Hrátky s R

Arnošt Komárek

Katedra pravděpodobnosti a matematické statistiky
Matematicko-fyzikální fakulta Univerzity Karlovy v Praze

Poslední úprava: 13. září 2012

Obsah

1 Základy

1.1 Stažení a instalace

Program R je (zdarma při dodržení podmínek GNU GPL licence) ke stažení na
http://www.R-project.org,
respektive z vhodného CRAN zrcadla (pro stahování z Česko-Slovenska doporučuji
http://cran.at.r-project.org,
které leží za humny a navíc se nejedná o zrcadlo, ale o vzor pro ostatní zrcadla). Uživatelé Linuxu si mohou R sami zkompilovat ze zdroje (`R-2.15.1.tar.gz`), případně si stáhnout balíček pro svoji distribuci (podporované jsou Debian, RedHat, SuSe, Ubuntu). Uživatelé MS Windows mohou též kompilovat ze zdroje (ale patrně jim to dá více práce) anebo instalovat klasicky ze souboru `R-2.15.1-win32.exe`.

1.2 Přídavné balíčky

Počet uživatelů R neustále roste a mnozí z nich R doplňují o balíčky vlastních funkcí.[1] Některé z těchto rozšiřujících balíčků se staly standardem a instalují se automatricky se základním Rkem [2] Ostatní je nutno nainstalovat, chceme-li je používat. Známe-li název balíčku, lze ho (na počítači připojeném k internetu) nainstalovat např. příkazem

```
install.packages("NAZEV_BALICKU")
```

Po odeslání tohoto příkazu se objeví dotaz na volbu zrcadla a poté proběhne vše automaticky. Automaticky se též nainstalují všechny (dosud nenainstalované) balíčky, na nichž námi zvolený balíček závisí. S ohledem na potřeby výuky na MFF UK doporučuji nainstalovat balíček `Rcmdr`. Jedná se o GUI rozšíření Rka, které sice příliš používat nebudeme, ale balíček samotný závisí na mnoha jiných balíčcích, z nichž mnohé později využijeme. Instalací `Rcmdr` si tedy zajistíme nainstalování mnohých užitečných doplňků. Příklady použité dále v tomto manuálu využívají balíčků `mvtnorm` a `colorspace`.

1.3 Editace skriptů

Pro samotnou práci se navíc hodí nějaký trochu inteligentní editor. Nejlepší (dle názoru autora) je Xemacs (`http://www.xemacs.org`) – funguje jak v Linuxu tak v MS Windows – nebo GNU Emacs, ke kterým se dá přidat ESS (Emacs Speaks Statistics) (`http://ess.r-project.org/`) a poté najednou editovat skripty a (jejich části) současně spouštět. Neznalého uživatele občas odradí trochu jiné klávesové zkratky pro běžné úkony (např. Copy-Paste) než na jaké je zvyklý (bohužel, Bill Gates v začátcích svého podnikání nedodržel Emacsovou konvenci a vymyslel si svoje zkratky...). Obdobně funguje (pouze v MS Windows) též WinEdt (`http://www.winedt.com`) po nainstalování R přídavného balíku `RWinEdt`. WinEdt lze nainstalovat zdarma na dobu 30 dnů. Po uplynutí této doby se editor začne sám od sebe zavírat bez uložení rozdělané práce. Nicméně jeho cena se pohybuje na rozumné výši (cca 25$) a klesá s počtem zakoupených licencí.

[1] K 13.9.2012 jich je 4 044 (nárust o 22 % oproti stejnému období roku 2011).

[2] Ve verzi 2.15.1 se jedná o balíčky `KernSmooth`, `MASS`, `Matrix`, `boot`, `class`, `cluster`, `codetools`, `compiler`, `foreign`, `grid`, `lattice`, `mgcv`, `nlme`, `nnet`, `parallel`, `rpart`, `spatial`, `splines`, `stats4`, `survival`, `tcltk`, `tools`.

Kromě možnosti propojit některý ze standardních editorů s konzolí, ve které běží R, existují též integrovaná prostředí vytvořená speciálně pro práci s R. Jde například o RStudio (http://rstudio.org/, distribuce zdarma pod GNU GPL licencí). Pro použití R k řešení základních statistických úloh neprofesionálními statistiky existují též nadstavby R, ve kterých uživatel vybírá požadovaný typ analýzy z nabídky v menu, viz např. R balíček Rcmdr. R samotné má ve své MS Windows verzi zabudován též jednoduchý editor (černobílý bez zvýrazňování syntaxe). Příkazy je z něj možné do R konzole přenášet pomocí kombinace kláves `Ctrl-r`.

1.4 Tento manuál

Data použitá v tomto manuálu lze stáhnout z autorova webu:

http://www.karlin.mff.cuni.cz/~komarek/vyuka/dataRko/auta04.dat
http://www.karlin.mff.cuni.cz/~komarek/vyuka/dataRko/auta04.csv
http://www.karlin.mff.cuni.cz/~komarek/vyuka/dataRko/auta04.xls

Skript s kódem z tohoto manuálu lze nalézt na

http://www.karlin.mff.cuni.cz/~komarek/vyuka/2012_13/introR/introR-2012.R

1.5 Úplný začátek

Na začátku každé práce doporučuji nastavit pracovní adresář tak. Neuvede-li se dále explicitně cesta, bude program odsud načítat data, ukládat sem souboru s obrázky atp. Předpokládejme, že chceme mít pracovní adresář `/home/User/Moje/Veci`, respektive `C:\Moje\Veci`. Jako pracovní je nastavíme pomocí (povšimněte si běžných, nikoliv zpětných lomítek, pro uživatele operačních systémů založených na UNIXu jistě žádné překvapení):

```
setwd("/home/User/Moje/Veci")
setwd("C:/Moje/Veci")
```

Dále doporučuji (zejména při práci na veřejných počítačích) „vyčistit“ R prostředí od proměnných vytvořených předchozím uživatelem:

```
rm(list=ls())
```

1.6 Několik základních operací s vektory a maticemi

✧ Operátor přiřazení má tvar `<-` (menšítko a pomlka) anebo `=` (rovnítko).

```
x <- 10
x
x = 10
x
```

✧ Vektor vytvoříme pomocí funkce `c` (concatenate).

```
x <- c(1, 2, 3, 4, 5)
x
```

✧ Vektor mající tvar aritmetické posloupnosti vytvoříme pomocí funkce `seq` (sequence), vektor ve tvaru aritmetické posloupnosti s krokem 1 též pomocí operátoru `:` (dvojtečka).

```
x <- seq(1, 5, by=1)
x
x <- seq(1, 5, length=5)
x
x <- 1:5
x
```

✧ Matici vytvoříme pomocí funkce `matrix`. Data do matice se vyplňují standardně po sloupcích. Chceme-li je vyplňovat po řádcích, musíme nastavit argument byrow funkce `matrix` na TRUE.

```
x <- matrix(seq(1, 11, by=2), nrow=2, ncol=3)
x
x <- matrix(seq(1, 11, by=2), nrow=2, ncol=3, byrow=TRUE)
x
```

✧ Matice vytvořená složením řádků:

```
x2 <- rbind(x, c(0.5, 0.6, 0.7))
x2
```

✧ Matice vytvořená složením sloupců:

```
x3 <- cbind(x, c(0.5, 0.6))
x3
```

✧ Vytvoření jednotkové matice:

```
x4 <- diag(4)
x4
```

✧ Přístup ke složkám vektoru/matice:

```
v <- seq(10, 60, by=10)
v[3]
v[c(1, 4)]
x
x[1,2]
x[,3]
x[2,]
x[1, c(1, 3)]
```

✧ Operace konstanta – matice:

```
10 * x
x / 10
x + 10
x - 10
```

✧ čistě maticové operace.

```
y <- matrix(seq(0, 100, length=6), nrow=2, ncol=3)
y
t(y)
x + y
x - y
```

✧ Příkazy lze též skládat dohromady.

```
print(M <- x %*% t(y))
print(Minv <- solve(x %*% t(y)))
M %*% Minv
round(M %*% Minv, 10)
```

1.7 Třídy objektů

Každý objekt v R má svoji třídu. Základní třídy jsou

✧ `numeric`, resp. `integer`.

```
x1 <- 1:10
class(x1)
x1b <- seq(0, 10, by=0.5)
class(x1b)
```

✧ matrix

```
x2 <- cbind(1:10, seq(10, 100, by=10))
class(x2)
```

✧ character

```
x3 <- c("jaro", "leto", "podzim", "zima", "zima", "leto")
class(x3)
```

✧ logical

```
x4 <- c(TRUE, TRUE, FALSE, FALSE, TRUE, TRUE)
x4 <- c(T, T, F, F, T, T)
class(x4)
print(x4b <- (x3 == "jaro"))
print(x4c <- !x4b)
print(x4d <- (x4 & x4b))
print(x4e <- (x4 && x4b))
print(x4f <- (x4 | x4b))
print(x4g <- (x4 || x4b))
```

✧ factor

```
x5 <- factor(x3)
x5
class(x5)
```

2 Jednorozměrná rozdělení

Pro běžně používaná rozdělení existují v Rku funkce počítající hodnoty hustoty, distribuční funkce, kvantilové funkce a generující pseudonáhodná čísla ze zvoleného rozdělení. Názvy jednotlivých funkcí začínají postupně písmeny `d` (hustota – *density*), `p` (distribuční funkce – *probability*), `q` (kvantilová funkce – *quantile*), `r` (pseudonáhodná čísla – *random*) a dále pokračují názvem (či zkratkou) příslušného rozdělení (např. `binom` pro binomické, `norm` pro normální, `unif` pro rovnoměrné atp.).

2.1 Diskrétní rozdělení

2.1.1 Binomické rozdělení

✧ Hustota (pravděpodobnostní funkce) (`dbinom`), distribuční funkce (`pbinom`) a kvantilová funkce (`qbinom`).

