 6 8 10
537 -
538 - [1,] 1 2 3 4 5
539 - [2,] 2 4 6 8 10
540 -
541 - [1,] 1 2 3 4 5
542 - [2,] 2 4 6 8 10
543 -
544 - [1,] 1 2 3 4 5
545 - [2,] 2 4 6 8 10
546 -
547 - [1,] 1 2 3 4 5
548 - [2,] 2 4 6 8 10
549 -
550 - [1,] 1 2 3 4 5
551 - [2,] 2 4 6 8 10
552 -
553 - [1,] 1 2 3 4 5
554 - [2,] 2 4 6 8 10
555 -
556 - [1,] 1 2 3 4 5
557 - [2,] 2 4 6 8 10
558 -
559 - [1,] 1 2 3 4 5
560 - [2,] 2 4 6 8 10
561 -
562 - [1,] 1 2 3 4 5
563 - [2,] 2 4 6 8 10
564 -
565 - [1,] 1 2 3 4 5
566 - [2,] 2 4 6 8 10
567 -
568 - [1,] 1 2 3 4 5
569 - [2,] 2 4 6 8 10
570 -
571 - [1,] 1 2 3 4 5
572 - [2,] 2 4 6 8 10
573 -
574 - [1,] 1 2 3 4 5
575 - [2,] 2 4 6 8 10
576 -
577 - [1,] 1 2 3 4 5
578 - [2,] 2 4 6 8 10
579 -
580 - [1,] 1 2 3 4 5
581 - [2,] 2 4 6 8 10
582 -
583 - [1,] 1 2 3 4 5
584 - [2,] 2 4 6 8 10
585 -
586 - [1,] 1 2 3 4 5
587 - [2,] 2 4 6 8 10
588 -
589 - [1,] 1 2 3 4 5
590 - [2,] 2 4 6 8 10
591 -
592 - [1,] 1 2 3 4 5
593 - [2,] 2 4 6 8 10
594 -
595 - [1,] 1 2 3 4 5
596 - [2,] 2 4 6 8 10
597 -
598 - [1,] 1 2 3 4 5
599 - [2,] 2 4 6 8 10
600 -
601 - [1,] 1 2 3 4 5
602 - [2,] 2 4 6 8 10
603 -
604 - [1,] 1 2 3 4 5
605 - [2,] 2 4 6 8 10
606 -
607 - [1,] 1 2 3 4 5
608 - [2,] 2 4 6 8 10
609 -
610 - [1,] 1 2 3 4 5
611 - [2,] 2 4 6 8 10
612 -
613 - [1,] 1 2 3 4 5
614 - [2,] 2 4 6 8 10
615 -
616 - [1,] 1 2 3 4 5
617 - [2,] 2 4 6 8 10
618 -
619 - [1,] 1 2 3 4 5
620 - [2,] 2 4 6 8 10
621 -
622 - [1,] 1 2 3 4 5
623 - [2,] 2 4 6 8 10
624 -
625 - [1,] 1 2 3 4 5
626 - [2,] 2 4 6 8 10
627 -
628 - [1,] 1 2 3 4 5
629 - [2,] 2 4 6 8 10
630 -
631 - [1,] 1 2 3 4 5
632 - [2,] 2 4 6 8 10
633 -
634 - [1,] 1 2 3 4 5
635 - [2,] 2 4 6 8 10
636 -
637 - [1,] 1 2 3 4 5
638 - [2,] 2 4 6 8 10
639 -
640 - [1,] 1 2 3 4 5
641 - [2,] 2 4 6 8 10
642 -
643 - [1,] 1 2 3 4 5
644 - [2,] 2 4 6 8 10
645 -
646 - [1,] 1 2 3 4 5
647 - [2,] 2 4 6 8 10
648 -
649 - [1,] 1 2 3 4 5
