PNADC - PESQUISA NACIONAL DE AMOSTRA DE DOMICÍLIOS CONTÍNUA

Marcelo Prudente

03/03/2020

1 PNAD CONTÍNUA

A lógica computacional da análise dos dados da PNAD Contínua aqui adotada é muito próxima àquela observada com o *dplyr* do *tidyverse*. Em resumo, utilizamos os verbos *filter, select, mutate, group_by e summarise*. - Porém, há maior facilidade de acessar os seus dados: podemos fazer isso por meio de um pacote do R.

1.1 Survey?

Os grandes *surveys*, a exemplo das PNADs e PNAD Contínua, se diferenciam filosoficamente e substantivamente da aborgagem tradicional das amostras aleatórias. Em poucas palavras, a amostragem aleatória não produz estimadores corretos para *surveys* grandes e complexos. Em pesquisas domiciliares, se a estratificação envolvesse aleatoriamente indivíduos, seria necessário visitar milhões de lugares para uma pesquisa.

Por isso, as questões logísticas (custos!) levam a conglomerar as amostras - concentrar geograficamente as entrevistas é financeiramente mais efetivo.

Nas PNADs e PNAD Contínua há uma amostragem por conclomerados: + A unidade de seleção pode ser o município, que congrega setores censitários, que contém domicílios. + Seleciona-se uma amostra dos municípios, depois uma amostra dos setores censitários, em seguida dos domicílios.

1.2 Analisando surveys

Por conta do seu desenho, a análise de *surveys* não pode ser feita diretamente. AO invés disso, deve-se fazer a "correção"" das unidades primárias amostrais e dos estratos de amostragem.

Por sorte, para a PNAD Contínua, é possível baixar os dados diretamente já com o tratamento dos dados para a estrutura de *survey*.

Assim, é necessário baixar o pacote **PNADcIBGE**.

Você pode ver exemplos instrutivos na página de exemplos do autor do pacote, **Douglas Braga**. Você observará que a abordagem da análise é um pouco distinta da implementada aqui, pois o autor utiliza o pacote *survey* como base e nós utilizaremos o pacote *srvyr*, que adota a abordagem de análise do *tidyverse*.

```
# instalar pacote
install.packages("PNADcIBGE")
```

```
# carregar pacote
library(PNADcIBGE)
```

1.3 PNADCs

- As PNADs Contínuas têm dois tipos de microdados disponíveis para análise:
  - Trimestrais
  - Anuais

Este link apresenta as informações presentes em cada uma das visitas. Por exemplo, as condições de habitação são summarizadas nas 1ª vistas das pesquisas anuais. Por sua vez, as estatísticas oficiais sobre educação são apresentadas nas entrevistas do 2º trimestre das divulgações trimestrais.

1.4 Vantagens do PNADcIBGE

1. Os dados já são baixados com o desenho de survey.
2. É possível baixar dados anuais e trimestrais.
3. É possível baixar apenas algumas variáveis de interesse.
4. Variáveis carregam seus rótulos

1.5 Baixando os dados da PNADC

- Não há esforço para baixar a PNADC trimestral:

```
# Especifica o ano e o trimestre
pnadc = get_pnadc(year = 2019, quarter = 4)
class(pnadc)
```

- Tampouco a anual:

```
# Especifica o ano e a entrevista
pnadc_anual = get_pnadc(year = 2018, interview = 1)
```

Embora baixar automaticamente seja prático, pode ocorrer eventuais erros de conexão com o *ftp* do IBGE, frustrando a análise. Por isso, recomendo baixar os dados e a documentação *offline*, o que permitirá carregar os dados de forma mais rápida e pronta.

1.6 Carregando dados Off-line

- Entrar no **ftp do ibge**
- Baixar os arquivos da PNAD e o Dicionario_e_input

```
# diretório
setwd("C:/curso_r/dados/PNADC")

# baixar dados
pnadc <- read_pnadc("PNADC_042019.txt", "Input_PNADC_trimestral.txt")

# incluir rótulos
pnadc <- pnadc_labeller(pnadc,
↪  "dicionario_PNADC_microdados_trimestral.xls")
```

Depois de baixarmos os dados e colocarmos os rótulos, é necessário inputar o desenho de *survey* na PNADC.