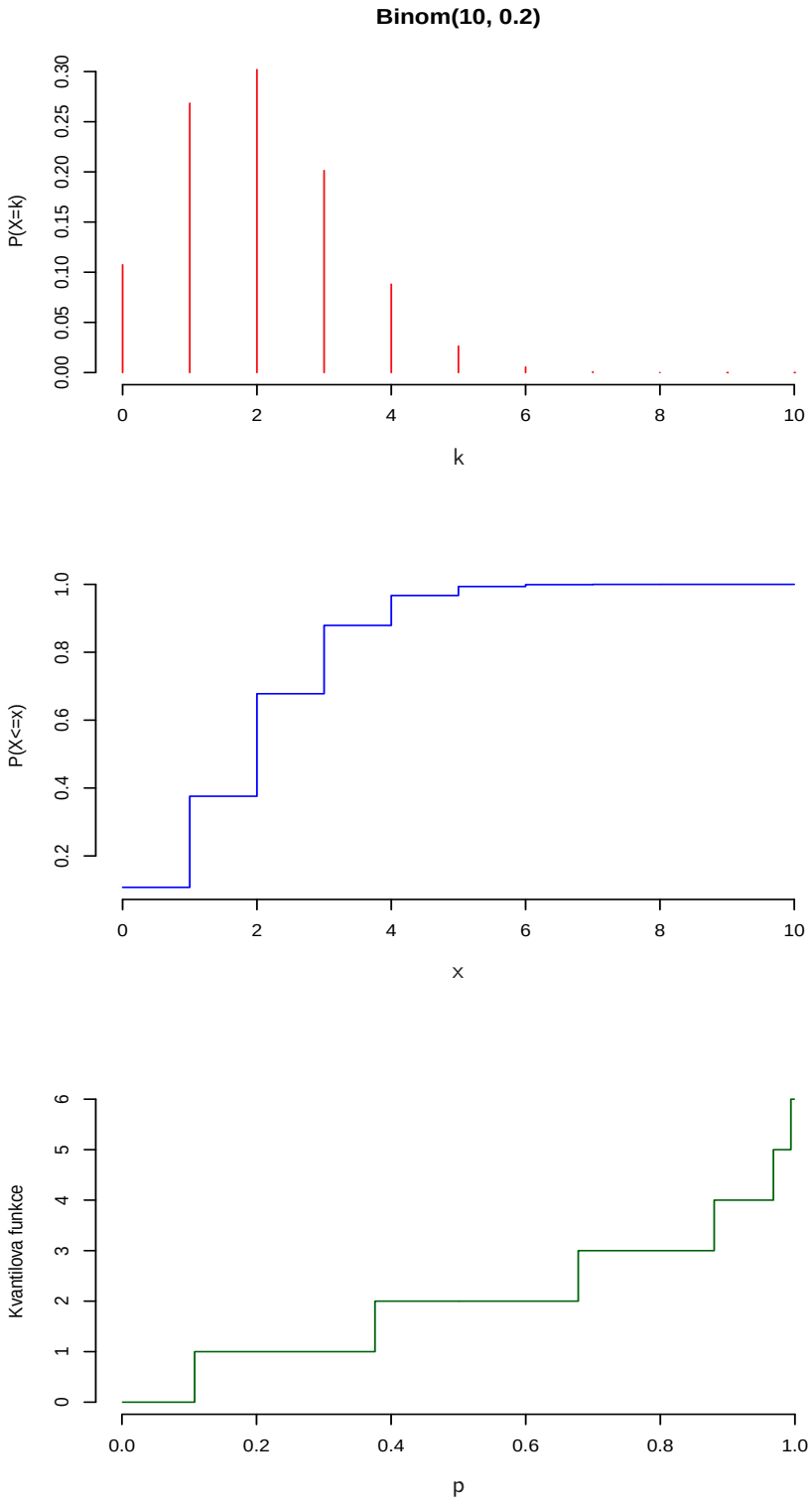
```
p <- 0.2                                ## Pravdepodobost uspechu
n <- 10                                 ## Pocet pokusu
x <- 0:10                               ## Hodnoty
pq <- seq(0.001, 1-0.001, by=0.001)     ## Pravdepodobnosti pro vypocet
                                        ## kvantilove funkce
px <- dbinom(x, size=n, prob=p)
Fx <- pbinom(x, size=n, prob=p)
Finvp <- qbinom(pq, size=n, prob=p)
ukaz <- data.frame(Hodnoty=x, px=px, Fx=Fx)
print(ukaz)
```

✧ Výše spočtené funkce si nakreslíme a uložíme v postscriptu v souboru fig01.ps (obrázek bude 5 palců široký a 10 palců vysoký). Soubor s obrázek bude uložen ve vašem pracovním adresáři (viz `getwd()`).

```
postscript("fig01.ps", width=5, height=10, horizontal=FALSE)
par(mfrow=c(3, 1))                      ## Bude kreslit 3 obrazky pod sebe.
plot(x, px, type="h", xlab="k", ylab="P(X=k)", col="red")
title(main=paste("Binom(", n, ", ", p, ")", sep=""))
plot(x, Fx, type="s", xlab="x", ylab="P(X<=x)", col="blue")
plot(pq, Finvp, type="s", xlab="p", ylab="Kvantilova funkce",
     col="darkgreen")
dev.off()
```

✧ Dále si můžeme vygenerovat 1 000 pseudonáhodných čísel z daného binomického rozdělení (nechcete-li generovat pokaždé stejné hodnoty, buď zakomentujte příkaz `set.seed(...)`, nebo změňte hodnotu seedu):

```
set.seed(18675)
rn <- rbinom(1000, size=n, prob=p)
rn[1:10]
```



Obrázek 1: Binomické rozdělení.

```
table(rn)
prop.table(table(rn))
round(px, 3)
```

2.2 Spojitá rozdělení

2.2.1 Normální rozdělení

✧ Hustota (`dnorm`), distribuční funkce (`pnorm`) a kvantilová funkce (`qnorm`).

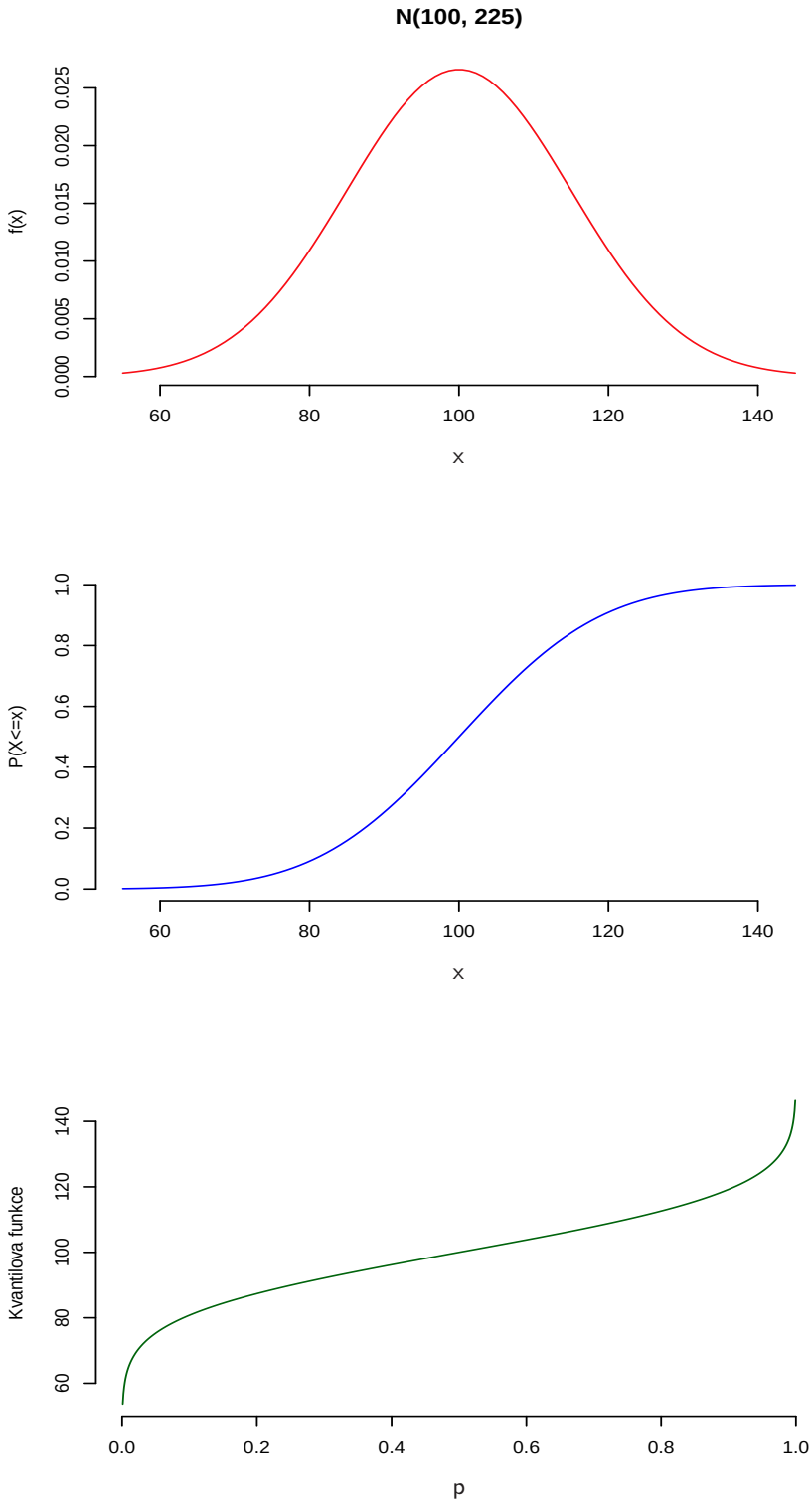
```
mu <- 100                                          ## Stredni hodnota
sigma <- 15                                        ## Smerodatna odchylka
x <- seq(mu-3*sigma, mu+3*sigma, length=100)       ## Hodnoty, kde je hustota viditelne
pq <- seq(0.001, 1-0.001, by=0.001)                ## Pravdepodobnosti pro vypocet kvan
fx <- dnorm(x, mean=mu, sd=sigma)
Fx <- pnorm(x, mean=mu, sd=sigma)
Finvp <- qnorm(pq, mean=mu, sd=sigma)
ukaz <- data.frame(Hodnoty=x, fx=fx, Fx=Fx)
print(ukaz[c(1:5, 45:55, 96:100),])
```

✧ Výše spočtené funkce si nakreslíme a uložíme v pdf v souboru fig02.pdf (obrázek bude 5 palců široký a 10 palců vysoký). Soubor s obrázek bude opět uložen ve vašem pracovním adresáři (viz `getwd()`).

```
pdf("fig02.pdf", width=5, height=10)
par(mfrow=c(3, 1))                          ## Bude kreslit 3 obrazky pod sebe.
plot(x, fx, type="l", xlab="x", ylab="f(x)", col="red")
title(main=paste("N(", mu, ", ", sigma^2, ")", sep=""))
plot(x, Fx, type="l", xlab="x", ylab="P(X<=x)", col="blue")
plot(pq, Finvp, type="l", xlab="p", ylab="Kvantilova funkce",
     col="darkgreen")
dev.off()
```

✧ Kvantily normovaného normálního rozdělení, s kterými se často setkáváme:

```
qnorm(c(0.95, 0.975, 0.99, 0.995))
pp <- c(0.95, 0.975, 0.99, 0.995)
qq <- qnorm(pp)
names(qq) <- paste(pp*100, "%", sep="")
qq
```



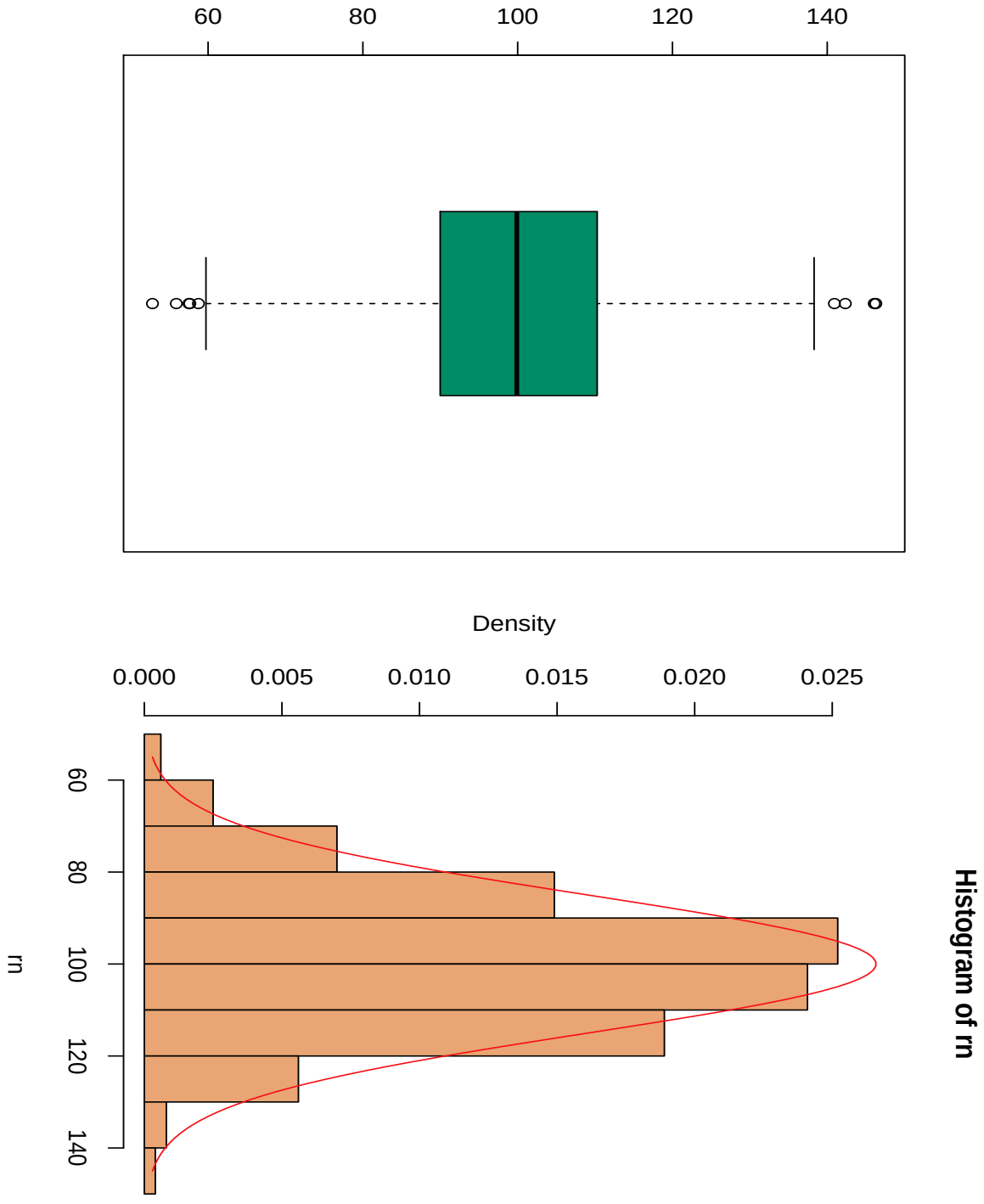
Obrázek 2: Normální rozdělení.

