Tarea06

Daniel Camarena

15/9/2017

1.ENIGH

Para este ejercicio usaremos los datos de la ENIGH (2014). En la tabla concentradohogar que vimos en clase se incluyen las variables alimentos, vestido, vivienda, salud, comunica, educacion y esparci (esparcimiento) que indican el gasto trimestral en cada una de las categorías.

```
concentrado_hogar <- read_csv("datos/concentradohogar.csv")
hogar <- concentrado_hogar %>%
    select(folioviv, foliohog, est_dis, upm, factor_hog, ing_cor, alimentos,
        vestido, vivienda, salud, transporte, comunica, educacion, esparci)
```

Nos interesa analizar los patrones de gasto por decil de ingreso, para ello sigue los siguientes pasos.

1. Calcula los deciles de ingreso usando la variable de ingreso corriente (ing_cor). Debes tomar en cuenta el diseño de la muestra, puedes usar la función survey_quantile() del paquete srvyr o svyquantile() del paquete survey. Reporta las estimaciones y sus errores estándar usando el bootstrap de Rao y Wu.

```
# 1. Definimos el diseño de la encuesta
library(survey); library(srvyr);library(gridExtra)
#> Warning: package 'survey' was built under R version 3.4.1
#> Warning: package 'srvyr' was built under R version 3.4.1
enigh_design <- hogar %>%
    as_survey_design(ids = upm, weights = factor_hog, strata = est_dis)

set.seed(7398731)
enigh_boot <- enigh_design %>%
    as_survey_rep(type = "subbootstrap", replicates = 500)

enigh_boot %>%
    srvyr::summarise(mean_ingcor = survey_mean(ing_cor))
#> # A tibble: 1 x 2
#>   mean_ingcor mean_ingcor_se
#>         <dbl>          <dbl>
#> 1       39719           1008

deciles<- svyquantile(~ing_cor, enigh_boot, quantiles = seq(0.1, 1, 0.1), interval.type = "quantile")
print(deciles)
#> Statistic:
#>      ing_cor
#> q0.1   10622
#> q0.2   14775
#> q0.3   18597
#> q0.4   22682
#> q0.5   27186
#> q0.6   32726
#> q0.7   40057
#> q0.8   51990
#> q0.9   76285
#> q1   4150377
```

```
#> SE:
#>       ing_cor
#> q0.1      144
#> q0.2      165
#> q0.3      193
#> q0.4      218
#> q0.5      239
#> q0.6      348
#> q0.7      484
#> q0.8      769
#> q0.9     1322
#> q1    1318629
```

2. Crea una nueva variable que indique el decil de ingreso para cada hogar. Tips: 1) una función que puede resultar útil es cut2() (de Hmisc)

```
library(Hmisc)
#> Loading required package: lattice
#> Loading required package: Formula
#> Warning: package 'Formula' was built under R version 3.4.1
#>
#> Attaching package: 'Hmisc'
#> The following object is masked from 'package:srvyr':
#>
#>     summarize
#> The following object is masked from 'package:survey':
#>
#>     deff
#> The following object is masked from 'package:gridExtra':
#>
#>     combine
#> The following objects are masked from 'package:dplyr':
#>
#>     combine, src, summarize
#> The following objects are masked from 'package:base':
#>
#>     format.pval, round.POSIXt, trunc.POSIXt, units
#### NOTA: COn cut2 no coinciden los deciles con los calculados con survey
#hog1<- hogar %>% mutate(decil = cut2(ing_cor, g=10))
hogar_decil<-hogar %>% mutate(decil = cut2(hogar$ing_cor, g=10))
dec<-levels(cut2(hogar$ing_cor,g=10))
dec<-as.data.frame(dec)
dec$decil_number<-c(1,2,3,4,5,6,7,8,9,10)

hogar_decil<-hogar_decil %>% left_join(dec, by = c("decil"="dec"))
```

3. Estima para cada decil, el porcentaje del gasto en cada categoría, reporta el error estándar de las estimaciones, usa el bootstrap de Rao y Wu. Tip: 1) agrega una variable que indica para cada hogar el porcentaje de gasto en cada categoría, 2) si usas srvyr puedes usar la función group_by() para estimar la media del porcentaje de gasto por decil.