Antes, necessitamos instalar o pacote *srvyr*.

```
# instalar pacote srvyr
install.packages("srvyr")
```

```
library(srvyr)
```

```
##
## Attaching package: 'srvyr'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
# desenho de survey
pnadc <- pnadc_design(pnadc)

# transformar em tbl_svy
pnadc <- as_survey(pnadc)
```

2 Exemplo de tabulações utilizando a PNAD Contínua

2.1 pacote svryr

Para a análise dos surveys, utilizaremos o pacote **srvyr** que utiliza a sintaxe do *dplyr* para executar as análises de pesquisas amostrais complexas. Porém, aqui sempre utilizaremos o *summarise*, precedido ou não pelo *group_by*.

Portanto, quando lidamos com a PNAD ou com a PNADC substituiremos os comandos usuais, a exemplo de *mean*, *sum* ou *quantile* pelos comandos abaixo.

- Há cinco comandos principais para a análise:
  - **survey_mean()**
  - **survey_ratio()**
  - **survey_total()**
  - **survey_quantile()**
  - **survey_median()**

2.2 Análises: totais survey_total()

Calcula os valores totais de uma variável em surveys complexos.

- **Algumas estimativas**:

```
# população estimada do Brasil em 2019
pnadc %>%
  mutate(one = 1) %>%
  summarise(pop_brasil = survey_total(one))
# população estimadad do Brasil por regiao em 2015
pnadc %>%
  mutate(one = 1,
         regiao = substr(UPA, 1, 1)) %>%
  group_by(regiao)%>%
  summarize(pop_brasil = survey_total(one))
```

- Tente extrair a população por Estado

2.3 Análises: média survey_mean()

Calcula as médias ou proporções em um survey complexo. No primeiro caso, você deve especificar a variável numérica a ter a média calculada (por exemplo, rendimento médio). No segundo caso, o comando calcula o percentual de observações por grupo (percentual de analfabetos no país).

```
# rendimento por sexo: rendimento é numerico
pnadc %>%
  group_by(V2007)%>%
  summarize(rendimento = survey_mean(VD4020, na.rm = TRUE))
```

```
## # A tibble: 2 x 3
##   V2007  rendimento rendimento_se
##   <fct>       <dbl>         <dbl>
```

```
## 1 Homem       2655.          35.7
## 2 Mulher      2107.          29.0
```

```
# percentual de analfabetos
pnadc %>%
  group_by(V3001) %>%
  summarise(perc_analfabetos = survey_mean( na.rm = TRUE))
```

```
## Warning: Factor `V3001` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```
## # A tibble: 3 x 3
##   V3001 perc_analfabetos perc_analfabetos_se
##   <fct>            <dbl>               <dbl>
## 1 Sim             0.921             0.000660
## 2 Não             0.0789            0.000660
## 3 <NA>           NA                 NA
```

Observe que você pode criar novas variáveis para fazer suas estimativas.

```
# rendimento por faixa etária
source("C:/curso_r/funcoes/age_cat.R")
pnadc %>%
  mutate(fx_idade = age.cat(V2009, upper = 90)) %>%
  group_by(fx_idade)%>%
  summarise(perc_por_idade = survey_mean(, na.rm = TRUE),
            rendimento = survey_mean(VD4020, na.rm = TRUE))
```

- Qual o rendimento médio por sexo e raça?