✧ Dále si můžeme vygenerovat 1 000 pseudonáhodných čísel z normálního rozdělení $\mathcal{N}(100,\ 15^2)$:

```
set.seed(221913282)
rn <- rnorm(1000, mean=mu, sd=sigma)
rn[1:10]
mean(rn)
sd(rn)
var(rn)
```

✧ Nakresleme si krabičkový graf (boxplot) a histogram. K histogramu ještě přidáme hustotu příslušného normálního rozdělení. Obrázek tentokrát uložíme jako jpeg v souboru fig03.jpg (jeho rozlišení bude 1 280 × 1 024 bodů). Soubor s obrázek bude uložen ve vašem pracovním adresáři (viz `getwd()`).

```
jpeg("fig03.jpg", width=1280, height=1024)
par(mfrow=c(1, 2))                          ## Bude kreslit 2 obrazky vedle sebe.
boxplot(rn, col="seagreen")
hist(rn, prob=TRUE, col="sandybrown", ylim=range(fx))
lines(x, fx, col="red")
dev.off()
```



Obrázek 3: Náhodný výběr z normálního rozdělení.

3 Vícerozměrná rozdělení

3.1 Dvourozměrné normální rozdělení

✧ Napišme si nejprve funkci, která nám ze zadaných směrodatných odchylek σ_1, σ_2 a korelace ϱ vytvoří příslušnou varianční matici:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \varrho\sigma_1\sigma_2 \\ \varrho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

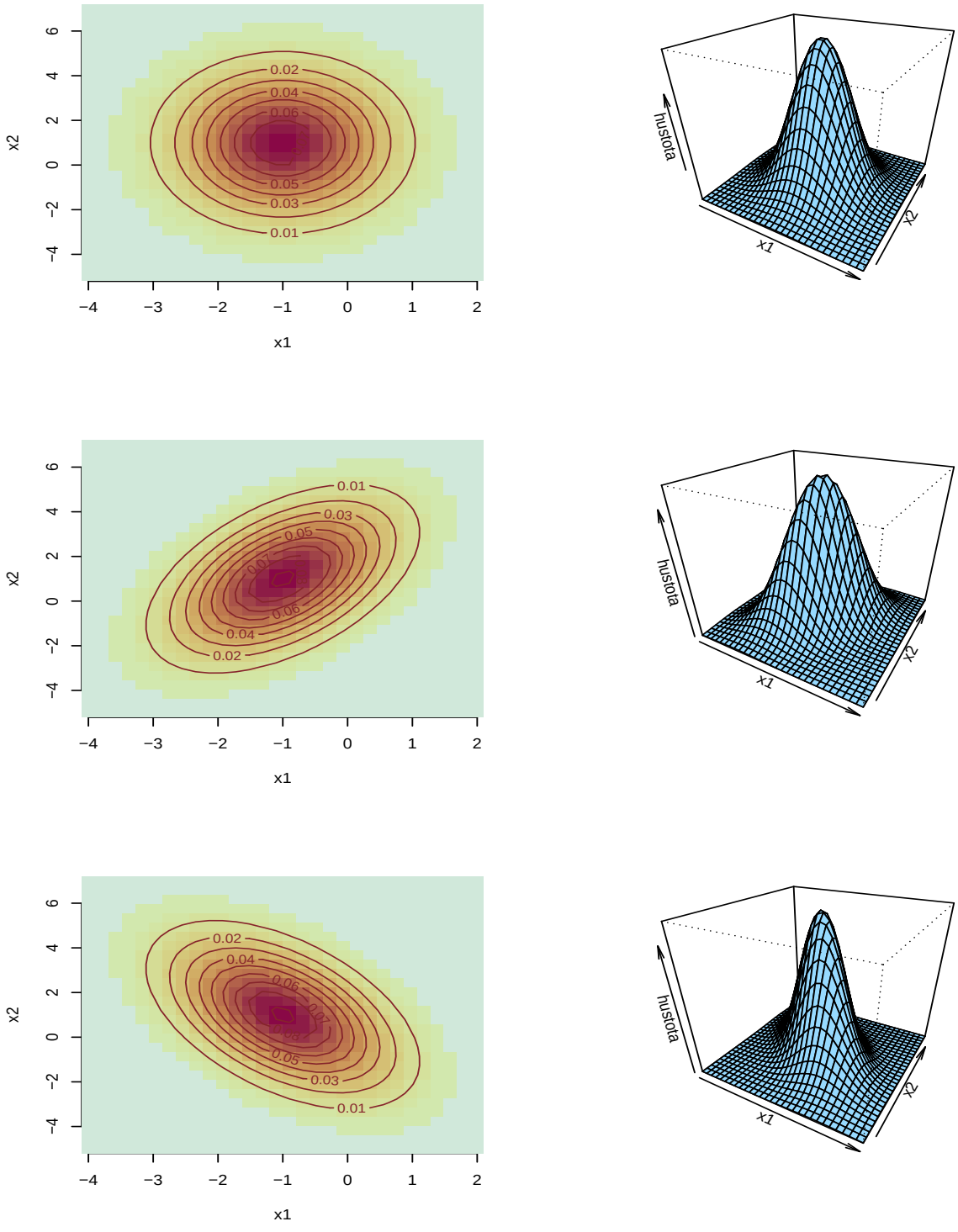
```
CovMat <- function(sigma1, sigma2, rho){
  if (sigma1 <= 0 | sigma2 <= 0 | rho < -1 | rho > 1)
    stop("Nespravne vstupni parametry.")
  Sigma <- matrix(c(sigma1^2, rho*sigma1*sigma2, rho*sigma1*sigma2, sigma2^2),
                  nrow=2)
  return(Sigma)
}
```

✧ Spočtěte pomocí této funkce varianční matice několika různých dvourozměrných normálních rozdělení.

✧ Nakreslete hustoty dvourozměrných normálních rozdělení (hustotu umí počítat funkce `dmvnorm` z balíku `mvtnorm`) pro $\boldsymbol{\mu} = (-1, 1)$, $\sigma_1 = 1$, $\sigma_2 = 2$ a $\varrho \in \{0,\ 0{,}5,\ -0{,}5\}$.

✧ Tušíte, proč se x1 a x2 zvolilo tak, jak se zvolilo? Do objektu BARVY jsme si uložili „pěknou“ paletu barev pro kreslení „mapy“ vytvořenou funkcemi z balíčku `colorspace`.

```
library("mvtnorm")
library("colorspace")
BARVY <- rev(heat_hcl(33, c=c(80, 30), l=c(30, 90), power=c(1/5, 1.3)))
mu <- c(-1, 1)
sigma <- c(1, 2)
rho <- c(0, 0.5, -0.5)
x1 <- seq(mu[1] - 3*sigma[1], mu[1] + 3*sigma[1], length=30)
x2 <- seq(mu[2] - 3*sigma[2], mu[2] + 3*sigma[2], length=30)
XX <- cbind(rep(x1, length(x2)), rep(x2, each=length(x1)))
par(mfrow=c(3, 2), bty="n")
for (i in 1:length(rho)){
  Sigma <- CovMat(sigma[1], sigma[2], rho[i])
  hustota <- matrix(dmvnorm(XX, mean=mu, sigma=Sigma),
                    nrow=length(x1), ncol=length(x2))
  image(x1, x2, hustota, col=BARVY, xlab="x1", ylab="x2")
  contour(x1, x2, hustota, col="brown4", add=TRUE)
  persp(x1, x2, hustota, col="lightblue", theta=30, phi=30)
}
```



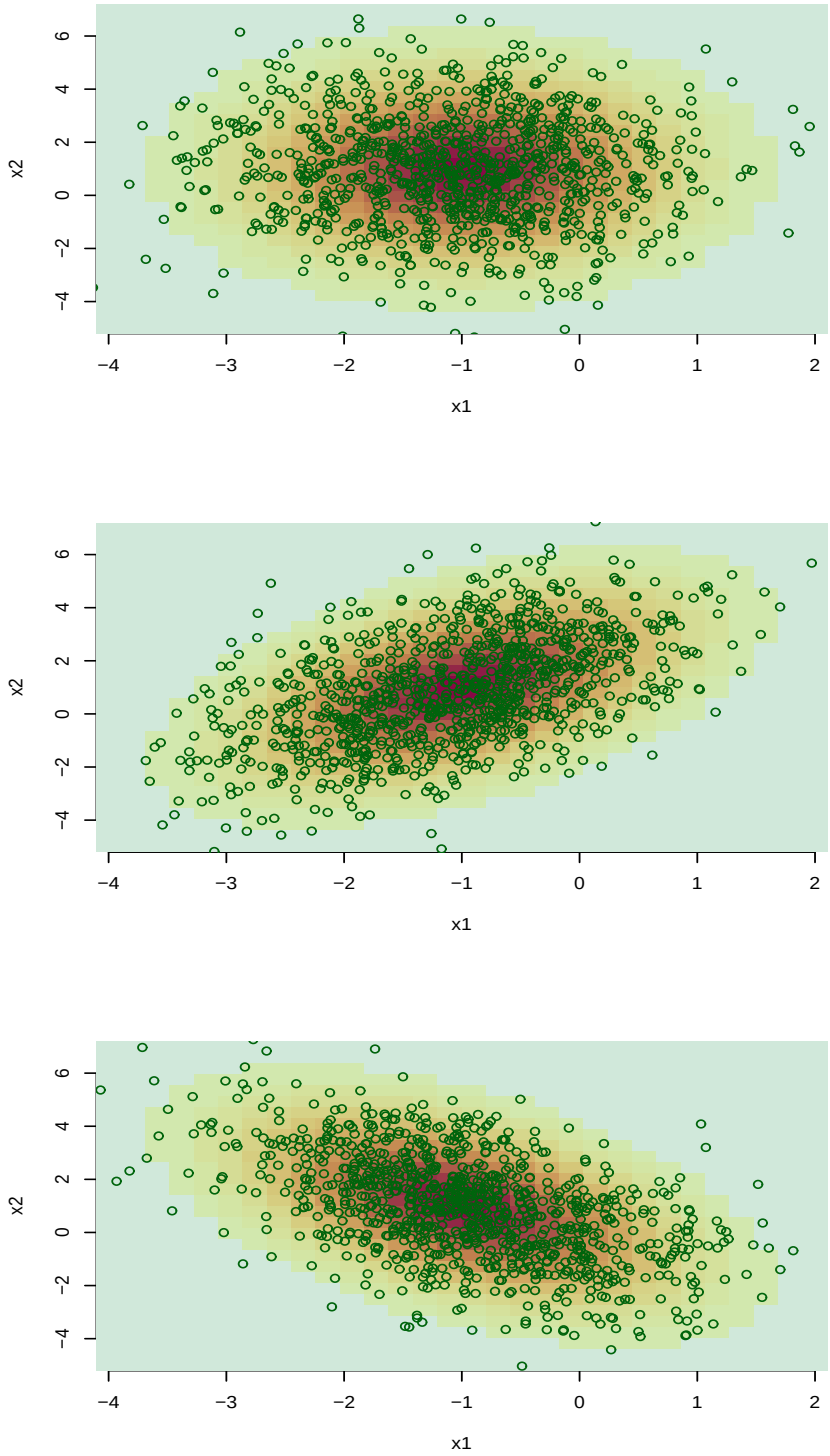
Obrázek 4: Dvourozměrná normální rozdělení.