650 - [2,] 2 4 6 8 10
651 -
652 - [1,] 1 2 3 4 5
653 - [2,] 2 4 6 8 10
654 -
655 - [1,] 1 2 3 4 5
656 - [2,] 2 4 6 8 10
657 -
658 - [1,] 1 2 3 4 5
659 - [2,] 2 4 6 8 10
660 -
661 - [1,] 1 2 3 4 5
662 - [2,] 2 4 6 8 10
663 -
664 - [1,] 1 2 3 4 5
665 - [2,] 2 4 6 8 10
666 -
667 - [1,] 1 2 3 4 5
668 - [2,] 2 4 6 8 10
669 -
670 - [1,] 1 2 3 4 5
671 - [2,] 2 4 6 8 10
672 -
673 - [1,] 1 2 3 4 5
674 - [2,] 2 4 6 8 10
675 -
676 - [1,] 1 2 3 4 5
677 - [2,] 2 4 6 8 10
678 -
679 - [1,] 1 2 3 4 5
680 - [2,] 2 4 6 8 10
681 -
682 - [1,] 1 2 3 4 5
683 - [2,] 2 4 6 8 10
684 -
685 - [1,] 1 2 3 4 5
686 - [2,] 2 4 6 8 10
687 -
688 - [1,] 1 2 3 4 5
689 - [2,] 2 4 6 8 10
690 -
691 - [1,] 1 2 3 4 5
692 - [2,] 2 4 6 8 10
693 -
694 - [1,] 1 2 3 4 5
695 - [2,] 2 4 6 8 10
696 -
697 - [1,] 1 2 3 4 5
698 - [2,] 2 4 6 8 10
699 -
700 - [1,] 1 2 3 4 5
701 - [2,] 2 4 6 8 10
702 -
703 - [1,] 1 2 3 4 5
704 - [2,] 2 4 6 8 10
705 -
706 - [1,] 1 2 3 4 5
707 - [2,] 2 4 6 8 10
708 -
709 - [1,] 1 2 3 4 5
710 - [2,] 2 4 6 8 10
711 -
712 - [1,] 1 2 3 4 5
713 - [2,] 2 4 6 8 10
714 -
715 - [1,] 1 2 3 4 5
716 - [2,] 2 4 6 8 10
717 -
718 - [1,] 1 2 3 4 5
719 - [2,] 2 4 6 8 10
720 -
721 - [1,] 1 2 3 4 5
722 - [2,] 2 4 6 8 10
723 -
724 - [1,] 1 2 3 4 5
725 - [2,] 2 4 6 8 10
726 -
727 - [1,] 1 2 3 4 5
728 - [2,] 2 4 6 8 10
729 -
730 - [1,] 1 2 3 4 5
731 - [2,] 2 4 6 8 10
732 -
733 - [1,] 1 2 3 4 5
734 - [2,] 2 4 6 8 10
735 -
736 - [1,] 1 2 3 4 5
737 - [2,] 2 4 6 8 10
738 -
739 - [1,] 1 2 3 4 5
740 - [2,] 2 4 6 8 10
741 -
742 - [1,] 1 2 3 4 5
743 - [2,] 2 4 6 8 10
744 -
745 - [1,] 1 2 3 4 5
746 - [2,] 2 4 6 8 10
747 -
748 - [1,] 1 2 3 4 5
749 - [2,] 2 4 6 8 10
750 -
751 - [1,] 1 2 3 4 5
752 - [2,] 2 4 6 8 10
753 -
754 - [1,] 1 2 3 4 5
755 - [2,] 2 4 6 8 10
756 -
757 - [1,] 1 2 3 4 5
758 - [2,] 2 4 6 8 10
759 -
760 - [1,] 1 2 3 4 5
761 - [2,] 2 4 6 8 10
762 -