2.4 Exemplo: a distribuição salarial do servidor público federal por nível de ensino

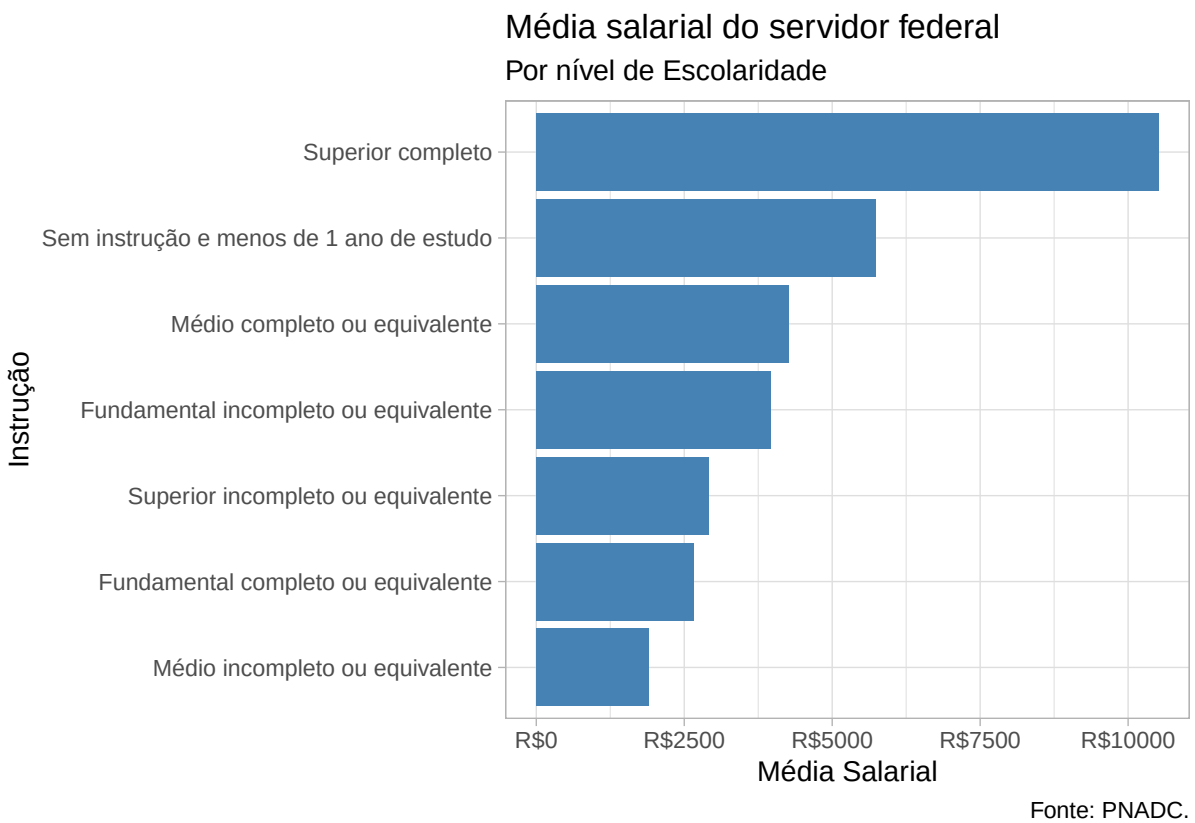
Precisamos encontrar as variáveis que indicam o nível de ensino e as que identificam os servidores.

```
servidor <- pnadc %>%
  filter(V4012 == levels(V4012)[4] &
         V4014 == levels(V4014)[1]) %>%
  group_by(VD3004) %>%
  summarise(numero_de_servidores = survey_total(na.rm = T),
            media_salarial = survey_mean(VD4020, na.rm = T),
            percentual_no_grupo = survey_mean(na.rm = T))
```

```
## Warning: Factor `VD3004` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

Podemos demonstrar graficamente a relação.

```
library(tidyverse)
servidor %>%
  select(-ends_with("_se")) %>%
  filter(!is.na(VD3004)) %>%
  ggplot(aes(x = reorder(VD3004, media_salarial), y = media_salarial)) +
  geom_col(fill = "steelblue") +
  labs(y = "Média Salarial", x = "Instrução",
       title = "Média salarial do servidor federal",
       subtitle = "Por nível de Escolaridade",
       caption = "Fonte: PNADC.") +
  scale_y_continuous(labels = function(x) paste0("R$", x))+
  coord_flip() +
  theme_light()
```



2.5 Análises mediana survey_median()

Calcula a mediana em surveys complexos.

