

# Analizė naudojant R. 3 užduotis

Matas Amšiejus

4/24/2021

## 1.

(41 pratimas 163. Pustl; forest.dat). Išrinkta paprastoji atsitiktinė didumo 2000 imtis.

- Įvertinkite šešėlio (hillshade) 9 val. ryte ir vidurdienį (noon) santykį, raskite 95 % pasiklovimo intervalą.
- Įvertinti kiekvieno tipo (cover type) miško aukštį (elevation), raskite 95 % pasiklovimo intervalą.

### Sprendimas

a) Nuskaitome duomenis ir priskiriame pavadinimus:

```
forest <- read.csv("forest.csv",header=F)
names(forest)=c("Elevation","Aspect","Slope","Horiz","Vert","HorizRoad","Hillshade_9am","Hillshade_Noon")
```

Randame ir priskiriame populiacijos dydį. Randame imties elemento priklausymo imčiai tikimybę, tada ėmimo svorį ir priskiriame jį lentelę.

```
N<-nrow(forest)# populiacijos dydis
n<-2000# imties dydis
forest$N<-N
pi_k<-n/N# imties elemento priklausymo imčiai tikimybė
w<-1/pi_k# emimo svoris
forest$w<-w
```

Imame paprastąją negražintinę atsitiktinę imtį su funkcija *sample*:

```
forest_imtis <- forest[sample(N,n),]
```

Sudarome paprastosios atsitiktinės imties planą su paketu *SURVEY*:

```
imties_planas<-svydesign(ids=~1, data = forest_imtis, weights = ~w, fpc = ~N)
```

Naudojame santykinio įvertinio funkciją (lygindami šešėlį 9 ryto ir vidurdienį) *svyratio*

```
santykis<-svyratio(~Hillshade_9am, ~Hillshade_Noon, imties_planas);santykis
```

```
## Ratio estimator: svyratio.survey.design2(~Hillshade_9am, ~Hillshade_Noon, imties_planas)
## Ratios=
##           Hillshade_Noon
## Hillshade_9am      0.9504275
## SEs=
##           Hillshade_Noon
## Hillshade_9am      0.003267598
```

Randame 95% pasiklovimo intervalą:

```
confint(santykis,level = 0.95)
```

```
##                2.5 %      97.5 %
```

```
## Hillshade_9am/Hillshade_Noon 0.9440231 0.9568318
```

b) Įvertiname medžių aukštį pagal tipą (cover). svyby padeda suskirstyti pagal grupes, formula yra vertinamas dydis, by yra grupių skirstymo atributas, design yra imties planas, o FUN yra vertinimo būdas:

```
aukstis<-svyby(formula = ~Elevation, by = ~Cover, design = imties_planas, FUN = svymean);aukstis
```

```
##      Cover Elevation      se
## 1      1  3126.453  5.848580
## 2      2  2916.467  6.141048
## 3      3  2366.250 19.131102
## 4      4  2286.846 31.176981
## 5      5  2784.344 16.317774
## 6      6  2404.267 24.245193
## 7      7  3369.800 18.580031
```

Ieškome 95% pasiklovimo intervalo:

```
confint(aukstis,level = 0.95)
```

```
##      2.5 %   97.5 %
## 1 3114.990 3137.916
## 2 2904.431 2928.504
## 3 2328.754 2403.746
## 4 2225.740 2347.952
## 5 2752.362 2816.326
## 6 2356.747 2451.786
## 7 3333.384 3406.216
```

## 2.

(42 pratimas 164 pusl.; vius.dat).

a) Kintamasis Business žymi pagrindinę komercinę veiklą, kuriai buvo naudotas sunkvežimis. Įvertinkite mylių skaičių kiekvienos komercinės veiklos atveju, raskite 95 % pasiklovimo intervalą.

b) Įvertinkite kiek vidutiniškai mylių galima nuvažiuoti su vienu galonu kuro (MPG) kiekvienam transmisijos tipui.

c) Apskaičiuokite santykį 2002 metais nuvažiuotų mylių (miles\_annl) su per visą transporto priemonės naudojimo laiką nuvažiuotų mylių (miles\_life) skaičiumi, raskite 95 % pasiklovimo intervalą.

