Statistinės duomenų analizės praktinės užduotys

2017

3. **Imčių generavimas.**

```r
library(knitr)
set.seed(42)
n <- 300
```

(a) Sugeneruoti imtį $X_1, \ldots, X_{300}$, $X_i \sim N(65, 11)$ (kintamasis `norm1`)

```r
norm1 <- rnorm(n, mean=65, sd=11) # Dėstytojo sprendiniuose irgi imta 11, o ne srqt(11)
head(norm1)
```

```
## [1] 80.08054 58.78832 68.99441 71.96149 69.44695 63.83263
```

(b) Su tuo pačiu generatoriumi sugeneruoti imtį $Y_1, \ldots, Y_{300}$, $Y_i \sim N(65, 1)$ (kintamasis `norm2`). Palyginti empirines charakteristikas.

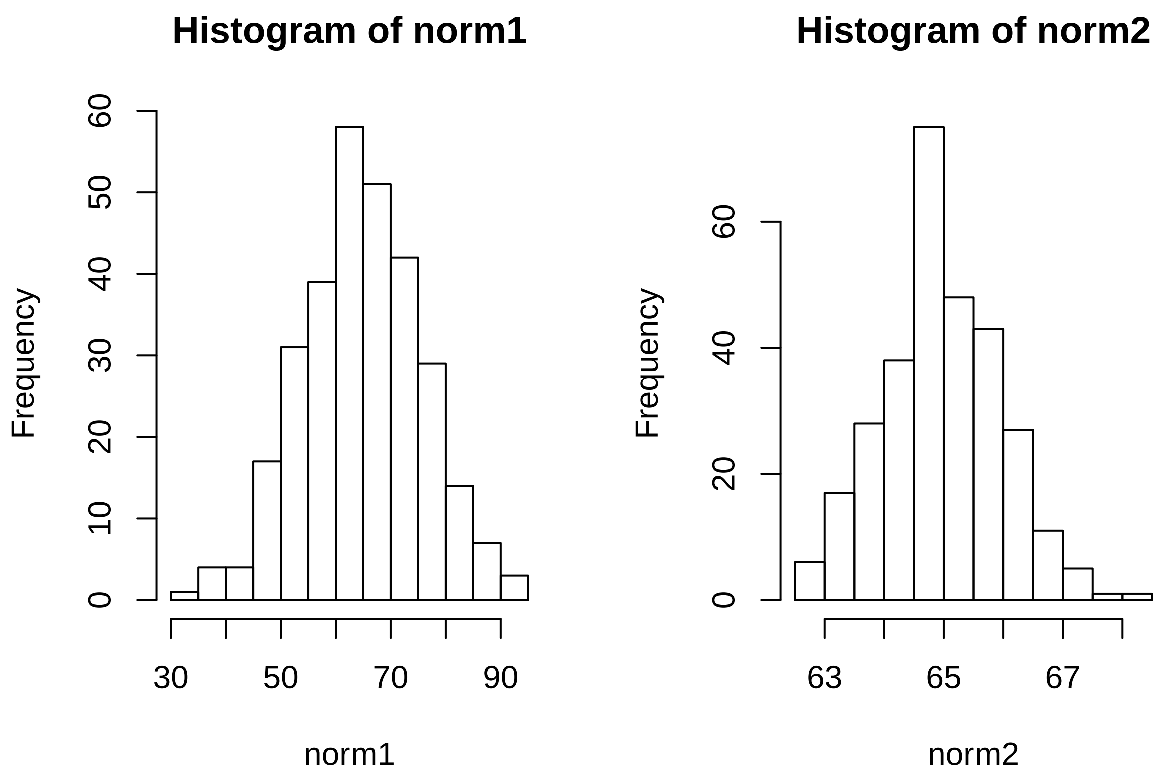
```r
norm2 <- rnorm(n, mean=65, sd=1)
head(norm2)
```

```
## [1] 64.99538 65.76024 65.03899 65.73507 64.85353 64.94211
```

```r
desc_df <- function(df) {
  # Calculates descriptive statistics, returns data frame
  std <- function(x) sd(x) / sqrt(length(x))

  do.call(data.frame,
          list(
                mean = apply(df, 2, mean),
                median = apply(df, 2, median),
                std.deviation = apply(df, 2, sd),
                variance = apply(df, 2, var),
                min = apply(df, 2, min),
                max = apply(df, 2, max),
                s.e.mean = apply(df, 2, std),
                n = apply(df, 2, length)
            )
         )

}

kable(desc_df(data.frame(norm1=norm1, norm2=norm2)))
```