✧ Zkuste též generovat vždy 1 000 pseudonáhodných čísel z příslušných normálních rozdělení.

✧ Pro nagenerované hodnoty spočtěte výběrové průměry, výběrové směrodatné odchylky a výběrové korelační koeficienty.

✧ Pro nagenerované hodnoty vytvořte bodový graf a podbarvěte ho mapou s příslušnou hustotou. Myslíte, že to je náhoda, že kolečka leží z větší části v „hornaté“ oblasti?

```
set.seed(495265835)
charakt <- matrix(NA, ncol=5, nrow=3)
colnames(charakt) <- c("prum1", "prum2", "sd1", "sd2", "r")
par(mfrow=c(3, 1), bty="n")
for (i in 1:length(rho)){
  Sigma <- CovMat(sigma[1], sigma[2], rho[i])
  hustota <- matrix(dmvnorm(XX, mean=mu, sigma=Sigma), nrow=length(x1), ncol=length(x2))
  xxR <- rmvnorm(1000, mean=mu, sigma=Sigma)
  image(x1, x2, hustota, col=BARVY, xlab="x1", ylab="x2")
  points(xxR[,1], xxR[,2], col="darkgreen")
  #contour(x1, x2, hustota, col="brown4", add=TRUE)
  charakt[i,] <- c(apply(xxR, 2, mean), apply(xxR, 2, sd), cor(xxR)[1,2])
}
print(charakt)
```



Obrázek 5: Náhodné výběry z dvourozměrných normálních rozdělení.

✧ Uměli byste generovat náhodná čísla z vícerozměrného normálního rozdělení bez použití funkce `rmvnorm` z balíku `mvtnorm`?

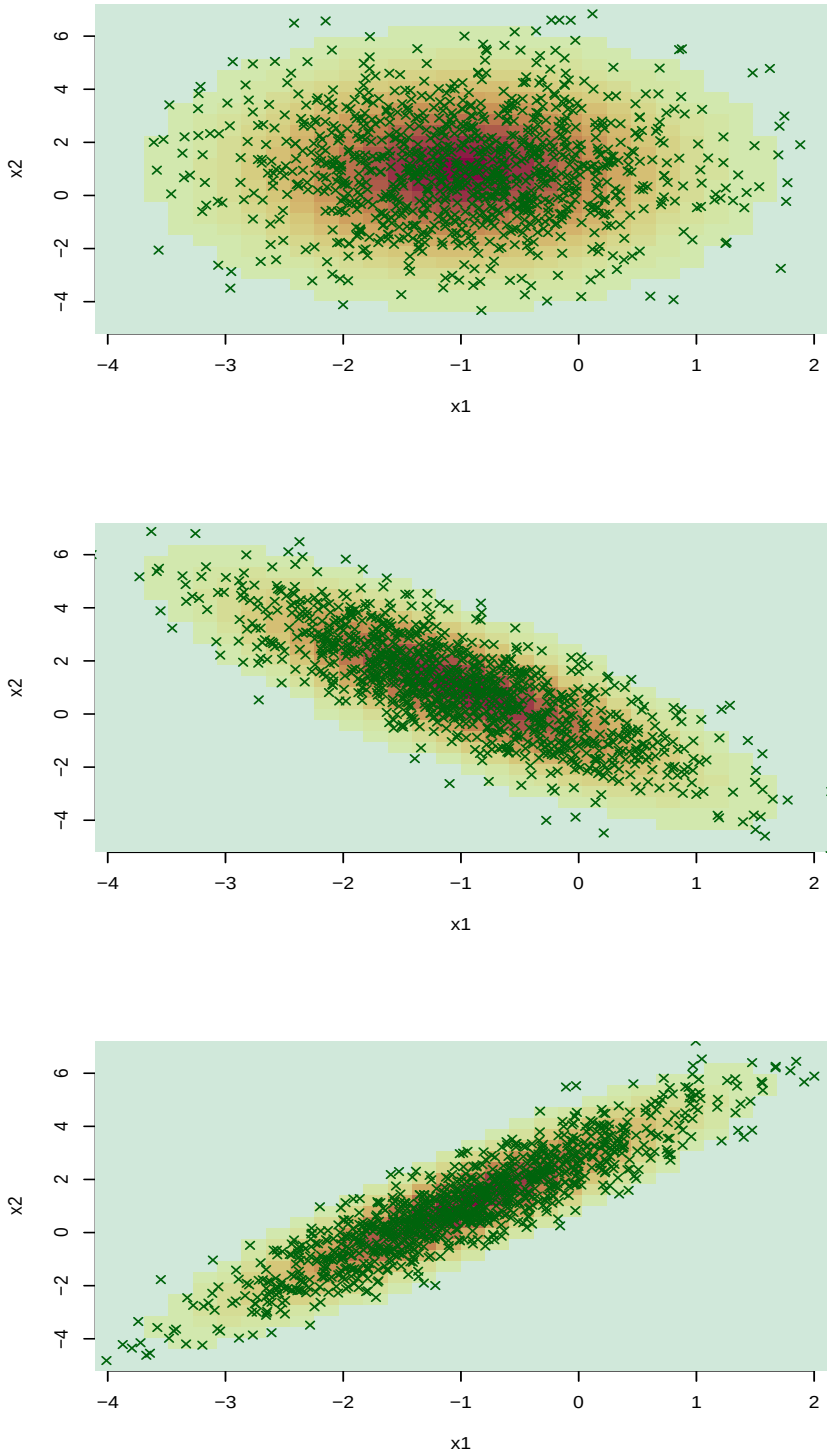
✧ **Rada:** Každou pozitivně definitní matici Σ lze rozložit na $\Sigma = \mathbb{U}'\mathbb{U}$, kde $\mathbb{U}$ je horní trojúhelníková matice (Choleského dekompozice). Dále platí: $\boldsymbol{Z} \sim \mathcal{N}_p(\mathbf{0}, I_p) \Rightarrow \boldsymbol{X} = \boldsymbol{\mu} + \mathbb{U}'\boldsymbol{Z} \sim \mathcal{N}_p(\boldsymbol{\mu}, \Sigma)$.

✧ Choleského dekompozici počítá funkce `chol`. Myslíte, že následující funkce dělá to co chceme?

```
rmvnormMFF <- function(n, mean, sigma){
  p <- length(mean)
  Z <- matrix(rnorm(p*n, mean=0, sd=1), nrow=n, ncol=p)
  U <- chol(sigma)
  X <- rep(mean, each=n) + Z %*% U
  return(X)
}
```

✧ Zkuste znovu generovat z našich dvourozměrných normálních rozdělení, tentokrát s použitím funkce `rmvnormMFF`. Změňte hodnoty korelací mezi jednotlivými složkami náhodného vektoru. Hustotu tentokrát nakreslíme jako mapu, v níž jsou nadmořské výšky odlišeny barvami:

```
rho <- c(0, -0.8, 0.9)
set.seed(16336886)
charakt <- matrix(NA, ncol=5, nrow=3)
colnames(charakt) <- c("prum1", "prum2", "sd1", "sd2", "r")
par(mfrow=c(3, 1), bty="n")
for (i in 1:length(rho)){
  Sigma <- CovMat(sigma[1], sigma[2], rho[i])
  hustota <- matrix(dmvnorm(XX, mean=mu, sigma=Sigma),
                    nrow=length(x1), ncol=length(x2))
  xxR <- rmvnormMFF(1000, mean=mu, sigma=Sigma)
  image(x1, x2, hustota, col=BARVY, xlab="x1", ylab="x2")
  points(xxR[,1], xxR[,2], pch=4, col="darkgreen")
  charakt[i,] <- c(apply(xxR, 2, mean), apply(xxR, 2, sd), cor(xxR)[1,2])
}
print(charakt)
```



Obrázek 6: Náhodné výběry z dvourozměrných normálních rozdělení.

4 Transformace v praxi

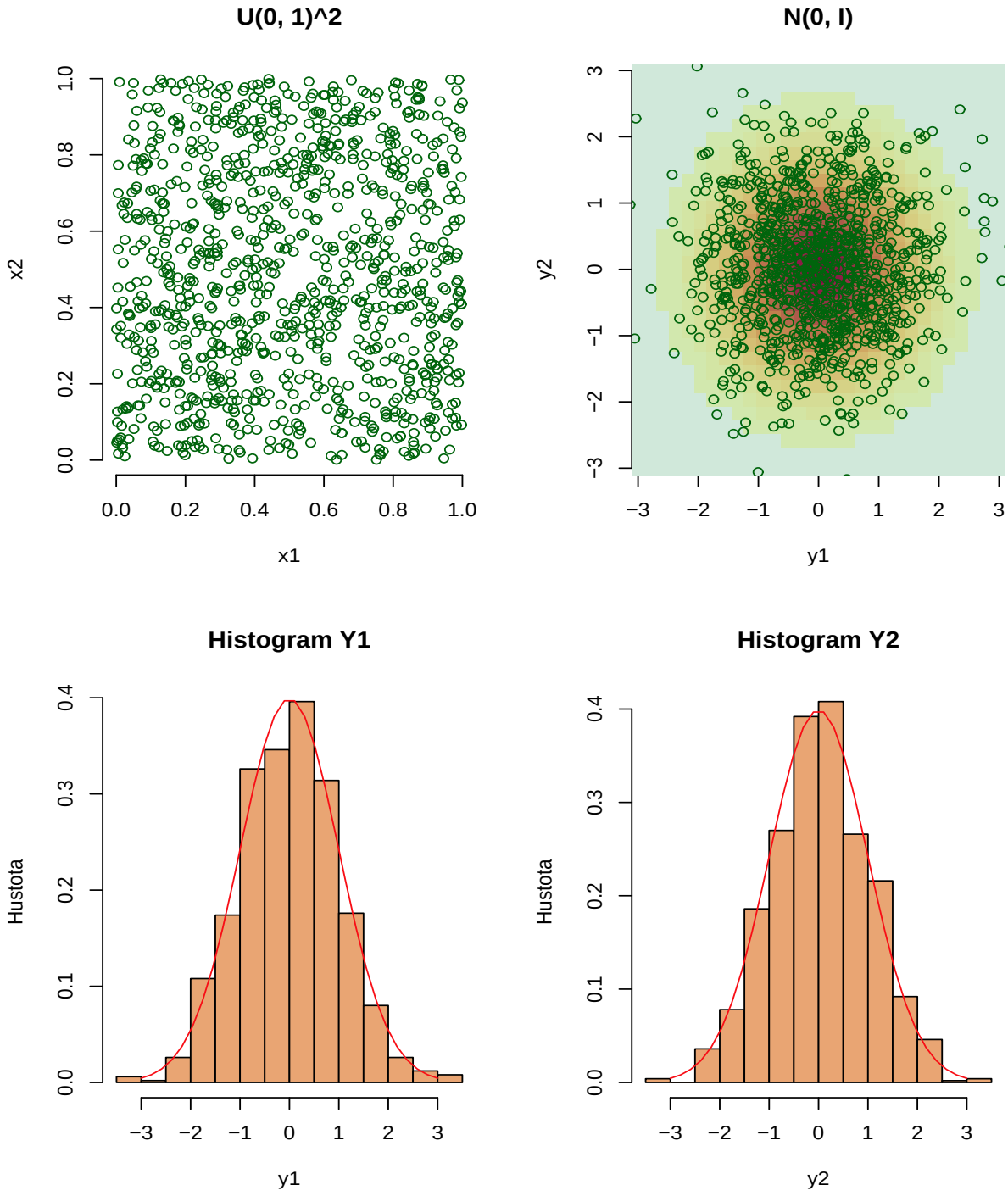
4.1 Transformace rovnoměrného rozdělení na normální

