Trabalho de Amostragem

Arthur Marchito, Lucas Ávila e Natasha Ferrari

2023-07-09

Contents

1 Resumo

O estudo de simulação proposto tem como objetivo avaliar a Proficiência em Língua Portuguesa de uma população fictícia de alunos matriculados em escolas das redes Municipal, Estadual e Federal. Para isso, será utilizado um banco de dados chamado 'Alunos.txt', que contém informações sobre cada aluno, como a rede de ensino, escola, turma, proficiência em português e sexo. A análise será conduzida utilizando métodos de amostragem, mais especificamente a Amostragem Estratificada, Amostragem por Conglomerados e Amostragem por Conglomerados com Probabilidade Proporcional ao Tamanho (PPT) de Poisson. Esses métodos de amostragem são comumente utilizados em estudos de pesquisa para obter uma amostra representativa da população de interesse. Ao realizar a simulação com esses métodos de amostragem, será possível comparar seus desempenhos em relação à Proficiência em Língua Portuguesa dos alunos. Essa análise ajudará a identificar qual método é mais eficiente na obtenção de uma amostra representativa da população e poderá fornecer insights importantes para futuras pesquisas e intervenções educacionais.

2 Introdução

A avaliação da aptidão em Língua Portuguesa dos alunos matriculados em escolas é uma questão de extrema importância para o aprimoramento da qualidade educacional. Para obter informações sobre o desempenho desses alunos, é crucial utilizar métodos de amostragem adequados. Neste artigo, propomos um estudo de simulação com o objetivo de investigar qual dos três métodos produz resultados não viesados e com menor erro padrão para a média amostral da variável de interesse 'port', referente a proficiência em Língua Portuguesa da população, que são eles: Amostragem Estratificada, Amostragem por Conglomerados e Amostragem por Conglomerados com PPT de Poisson. A comparação entre os métodos permitirá identificar qual deles é mais adequado para fornecer informações precisas e embasar decisões educacionais fundamentadas.

A motivação para este estudo é impulsionada pela necessidade de compreender a proficiência média em Língua Portuguesa, a fim de identificar lacunas no sistema educacional e implementar intervenções pedagógicas eficazes. O domínio da Língua Portuguesa é essencial para o desenvolvimento intelectual, a comunicação e o sucesso acadêmico dos estudantes. Portanto, a obtenção de dados precisos sobre este domínio é fundamental para promover a igualdade de oportunidades e aprimorar o ensino.

A estrutura deste artigo segue uma abordagem metodológica, dividida em seções que abordam os aspectos essenciais do estudo, são elas: introdução, metodologia, estudo de simulação, considerações finais, referências bibliográficas e anexos do estudo.

Espera-se que este artigo contribua para o avanço do conhecimento na área de avaliação educacional, fornecendo insights sobre a seleção apropriada de métodos de amostragem para avaliar a proficiência em Língua Portuguesa, bem como aprofundar a compreensão da proficiência média dos alunos matriculados em escolas das redes de ensino. Além disso, busca-se analisar como os alunos se posicionam em relação a essa maestria, identificando possíveis lacunas e desafios enfrentados no ensino da Língua Portuguesa.

Dessa forma, espera-se que os resultados obtidos neste estudo possam embasar políticas e práticas educacionais mais eficazes, direcionadas ao aprimoramento da qualidade da educação em nosso contexto. Com uma compreensão mais aprofundada da proficiência em Língua Portuguesa e de como os alunos estão se desenvolvendo nessa área, será possível direcionar esforços e recursos de forma mais direcionada e eficiente, visando melhorar o ensino e promover melhores resultados educacionais para os alunos.

3 Metodologia

A partir dos dados fictícios da população de alunos das diferentes redes de ensino citadas, selecionamos 1000 amostras de tamanhos 500 e 750 utilizando os seguintes métodos de Amostragem para a análise da variável 'port' (Proficiência na Língua Portuguesa):

3.1 Amostragem Estratificada com estrato definido por "rede":

Este método de amostragem divide a população de alunos matriculados em escolas das redes Municipal, Estadual e Federal em estratos com base na rede de ensino, onde é selecionada uma amostra em cada estrato pelas técnicas de alocação (Uniforme, Proporcional e Ótima de Neyman) .

3.1.1 Alocação Uniforme:

Neste método, faremos uma alocação uniforme de alunos em cada estrato da amostra, garantindo que todas as escolas, independentemente da rede de ensino, tenham o mesmo tamanho na amostra.

3.1.2 Alocação Proporcional:

Neste método, faremos uma alocação proporcional de alunos em cada estrato da amostra, levando em consideração o tamanho relativo de cada rede de ensino. Dessa forma, o tamanho da amostra das redes de ensino será proporcional ao número de alunos em cada uma delas.

