Základní popisné statistiky a grafy

8. cvičení

Úvodní nastavení.

- Založte speciální adresář na toto cvičení.
- Stáhněte si datový soubor Hosi.txt a uložte si jej do tohoto adresáře. Můžete si také stáhnout zdrojový kód k dnešnímu cvičení cviceni8.R.
- Otevřete si program R Studio.
- Změňte si pracovní adresář pomocí Session →Set working directory →Choose directory na Váš právě založený adresář nebo použijte příkaz `setwd`, do kterého zadáte cestu do tohoto adresáře.
- Nechejte si vypsat seznam objektů, které jsou aktivní:
  `ls()`
  a případně proveďte před další prací vyčištění
  `rm(list=ls())`

Základní operace v programu R

1. R používáme buď tak, že píšeme příkazy přímo do okna Console nebo (což je preferováno) si příkazy píšeme do zvláštního souboru a odtud je spouštíme.

2. Použijte R jako kalkulačku a spočítejte následující výrazy:

$$1+1, \quad \frac{2}{3}, \quad 3^4, \quad \sqrt{3}, \quad \log(10), \quad \exp(10), \quad \sin\left(\frac{\pi}{2}\right).$$

   Nechte si vypsat nápovědu k funkci log tak, že zadáte ?log. Stejným způsobem si lze zavolat nápovědu ke každé funkci v R.

3. Posloupnost čísel můžeme v R zadat různými způsoby. Vyzkoušejte následující:

```
c(5,6,7,8,9,10)
5:10
(5:10)/5
seq(1,2,by=0.2)
seq(1,2,length=10)
rep(1,5)
```

4. Minulý týden psaly zápočtovou písemku dvě skupiny studentů: pondělní skupina a úterní skupina. Bodové zisky odevzdaných prací (uvedené v náhodném pořadí) si načteme do vektorů x (pondělí) a y (úterý):

```
x=c(95, 98, 95, 56, 81, 73, 98, 75, 55, 97, 69)
y=c(71, 99, 60, 56, 75, 50, 91, 69, 85, 85, 52, 94, 83, 45, 86, 60, 30)
```

   Chceme-li mít v proměnné z body pro obě skupiny dohromady, pak zavoláme

```
z=c(x,y)
```

Pomocí funkcí min, max, mean si spočítejte minimální, maximální a průměrné počty bodů v jednotlivých skupinách a pro obě skupiny společně. Kolik studentů odevzdalo písemku? (Použijte funkci length.)
Počet úspěšných řešitelů a procentuální úspěšnost bychom dostali jako

```
sum(z>=60)
mean(z>=60)*100
```

5. Samostatný úkol: Ve skupině, ve které jste písemku psali Vy, spočtěte úspěšnost a počet úspěšných řešitelů. Dále zjistěte, kolik studentů (bráno obě skupiny společně) dosáhlo lepšího výsledku než Vy. Vyjádřete to jak absolutním počtem, tak i procentuálně.

Práce s daty – popisné statistiky

6. Načtěte si data Hosi.txt. Buď naklikáním pomocí Import data set nebo pomocí příkazu

```
Hosi=read.table("Hosi.txt",header=TRUE)
```

(Pokud voláte příkazem, ujistěte se, že máte správně nastavený pracovní adresář, viz pokyny v úvodu.)

7. Základní prohlídka dat: Na data můžeme nahlédnout kliknutím na jejich název v seznamu proměnných vpravo nahoře. Užitečné příkazy jsou

```
head(Hosi)
dim(Hosi)
summary(Hosi)
```

Můžeme vidět, že v datech máme následující proměnné: porodní hmotnost v g, porodní délku v cm, věk matky, věk otce, hmotnost dítěte v 1 roce v g, délku v 1 roce a pořadové číslo dítěte.

8. Dále nás bude zajímat pouze porodní hmotnost. Můžeme předpokládat, že data odpovídají realizaci náhodného výběru. Pro jednoduchost si je uložíme do vektoru hmot a do n si uložíme rozsah výběru.

```
hmot= Hosi$por.hmot
(n=length(hmot))
```

9. Spočítejte si základní charakteristiky polohy: minimum, maximum, průměr.

10. Spočtěte si medián pomocí funkce median. Pozor ale, R počítá výběrový medián trochu jinak než bylo zavedeno na přednášce (viz dále u kvantilů).

11. Spočítejte si základní charakteristiky variability pomocí funkcí var, sd. Uvědomte si, v jakých jednotkách jsou tyto kvantity.
Variabilitu můžeme ještě charakterizovat rozpětím, které získáme jako rozdíl minima a maxima.