```
# rendimento por sexo
pnadc %>%
  group_by(V2007)%>%
  summarize(rendimento = survey_median(VD4020, na.rm = TRUE))

# rendimento entre os maiores de 30 e menores de 40
pnadc %>%
  filter(V2007>= 30 & V2007 <=40 )%>%
  summarize(rendimento = survey_median(VD4020, na.rm = TRUE))
```

- Compare a média e a mediana dos salários em uma mesma tabela

2.6 Análises dos quantis survey_quantile()

```
# distribuição da renda por quantis
pnadc %>%
  summarise(rendimento =
              survey_quantile(VD4020, c(0.25, 0.5, 0.75),
                              na.rm = TRUE, covmat = TRUE))
```

As tabulações dos dados da PNADC são muito parecidas com aquelas obtidas no *dplyr* do *tidyverse*.

2.7 Exemplo: Estimativas educacionais

- A referência do IBGE para estatísticas educacionais é a PNAD Contínua do segundo trimestre. Como estamos utilizando os dados do 4º trimestre de 2019, não podemos dizer que esses são os números oficiais, mas podemos ter uma boa ideia dos índices em estudo. Caso queira, entre no ftp do IBGE e baixe os dados do 2º trimestre de 2019 para verificar as informações oficiais.

2.7.1 Exemplo e exercícios

```
pnadc <- pnadc  %>%
  mutate(regiao = as.factor(substr(UPA, 1, 1)))

# número e percentual de pessoas com mais de 15 anos
# analfabetos
analfabetos_uf <- pnadc %>% filter(V2009 >=15) %>%
```

```
    group_by(UF, V3001)%>%
    summarise(analf = survey_total( na.rm = T),
              analf_perc = survey_mean(na.rm = T))
```

- **Exercícios**

2.8 Baixar pacotes

```
# Pacotes exigidos
pacotes <- c("survey", "tidyverse", "srvyr")

# carregar lista de uma só vez
lapply(pacotes, require, character.only = TRUE)
```

- Para usar o **srvyr**, é necessário transformar o desenho de survey em **tbl_svy**.

```
# Pacotes exigidos
pnadc <- as_survey(pnadc)
```

2.9 Algumas análises

- Tamanho da população estimada no trimestre.

```
# total
dadosPNADc %>% summarise(survey_total(one, na.rm = T))
# por uf
dadosPNADc %>%
  group_by(UF)%>%
  summarise(pop = survey_total( one ,na.rm = T))
```

2.10 Transformando tabelas

Podemos pivotar as tabelas obtidas pela análise das PNADCs para observar melhor os resultados.

```
# renda por sexo e raca
tot_sexo_raca <- pnadc %>%
  group_by(V2007, V2010) %>%
  summarise(total = survey_mean(VD4016, na.rm = T))

# spread: inverte a tabela
tot_sexo_raca <- tot_sexo_raca %>%
  select(V2007:total) %>%
```

```
  spread(V2007, total)

tot_sexo_raca
# gather: volta ao formato original
tot_sexo_raca <- gather(tot_sexo_raca, V2007, value, - V2010)
```

3 Modelagem com survey

3.1 Teste de Hipóteses

- Vamos testar se a diferença salarial entre homens e mulheres tem significância estatística.

```
# VD4020 - renda
# V2007 - sexo
svyttest(VD4020 ~ V2007, pnadc)
```

3.2 Regressão Linear

- A renda está associada ao nível educacional, à raça, à idade e ao sexo?
- Para regressão com surveys, utilizamos o *svyglm*.

```
# o  ~ separa a variável dependente das independentes
modeloLin <- svyglm(VD4020 ~ VD3001 + V2010 + V2009 + V2007, pnadc)
summary(modeloLin)
```

3.3 Regressões Logísticas

- O que está associado à conclusão de um curso de graduação.

```
modelo <- svyglm(V3007 ~ V2007 + V2010 + V2009 + regiao, pnadc, family =
↪  "binomial")
summary(modelo)
```

4 Concentração de renda

4.1 convey

- O Pacote convey permite estimar diversas medidas de concentração de renda para dados provenientes de pesquisas com planos amostrais complexos.