✧ Na cvičeních z matematické statistiky se často počítá následující transformace: $\boldsymbol{X} = (X_1, X_2) \sim \mathcal{U}(0, 1)^2$,

$$Y_1 = (-2 \log(X_1))^{1/2} \cos(2\pi X_2),$$
$$Y_2 = (-2 \log(X_1))^{1/2} \sin(2\pi X_2),$$

přičemž se dojde k závěru $\boldsymbol{Y} = (Y_1, Y_2) \sim \mathcal{N}_2(\mathbf{0}, I_p)$. Překvapují vás proto následující obrázky?

```
X <- matrix(runif(2000, 0, 1), nrow=1000, ncol=2)
Y <- cbind(sqrt(-2*log(X[,1])) * cos(2*pi*X[,2]),
           sqrt(-2*log(X[,1])) * sin(2*pi*X[,2]))
y1 <- seq(-3, 3, length=30)
y2 <- seq(-3, 3, length=30)
YY <- cbind(rep(y1, length(y2)), rep(y2, each=length(y1)))
hustota1 <- dnorm(y1)
hustota2 <- matrix(dmvnorm(YY, mean=rep(0, 2), sigma=diag(2)),
                   nrow=length(y1), ncol=length(y2))
par(mfrow=c(2, 2), bty="n")
plot(X, col="darkgreen", xlab="x1", ylab="x2", main="U(0, 1)^2", bg=BARVY[33])
image(y1, y2, hustota2, col=BARVY, xlab="y1", ylab="y2", main="N(0, I)")
#contour(y1, y2, hustota2, col="darkblue", xlab="y1", ylab="y2", main="N(0, I)")
points(Y, col="darkgreen")
hist(Y[,1], prob=TRUE, xlab="y1", ylab="Hustota",
     col="sandybrown", main="Histogram Y1")
lines(y1, hustota1, col="red")
hist(Y[,2], prob=TRUE, xlab="y2", ylab="Hustota",
     col="sandybrown", main="Histogram Y2")
lines(y2, hustota1, col="red")
```

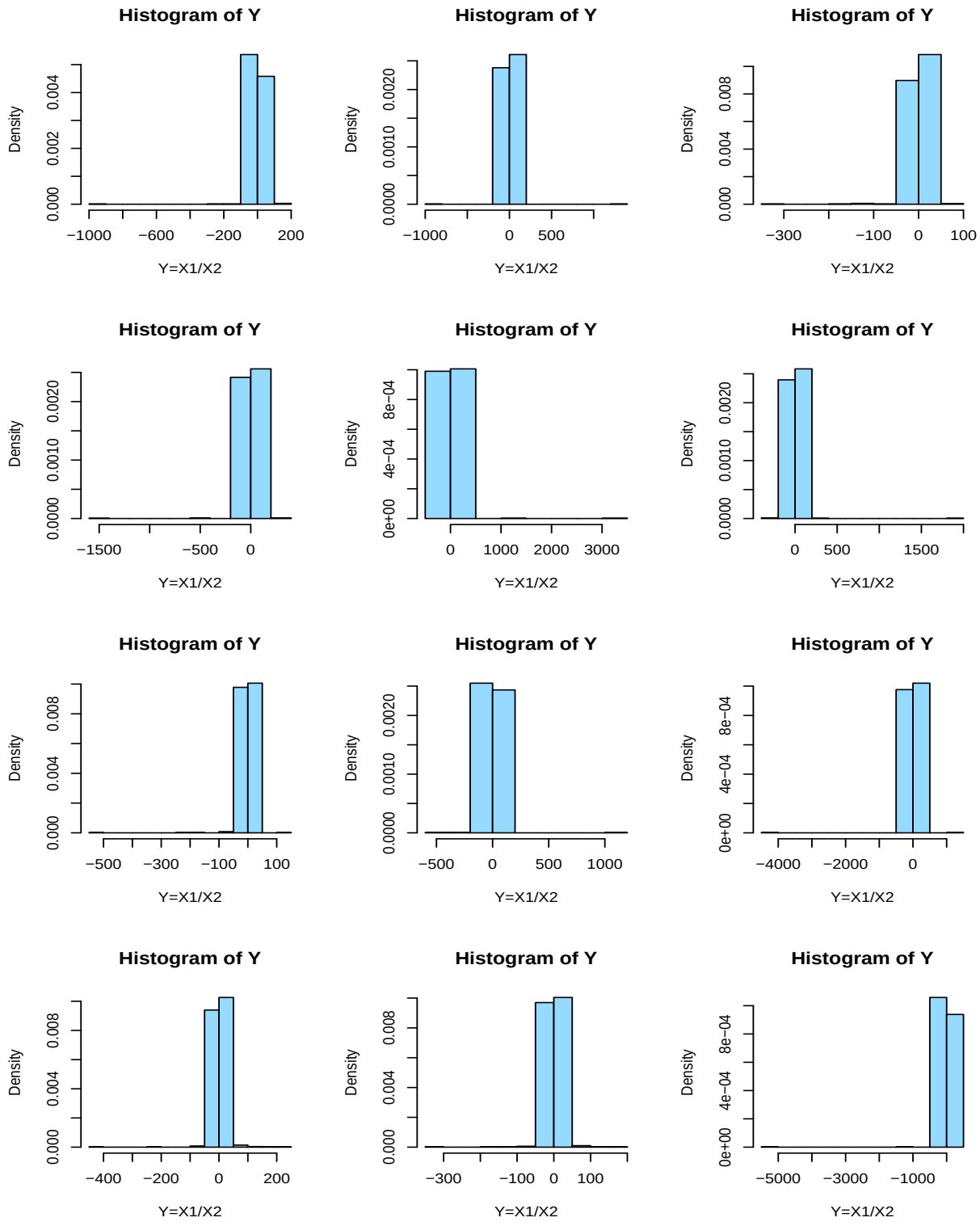


Obrázek 7: Transformace vedoucí k normalitě.

4.2 Transformace vedoucí ke Cauchyho rozdělení

✧ Další transformací často počítanou na cvičeních z matematické statistiky je následující transformace. Jsou-li $X_1 \sim \mathcal{N}(0, 1)$, $X_2 \sim \mathcal{N}(0, 1)$, nezávislé, potom $Y = X_1/X_2$ má Cauchyho rozdělení. Vygenerujte si několikrát po sobě následující obrázek. Překvapuje vás, že se výsledek poměrně dosti mění?

```
set.seed(221913273)
par(mfrow=c(4, 3), bty="n")
for (i in 1:12){
  X <- matrix(rnorm(2000, 0, 1), nrow=1000, ncol=2)
  Y <- X[,1]/X[,2]
  hist(Y, prob=TRUE, col="lightblue", xlab="Y=X1/X2")
}
```



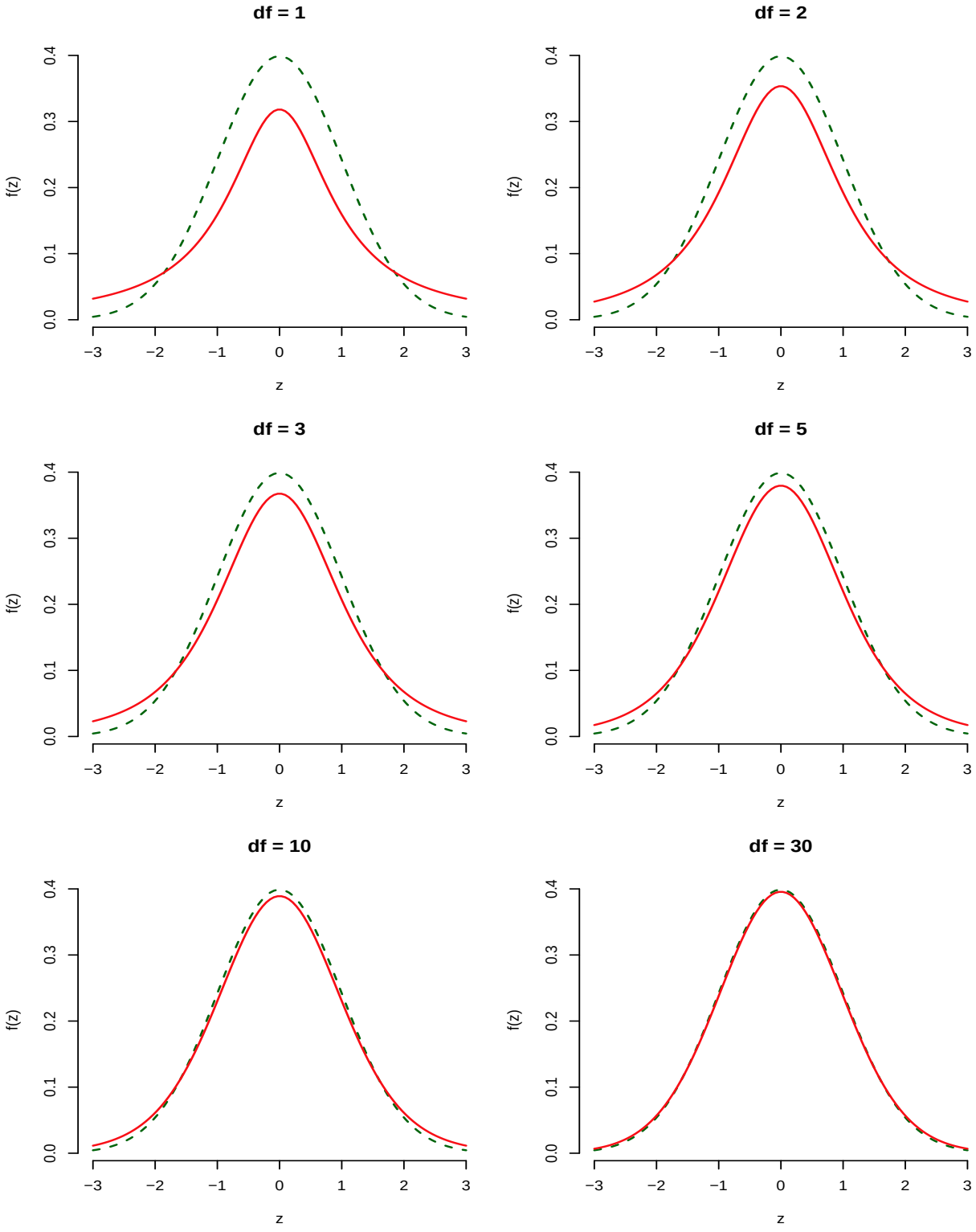
Obrázek 8: Cauchyho rozdělení.