763 - [1,] 1 2 3 4 5
764 - [2,] 2 4 6 8 10
765 -
766 - [1,] 1 2 3 4 5
767 - [2,] 2 4 6 8 10
768 -
769 - [1,] 1 2 3 4 5
770 - [2,] 2 4 6 8 10
771 -
772 - [1,] 1 2 3 4 5
773 - [2,] 2 4 6 8 10
774 -
775 - [1,] 1 2 3 4 5
776 - [2,] 2 4 6 8 10
777 -
778 - [1,] 1 2 3 4 5
779 - [2,] 2 4 6 8 10
780 -
781 - [1,] 1 2 3 4 5
782 - [2,] 2 4 6 8 10
783 -
784 - [1,] 1 2 3 4 5
785 - [2,] 2 4 6 8 10
786 -
787 - [1,] 1 2 3 4 5
788 - [2,] 2 4 6 8 10
789 -
790 - [1,] 1 2 3 4 5
791 - [2,] 2 4 6 8 10
792 -
793 - [1,] 1 2 3 4 5
794 - [2,] 2 4 6 8 10
795 -
796 - [1,] 1 2 3 4 5
797 - [2,] 2 4 6 8 10
798 -
799 - [1,] 1 2 3 4 5
800 - [2,] 2 4 6 8 10
801 -
802 - [1,] 1 2 3 4 5
803 - [2,] 2 4 6 8 10
804 -
805 - [1,] 1 2 3 4 5
806 - [2,] 2 4 6 8 10
807 -
808 - [1,] 1 2 3 4 5
809 - [2,] 2 4 6 8 10
810 -
811 - [1,] 1 2 3 4 5
812 - [2,] 2 4 6 8 10
813 -
814 - [1,] 1 2 3 4 5
815 - [2,] 2 4 6 8 10
816 -
817 - [1,] 1 2 3 4 5
818 - [2,] 2 4 6 8 10
819 -
820 - [1,] 1 2 3 4 5
821 - [2,] 2 4 6 8 10
822 -
823 - [1,] 1 2 3 4 5
824 - [2,] 2 4 6 8 10
825 -
826 - [1,] 1 2 3 4 5
827 - [2,] 2 4 6 8 10
828 -
829 - [1,] 1 2 3 4 5
830 - [2,] 2 4 6 8 10
831 -
832 - [1,] 1 2 3 4 5
833 - [2,] 2 4 6 8 10
834 -
835 - [1,] 1 2 3 4 5
836 - [2,] 2 4 6 8 10
837 -
838 - [1,] 1 2 3 4 5
839 - [2,] 2 4 6 8 10
840 -
841 - [1,] 1 2 3 4 5
842 - [2,] 2 4 6 8 10
843 -
844 - [1,] 1 2 3 4 5
845 - [2,] 2 4 6 8 10
846 -
847 - [1,] 1 2 3 4 5
848 - [2,] 2 4 6 8 10
849 -
850 - [1,] 1 2 3 4 5
851 - [2,] 2 4 6 8 10
852 -
853 - [1,] 1 2 3 4 5
854 - [2,] 2 4 6 8 10
855 -
856 - [1,] 1 2 3 4 5
857 - [2,] 2 4 6 8 10
858 -
859 - [1,] 1 2 3 4 5
860 - [2,] 2 4 6 8 10
861 -
862 - [1,] 1 2 3 4 5
863 - [2,] 2 4 6 8 10
864 -
865 - [1,] 1 2 3 4 5
866 - [2,] 2 4 6 8 10
867 -
868 - [1,] 1 2 3 4 5
869 - [2,] 2 4 6 8 10
870 -
871 - [1,] 1 2 3 4 5
872 - [2,] 2 4 6 8 10
873 -
874 - [1,] 1 2 3 4 5
875 - [2,] 2 4 6 8 10