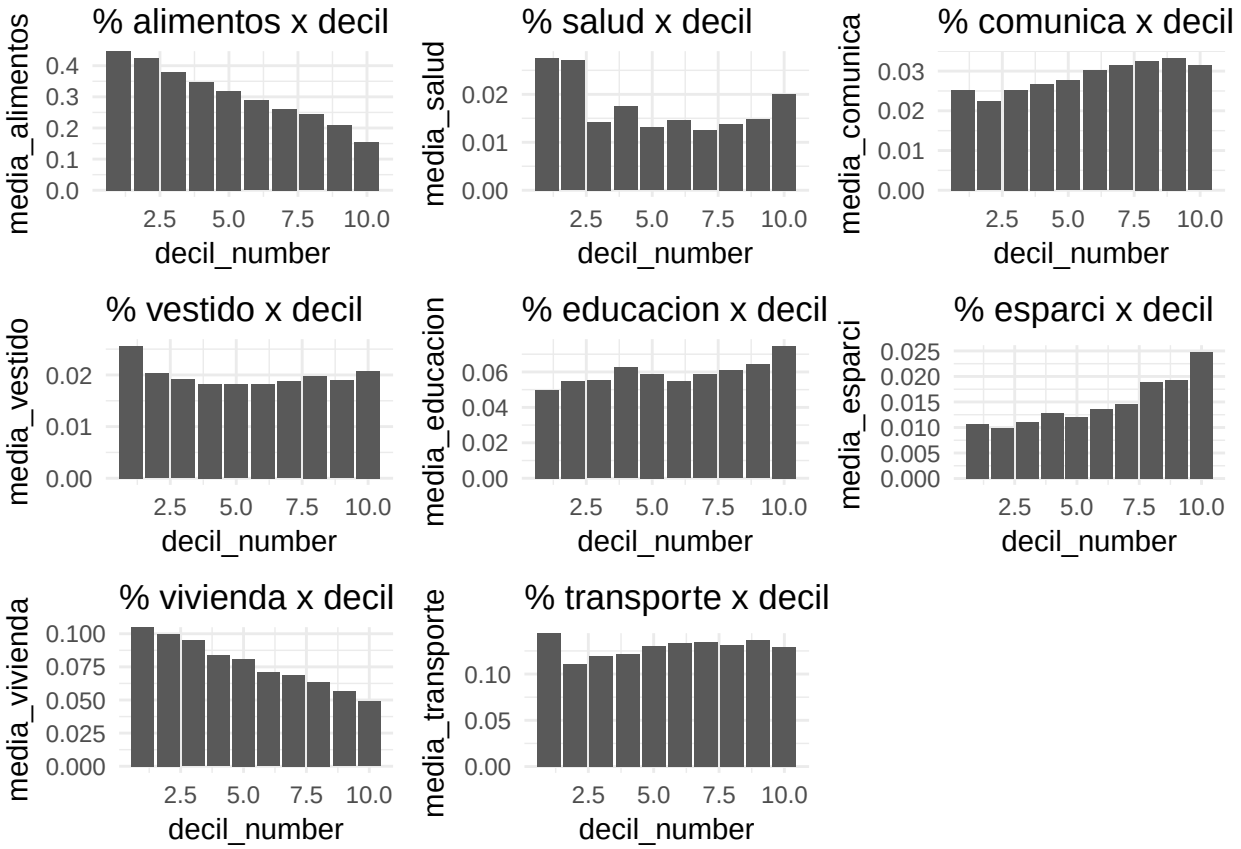
```
hogar_3<- hogar_decil %>% mutate(por_alimento= alimentos/ing_cor, por_vestido= vestido/ing_cor,
                                 por_vivienda= vivienda/ing_cor, por_salud=salud/ing_cor,
                                 por_transporte= transporte/ing_cor, por_comunica=comunica/ing_cor,
                                 por_educacion=educacion/ing_cor, por_esparci= esparci/ing_cor)
```

```r
enigh_design2 <- hogar_3 %>%
    as_survey_design(ids = upm, weights = factor_hog, strata = est_dis)

set.seed(7398731)
enigh_boot2 <- enigh_design2 %>%
    as_survey_rep(type = "subbootstrap", replicates = 500)

# Calculamos el porcentaje de gasto en alimentos por decil
gasto_alimento<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_alimentos = survey_mean(por
#> Warning in svrVar(repmeans, scale, rscales, mse = design$mse, coef = rval):
#> 191 replicates gave NA results and were discarded.
#### La columna con _SE indica el error estandar

# Calculamos el porcentaje de gasto en vestido por decil
gasto_vestido<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_vestido = survey_mean(por_v

# Calculamos el porcentaje de gasto en vivenda por decil
gasto_vivienda<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_vivienda = survey_mean(por
#> Warning in svrVar(repmeans, scale, rscales, mse = design$mse, coef = rval):
#> 191 replicates gave NA results and were discarded.

# Calculamos el porcentaje de gasto en salud por decil
gasto_salud<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_salud = survey_mean(por_salud

# Calculamos el porcentaje de gasto en educacion por decil
gasto_educacion<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_educacion = survey_mean(p

# Calculamos el porcentaje de gasto en transporte por decil
gasto_transporte<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_transporte = survey_mean
#> Warning in svrVar(repmeans, scale, rscales, mse = design$mse, coef = rval):
#> 191 replicates gave NA results and were discarded.

# Calculamos el porcentaje de gasto en comunica por decil
gasto_comunica<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_comunica = survey_mean(por

# Calculamos el porcentaje de gasto en esparci por decil
gasto_esparci<- enigh_boot2  %>% group_by(decil_number) %>% summarise(media_esparci = survey_mean(por_e
```