3.1.3 Alocação Ótima de Neyman:

Neste método, utilizaremos a alocação ótima proposta por Neyman, que leva em consideração a variabilidade da proficiência em Língua Portuguesa dentro de cada estrato. A alocação será feita de forma a minimizar a variância da estimativa da proficiência para a população total.

3.2 Amostragem por Conglomerados:

Este método é utilizado neste estudo para selecionar uma amostra de alunos matriculados em escolas das redes de ensino. Através dos estágios de seleção, é possível considerar as diferenças entre as escolas, turmas e alunos, obtendo estimativas mais precisas e representativas da proficiência em Língua Portuguesa.

3.2.1 1 estágio: Escolas:

Neste método, selecionamos aleatoriamente um conjunto de escolas de cada rede de ensino, e todos os alunos presentes nessas escolas serão incluídos na amostra. Como temos em média 31.41 alunos por escola, selecionamos n/31.41 escolas, onde n é o tamanho amostral esperado

3.2.2 1 estágio: Turmas:

Neste método, selecionamos aleatoriamente um conjunto de turmas de cada rede de ensino, e todos os alunos presentes nessas turmas serão incluídos na amostra. Analogamente ao caso anterior, temos em média 15.79 alunos por turma, então selecionamos n/15.79 escolas.

3.2.3 2 estágios: Escolas - Turmas:

Neste método, selecionamos aleatoriamente um conjunto de escolas de cada rede de ensino como primeira etapa, depois, dentro dessas escolas selecionadas, selecionamos aleatoriamente uma segunda amostra de turmas. Por fim, dentro das turmas selecionadas, selecionamos aleatoriamente uma terceira amostra de alunos, que serão incluídos na amostra. Nesse caso selecionamos n/31.41 escolas e 1 turma de cada escola.

3.2.4 3 estágios: Escolas - Turmas - Alunos:

Neste método, selecionamos aleatoriamente um conjunto de escolas de cada rede de ensino como primeira etapa, depois, dentro dessas escolas selecionadas, selecionamos aleatoriamente uma segunda amostra de turmas. Por fim, dentro das turmas selecionadas, selecionamos aleatoriamente uma terceira amostra de alunos, que serão incluídos na amostra. Aqui selecionamos n/31.41 escolas, 1 turma de cada escola e 16 alunos de cada turma.

3.3 Amostragem por Conglomerados com PPT de Poisson:

Nesse método, as escolas são consideradas conglomerados e são selecionadas aleatoriamente algumas delas. No entanto, a seleção dos conglomerados é realizada com base em uma distribuição de Poisson. Isso significa que as escolas com maior número de turmas/alunos têm uma maior probabilidade de serem selecionadas, enquanto as escolas com menor número, têm uma menor probabilidade de serem incluídas na amostra. Essa abordagem visa garantir que a amostra seja proporcional ao tamanho das escolas.

3.3.1 1 estágio: Escolas - Número de Turmas:

Neste método, selecionamos aleatoriamente n/31.41 escolas, com probabilidade de seleção proporcional ao número de turmas em cada escola

3.3.2 1 estágio: Escolas - Número de Alunos :

Neste método, selecionamos aleatoriamente n/31.41 escolas, com probabilidade de seleção proporcional ao número de alunos em cada escola

A partir dos resultados obtidos para a amostra selecionada em cada método, para ambos os tamanhos amostrais definidos, vamos calcular o estimador para a média populacional e verificar se ele é viesado e consistente, bem como seu intervalo de confiança de 95%. Essa estimativa será utilizada para fazer inferência sobre a proficiência em Língua Portuguesa da população de alunos matriculados nas diferentes redes de ensino.

4 Estudo de Simulação

Para n = 500

Table 1:

Método de alocação	Rede		
	1	2	3
Uniforme	167	167	167
Proporcional	163	284	55
Ótima de Neyman	160	286	55

Alocação	Fora do intervalo	Dentro do intervalo	$\hat{E}(\bar{y})$	$\hat{E}(\hat{EP}(\bar{y}))$	Tamanho médio
Uniforme	52	948	511.77	4.82	500
Proporcional	55	945	511.93	4.16	502
Ótima de Neyman	66	934	511.69	4.16	501
Conglomerado Escola	78	922	511.73	14.02	498.70
Conglomerado turma	63	937	511.15	11.55	506.19
UPA escolas USA turmas	51	949	511.79	12.47	554.09
UPA escolas USA turmas e UTA alunos	50	950	511.83	12.56	430.49
PPT turmas	75	925	511.26	14.58	600.32
PPT alunos	79	921	511.56	13.99	645.98

Para n = 750

Table 3:

Metodo de alocação	Rede		
	1	2	3
Uniforme	250	250	250
Proporcional	244	425	82
Ótima de Neyman	240	429	83