12. Budeme počítat výběrové kvantily porodní hmotnosti. Jak je známo z přednášky, existuje více definic výběrových kvantilů. Zjistíme si, jak je počítá R

```
?quantile
```

Připomeňme si, že na přednášce byly výběrové kvantily definované tak, že pro $\alpha \in (0,1)$ je $\hat{u}_n(\alpha) = X_{(k_\alpha)}$, kde $k_\alpha = \alpha n$, pokud αn je celé číslo, a $k_\alpha = \lfloor n\alpha \rfloor + 1$ pokud αn není celé číslo. Odtud vyčteme, že defaultně je nastaven jiný postup, než jaký byl na přednášce. Pro naše případy tedy budeme používat nastavení type=1. Spočteme si tedy několik výběrových kvantilů

```
(kvant <- quantile(hmot, prob = c(0.1, 0.25, 0.5, 0.75, 0.9), type=1))
```

Pro $\alpha = 0.25$ ověříme, že to opravdu odpovídá definici z přednášky:

```
quantile(hmot,prob=0.25,type=1)
sort(hmot)[floor(n*0.25)+1]
```

Pro $\alpha = 0.99$ vyzkoušejte zadat různé typy ve funkci quantile (1 až 9) a porovnejte výsledky.

13. Můžeme si nechat vykreslit obrázek empirické distribuční funkce a její souvislost s výběrovými kvantily:

```
plot.stepfun(ecdf(hmot), verticals=TRUE,  do.points=FALSE, ylab=expression(F[n](x)),
  main="Empiricka distribucni funkce")
abline(h=0.1, col="blue")
abline(h=0.25, col="blue")
abline(h=0.5, col="blue")
abline(h=0.75, col="blue")
abline(h=0.9, col="blue")
lines(rep(kvant[1], 2), c(-1,0.1), col="red")
lines(rep(kvant[2], 2), c(-1,0.25), col="red")
lines(rep(kvant[3], 2), c(-1,0.5), col="red")
lines(rep(kvant[4], 2), c(-1,0.75), col="red")
lines(rep(kvant[5], 2), c(-1,0.9), col="red")
text(rep(2000,5), c(0.1,0.25,0.5,0.75,0.9)+0.02, labels=c(0.1,0.25,0.5,0.75,0.9), col="blue")
```

Funkce ecdf počítá empirickou distribuční funkci. Tu si pak můžeme vykreslit (funkce plot nebo plot.stepfun – porovnejte výsledky) nebo můžeme chtít její hodnotu v nějakém bodě. Např. pro 4000 zjistíme výsledek ecdf(hmot)(4000). Připomeňte si, co touto kvantitou odhadujeme.

14. Další charakteristikou variability, založenou na kvantilech, je mezikvartilové rozpětí, které je rozdílem třetího a prvního kvartilu. Můžeme si ho spočítat pomocí funkce IQR. Toto číslo je pro popis dat někdy užitečnější než směrodatná odchylka. Uložte si ho do proměnné iqr a vyzkoušejte, že funkce IQR skutečně počítá to, co má.

Popisné grafy

13. Tzv. krabicový graf nám graficky znázorňuje některé popisné statistiky a také nám dává určitou představu o tvaru zkoumaného rozdělení

```
boxplot(hmot, ylab="Porodni hmotnost [g]")
```

Kdybychom chtěli znát souřadnice jednotlivých částí boxplotu:

```
bobj = boxplot(hmot)
bobj$stats
```

(Proměnná bobj je typu list. Pomocí funkce names(bobj) se můžeme nechat vypsat její složky.) Pomocí porovnání s popisnými statistikami hmotnosti summary(hmot) zkuste přijít na to, co je v grafu znázorněno.

Zdůvodnění proč právě takto: obrázek na wikipedii.

```
pnorm(qnorm(0.75)+1.5*(qnorm(0.75)-qnorm(0.25))) -
     pnorm(qnorm(0.25)-1.5*(qnorm(0.75)-qnorm(0.25)))
```

14. Kdybychom chtěli vědět, které hodnoty leží mimo fousy a jaká je jejich relativní četnost:

```
sort(bobj$out)
length(bobj$out)/n
```

15. Do funkce `boxplot` můžeme jako argument dát i více vektorů a získáme tak graf jejich boxplotů vedle sebe. Vyzkoušejte na bodové zisky z písemky.

16. Nyní se budeme zabývat histogramem, který nám slouží jako odhad hustoty rozdělení.

```
hist(hmot)
hist(hmot, prob = TRUE)
```

Jaký je rozdíl mezi výše uvedenými dvěma obrázky?