4. Realiza una gráfica con las estimaciones del paso 3.

```r
p1<-ggplot(data = gasto_alimento, aes(x = decil_number, y = media_alimentos)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% alimentos x decil")

p2<-ggplot(data = gasto_vestido, aes(x = decil_number, y = media_vestido)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% vestido x decil")

p3<-ggplot(data = gasto_vivienda, aes(x = decil_number, y = media_vivienda)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% vivienda x decil")

p4<-ggplot(data = gasto_salud, aes(x = decil_number, y = media_salud)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% salud x decil")

p5<-ggplot(data = gasto_educacion, aes(x = decil_number, y = media_educacion)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% educacion x decil")
```

```
p6<-ggplot(data = gasto_transporte, aes(x = decil_number, y = media_transporte)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% transporte x decil")

p7<-ggplot(data = gasto_comunica, aes(x = decil_number, y = media_comunica)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% comunica x decil")

p8<-ggplot(data = gasto_esparci, aes(x = decil_number, y = media_esparci)) +
  geom_bar(stat = "identity", position = "dodge") + ggtitle("% esparci x decil")

multiplot(p1, p2, p3, p4,p5,p6,p7,p8,cols=3)
```



2. Cobertura de intervalos

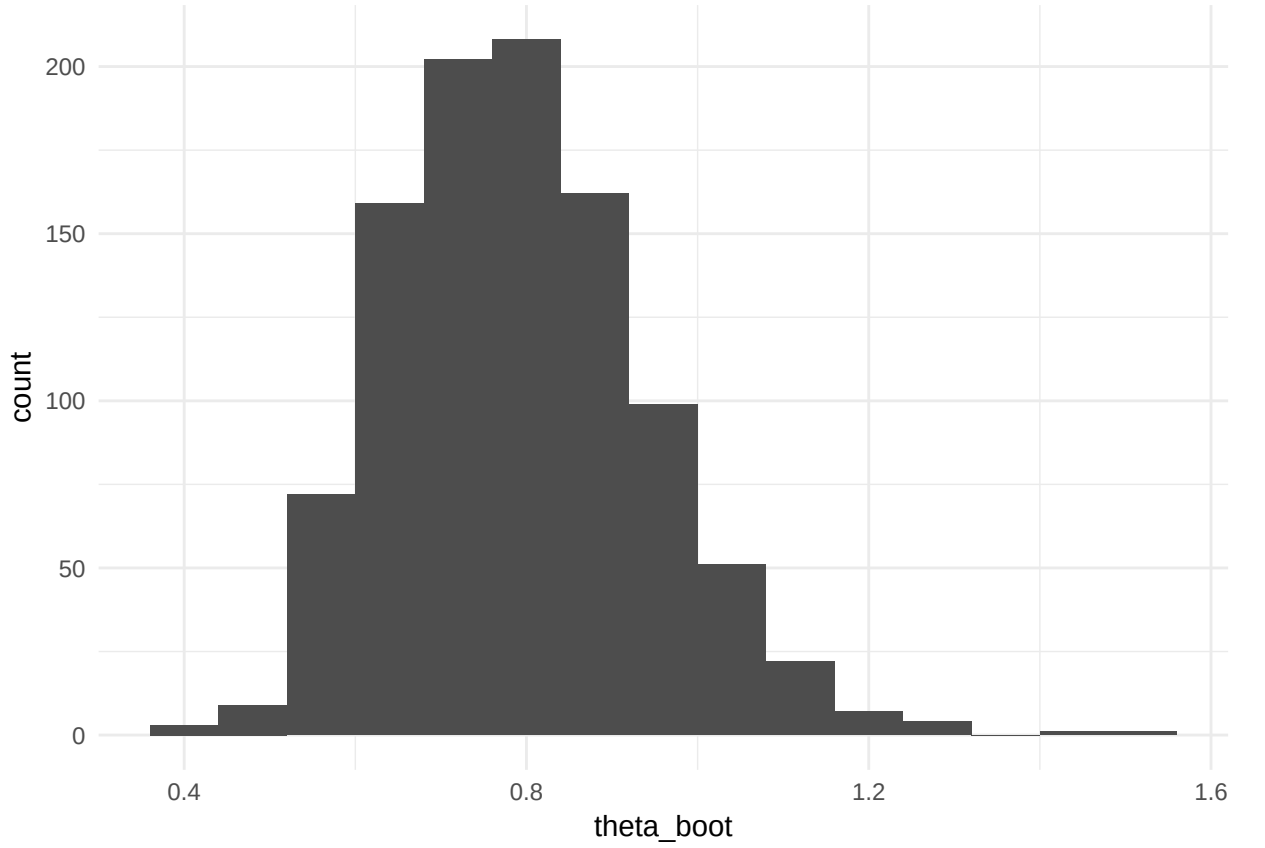
Vamos a retomar de simulación que vimos en clase, donde comparamos los intervalos de confianza construidos con el método de percentiles y usando la aproximación normal ($\hat{\theta} \pm 1.96\hat{se}$).

Generamos una muestra de tamaño 30 (en clase era 10) de una distribución normal estándar, el parámetro de interés es e^{μ} donde μ es la media poblacional.