```
library(convey)
pnadc <- convey_prep(pnadc)
```

4.2 Índice de Gini

- Para medir a concentração de renda no país por meio do índice de gini, podemos:

```
giniHab <- svygini(~VD4020, pnadc, na.rm  =  TRUE)
giniHab

giniUF <- svyby(~VD4020, by = ~UF,
                pnadc, svygini, na.rm  =  TRUE)
giniUF

gini_regiao <- svyby(~VD4020, by = ~ regiao,
                     pnadc, svygini, na.rm  =  TRUE)
gini_regiao
```

4.3 Curva de Lorenz

- A Curva de Lorenz é um gráfico utilizado para relacionar a distribuição relativa de renda pelas pessoas. A área entre essa curva e a reta identidade, é uma das formas de definir o coeficiente de Gini.

```
curvaLorenz <- svylorenz(~VD4020, pnadc,
                         quantiles = seq(0, 1, .05), na.rm  =  TRUE)
```

5 Calcular decis de renda

Este artigo documenta como calcular as rendas de acordo com os decis populacionais na PNAD Contínua. Essas são informações relevantes para entender a natureza distributiva das políticas públicas.

5.1 Qual arquivo devo utilizar?

A referência de rendimentos de todas as fontes é a 1ª e 5ª entrevista anual. São esses os dados que balizam as publicações oficiais do IBGE. Assim, vamos inicialmente seguir os passos abaixo.

Primeiro passo: baixar os pacotes

```
# baixar pacotes
library(srvyr); library(tidyverse); library(PNADcIBGE); library(lubridate)
library(convey); library(knitr); library(kableExtra)
```

Segundo passo: ler os arquivos

```
# fixar diretório
setwd("C:/curso_r/dados/PNADC")
pnadc <- read_pnadc("PNADC_2018_visita1.txt",
↪  "Input_PNADC_2018_visita1.txt")
pnadc <- pnadc_labeller(pnadc,
↪  "dicionario_PNADC_microdados_2018_visita1.xls")
deflator <- readxl::read_excel("deflator_PNADC_2018.xls")
```

Terceiro passo: aplicar os deflatores

```
# ajustar classes das variaveis para cruzamento
deflator <- deflator %>%
  mutate(Ano = as.character(ano),
    Trimestre = as.character(trim),
         uf = as.character(uf))
pnadc$uf <- substr(pnadc$Estrato, 1, 2)
# cruzar pnadc com deflatores
pnadc <- inner_join(pnadc, deflator,
          by =c("uf", "Ano", "Trimestre"))
```

Quarto passo: estruturar os bancos para o survey

```
# transformar banco em classes de survey
pnadc <- pnadc_design(pnadc)
pnadc <- as_survey(pnadc)
pnadc <- convey_prep(pnadc)
```

5.2 DECIS DA RENDA DOMICILIAR

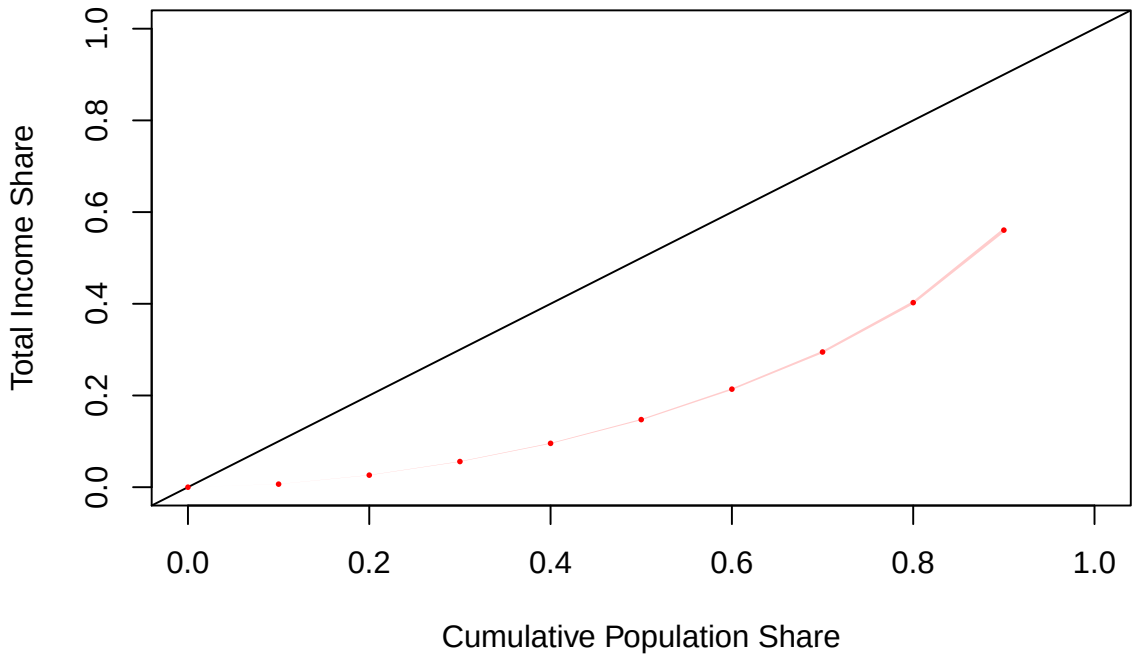
Para observar a distribuição da renda na sociedade brasileira, é necessário extrair os decis de renda. Isso é facilmente realizável por meio do pacote *convey* que extrai a curva de lorenz de uma distribiuição - no caso específico, a renda.

Antes, vamos criar a variável renda domiciliar per capita deflacionada, de acordo com o anexo do IBGE sobre o deflacionamento dos valores da PNADC.