Alocação	Fora do intervalo	Dentro do intervalo	$\hat{E}(\bar{y})$	$\hat{E}(\hat{EP}(\bar{y}))$	Tamanho médio
Uniforme	56	944	511.81	3.88	750
Proporcional	52	948	511.72	3.32	751
Ótima de Neyman	42	958	511.75	3.31	752
Conglomerado Escola	75	925	511.31	11.27	752.38
Conglomerado turma	63	937	511.35	9.22	757.29
UPA escolas USA turmas	42	958	511.76	10.25	829.81
UPA escolas USA turmas e UTA alunos	41	959	511.77	10.34	645.64
PPT turmas	98	902	511.57	11.58	902.33
PPT alunos	68	932	511.79	11.09	966.27

Após realizar a simulação de acordo com os passos da metodologia, observa-se que, para o mesmo método, o erro padrão é menor nas amostras (tabelas 2 e 4) de tamanho 750 do que nas de tamanho 500. Além disso, destaca-se que a amostra estratificada demonstrou um erro padrão significativamente menor em comparação com a amostra por conglomerado.

Na amostragem estratificada, observou-se que a alocação proporcional apresentou um desempenho superior em relação à alocação uniforme, embora ambas tenham sido inferiores à alocação Ótima de Neyman. Vale ressaltar que as alocações Ótima de Neyman e proporcional obtiveram resultados bastante semelhantes, uma vez que a atribuição desses métodos acabou sendo parecida (tanto para n = 500 quanto para n = 750, conforme apresentado na tabela 1).

Na análise para determinar o método mais adequado (tabelas 2 e 4), constatou-se que o uso do PPT de Poisson apresentou resultados desfavoráveis, pois, além de ter os maiores tamanhos amostrais, apresenta um erro padrão grande. Pelos mesmos motivos, usar amostragem por conglomerado com unidade primária sendo as escolas e unidade secundária turmas foi descartada.

A estratégia de conglomerar por escolas também foi considerada desfavorável, devido ao erro padrão. Em contrapartida, a abordagem de conglomerar por turmas mostrou-se mais adequada, uma vez que resultou em um menor erro padrão quando aplicada a amostragem por conglomerado. Outra opção viável é a utilização da UPA (Unidade Primária de Amostragem) em escolas, USA (Unidade Secundária Amostral) em turmas e UTA (Unidade Terciária Amostral) em alunos,pois teve o menor tamanho amostral requerido e um erro padrão relativamente pequeno.

O método mais eficaz foi a estratificação por alocação ótima de Neyman, no entanto, na prática, devido à falta de variáveis auxiliares para sua aplicação, a alocação proporcional é fortemente recomendada.

No método UPA para escolas, USA para turmas e UTA para alunos, o tamanho amostral foi considerado pequeno devido à abordagem adotada. Após a seleção das turmas, foi estabelecido um critério de amostragem no qual foram selecionados 16 alunos de cada turma. No caso de turmas com menos de 16 alunos, a amostragem contemplou todos os alunos presentes. A obtenção de um erro padrão (EP) ainda sim reduzido indica que não era muito necessário pesquisar todos os estudantes de uma determinada turma, levando à conclusão de que as turmas são homogêneas.

5 Considerações Finais

Nossos resultados são consistentes com as expectativas iniciais de que um aumento no tamanho da amostra levaria a um menor erro padrão. Portanto, fica evidente que a escolha adequada do tamanho amostral é crucial para obter estimativas mais precisas.

A escolha do método de amostragem depende das características da população, dos objetivos do estudo e das restrições logísticas e de recursos. Em geral, e no nosso caso, a amostragem estratificada com alocação Ótima de Neyman é preferível, pois oferece estimativas mais precisas e representativas. No entanto, a estratégia de conglomerar por turmas, com alocação proporcional, pode ser uma alternativa viável em situações em que a homogeneidade dentro dos conglomerados é alta e as variáveis auxiliares não estão disponíveis. Para poupar recursos, é importante ter em mente o resultado de que os alunos das turmas são homogêneos, devendo ser avaliada a viabilidade da aplicação para cada problema.

6 Referências

- Bolfarine, H; Bussab, W. O. (2005) Elementos de Amostragem. São Paulo, Edgard Blucher.
- Cochran, W. G. (1977) Sampling Techniques. 3a ed. New York, John Wiley & Sons.
- Vicente, P.; Reis, E.; Ferrão, F. (2001) Sondagens: a Amostragem como Factor Decisivo de Qualidade. 2a ed. Lisboa, Edições Sílabo.