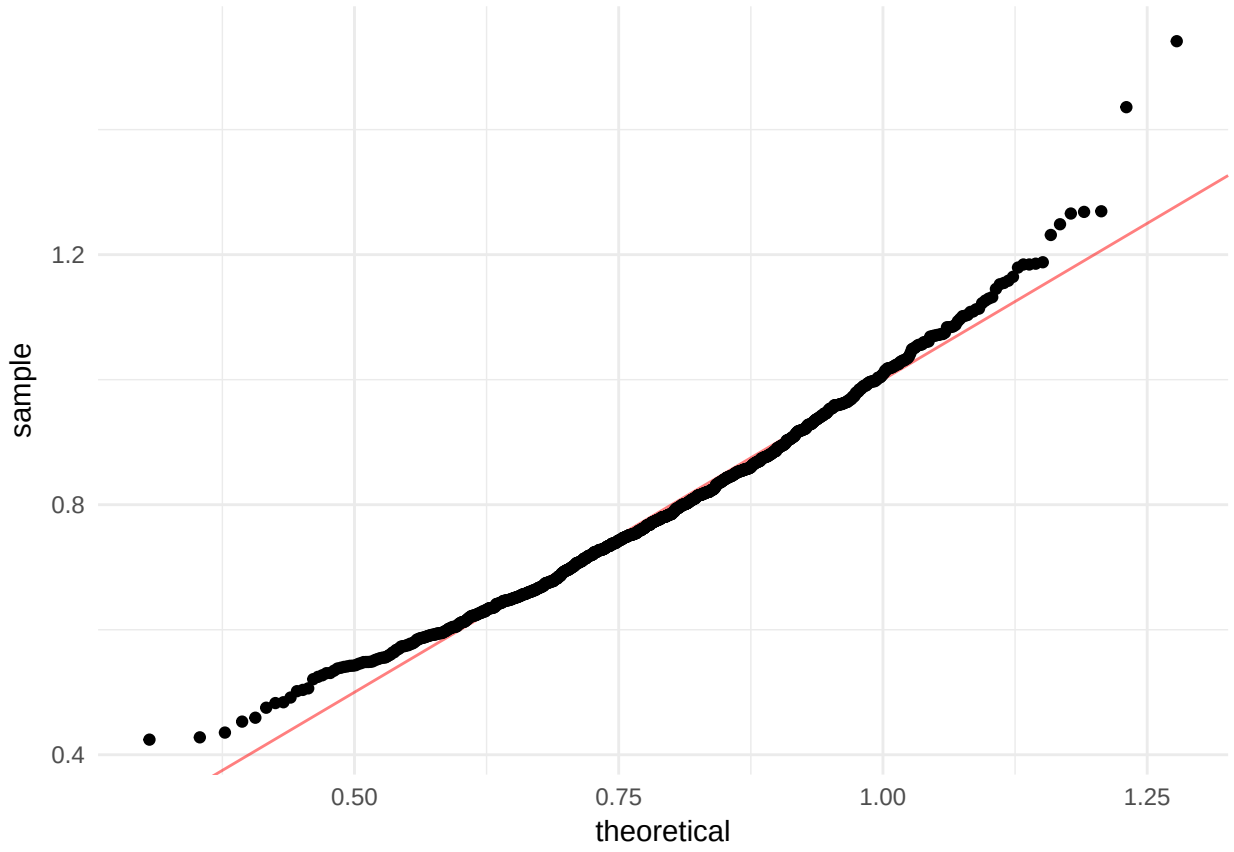
1. Construye intervalos de confianza con el método de percentiles y de aproximación normal.

```
set.seed(766587)
x <- rnorm(30)

boot_sim_exp <- function(){
  x_boot <- sample(x, size = 30, replace = TRUE)
  exp(mean(x_boot))
```

```
}

theta_boot <- rerun(1000, boot_sim_exp()) %>% flatten_dbl()
theta_boot_df <- data_frame(theta_boot)

ggplot(theta_boot_df, aes(x = theta_boot)) +
    geom_histogram(fill = "gray30", binwidth = 0.08)
```




```
ggplot(theta_boot_df) +
    geom_abline(color = "red", alpha = 0.5) +
    stat_qq(aes(sample = theta_boot),
        dparams = list(mean = mean(theta_boot), sd = sd(theta_boot)))
```




```
# Normal
round(exp(mean(x)) - 1.96 * sd(theta_boot), 2)
#> [1] 0.48
round(exp(mean(x)) + 1.96 * sd(theta_boot), 2)
#> [1] 1.06
# Percentil
round(quantile(theta_boot, prob = 0.025), 2)
#> 2.5%
#> 0.55
round(quantile(theta_boot, prob = 0.975), 2)
#> 97.5%
#>   1.1
```

2. ¿Cuál tiene mejor cobertura? Realiza 500 simulaciones de vectores de tamaño 30 de una normal estándar, para cada simulación calcula $\hat{\theta}$ y calcula el porcentaje de realizaciones que caen dentro de cada intervalo de confianza.

```
library(printr)
set.seed(766587)


simul<-function(){
  x <- rnorm(30,0,1)
  theta<-mean(x)
  return(comma(q_mean <- quantile(x, probs = c(0.025, 0.05, 0.1, 0.9, 0.95, 0.975))))
```

```
}

resultado<-rerun(500,simul())

df= as.data.frame(t(as.data.frame(resultado)))
rownames(df)<-NULL
intervalos<-as.data.frame(colnames(df)[apply(df,1,which.max)])
names(intervalos)<-c("interval")
intervalos<-intervalos %>% group_by(interval) %>%
                      summarise(n=n(),
                                prob=n/500)

intervalos
```

interval	n	prob
90%	1	0.002
95%	17	0.034
97.5%	482	0.964