17. U histogramu je poměrně zásadní počet uvažovaných intervalů. Porovnejte:

```
par(mfrow=c(1,2));
hist(hmot, prob = TRUE, breaks=seq(1750, 5100, by=10), main = "Delka intervalu 10");
hist(hmot, prob = TRUE, breaks=seq(1500, 5500, by=1000), main = "Delka intervalu 1000");
par(mfrow=c(1,1));
```

18. Porovnáme histogram našich dat s hustotou normálního rozdělení, které by mělo střední hodnotu rovnou průměru našich dat a směrodatnou odchylku rovnou výběrové směrodatné odchylce.

```
xbar <- mean(hmot)
smodch <- sd(hmot)
xgrid <- seq(xbar - 3.5*smodch, xbar + 3.5*smodch, length = 500)
fxgrid <- dnorm(xgrid, mean = xbar, sd = smodch)
hist(hmot, prob = TRUE, xlab = "Porodni hmotnost [g]", main = "")
lines(xgrid, fxgrid, col = "red", lwd = 2)
```

Nebo to lze provést následovně:

```
hist(hmot,prob=TRUE, xlab = "Porodni hmotnost [g]", main = "")
curve(dnorm(x,mean=mean(hmot),sd=sd(hmot)),from=min(hmot),to=max(hmot),
   add=T,col="red",lwd=2)
```

19. V budoucnu nás bude často zajímat, zda můžeme data považovat za náhodný výběr z (přibližně) normálního rozdělení. Z grafických metod můžeme použít výše uvedené srovnání histogramu s hustotou, ale z něho někdy nevidíme velmi dobře, jak nám normální rozdělení sedí „v krajích“. Proto je vhodnější podívat se na tzv. Q-Q graf.

```
qqnorm(hmot, cex=0.2)
qqline(hmot)
```

Q-Q graf obecně srovnává výběrové kvantily spočtené z dat (osa y) s teoretickými kvantily nějakého rozdělení. Zde v případě normálního Q-Q grafu odpovídají y souřadnice přímo uspořádaným datům a x-ové souřadnice kvantilům standardizovaného normálního rozdělení. V případě, že data pocházejí z normálního rozdělení, tak by body měly ležet přibližně na přímce. Umíte to teoreticky zdůvodnit?

20. Generování pseudonáhodných čísel: Podíváme se na Q-Q graf a histogram spočtený pro data generovaná přímo z normálního rozdělení:

```
data=rnorm(n=100,mean=20,sd=1)
qqnorm(data)
qqline(data)

hist(data)
```

Pro srovnání si stejný obrázek vytvoříme i pro data generovaná z exponenciálního rozdělení se střední hodnotou :

```
data2=rexp(n=100,rate=1)
qqnorm(data2)
qqline(data2)

hist(data2)
```

Nakonec se podíváme na grafy dat generovaných z t-rozdělení s 2 stupni volnosti:

```
data3=rt(n=100,df=2)
qqnorm(data3)
qqline(data3)

hist(data3)
```

21. Podíváme se na to, že i data generovaná přímo z normálního rozdělení nemusí pro malé n vypadat „ideálně“:

```
n=50
data=rnorm(n,mean=20,sd=1)
hist(data,prob=T)
curve(dnorm(x,mean=mean(data),sd=sd(data)),from=min(data),to=max(data),
      add=TRUE,col="red")

qqnorm(data)
qqline(data)
```

    Zkuste zvyšovat n a dívat se, jaké obrázky dostáváte.

22. Samostatná práce (vyzkoušejte si některé z probraných funkcí): Spočtěte výběrový 90% kvantil počtu bodů z písemky (podle definice z přednášky). Vykreslete si histogram počtu bodů z písemky. Přidejte do obrázku hustotu normálního rozdělení s vhodnými parametry.

Doplňující informace pro zájemce.

1. Funkce hist používá následující výpočet intervalů histogramu: Nejprve se použije tzv. Sturgesovo pravidlo, které říká, že optimální počet intervalů je roven horní celé části z $\log_2(n)$ plus jedna. Pak se použije funkce, která vytvoří „hezké intervaly“

```
(k <- ceiling(1+log2(length(hmot))))
pretty(hmot, k)
```

    Srovnáme tento výsledek s tím, co dělá funkce hist. Uložíme si celý objekt (typu list) do proměnné hobj a podíváme se na jeho složky.

```
hobj <- hist(hmot, prob = TRUE)
hobj$breaks
hobj$counts
```

    Spočtěte si pomocí funkce length kolik tedy máme v histogramu sloupců.

2. Pro názornost budeme uvažovat Q-Q graf pro prvních 25 pozorování hmotnosti a necháme si vypsat souřadnice bodů

```
h=hmot[1:25]
qqnorm(h)
qqline(h)

qobj=qqnorm(h)

sort(qobj$x)
sort(qobj$y)
sort(h)

qnorm(((1:length(h))-1/2)/length(h))
```

Pro $n > 10$ jsou kvantily na ose x počítané na hladině $\frac{i-1/2}{n}$ pro $i = 1, \ldots, n$. Viz také funkce ppoints.