7 Anexos (códigos)

7.1 Para n = 500

```
library(tidyverse)
library(ggplot2)
library(survey)
library(sampling)
library(kableExtra)
library(tidyverse)
dados <- read.table("Alunos.txt", header = TRUE)
dados <- tibble::tibble(dados)
dados
```

7.1.1 Alocação uniforme

```
valorverdadeiro <- mean(dados$port)
amostrasunif <- lapply(1:1000, function(x){ IAESs=sampling::strata(dados, stratanames=c(
AESs=getdata(dados,IAESs)
fpc=rep(as.numeric(table(dados$rede)),c(501/3, 501/3, 501/3))
Plano=svydesign(~1, strata=~rede, data = AESs, probs=~IAESs$Prob, fpc=~fpc)
svymean(~port,Plano)})

IC <- lapply(amostrasunif,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
propunif <- unlist(quantidade)

mediaunif <- mean(unlist(lapply(amostrasunif, function(x) x[1]))) # média das estimações
epunif <- mean(unlist(lapply(amostrasunif, function(x) SE(x)))) # média dos erros padrõe
```

7.1.2 Proporcional

```
amostrasprop <- lapply(1:1000, function(x){ IAESs=sampling::strata(dados, stratanames=c(
AESs=getdata(dados,IAESs)
fpc=rep(as.numeric(table(dados$rede)),as.numeric(ceiling(500*(prop.table(table(dados$red
Plano=svydesign(~1, strata=~rede, data = AESs, probs=~IAESs$Prob, fpc=~fpc)
svymean(~port,Plano)})
```

```
IC <- lapply(amostrasprop,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
propprop <- unlist(quantidade)

mediaprop <- mean(unlist(lapply(amostrasprop, function(x) x[1])))# média das estimações
epprop <- mean(unlist(lapply(amostrasprop, function(x) SE(x))))# média dos erros padrões
```

7.1.3 Ótima de Neyman

```
n <- 500
s <- dados %>%
  group_by(rede) %>%
  summarise(s = sd(port))
s <- s$s
N <- as.numeric(table(dados$rede))
tamanhos <- n*(N*s/sum(N*s))
tamanhos <- ceiling(tamanhos)
amostrasotimas <- lapply(1:1000, function(x){ IAESs=sampling::strata(dados, stratanames=
AESs=getdata(dados,IAESs)
fpc=rep(as.numeric(table(dados$rede)), tamanhos)
Plano=svydesign(~1, strata=~rede, data = AESs, probs=~IAESs$Prob, fpc=~fpc)
svymean(~port,Plano)})

IC <- lapply(amostrasotimas,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
propney <- unlist(quantidade)

medianey <- mean(unlist(lapply(amostrasotimas, function(x) x[1]))) # média das estimaçõe
epney <- mean(unlist(lapply(amostrasotimas, function(x) SE(x)))) # média dos erros padrõ


todos <- rbind(rep(501/3, 3),
               as.numeric(ceiling(500*(prop.table(table(dados$rede))))), tamanhos)
todos <- as.data.frame(todos)
colnames(todos) <- NULL
rownames(todos) <- c("Uniforme", "Proporcional", "Ótima de Neyman")
# kbl(todos, format = "latex") %>%  kable_classic() %>% add_header_above(c("", "1", "2",
#
```

```
# kbl(todos, format = "latex") %>%  kable_classic() %>% add_header_above(c("", "1", "2",
```

7.1.4 Unidade primária amostral: Escolas

```
mean(table(dados$escola))
```

tamanho médio dos clusters de escola é 31.41, se queremos uma amostra de tamanho aproximadamente igual a 500, então devemos pegar 500/31.41 clusters, ou seja, 16 clusters

```
conglomerado_Escola <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("escola"),
size=16, method=c("srswor"))

ACSs=getdata(dados,IACSs)

fpc2=rep(191,dim(IACSs)[1]) # pois são 191 escolas
PlanoC=svydesign(id=~escola, data = ACSs, probs=~IACSs$Prob,
fpc=~fpc2)

svymean(~port,PlanoC)
})

IC <- lapply(conglomerado_Escola,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_conglomerado_escola <- unlist(quantidade)

media_conglomerado_escola <- mean(unlist(lapply(conglomerado_Escola, function(x) x[1])))
ep_conglomerado_escola <- mean(unlist(lapply(conglomerado_Escola, function(x) SE(x)))) #
```

```
conglomerado_Escola_tamanho <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("escola"),
size=16, method=c("srswor"))

ACSs=getdata(dados,IACSs)

nrow(ACSs)
})

mean(unlist(conglomerado_Escola_tamanho))
```

7.1.5 Unidade primária amostral: Turmas

```
mean(table(dados$turma))
```

tamanho médio dos clusters de turma é 15.78947, se queremos uma amostra de tamanho aproximadamente igual a 500, então devemos pegar 500/15.78947 clusters, ou seja, 32 clusters

```
conglomerado_Turma <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("turma"),
size=32, method=c("srswor"))

ACSs=getdata(dados,IACSs)

fpc2=rep(380,dim(IACSs)[1]) # pois são 380 escolas
PlanoC=svydesign(id=~turma, data = ACSs, probs=~IACSs$Prob,
fpc=~fpc2)

svymean(~port,PlanoC)
})

IC <- lapply(conglomerado_Turma,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_conglomerado_turma <- unlist(quantidade)

media_conglomerado_turma <- mean(unlist(lapply(conglomerado_Turma, function(x) x[1]))) #
ep_conglomerado_turma <- mean(unlist(lapply(conglomerado_Turma, function(x) SE(x)))) # m
```

```
conglomerado_Turma_tamanho <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("turma"),
size=32, method=c("srswor"))

ACSs=getdata(dados,IACSs)

nrow(ACSs)
})
mean(unlist(conglomerado_Turma_tamanho))
```