5 Limitní věty v praxi

5.1 Studentovo rozdělení pro rostoucí stupně volnosti

✧ Jakou větu/věty z matematické statistiky ilustruje následující obrázek?

```
par(mfrow=c(3, 2), bty="n", mar=c(4, 4, 4, 1)+0.1)
grid <- seq(-3, 3, length=100)
df <- c(1, 2, 3, 5, 10, 30)
ynorm <- dnorm(grid)
for (i in 1:length(df)){
  plot(grid, ynorm, xlab="z", ylab="f(z)", type="l",
       col="darkgreen", lty=2, main=paste("df = ", df[i], sep=""), lwd=1.5)
  lines(grid, dt(grid, df=df[i]), col="red", lty=1, lwd=1.5)
}
```



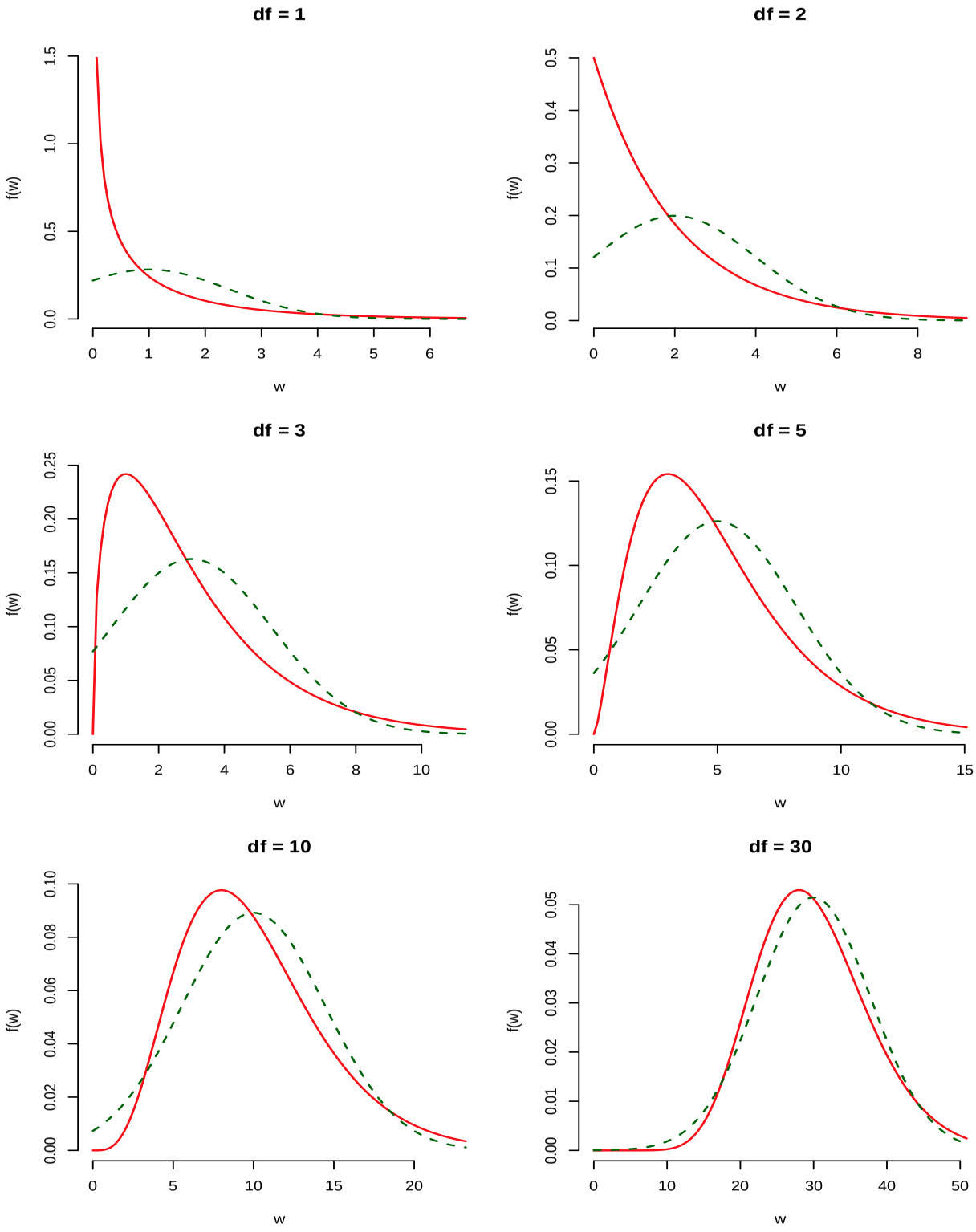
Obrázek 9: Normální (zeleně čárkovaně) a Studentovo (červeně plně) t-rozdělení.

5.2 χ^2 rozdělení pro rostoucí stupně volnosti

✧ Připomeňme, že pro $X \sim \chi^2_n$ platí $\mathrm{E}(X) = n$, $\mathrm{var}(X) = 2n$.

✧ Jakou větu/věty z matematické statistiky ilustruje následující obrázek?

```
par(mfrow=c(3, 2), bty="n", mar=c(4, 4, 4, 1)+0.1)
df <- c(1, 2, 3, 5, 10, 30)
for (i in 1:length(df)){
  ymax <- qchisq(0.99, df=df[i])
  grid <- seq(0, ymax, length=100)
  ynorm <- dnorm(grid, mean=df[i], sd=sqrt(2*df[i]))
  plot(grid, dchisq(grid, df=df[i]), xlab="w", ylab="f(w)",
       type="l", col="red", lty=1, main=paste("df = ", df[i], sep=""), lwd=1.5)
  lines(grid, ynorm, lty=2, col="darkgreen", lwd=1.5)
}
```



Obrázek 10: Normální (zeleně čárkovaně) a χ^2 (červeně plně) rozdělení.

6 Základy práce s daty

6.1 Načtení dat

Data bývají od „zákazníků“ dodávána v rozdílných formátech (v praxi dosti oblíbeným je MS Excel). Většinu z běžně používaných formátů lze s větším či menším úsilím převézt do ASCII podoby, v které jsou hodnoty na řádku odděleny mezerou, čárkou nebo středníkem. Na jednom konkrétním příkladu se tedy podíváme na načítání dat z ASCII souboru.

Soubory auta2004.dat (ASCII s hodnotami oddělenými mezerou), auta2004.csv (ASCII s hodnotami oddělenými středníky) a auta2004.xls (MS Excel) obsahují informace o výběru 428 nových automobilů prodávaných na trhu v USA v roce 2004. Každý soubor obsahuje následující proměnné (sloupce).

typ: *slovní proměnná udávající typ vozidla (např. Ford Focus LX 4dr)*;

druh: *druh vozidla: 1 = osobní, 2 = kombi, 3 = SUV, 4 = pickup, 5 = sportovní, 6 = minivan*;

nahon: *typ náhonu: 1 = přední, 2 = zadní, 3 = 4×4*;

cena.prodej: *doporučená prodejní cena v USD*;

cena.dealer: *cena v USD, za kterou prodejce odebírá vozidlo od výrobce*;

objem: *objem motoru v l*;

n.valec: *počet válců motoru. Hodnota −1 značí rotační motor*;

konska.sila: *koňská síla motoru*;

spotreba.mesto: *průměrná spotřeba v městském provozu v l/100 km*;

spotreba.dalnice: *průměrná spotřeba na dálnici v l/100 km*;

hmotnost: *hmotnost vozidla v kg*;

obvod.kola: *obvod kola v cm*;

delka: *délka vozidla v cm*;

sirka: *šířka vozidla v cm.*

- ✧ Chybějící hodnoty jsou označeny pomocí znakového řetězce NA.
- ✧ Zdrojem původních dat je Kiplinger's Personal Finance, December 2003, vol. 57, no. 12, pp. 104 – 123,http://www.kiplinger.com. Původní data jsou k dispozici též na stránkách časopisu *Journal of Statistical Education*, http://www.amstat.org/publications/jse/jse_data_archive.htm, soubor 04cars.dat. Transformace veličin uvedených původně v jednotkách běžných v USA do jednotek běžně používaných v kontinentální Evropě, překlad názvů veličin a označení hodnot kategoriálních proměnných byl proveden autory tohoto textu.

Data lze do R načíst následovně:

```
auta2004 <- read.table("auta2004.dat", header=TRUE, as.is=1)
```

- ✧ Argument header nastavený na TRUE poukazuje na fakt, že v souboru auta2004.dat jsou na prvním řádku uvedeny názvy jednotlivých proměnných.
- ✧ Při načítání dat konvertuje R automaticky všechny znakové proměnné na faktory (class factor). Zabránit tomu lze pomocí argumentu as.is, v kterém lze zadat čísla sloupců, které mají zůstat znakovými (class character). Zde jsme toho využili u proměnné typ (název auta), která není skutečnou proměnnou. Jedná se o identifikátor jednotlivých jednotek v datech.

Ze souboru, v kterém jsou hodnoty odděleny středníky načteme data pomocí

```
auta2004 <- read.csv("auta2004.csv", sep=";", header=TRUE, as.is=1)
auta2004 <- read.csv2("auta2004.csv", dec=".", header=TRUE, as.is=1)
```

Další možnosti načítání dat ze souborů zjistíte po prohlédnutí helpu pro související funkce:

```
?read.table
help(read.csv)
help(read.csv2, htmlhelp=TRUE)
```

6.2 data.frame

```
class(auta2004)
```

Třídou dat je data.frame, což je skoro matice. Na rozdíl od objektů třídy matrix má však každý sloupec svoji vlastní třídu, která může být rozdílná pro jednotlivé sloupce:

```
class(auta2004[,1])
class(auta2004[,2])
class(auta2004[,4])
class(auta2004[,6])
```

část datové tabulky (nebo celou datovou tabulku) si můžeme vypsat podobně jako u matice:

```
auta2004[1:5,]
```

6.3 Přístup k jednotlivým sloupcům datové tabulky

Povšimněte si použití operátorů [[]] a $.

```
Cena.Prodej <- auta2004[,4]
Cena.Prodej[1:10]
Cena.Prodej <- auta2004[[4]]
Cena.Prodej[1:10]
Cena.Prodej <- auta2004$cena.prodej
Cena.Prodej[1:10]
```

6.4 Kvalitativní proměnné

Zvláštní pozornost zasluhují proměnné, které jsou v datech sice uloženy jako numerické, ale ve skutečnosti se jedná o proměnné kategoriální (kvalitativní). Pro tyto je vhodné vytvořit nové proměnné, o kterých bude R vědět, že se jedná o proměnné kvalitativní (`factor`). Ve zpracovávaných datech se jedná zejména o proměnné druh a nahon. Vytvoříme nové proměnné fdruh a fnahon:

```
auta2004$fdruh <- factor(auta2004$druh, levels=1:6,
              labels=c("osobni", "combi", "SUV", "pickup", "sport", "minivan"))
auta2004$fnahon <- factor(auta2004$nahon, levels=1:3,
                          labels=c("predni", "zadni", "4x4"))
```

Uvedení popisek jednotlivých hodnot pomocí argumentů `levels` a `labels` je nepovinné.

6.5 Uložení a znovunačtení dat v R formátu

Data lze po provedení transformací či jiných datových operací uložit na disk v Rkovém formátu, z něhož lze upravená data v budoucnu snadno načíst (bez nutnosti spouštět skript provádějící úpravy dat).

Data uložíme následovně:

```
save(auta2004, file="auta2004.RData")
```

Následné načtení provedeme pomocí:

```
load("auta2004.RData")
```

6.6 Výběr podmnožiny dat

často potřebujeme zpracovávat pouze podmnožinu dat. V následujících příkladech si vybereme pouze auta druhu „combi“:

```
a04.Combi <- auta2004[auta2004$fdruh=="combi",]
a04.Combi[1:5,]
a04.Combi <- subset(auta2004, fdruh=="combi")
a04.Combi[1:5,]
```

Obdobně lze vybrat též jenom některé sloupce:

```
a04.sl124 <- auta2004[, c(1, 2, 4)]
a04.sl124[1:5,]
a04.sl124 <- auta2004[, c("typ", "fdruh", "cena.prodej")]
a04.sl124[1:5,]
a04.sl124 <- subset(auta2004, select=c("typ", "fdruh", "cena.prodej"))
a04.sl124[1:5,]
```

6.7 Základní popisné statistiky a prohlídka dat

Základní popisné statistiky získáme příkazem summary. Povšimněte si, že typ spočtených popisných statistik závisí na třídě jednotlivých sloupců (tabulky četností pro kvalitativní veličiny, průměr a vybrané kvantily pro kvantitativní veličiny):

```
summary(auta2004)
```

Povšimněte si rozdílného výstupu u proměnných druh, fdruh, respektive nahon, fnahon. Jsou rozumně interpretovatelné výsledky uvedené u proměnných druh a nahon?