```
# variavel deflaciona: rdpc_co2e
pnadc <- pnadc %>%  mutate(rdpc_co2e = VD5002 * CO2e)
```

Em seguida, vamos extrair a curva de lorenz. Atenção: é preciso que o objeto seja da classe convey: ver *?convey_prep*. Abaixo, estamos tirando apenas os decis, por isso fizemos uma sequencia de 0 a 0.9 - o equivalente a 10 decis.

```
# a distribuicao da renda domiciliar per capita
curvaLorenz <- svylorenz(~ rdpc_co2e, pnadc, quantiles = seq(0, .9, 0.1),
                         na.rm  =  TRUE, plot = TRUE)
```




```
curvaLorenz
```

```
## $quantiles
##                0         0.1        0.2        0.3        0.4       0.5       0.6
## rdpc_co2e 0 0.006646624 0.02624461 0.05591789 0.09559275 0.1473845 0.2138203
##                  0.7       0.8       0.9
## rdpc_co2e 0.2948319 0.4024786 0.5606674
##
## $CIs
```

```
## , , rdpc_co2e
##
##         0         0.1        0.2        0.3        0.4       0.5       0.6
## (lower 0 0.006567273 0.02600516 0.05542283 0.09475292 0.1460847 0.2119466
## upper) 0 0.006725976 0.02648407 0.05641295 0.09643258 0.1486843 0.2156940
##               0.7       0.8       0.9
## (lower 0.2922131 0.3989523 0.5557945
## upper) 0.2974507 0.4060049 0.5655403
```

Por meio da distribuição, nota-se o seguinte: os 10% mais pobres apropriam 0,7% da renda, enquanto os mais ricos levam 43,3% (1 - 0,567).

Mas como calcular a renda média dos indivíduos em cada faixa de renda e até aquela faixa de renda? Fácil: replicando esses dados para a população. Como os decis representam a 10% da população, vamos encontrar a população geral e calcular o seu decil.

```
# criar variavel para calcular a população
pnadc <- pnadc %>% mutate(contagem = 1)
# calcular a população
pop <- pnadc %>%
  summarise(pop = survey_total(contagem , na.rm = TRUE))
# Cada 10% representa 1/10 da populacao
pop_decil <- pop$pop/10
```

Ora, se os 10% mais pobres apropriam 0,7% da renda, então apropriam 0,7% de toda a massa de rendimentos. Então, vamos calcular a massa de rendimento domiciliar per capita (nossa variável de interesse).

```
# massa de rendimento domiciliar per capita
massa_ren_dom_pc <- pnadc %>%
  summarise(rend_dom_pc = survey_total(VD5002*CO2e, na.rm = TRUE))
```

Agora, fica fácil calcular as fatias de renda para cada grupo. Primeiro, vamos extrair a informação da curva de Lorenz.

```
# extrai apenas distribuição da renda por decil
distr_rdpc <- data.frame(curvaLorenz$quantiles)

# transpor o vetor da distribuição percentual da renda
# e transformá-lo em um dataframe
rd <- as.data.frame(t(distr_rdpc))
```

Com esse vetor em um dataframe, podemos encadear algums cálculos.