7.1.6 Escolas como unidade primária amostral e turmas como unidade secundária amostral

7.1.6.1 Selecionaremos 500/15.8 (32) escolas e 1 turma de cada escola Já que cada turma possui em média 15.8 alunos.

```
conglomerado2estagios <- lapply(1:1000, function(x){
n2=rep(1,32[1])
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(32,n2), # Selecionaremos 500/15.8 (32) escol
method=list("srswor", "srswor"))
AC2=getdata(dados,IAC2)[[2]]
PlanoC2=svydesign(data=AC2,ids=~escola+turma, nest=TRUE,
probs=AC2$Prob)
svymean(~port,PlanoC2)})


IC <- lapply(conglomerado2estagios,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_2estagios <- unlist(quantidade)

media_2estagios <- mean(unlist(lapply(conglomerado2estagios, function(x) x[1]))) # média
ep_2estagios <- mean(unlist(lapply(conglomerado2estagios, function(x) SE(x)))) # média d
```

```
conglomerado2estagiosTamanho <- lapply(1:1000, function(x){
n2=rep(1,32[1])
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(32,n2), # Selecionaremos 500/15.8 (32) escol
method=list("srswor", "srswor"))
AC2=getdata(dados,IAC2)[[2]]
nrow(AC2)})
mean(unlist(conglomerado2estagiosTamanho))
```

7.1.7 Escolas como unidade primária amostral e escolas como unidade secundária amostral e turmas como unidade terciária amostral

```
n2 <- rep(1, 32)
tamanho_amostra <- 0
conglomerado3estagios <- lapply(1:1000, function(x){
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(32,n2),
method=list("srswor", "srswor")) # seleciona a escola e a turma primeiro
```

```
AC2=getdata(dados,IAC2)[[2]]

tamanho_turma <- AC2 %>% group_by(turma) %>% reframe(n = n(), probabilidade = Prob)
tamanho_turma <- unique(tamanho_turma)
amostrar <- tamanho_turma[which(tamanho_turma$n >= 16), ]
pegar_todos <- tamanho_turma[which(tamanho_turma$n < 16), ]

amostrar$probabilidade <- amostrar$probabilidade*16/amostrar$n

IAC_final=mstage(dados %>% filter(turma %in% amostrar$turma), stage=list("cluster", "clu
varnames=list("turma","aluno"), size=list(nrow(amostrar), rep(16, nrow(amostrar))),
method=list("srswor", "srswor"))

AC_final=getdata(dados %>% filter(turma %in% amostrar$turma),IAC_final)[[2]]

df_merged <- merge(AC_final, amostrar %>% select(turma, probabilidade), by = "turma", al
df_merged$Prob <- df_merged$probabilidade

df_merged2 <- merge(dados %>% filter(turma %in% pegar_todos$turma), pegar_todos %>% sele
df_merged2$Prob <- df_merged2$probabilidade
df_merged2 <- df_merged2[,-7]

final <- rbind(df_merged[,c(1,6,2,3,4,5,9)],df_merged2)
Plano=svydesign(data=final,
                  ids=~escola+turma+aluno, nest=TRUE,probs=final$Prob)
tamanho_amostra[x] <- nrow(final)
svymean(~port,Plano)})


IC <- lapply(conglomerado3estagios,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_3estagios <- unlist(quantidade)

media_3estagios <- mean(unlist(lapply(conglomerado3estagios, function(x) x[1]))) # média
ep_3estagios <- mean(unlist(lapply(conglomerado3estagios, function(x) SE(x)))) # média d
```

Para ver o tamanho da amostra coletada anteriormente:

```
tam <- lapply(1:1000, function(x){
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(32,n2),
```

```
method=list("srswor", "srswor")) # seleciona a escola e a turma primeiro

AC2=getdata(dados,IAC2)[[2]]

tamanho_turma <- AC2 %>% group_by(turma) %>% reframe(n = n(), probabilidade = Prob)
tamanho_turma <- unique(tamanho_turma)
amostrar <- tamanho_turma[which(tamanho_turma$n >= 16), ]
pegar_todos <- tamanho_turma[which(tamanho_turma$n < 16), ]

tamanho <- sum(pegar_todos$n) + nrow(amostrar)*16
tamanho})
mean(unlist(tam))
```