876 -
877 - [1,] 1 2 3 4 5
878 - [2,] 2 4 6 8 10
879 -
880 - [1,] 1 2 3 4 5
881 - [2,] 2 4 6 8 10
882 -
883 - [1,] 1 2 3 4 5
884 - [2,] 2 4 6 8 10
885 -
886 - [1,] 1 2 3 4 5
887 - [2,] 2 4 6 8 10
888 -
889 - [1,] 1 2 3 4 5
890 - [2,] 2 4 6 8 10
891 -
892 - [1,] 1 2 3 4 5
893 - [2,] 2 4 6 8 10
894 -
895 - [1,] 1 2 3 4 5
896 - [2,] 2 4 6 8 10
897 -
898 - [1,] 1 2 3 4 5
899 - [2,] 2 4 6 8 10
900 -
901 - [1,] 1 2 3 4 5
902 - [2,] 2 4 6 8 10
903 -
904 - [1,] 1 2 3 4 5
905 - [2,] 2 4 6 8 10
906 -
907 - [1,] 1 2 3 4 5
908 - [2,] 2 4 6 8 10
909 -
910 - [1,] 1 2 3 4 5
911 - [2,] 2 4 6 8 10
912 -
913 - [1,] 1 2 3 4 5
914 - [2,] 2 4 6 8 10
915 -
916 - [1,] 1 2 3 4 5
917 - [2,] 2 4 6 8 10
918 -
919 - [1,] 1 2 3 4 5
920 - [2,] 2 4 6 8 10
921 -
922 - [1,] 1 2 3 4 5
923 - [2,] 2 4 6 8 10
924 -
925 - [1,] 1 2 3 4 5
926 - [2,] 2 4 6 8 10
927 -
928 - [1,] 1 2 3 4 5
929 - [2,] 2 4 6 8 10
930 -
931 - [1,] 1 2 3 4 5
932 - [2,] 2 4 6 8 10
933 -
934 - [1,] 1 2 3 4 5
935 - [2,] 2 4 6 8 10
936 -
937 - [1,] 1 2 3 4 5
938 - [2,] 2 4 6 8 10
939 -
940 - [1,] 1 2 3 4 5
941 - [2,] 
```

# 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=10000
data <- data.frame("id"=paste("cpf", 1:sample_n, sep="_"),
                   "tipo_local"=sample(c("urbano", "rural"),
                                       size=sample_n,
                                       prob=c(0.84, 1-0.84),
                                       replace=TRUE),
                   "cor"=sample(c("branca", "preta", "parda"),
                               size=sample_n,
                               prob=c(0.48, 0.08, 0.44),
                               replace=TRUE))

head(data, 5)
```

```
##      id tipo_local  cor
## 1 cpf_1      rural  parda
## 2 cpf_2      urbano  parda
## 3 cpf_3      urbano branca
## 4 cpf_4      urbano  parda
## 5 cpf_5      rural  branca
```