	mean	median	std.deviation	variance	min	max	s.e.mean	n
norm1	64.76043	64.73679	10.8598608	117.9365771	32.07601	94.72080	0.6269944	300
norm2	64.97249	64.93875	0.9885735	0.9772775	62.53866	68.22907	0.0570753	300

```r
oldpar <- par(mfrow=c(1, 2))
hist(norm1)
hist(norm2)
```




```r
par(oldpar)
```

(c) Sugeneruoti dar šešias tūrio $n = 300$ imtis (kintamieji `eksp`, `tolyg`, `ber`, `bin`, `puas`, `geom`) iš atitinkamai eksponentinio $E(65)$, tolygaus $U(54, 76)$, Bernulio $Bern(0.4)$, binominio $Bin(10, 0.4)$ bei Puasono $P(65)$ skirstinių

```r
eksp <- rexp(n, rate=1 / 65)
head(eksp)
```

```
## [1] 84.54347 53.27114 43.98404 22.66498 46.21404 21.74191
```

```r
tolyg <- runif(n, min=54, max=76)
head(tolyg)
```

```
## [1] 73.73343 56.33967 72.45799 72.59813 54.59216 70.83766
```

```r
ber <- rbinom(n, size=1, prob=0.4)
#    binominio a.d atvejis, kai size parametras lygus 1.
head(ber)
```

```
## [1] 1 1 0 0 0 0
```

```r
bin <- rbinom(n, size=10, prob=0.4)
#    size: number of trials (zero or more).
#    prob: probability of success on each trial.
head(bin)
```

```
## [1] 3 3 5 7 3 3
```

```r
puas <- rpois(n, lambda=65)
#    lambda: vector of (non-negative) means.
head(puas)
```

```
## [1] 50 66 70 55 45 69
```

```r
geom <- rgeom(n, prob=0.4) # Dėstytojas praleido šitą salygoje. L. p. parametrą.
#    prob: probability of success in each trial. '0 < prob <= 1'.
head(geom)
```

```
## [1] 1 7 0 4 1 1
```

(d) Visų skirstinių atveju palyginkite teorinius ir empirinius vidurkius bei dispersijas

```r
df_dist <- data.frame(
                        norm1=norm1,
                        norm2=norm2,
                        eksp=eksp,
                        tolyg=tolyg,
                        ber=ber,
                        bin=bin,
                        puas=puas,
                        geom=geom
                      )

comparisons <- do.call(data.frame, list(
                   empiric.mean = apply(df_dist, 2, mean),
                   empiric.var = apply(df_dist, 2, var),
                   theor.mean = c(65, 65, 65, (54+76)/2, 0.4, 0.4 * 10, 65, 1/0.4),
                   theor.var = c(
                                 121,
                                 1,
                                 65^2,
                                 1/12 * (76-54)^2,
                                 0.4 * (1-0.4),
                                 10 * 0.4 * (1-0.4),
                                 65,
                                 (1-0.4)/0.4^2
                                )
                ))
kable(comparisons)
```

	empiric.mean	empiric.var	theor.mean	theor.var
norm1	64.7604308	117.9365771	65.0	121.00000
norm2	64.9724898	0.9772775	65.0	1.00000
eksp	67.3080902	4576.0021214	65.0	4225.00000
tolyg	65.4629650	44.2780883	65.0	40.33333
ber	0.4166667	0.2438685	0.4	0.24000
bin	4.1633333	2.3444705	4.0	2.40000
puas	64.0366667	59.1391193	65.0	65.00000
geom	1.5266667	3.5076477	2.5	3.75000

Padaryta su R version 3.4.2 (2017-09-28), x86_64-pc-linux-gnu.