```
# renda por decil - rd
rd <- rd %>%
  mutate(
    decil = 1:10, # numerar decis
    x = lead(rdpc_co2e, default = 1), # o primero decil não é 0, mas sim o
    ↪  valor subsequente
    var = x - rdpc_co2e, # a variacao a cada decil
    massa = var*massa_ren_dom_pc$rdpc_co2e, # ver a massa de renda de cada
    ↪  decil
    renda_entre_decis = massa/pop_decil,
    renda_ate_decil = (x*massa_ren_dom_pc$rdpc_co2e)/(decil*pop_decil),
    massa_acum = cumsum(massa)) # apenas para conferir
```

Assim, temos a nossa tabela com os valores

```
kable(rd %>% select(-x, - var, )) %>%
  kable_styling(bootstrap_options = c("striped", "bordered"), full_width =
↪  F, font_size = 10)
```

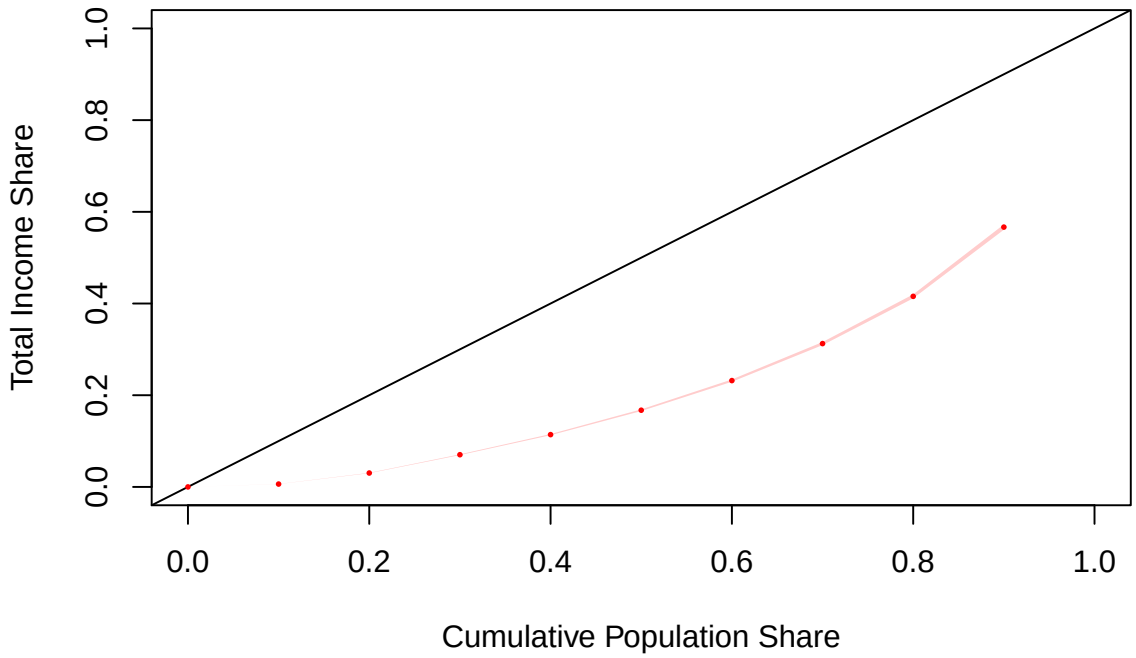
5.3 DECIS DO RENDIMENTO MENSAL EFETIVO DE TODOS OS TRABALHOS PARA PESSOAS DE 14 ANOS OU MAIS DE IDADE

O cálculo dos decis do rendimento efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade segue a mesma lógica do exercício anterior. Porém, há um pequeno detalhe: a população a ser considerada é distinta. Vamos repetir os procedimentos acima.

Antes, vamos criar a variável de todos os rendimentos (VD4020) deflacionada, de acordo com o anexo do IBGE sobre o deflacionamento dos valores da PNADC.