Absolutní a relativní četnosti pro kvalitativní veličinu získáme například takto:

```
table(auta2004$fdruh)
prop.table(table(auta2004$fdruh))
```

Jednotlivé popisné statistiky pro kvantitativní veličinu dostaneme takto (argument na.rm je potřeba nastavovat na TRUE pouze tehdy, když se v datech vyskytují nějaké chybějící hodnoty):

```
mean(auta2004$spotreba.mesto, na.rm=TRUE)
median(auta2004$spotreba.mesto, na.rm=TRUE)
quantile(auta2004$spotreba.mesto, probs=c(0, 0.25, 0.5, 0.75, 1), na.rm=TRUE)
sd(auta2004$spotreba.mesto, na.rm=TRUE)
var(auta2004$spotreba.mesto, na.rm=TRUE)
```

Konkrétní popisnou statistiku pro všechny kvantitativní proměnné z datové tabulky lze dostat např. následujícím způsobem:

```
a04.kvantita <- subset(auta2004, select=c("cena.prodej", "cena.dealer", "objem",
                            "konska.sila", "spotreba.mesto", "spotreba.dalnice",
                            "hmotnost", "obvod.kola", "delka", "sirka"))
sapply(a04.kvantita, sd, na.rm=TRUE)
lapply(a04.kvantita, sd, na.rm=TRUE)
```

Často nás též zajímají podmíněné popisné statistiky, např. průměrná spotřeba pro jednotlivé druhy aut. K výsledku se lze dopracovat pomocí funkcí tapply nebo by bez nutnosti vytvářet ručně podmnožiny dat:

```
tapply(auta2004$spotreba.mesto, auta2004$fdruh, mean, na.rm=TRUE)
by(auta2004$spotreba.mesto, auta2004$fdruh, mean, na.rm=TRUE)
```

6.8 Základní obrázky

Zamýšíme-li provádět s daty statistickou analýzu, měli bychom si nejprve data graficky prohlédnout. Obrázky lépe než čísla odhalí případné chyby v datech a upozorní nás na nástrahy zamýšlených analýz.

6.8.1 Obrázky pro kvalitativní proměnnou (factor)

Několik obrázků, které se mohou hodit při práci s kvalitativní proměnnou (víte proč?):

```
par(mfrow=c(2, 2), bty="n")
plot(auta2004$fdruh, ylab="Cetnost")
barplot(table(auta2004$fdruh), ylab="Cetnost")
barplot(prop.table(table(auta2004$fdruh)), ylab="Proporce")
pie(table(auta2004$fdruh))
```

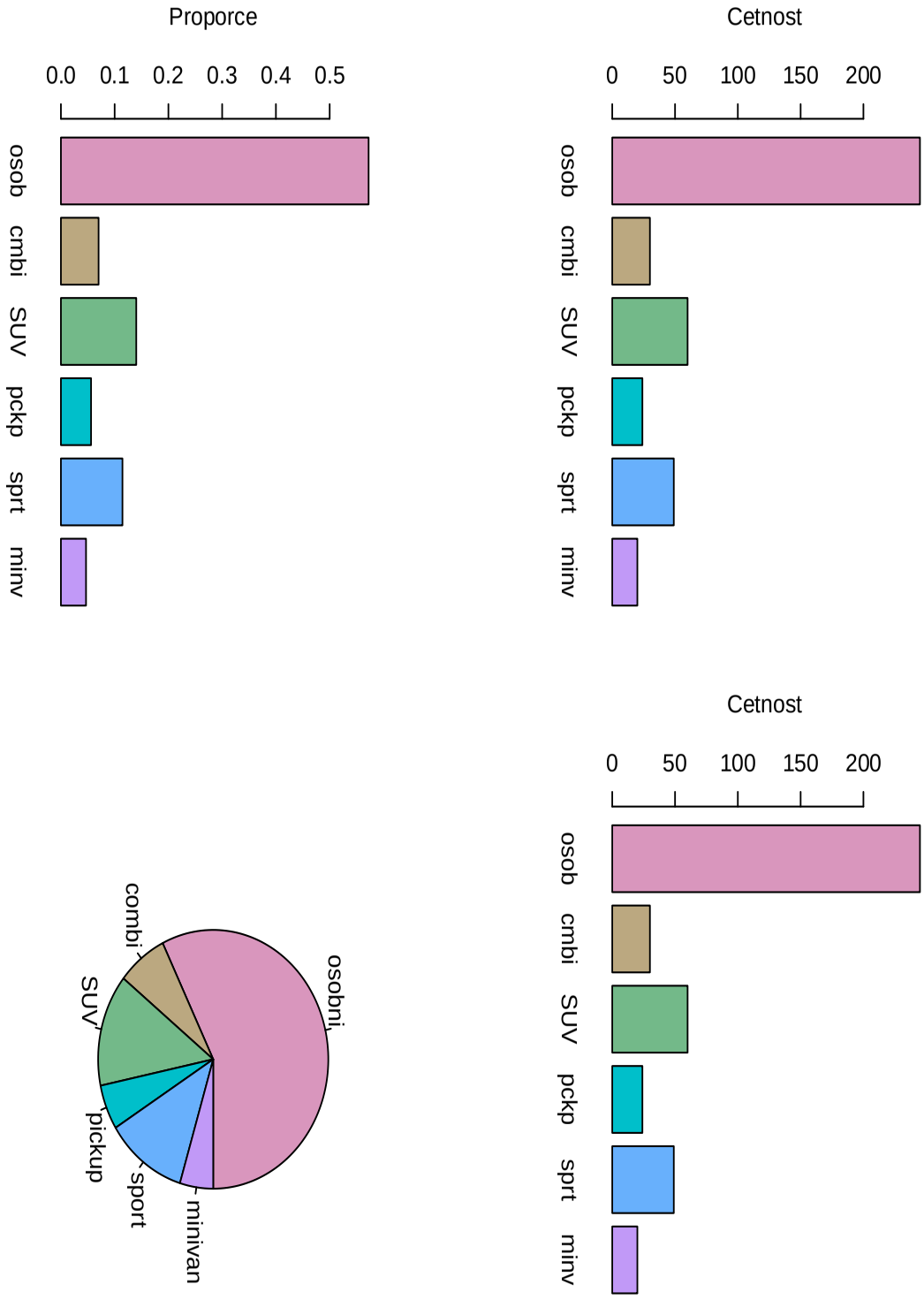
Ještě jednou, s trochu vylepšeným obarvením a zkrácenými popisky, aby se vešly do obrázku(viz obr. 11):

```
LABSHORT <- c("osob", "cmbi", "SUV", "pckp", "sprt", "minv")
Barvicky <- rainbow_hcl(6)
par(mfrow=c(2, 2), bty="n")
plot(auta2004$fdruh, ylab="Cetnost", col=Barvicky, names.arg=LABSHORT)
barplot(table(auta2004$fdruh), ylab="Cetnost", col=Barvicky,
        names.arg=LABSHORT)
barplot(prop.table(table(auta2004$fdruh)), ylab="Proporce", col=Barvicky,
        names.arg=LABSHORT)
pie(table(auta2004$fdruh), col=Barvicky)
```

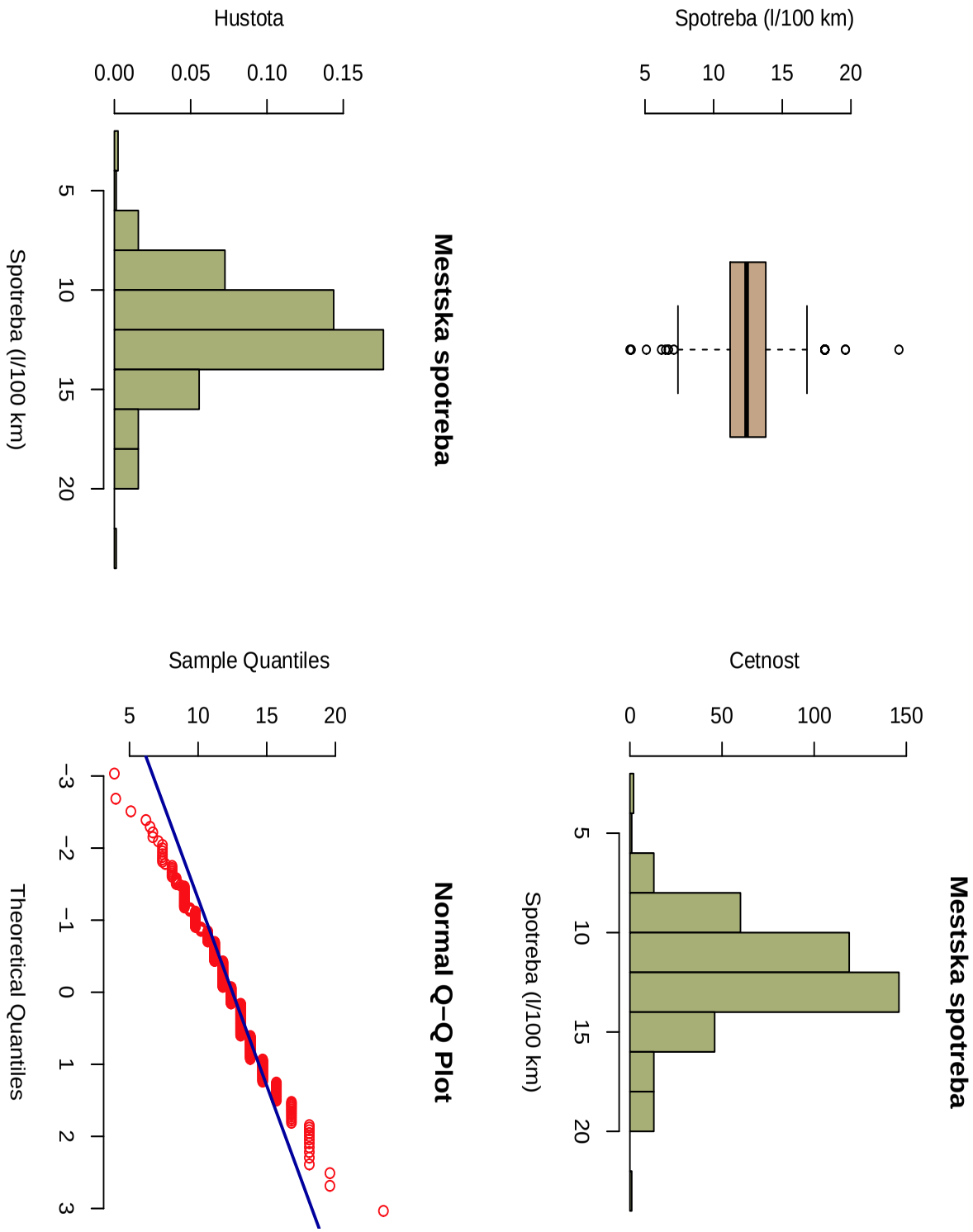
6.8.2 Obrázky pro kvantitativní proměnnou (numeric)

Několik obrázků, které se mohou hodit při práci s kvantitativní proměnnou (víte proč?) (viz obr. 12):

```
par(mfrow=c(2, 2), bty="n")
boxplot(auta2004$spotreba.mesto, ylab="Spotreba (l/100 km)",
        col=rainbow_hcl(1, start=50))
hist(auta2004$spotreba.mesto, xlab="Spotreba (l/100 km)", ylab="Cetnost",
     main="Mestska spotreba", col=rainbow_hcl(1, start=80))
hist(auta2004$spotreba.mesto, prob=TRUE, xlab="Spotreba (l/100 km)",
     ylab="Hustota", main="Mestska spotreba", col=rainbow_hcl(1, start=80))
qqnorm(auta2004$spotreba.mesto, col="red")
qqline(auta2004$spotreba.mesto, col="darkblue", lwd=2)
```


Proporce
0.0 0.1 0.2 0.3 0.4 0.5
osob
cmbi
SUV
pckp
sprt
minv
Cetnost
0 50 100 150 200
osob
cmbi
SUV
pckp
sprt
minv
osobni
combi
SUV
pickup
sport
minivan
Cetnost
0 50 100 150 200
osob
cmbi
SUV
pckp
sprt
minv

Obrázek 11:


Hustota
0.00
0.05
0.10
0.15
Spotreba (l/100 km)
5
10
15
20
Mestska spotreba
Spotreba (l/100 km)
Spotreba (l/100 km)
5
10
15
20
Cetnost
0
50
100
150
Mestska spotreba
Spotreba (l/100 km)
Sample Quantiles
5
10
15
20
Normal Q–Q Plot
Theoretical Quantiles
−3
−2
−1
0
1
2
3