# Exportando bases no R

## Code Chunk 6: Exportando bases

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

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

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

## 'data.frame':    10000 obs. of  3 variables:
## $ id           : chr  "cpf_1" "cpf_2" "cpf_3" "cpf_4" ...
## $ tipo_local   : chr  "rural" "urbano" "urbano" "urbano" ...
## $ cor          : chr  "parda" "parda" "branca" "parda" ...
```

# Download de arquivos

## Code Chunk 7: Download de arquivos

```
# Dados de Rendimento domiciliar per capita do Censo
tab3278 <- get_sidra(api="/t/3278/n1/all/v/allxp/p/all/c386/allxt/c1/allxt/c86/
save(tab3278, file="01 - Bases/Tab3278.Rdata")
tab3278 <- tab3278[,c(5, 12, 14, 16)]
names(tab3278) <- c("qtd", "classe_renda", "tipo_local", "cor")
head(tab3278)
```

##		qtd	classe_renda	tipo_local	cor
##	2	3603733	9681	1	2776
##	3	1436581	9681	1	2777
##	4	139021	9681	1	2778
##	5	7837682	9681	1	2779
##	6	48947	9681	1	2780
##	7	1944916	9681	2	2776

# mutate

```
tab3278 %<>%
  mutate(classe_renda = case_when(classe_renda<=9683 ~ "1sm",
                                   classe_renda>9683 ~ "1sm+",
                                   TRUE ~ "2sm+"),
         tipo_local = case_when(tipo_local==1 ~ "urbano",
                                   TRUE ~ "rural"),
         cor = case_when(cor==2776 ~ "branca",
                           cor==2777 ~ "preta",
                           TRUE ~ "parda"))

head(tab3278)
```

##	qtd	classe_renda	tipo_local	cor
## 2	3603733	1sm	urbano	branca
## 3	1436581	1sm	urbano	preta
## 4	139021	1sm	urbano	parda
## 5	7837682	1sm	urbano	parda
## 6	48947	1sm	urbano	parda
## 7	1944916	1sm	rural	branca

## Code Chunk 9: agregando a base de dados

```
tab.probs <- tab3278 %>%  
  filter(!is.na(qtd)) %>%  
  group_by(cor, tipo_local, classe_renda) %>%  
  summarise(qtd = sum(qtd)) %>%  
  group_by(cor, tipo_local) %>%  
  mutate(qtd_grupo = sum(qtd)) %>%  
  ungroup() %>%  
  mutate(prob_renda_cond = qtd / qtd_grupo)  
head(tab.probs[tab.probs$cor=="parda",])
```

```
## # A tibble: 4 x 6  
##   cor   tipo_local classe_renda      qtd qtd_grupo prob_renda_cond  
##   <chr> <chr>      <chr>      <dbl>    <dbl>         <dbl>  
## 1 parda rural      1sm        11763578  13543289      0.869  
## 2 parda rural      1sm+        1779711  13543289      0.131  
## 3 parda urbano     1sm        40286442  57265598      0.704  
## 4 parda urbano     1sm+        16979156  57265598      0.296
```

## Code Chunk 10: bases de dados relacionadas

```
# criando informação de elegibilidade
tab.probs %<>%
  filter(classe_renda=="1sm") %>%
  select(cor, tipo_local, prob_renda_cond)

data %<>%
  left_join(tab.probs, by=c("cor", "tipo_local"))

# existem outras opções de join: (inner_join, right_join, full_join)
head(data)
```

##	id	tipo_local	cor	prob_renda_cond
## 1	cpf_1	rural	parda	0.8685909
## 2	cpf_2	urbano	parda	0.7035016
## 3	cpf_3	urbano	branca	0.5279404
## 4	cpf_4	urbano	parda	0.7035016
## 5	cpf_5	rural	branca	0.7619917
## 6	cpf_6	urbano	branca	0.5279404

# Estrutura para looping

## Code Chunk 13: Usando for para implementar looping

```
# Criando status de renda
data$baixa.renda <- NA
for (i in 1:nrow(data)){
  data$baixa.renda[i] <- sample(0:1, size=1, prob=c(1-data$prob_renda_cond[i],
})
head(data)
```

##	id	tipo_local	cor	prob_renda_cond	baixa.renda
## 1	cpf_1	rural	parda	0.8685909	1
## 2	cpf_2	urbano	parda	0.7035016	1
## 3	cpf_3	urbano	branca	0.5279404	1
## 4	cpf_4	urbano	parda	0.7035016	1
## 5	cpf_5	rural	branca	0.7619917	1
## 6	cpf_6	urbano	branca	0.5279404	1

# Criando functions

## Code Chunk 14: Criando functions

*# Criando function*

```
gen_acesso_digital <- function(local, status.renda){  
  
  prob <- case_when(status.renda==1 & local=="urbano"~ 0.4,  
                    status.renda==1 & local=="rural"~ 0.2,  
                    status.renda==0 & local=="urbano"~ 0.75,  
                    status.renda==0 & local=="rural"~ 0.45,  
                    TRUE ~ 0.8)  
  
}  
dump("gen_acesso_digital", file = "gen_acesso_digital.R")  
rm(gen_acesso_digital)  
source("gen_acesso_digital.R")  
  
data %<>%  
  mutate(acesso_digital = gen_acesso_digital(tipo_local, baixa.renda))  
head(data, 2)
```