7.1.8 1 estágio-> upa: escolas; tamanho = número de turmas

```
ppt_turmas <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = length(unique(turma)))
pi=inclusionprobabilities(probabilidades$n, 16) # tem 31.41 alunos em media por escola,
set.seed(x)
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=16, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)

fpc2=rep(191,dim(IACSP)[1]) # pois são 191 escolas
PlanoPPT=svydesign(id=~escola, data = ACSP, probs=~IACSP$Prob,
fpc=~fpc2)
svymean(~port,PlanoPPT)})


IC <- lapply(ppt_turmas,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_ppt_turmas <- unlist(quantidade)

media_ppt_turmas <- mean(unlist(lapply(ppt_turmas, function(x) x[1]))) # média das estim
ep_ppt_turmas <- mean(unlist(lapply(ppt_turmas, function(x) SE(x)))) # média dos erros p
```

```
ppt_turmas_tamanho <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = length(unique(turma)))
pi=inclusionprobabilities(probabilidades$n, 16)
set.seed(x)
```

```
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=16, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)

nrow(ACSP)})

mean(unlist(ppt_turmas_tamanho))
```

7.1.9 1 estágio-> upa: escolas; tamanho = número de alunos

```
ppt_alunos <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = n())

pi=inclusionprobabilities(probabilidades$n, 16)
set.seed(x)
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=16, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)

fpc2=rep(191,dim(IACSP)[1]) # pois são 191 escolas
PlanoPPT=svydesign(id=~escola, data = ACSP, probs=~IACSP$Prob,
fpc=~fpc2)
svymean(~port,PlanoPPT)})


IC <- lapply(ppt_alunos,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_ppt_alunos <- unlist(quantidade)

media_ppt_alunos <- mean(unlist(lapply(ppt_alunos, function(x) x[1]))) # média das estim
ep_ppt_alunos <- mean(unlist(lapply(ppt_alunos, function(x) SE(x)))) # média dos erros p
```

```
ppt_alunos_tamanho <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = n())
pi=inclusionprobabilities(probabilidades$n, 16)
set.seed(x)
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=16, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)
```

```
nrow(ACSP)})
mean(unlist(ppt_alunos_tamanho))
```

7.2 Agora para tamanho amostral de 750

7.2.1 Alocação uniforme

```
valorverdadeiro <- mean(dados$port)
amostrasunif <- lapply(1:1000, function(x){ IAESs=sampling::strata(dados, stratanames=c(
AESs=getdata(dados,IAESs)
fpc=rep(as.numeric(table(dados$rede)),c(750/3, 750/3, 750/3))
Plano=svydesign(~1, strata=~rede, data = AESs, probs=~IAESs$Prob, fpc=~fpc)
svymean(~port,Plano)})

IC <- lapply(amostrasunif,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
propunif <- unlist(quantidade)

mediaunif <- mean(unlist(lapply(amostrasunif, function(x) x[1]))) # média das estimações
epunif <- mean(unlist(lapply(amostrasunif, function(x) SE(x)))) # média dos erros padrõe
```

7.2.2 Proporcional

```
amostrasprop <- lapply(1:1000, function(x){ IAESs=sampling::strata(dados, stratanames=c(
AESs=getdata(dados,IAESs)
fpc=rep(as.numeric(table(dados$rede)),as.numeric(ceiling(750*(prop.table(table(dados$red
Plano=svydesign(~1, strata=~rede, data = AESs, probs=~IAESs$Prob, fpc=~fpc)
svymean(~port,Plano)})

IC <- lapply(amostrasprop,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
propprop <- unlist(quantidade)

mediaprop <- mean(unlist(lapply(amostrasprop, function(x) x[1])))# média das estimações
epprop <- mean(unlist(lapply(amostrasprop, function(x) SE(x))))# média dos erros padrões
```

7.2.3 Ótima de Neyman

```
n <- 750
s <- dados %>%
  group_by(rede) %>%
  summarise(s = sd(port))
s <- s$s
N <- as.numeric(table(dados$rede))
tamanhos <- n*(N*s/sum(N*s))
tamanhos <- ceiling(tamanhos)
amostrasotimas <- lapply(1:1000, function(x){ IAESs=sampling::strata(dados, stratanames=
AESs=getdata(dados,IAESs)
fpc=rep(as.numeric(table(dados$rede)), tamanhos)
Plano=svydesign(~1, strata=~rede, data = AESs, probs=~IAESs$Prob, fpc=~fpc)
svymean(~port,Plano)})

IC <- lapply(amostrasotimas,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
propney <- unlist(quantidade)

medianey <- mean(unlist(lapply(amostrasotimas, function(x) x[1]))) # média das estimaçõe
epney <- mean(unlist(lapply(amostrasotimas, function(x) SE(x)))) # média dos erros padrõ


todos <- rbind(rep(750/3, 3),
               as.numeric(ceiling(750*(prop.table(table(dados$rede))))), tamanhos)
todos <- as.data.frame(todos)
colnames(todos) <- NULL
rownames(todos) <- c("Uniforme", "Proporcional", "Ótima de Neyman")
# kbl(todos, format = "latex") %>%  kable_classic() %>% add_header_above(c("", "1", "2",
#
# kbl(todos, format = "latex") %>%  kable_classic() %>% add_header_above(c("", "1", "2",
```