### Sprendimas

a) Nuskaitome duomenis:

```
vius<-read.csv("vius.csv")
head(vius)
```

```
##      STRATUM ADM_STATE STATE TRUCKTYPE TABTRUCKS HB_STATE BODYTYPE ADM_MODELYEAR
## 1      11      1      AL      1  3625.768      AL      1      7
## 2      11      1      AL      1  3625.768      AL      1     16
## 3      11      1      AL      1  3625.768      AL      1      5
## 4      11      1      AL      1  3625.768      AL      1      3
## 5      11      1      AL      1  3625.768      AL      1     16
## 6      11      1      AL      1  3625.768      AL      1      5
##      VIUS_GVW MILES_ANNL MILES_LIFE  MPG OPCLASS OPCLASS_MTR OPCLASS_OWN
## 1      1      25000    104875   17      1      0      0
## 2      1      500    115000    .      5      0      0
## 3      2     10000    45000 14.8      5      0      0
## 4      1     40000    97000   18      1      0      0
## 5      1      365    136000   15      1      0      0
```

```
## 6      1      20000      77000      19      1      0      0
##      OPCLASS_PSL OPCLASS_PVT OPCLASS_RNT TRANSMSSN TRIP_PRIMARY TRIP0_50
## 1      10      90      0      1      5      30
## 2      100      0      0      1      2      100
## 3      50      50      0      1      2      50
## 4      1      99      0      1      2      50
## 5      0      100      0      1      2      100
## 6      0      100      0      2      2      99
##      TRIP051_100 TRIP101_200 TRIP201_500 TRIP500MORE ADM_MAKE BUSINESS
## 1      20      0      50      0      1      14
## 2      0      0      0      0      3      NA
## 3      20      25      5      0      4      NA
## 4      50      0      0      0      1      10
## 5      0      0      0      0      1      14
## 6      1      0      0      0      4      14
```

Randame ir priskiriame populiacijos dydį. Randame ėmimo svorį ir priskiriame jį lentelę.

```
N<-sum(vius$TABTRUCKS) # nezinau kodėl, pdf apraše nepaaiškinta, ka reiskia TABTRUCKS, bet tarkime
vius$N<-N
w<-N/nrow(vius) # randame emimo svorius
vius$w<-w
```

Sudarome ėmimo planą su paketu *Survey*:

```
imties_planas2<-svydesign(ids=~1, data = vius, weights = ~w, fpc = ~N)
```

Kaip ir pirmoje užduotyje b dalyje naudojame svyby funkciją (*na.rm* (šiaip pagal aprašą turėtų būti *na.rm.by*, bet jis net neveikia) atmeta grupes kurios yra tuščia vertė (NA)):

```
truck_business<-svyby(formula = ~MILES_LIFE, by = ~BUSINESS, design = imties_planas2, FUN = svytotal, na.rm=TRUE)
```

```
##      BUSINESS      MILES_LIFE      se
## 1      1 5.216605e+12 52542798200
## 2      2 5.821797e+11 13118150223
## 3      3 3.650287e+12 45051634835
## 4      4 4.566603e+11 15490700927
## 5      5 2.762277e+11 10074281876
## 6      6 3.338308e+12 38604614782
## 7      7 7.242084e+11 18300890549
## 8      8 8.345894e+11 18638826781
## 9      9 8.952125e+11 18935229773
## 10     10 4.378450e+10 2414432837
## 11     11 7.392129e+11 17858118894
## 12     12 7.269057e+10 5403873095
## 13     13 2.018296e+11 9098501306
## 14     14 2.904084e+11 9621163406
```

Randame 95% pasiklioavimo intervalą:

```
confint(truck_business, level=0.95)
```

```
##      2.5 %      97.5 %
## 1 5.113623e+12 5.319587e+12
## 2 5.564686e+11 6.078908e+11
## 3 3.561988e+12 3.738587e+12
## 4 4.262991e+11 4.870215e+11
## 5 2.564824e+11 2.959729e+11
```

```
## 6 3.262644e+12 3.413971e+12
## 7 6.883393e+11 7.600775e+11
## 8 7.980580e+11 8.711208e+11
## 9 8.581001e+11 9.323249e+11
## 10 3.905230e+10 4.851670e+10
## 11 7.042117e+11 7.742142e+11
## 12 6.209917e+10 8.328196e+10
## 13 1.839969e+11 2.196624e+11
## 14 2.715513e+11 3.092656e+11
```

b)