```
##      id tipo_local   cor prob_renda_cond baixa.renda acesso_digital  
## 1 cpf_1      rural parda      0.8685909          1          0.2  
## 2 cpf_2      urbano parda      0.7035016          1          0.4
```



# Hands on!

# Desenho do programa

- ❖ Desenhamos um programa de combate à pobreza. A população-alvo são indivíduos em famílias com renda per capita inferior a 1sm, mas não temos informação sobre a renda.
- ❖ Na prática, o programa consistirá em uma transferência incondicional de 1sm para qualquer indivíduo que aderir ao programa.
  - ❖ 90% dos indivíduos com renda de até 1sm estão propensos a aderir.
  - ❖ 40% dos indivíduos com renda superior a 1sm estão propensos a aderir.

# Desenho do programa

- ❖ Existem duas opções para o processo de adesão ao programa: (i) presencial ou (ii) digital (via Govbr).
- ❖ No caso da adesão presencial:
  - ❖ apenas 30% dos indivíduos em área rural (independente de renda) conseguirão se inscrever no programa pelas dificuldades de deslocamento;
  - ❖ 100% dos indivíduos em área urbana poderão se inscrever.

# Falha de focalização

- ❖ No caso da adesão por canal digital:
  - ❖ apenas indivíduos com acesso a conectividade (cuja probabilidade é dependente de renda e localidade) terão condições de aderir ao programa.
- ❖ Considerando as probabilidades, qual desenho deveria levar à menor falha de focalização?
  - ❖ Vamos definir falha de focalização como Qtd. de beneficiários que não fazem parte da população-alvo sobre o total de beneficiários.

# Falha de focalização

## Code Chunk 15: Estimando probabilidade de adesao

```
data %<>%  
  mutate(prop_adesao= case_when(baixa.renda==1 ~ 0.9,  
                                TRUE ~ 0.4),  
         viabilidade_presencial = case_when(tipo_local=="rural" ~ 0.3,  
                                             TRUE ~ 1),  
         adesao_presencial = prop_adesao*viabilidade_presencial,  
         adesao_digital = prop_adesao*acesso_digital)  
head(data, 3)
```

```
##      id tipo_local   cor prob_renda_cond baixa.renda acesso_digital  
## 1 cpf_1      rural  parda    0.8685909         1         0.2  
## 2 cpf_2      urbano  parda    0.7035016         1         0.4  
## 3 cpf_3      urbano branca    0.5279404         1         0.4  
##   prop_adesao viabilidade_presencial adesao_presencial adesao_digital  
## 1          0.9                0.3          0.27         0.18  
## 2          0.9                1.0          0.90         0.36  
## 3          0.9                1.0          0.90         0.36
```

# Falha de focalização

## Code Chunk 16: Estimando probabilidade de adesao

```
data$beneficiario.digital <- NA
data$beneficiario.presencial <- NA
for (i in 1:nrow(data)){
  data$beneficiario.digital[i] <- sample(0:1, size=1,
                                         prob=c(1-data$adesao_digital[i],
                                                  data$adesao_digital[i]))
  data$beneficiario.presencial[i] <- sample(0:1, size=1,
                                             prob=c(1-data$adesao_presencial[i],
                                                    data$adesao_presencial[i]))
}
head(data[,c("id", "baixa.renda",
             "beneficiario.digital", "beneficiario.presencial")], 3)
```

```
##      id baixa.renda beneficiario.digital beneficiario.presencial
## 1 cpf_1           1                0                0
## 2 cpf_2           1                0                1
## 3 cpf_3           1                1                0
```

# Falha de focalização

```
table(data$baixa.renda, data$beneficiario.digital,  
      dnn = c("Baixa Renda", "Beneficiario"))
```

```
##              Beneficiario  
## Baixa Renda    0    1  
##              0 2439  986  
##              1 4434 2141
```

```
table(data$baixa.renda, data$beneficiario.presencial,  
      dnn = c("Baixa Renda", "Beneficiario"))
```

```
##              Beneficiario  
## Baixa Renda    0    1  
##              0 2153 1272  
##              1 1460 5115
```

# Encerramento



# 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*