7.2.4 Unidade primária amostral: Escolas

```
mean(table(dados$escola))
```

tamanho médio dos clusters de escola é 31.41, se queremos uma amostra de tamanho aproximadamente igual a 750, então devemos pegar 750/31.41 clusters, ou seja, 24 clusters

```
conglomerado_Escola <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("escola"),
size=24, method=c("srswor"))

ACSs=getdata(dados,IACSs)

fpc2=rep(191,dim(IACSs)[1]) # pois são 191 escolas
PlanoC=svydesign(id=~escola, data = ACSs, probs=~IACSs$Prob,
fpc=~fpc2)

svymean(~port,PlanoC)
})

IC <- lapply(conglomerado_Escola,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_conglomerado_escola <- unlist(quantidade)

media_conglomerado_escola <- mean(unlist(lapply(conglomerado_Escola, function(x) x[1])))
ep_conglomerado_escola <- mean(unlist(lapply(conglomerado_Escola, function(x) SE(x)))) #
```

```
conglomerado_Escola_tamanho <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("escola"),
size=24, method=c("srswor"))

ACSs=getdata(dados,IACSs)

nrow(ACSs)
})

mean(unlist(conglomerado_Escola_tamanho))
```

7.2.5 Unidade primária amostral: Turmas

```
mean(table(dados$turma))
```

tamanho médio dos clusters de turma é 15.78947, se queremos uma amostra de tamanho aproximadamente igual a 750, então devemos pegar 750/15.78947 clusters, ou seja, 48 clusters

```
conglomerado_Turma <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("turma"),
size=48, method=c("srswor"))

ACSs=getdata(dados,IACSs)

fpc2=rep(380,dim(IACSs)[1]) # pois são 380 escolas
PlanoC=svydesign(id=~turma, data = ACSs, probs=~IACSs$Prob,
fpc=~fpc2)

svymean(~port,PlanoC)
})

IC <- lapply(conglomerado_Turma,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_conglomerado_turma <- unlist(quantidade)

media_conglomerado_turma <- mean(unlist(lapply(conglomerado_Turma, function(x) x[1]))) #
ep_conglomerado_turma <- mean(unlist(lapply(conglomerado_Turma, function(x) SE(x)))) # m
```

```
conglomerado_Turma_tamanho <- lapply(1:1000, function(x){
IACSs=sampling::cluster(dados, clustername=c("turma"),
size=48, method=c("srswor"))

ACSs=getdata(dados,IACSs)

nrow(ACSs)
})
mean(unlist(conglomerado_Turma_tamanho))
```

7.2.6 Escolas como unidade primária amostral e turmas como unidade secundária amostral

7.2.6.1 Selecionaremos 750/15.8 (48) escolas e 1 turma de cada escola Já que cada turma possui em média 15.8 alunos.

```
conglomerado2estagios <- lapply(1:1000, function(x){
n2=rep(1,48[1])
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(48,n2), # Selecionaremos 750/15.8 (48) escol
```

```
method=list("srswor", "srswor"))
AC2=getdata(dados,IAC2)[[2]]
PlanoC2=svydesign(data=AC2,ids=~escola+turma, nest=TRUE,
probs=AC2$Prob)
svymean(~port,PlanoC2)})


IC <- lapply(conglomerado2estagios,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_2estagios <- unlist(quantidade)

media_2estagios <- mean(unlist(lapply(conglomerado2estagios, function(x) x[1]))) # média
ep_2estagios <- mean(unlist(lapply(conglomerado2estagios, function(x) SE(x)))) # média d
```

```
conglomerado2estagiosTamanho <- lapply(1:1000, function(x){
n2=rep(1,48[1])
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(48,n2), # Selecionaremos 750/15.8 (48) escol
method=list("srswor", "srswor"))
AC2=getdata(dados,IAC2)[[2]]
nrow(AC2)})
mean(unlist(conglomerado2estagiosTamanho))
```