Pirma susitvarkome duomenis, nes transmisija dabar yra skaičius, o MPG - raidinė vertė. Konvertuojame:

```
vius$TRANSMSSN <- as.character(vius$TRANSMSSN)
vius$MPG <- as.numeric(vius$MPG)
```

Naudosime tą patį imties planą (tik paleisime iš naujo, nes reikia atnaujinti duomenis), o svyby dabar bus skirstoma pagal transmisiją, vertinamas bus MPG vidurkis:

```
imties_planas2<-svydesign(ids=~1, data = vius, weights = ~w, fpc = ~N)
svyby(~MPG,~TRANSMSSN,imties_planas2,svymean,na.rm=T)
```

```
## TRANSMSSN MPG se
## 1 1 13.489937 0.03155550
## 2 2 7.767717 0.02003537
## 3 3 6.871671 0.16518684
## 4 4 7.773077 0.33321308
```

c) Randame santykinį įvertinį su funkcija *svyratio*:

```
truck_ratio<-svyratio(~MILES_ANNL,~MILES_LIFE,imties_planas2)
```

Randame jo pasiklovimo intervalą:

```
confint(truck_ratio,level=0.95)
```

```
## 2.5 % 97.5 %
## MILES_ANNL/MILES_LIFE 0.09629722 0.09807775
```

### 3.

(43 pratimas 164 pusl.; baseball.dat). Išrinkite paprastąją atsitiktinę 150 žaidėjų imtį.

a) Įvertinkite vidutinį kiekvienos pozicijos žaidėjo atlyginimą, pateikite standartines paklaidas.

b) Įvertinkite kokią dalį visų taškų sudaro namuose surinkti taškai, raskite 95 % pasiklovimo intervalą.

#### Sprendimas

a)

Nuskaitome, sutvarkome duomenis

```
baseball<-read.csv("baseball.csv", header = F)
names(baseball)[4]<-"Alga"#a) uzduociai
names(baseball)[5]<-"Poz"
names(baseball)[16]<-"Viso_taskai"#b) uzduociai
names(baseball)[20]<-"HM_taskai"
```

Duomenims priskiriame populiaciją N ir imties svorį w (su imtimi n = 150):

```
n<-150
N<-nrow(baseball)
baseball$N<-N
```

```
w<-N/n
baseball$w<-w
```

Sudarome 150 žaidėjų imtį:

```
baseball_imtis<-baseball[sample(N, n), ]
```

Sukuriame imties planą:

```
imties_planas3<-svydesign(ids=~1,data = baseball_imtis, weights = ~w, fpc = ~N)
```

Vertiname vidutinį žaidėjo atlyginimą (ir standartinį nuokrypį) pagal poziciją:

```
vid_alg<-svyby(~Alga, ~Poz, imties_planas3, svymean);vid_alg
```

```
##      Poz      Alga      se
## 1B   1B 4750260 1215256.1
## 2B   2B  892350  265825.2
## 3B   3B 1338015  345020.2
## C    C 1675369  670847.4
## CF   CF 3036222 1124867.1
## LF   LF 2700455  676339.9
## P    P 1892896  308671.0
## RF   RF 4254729 1844984.1
## SS   SS 1775415  606938.9
```

b) Naudodami tą patį imties planą ieškome santykinio namų bazės taškų (HM) ir visų taškų įvertinio.

```
taskai_sant<-svyratio(~HM_taskai, ~Viso_taskai, imties_planas3);taskai_sant
```

```
## Ratio estimator: svyratio.survey.design2(~HM_taskai, ~Viso_taskai, imties_planas3)
## Ratios=
##          Viso_taskai
## HM_taskai  0.2278828
## SEs=
##          Viso_taskai
## HM_taskai  0.01395995
```

Tikriname pasikliautinį intervalą (pasirodo numatytasis yra 95%):

```
confint(taskai_sant)
```

```
##                2.5 %    97.5 %
## HM_taskai/Viso_taskai 0.2005218 0.2552438
```