```
# variavel deflaciona: rdpc_co2e
pnadc <- pnadc %>%  mutate(rd20_co2e = VD4020 * CO2e)
```

Em seguida, vamos extrair a curva de lorenz. Atenção: é preciso que o objeto seja da classe convey: ver *?convey_prep*. Abaixo, estamos tirando apenas os decis, por isso fizemos uma sequencia de 0 a 0.9 - o equivalente a 10 decis.




```
## $quantiles
##                0         0.1         0.2       0.3       0.4       0.5       0.6
## rd20_co2e 0 0.006226379 0.03031007 0.0702338 0.1140603 0.1672949 0.2318904
##                  0.7       0.8       0.9
## rd20_co2e 0.3125978 0.4156217 0.5666802
##
## $CIs
## , , rd20_co2e
##
##          0         0.1         0.2         0.3       0.4       0.5       0.6
## (lower 0 0.006104574 0.02986932 0.06939916 0.1127736 0.1654341 0.2293450
## upper) 0 0.006348185 0.03075083 0.07106845 0.1153471 0.1691558 0.2344358
##                 0.7       0.8       0.9
## (lower 0.3092140 0.4112882 0.5609030
## upper) 0.3159816 0.4199552 0.5724573
```

Aqui, o cálculo muda! Mas como calcular a renda média dos indivíduos em cada faixa de renda e até aquela faixa de renda? Fácil: replicando esses dados para a população. Como os decis representam a 10% da população, vamos encontrar a população geral e calcular o seu decil. No entanto, a população a ser considerada é distinta da população geral, pois estamos analisando agora a população ocupada com renda. Ou seja, devemos encontrar as informações apenas para essa população

```
# calcular a população
pop2 <- pnadc %>%
  filter(VD4002 == "Pessoas ocupadas" & VD4018 == levels(VD4018)[1]) %>%
  summarise(pop = survey_total(contagem))

# Cada 10% representa 1/10 da populacao
pop_decil_ocupada <- pop2$pop/10
```

Ora, se os 10% mais pobres apropriam 0,7% da renda, então apropriam 0,7% de toda a massa de rendimentos. Então, vamos calcular a massa de rendimento domiciliar per capita (nossa variável de interesse).

```
# massa de rendimento todos trabalhos
massa_ren_todos_trab <- pnadc %>%
  summarise(renda_todos_trab = survey_total(VD4020*CO2e, na.rm = TRUE))
```

Agora, fica fácil calcular as fatias de renda para cada grupo. Primeiro, vamos extrair a informação da curva de Lorenz.

```
# extrai apenas distribuição da renda por decil
distr_rdpc <- data.frame(curvaLorenz$quantiles)

# transpor o vetor da distribuição percentual da renda
# e transformá-lo em um dataframe
rd <- as.data.frame(t(distr_rdpc))
```

```
rd <- rd %>%
  mutate(
         decil = 1:10, # numerar decis
         x = lead(rd20_co2e, default = 1), # o primero decil não é 0, mas
         ↪  sim o valor subsequente
         var = x - rd20_co2e, # a variacao a cada decil
         massa = var*massa_ren_todos_trab$renda_todos_trab, # ver a massa
         ↪  de renda de cada decil
         renda_entre_decis = massa/pop_decil_ocupada,
         renda_ate_decil =
         ↪  (x*massa_ren_todos_trab$renda_todos_trab)/(decil*pop_decil_ocupada),
         massa_acum = cumsum(massa)) # apenas para conferir
```

rd20_co2e	decil	massa	renda_entre_decis	renda_ate_decil	massa_acum
0.0000000	1	1302309925	144.5155	144.5155	1302309925
0.0062264	2	5037346851	558.9871	351.7513	6339656776
0.0303101	3	8350449745	926.6374	543.3800	14690106521
0.0702338	4	9166761034	1017.2222	661.8405	23856867555
0.1140603	5	11134549189	1235.5849	776.5894	34991416744
0.1672949	6	13510796101	1499.2735	897.0368	48502212845
0.2318904	7	16880758800	1873.2334	1036.4934	65382971644
0.3125978	8	21548483808	2391.2042	1205.8323	86931455452
0.4156217	9	31595410179	3506.0971	1461.4173	118526865631
0.5666802	10	90633199324	10057.4354	2321.0191	209160064955