7.2.7 Escolas como unidade primária amostral e escolas como unidade secundária amostral e turmas como unidade terciária amostral

```
n2 <- rep(1, 48)
tamanho_amostra <- 0
conglomerado3estagios <- lapply(1:1000, function(x){
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(48,n2),
method=list("srswor", "srswor")) # seleciona a escola e a turma primeiro

AC2=getdata(dados,IAC2)[[2]]

tamanho_turma <- AC2 %>% group_by(turma) %>% reframe(n = n(), probabilidade = Prob)
tamanho_turma <- unique(tamanho_turma)
amostrar <- tamanho_turma[which(tamanho_turma$n >= 16), ]# tem 48 turmas, para amostra d
```

```
pegar_todos <- tamanho_turma[which(tamanho_turma$n < 16), ]

amostrar$probabilidade <- amostrar$probabilidade*16/amostrar$n

IAC_final=mstage(dados %>% filter(turma %in% amostrar$turma), stage=list("cluster", "clu
varnames=list("turma","aluno"), size=list(nrow(amostrar), rep(16, nrow(amostrar))),
method=list("srswor", "srswor"))

AC_final=getdata(dados %>% filter(turma %in% amostrar$turma),IAC_final)[[2]]

df_merged <- merge(AC_final, amostrar %>% select(turma, probabilidade), by = "turma", al
df_merged$Prob <- df_merged$probabilidade

df_merged2 <- merge(dados %>% filter(turma %in% pegar_todos$turma), pegar_todos %>% sele
df_merged2$Prob <- df_merged2$probabilidade
df_merged2 <- df_merged2[,-7]

final <- rbind(df_merged[,c(1,6,2,3,4,5,9)],df_merged2)
Plano=svydesign(data=final,
                  ids=~escola+turma+aluno, nest=TRUE,probs=final$Prob)
tamanho_amostra[x] <- nrow(final)
svymean(~port,Plano)})


IC <- lapply(conglomerado3estagios,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_3estagios <- unlist(quantidade)

media_3estagios <- mean(unlist(lapply(conglomerado3estagios, function(x) x[1]))) # média
ep_3estagios <- mean(unlist(lapply(conglomerado3estagios, function(x) SE(x)))) # média d
```

Para ver o tamanho da amostra coletada anteriormente:

```
tam <- lapply(1:1000, function(x){
set.seed(x)
IAC2=mstage(dados, stage=list("cluster","cluster"),
varnames=list("escola", "turma"), size=list(48,n2),
method=list("srswor", "srswor")) # seleciona a escola e a turma primeiro

AC2=getdata(dados,IAC2)[[2]]

tamanho_turma <- AC2 %>% group_by(turma) %>% reframe(n = n(), probabilidade = Prob)
```

```
tamanho_turma <- unique(tamanho_turma)
amostrar <- tamanho_turma[which(tamanho_turma$n >= 16), ]
pegar_todos <- tamanho_turma[which(tamanho_turma$n < 16), ]

tamanho <- sum(pegar_todos$n) + nrow(amostrar)*16
tamanho})
mean(unlist(tam))
```

7.2.8 1 estágio-> upa: escolas; tamanho = número de turmas

```
ppt_turmas <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = length(unique(turma)))
pi=inclusionprobabilities(probabilidades$n, 24) # tem 31.41 alunos em media por escola,
set.seed(x)
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=24, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)

fpc2=rep(191,dim(IACSP)[1]) # pois são 191 escolas
PlanoPPT=svydesign(id=~escola, data = ACSP, probs=~IACSP$Prob,
fpc=~fpc2)
svymean(~port,PlanoPPT)})


IC <- lapply(ppt_turmas,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_ppt_turmas <- unlist(quantidade)

media_ppt_turmas <- mean(unlist(lapply(ppt_turmas, function(x) x[1]))) # média das estim
ep_ppt_turmas <- mean(unlist(lapply(ppt_turmas, function(x) SE(x)))) # média dos erros p
```

```
ppt_turmas_tamanho <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = length(unique(turma)))
pi=inclusionprobabilities(probabilidades$n, 24)
set.seed(x)
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=24, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)

nrow(ACSP)})
```

```
mean(unlist(ppt_turmas_tamanho))
```

7.2.9 1 estágio-> upa: escolas; tamanho = número de alunos

```
ppt_alunos <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = n())

pi=inclusionprobabilities(probabilidades$n, 24)
set.seed(x)
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=24, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)

fpc2=rep(191,dim(IACSP)[1]) # pois são 191 escolas
PlanoPPT=svydesign(id=~escola, data = ACSP, probs=~IACSP$Prob,
fpc=~fpc2)
svymean(~port,PlanoPPT)})


IC <- lapply(ppt_alunos,
       function(x) c(x[1] - 1.96*SE(x), x[1] + 1.96*SE(x)))

quantidade <- lapply(IC, function(x) ifelse(x[1]<valorverdadeiro & x[2]>valorverdadeiro,
prop_ppt_alunos <- unlist(quantidade)

media_ppt_alunos <- mean(unlist(lapply(ppt_alunos, function(x) x[1]))) # média das estim
ep_ppt_alunos <- mean(unlist(lapply(ppt_alunos, function(x) SE(x)))) # média dos erros p
```

```
ppt_alunos_tamanho <- lapply(1:1000, function(x){
probabilidades <- dados %>% group_by(escola) %>% summarise(n = n())
pi=inclusionprobabilities(probabilidades$n, 24)
set.seed(x)
IACSP=sampling::cluster(dados, clustername=c("escola"),
size=24, method=c("poisson"),pik=pi)
ACSP=getdata(dados,IACSP)
nrow(ACSP)})
mean(unlist(ppt_alunos_tamanho))
```