Obrázek 12:

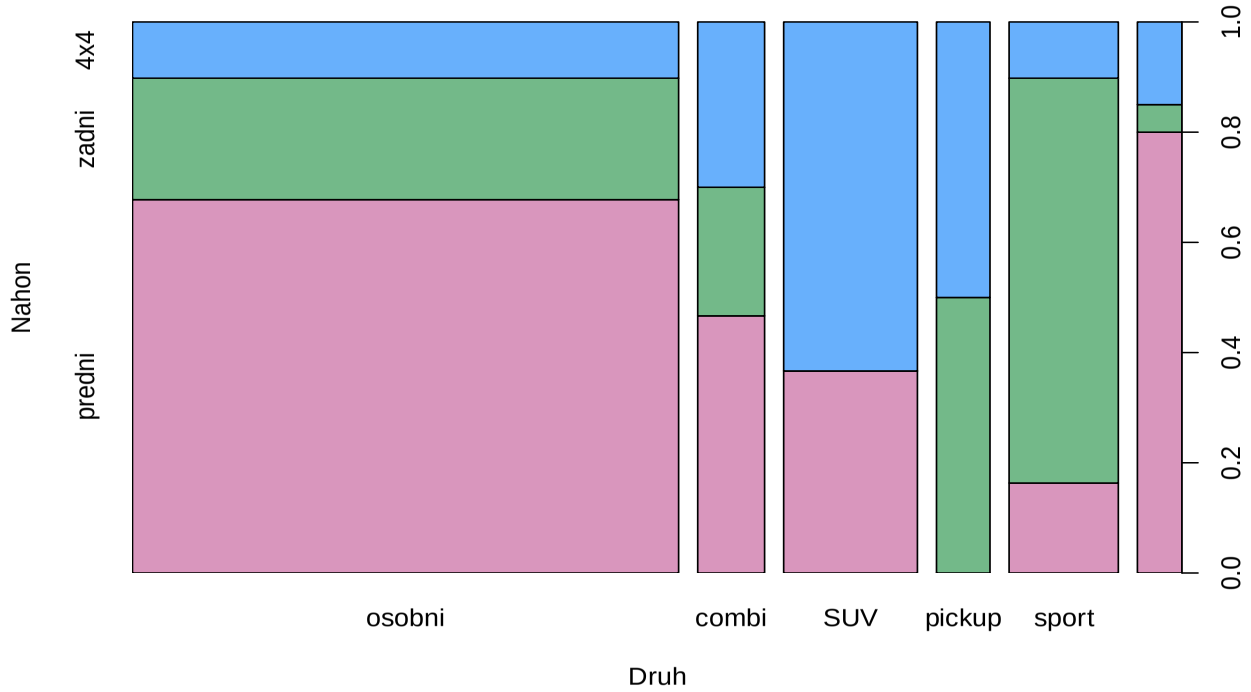
6.9 Obrázky při zkoumání vztahu mezi dvěma kvalitativními proměnnými

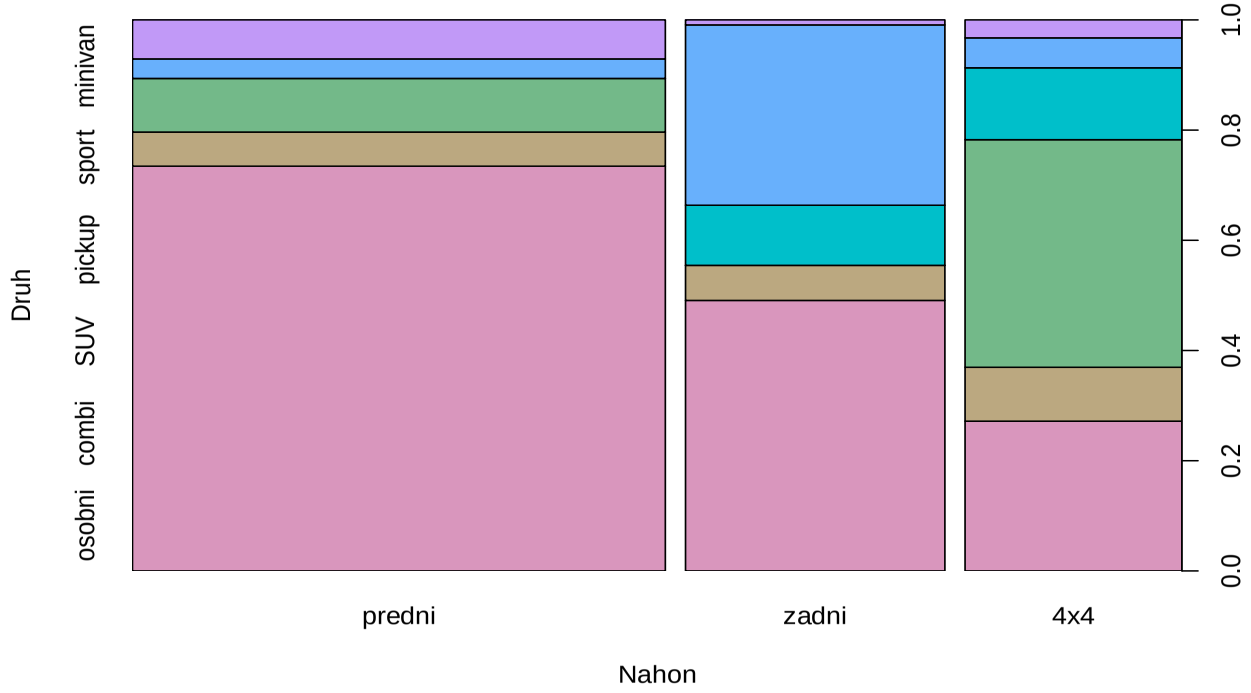
Při zkoumání vztahu mezi dvěma kvalitativními proměnnými se mohou hodit následující obrázky (co z nich vyčtete?) (viz obr. 13):

```
#Barvicky1 <- c("darkblue", "blue", "lightblue")
#Barvicky2 <- c("yellow", "orange", "brown", "darkblue", "red", "darkgreen")
Barvicky1 <- rainbow_hcl(3)
Barvicky2 <- rainbow_hcl(6)
par(mfcol=c(2, 1), bty="n")
plot(auta2004$fdruh, auta2004$fnahon, xlab="Druh", ylab="Nahon", col=Barvicky1)
plot(auta2004$fnahon, auta2004$fdruh, xlab="Nahon", ylab="Druh", col=Barvicky2)
```

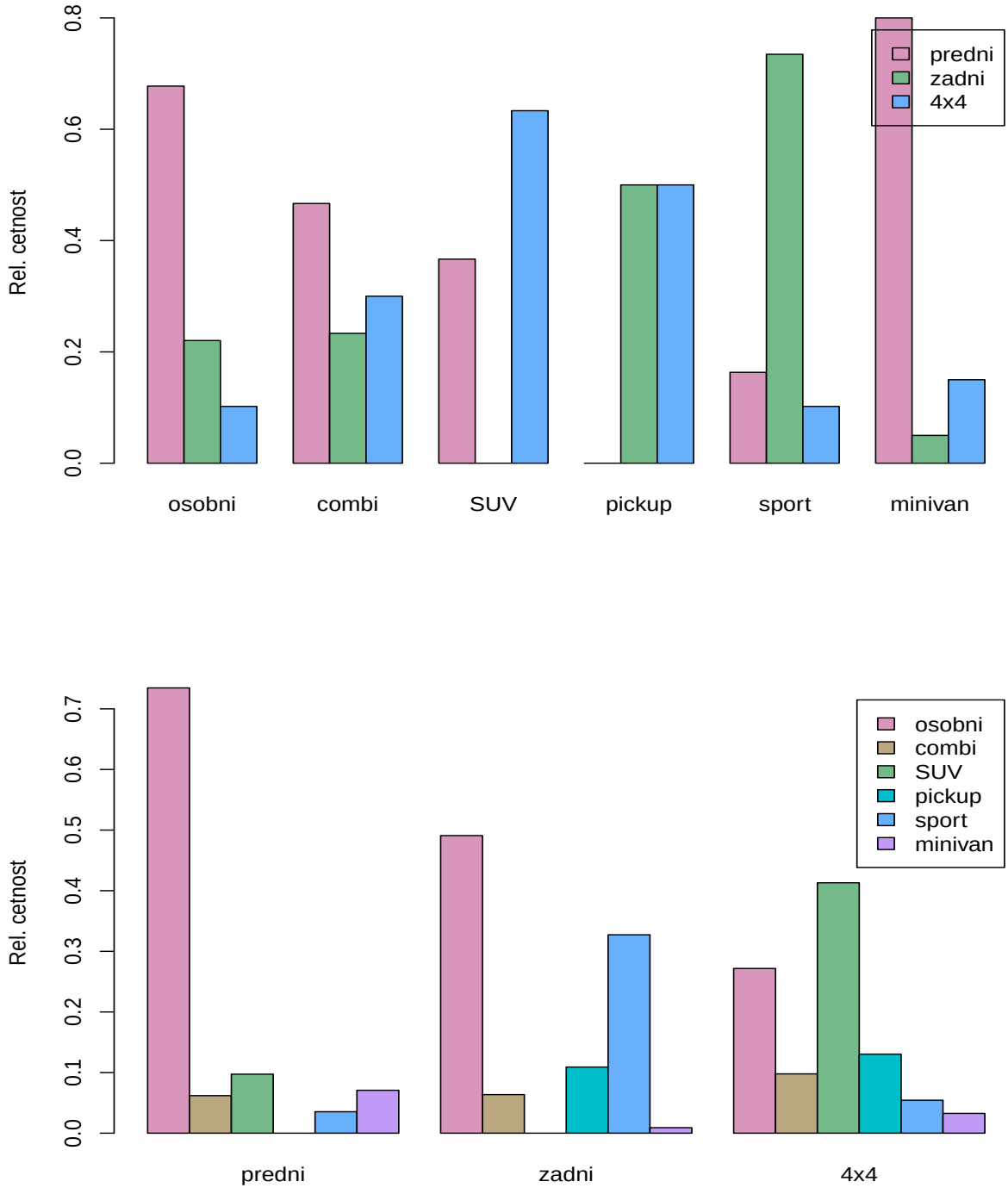
Jak jste jistě zjistili, funkce `plot` v tomto případě znázorňuje kumulativní relativní četnosti jedné proměnné při podmínění druhou proměnnou. Nicméně, kumulativní relativní četnosti nedávají příliš velký smysl, není-li proměnná, pro kterou jsou počítány, ordinální (kategorie lze smysluplně uspořádat). Bude tedy vhodnější znázornit pouze podmíněné relativní četnosti (nepormíněné). Toho lze dosáhnout například takto (viz obr. 14):

```
print(Tab <- table(auta2004$fdruh, auta2004$fnahon))
print(PropTab1 <- prop.table(Tab, margin=1))
print(PropTab2 <- prop.table(Tab, margin=2))
par(mfcol=c(2, 1), bty="n")
#barplot(t(PropTab1), legend.text=colnames(PropTab1),
#        ylab="Kumul. rel. cetnost", col=Barvicky1)
#barplot(PropTab2, legend.text=row.names(PropTab2),
#        ylab="Kumul. rel. cetnost", col=Barvicky2)
barplot(t(PropTab1), legend.text=colnames(PropTab1),
        ylab="Rel. cetnost", col=Barvicky1, beside=TRUE)
barplot(PropTab2, legend.text=row.names(PropTab2),
        ylab="Rel. cetnost", col=Barvicky2, beside=TRUE)
```


Nahon
4x4
zadni
predni
0.0
0.2
0.4
0.6
0.8
1.0
osobni
combi
SUV
pickup
sport
Druh


Druh
osobni
combi
SUV
pickup
sport
minivan
0.0
0.2
0.4
0.6
0.8
1.0
predni
zadni
4x4
Nahon

Obrázek 13:

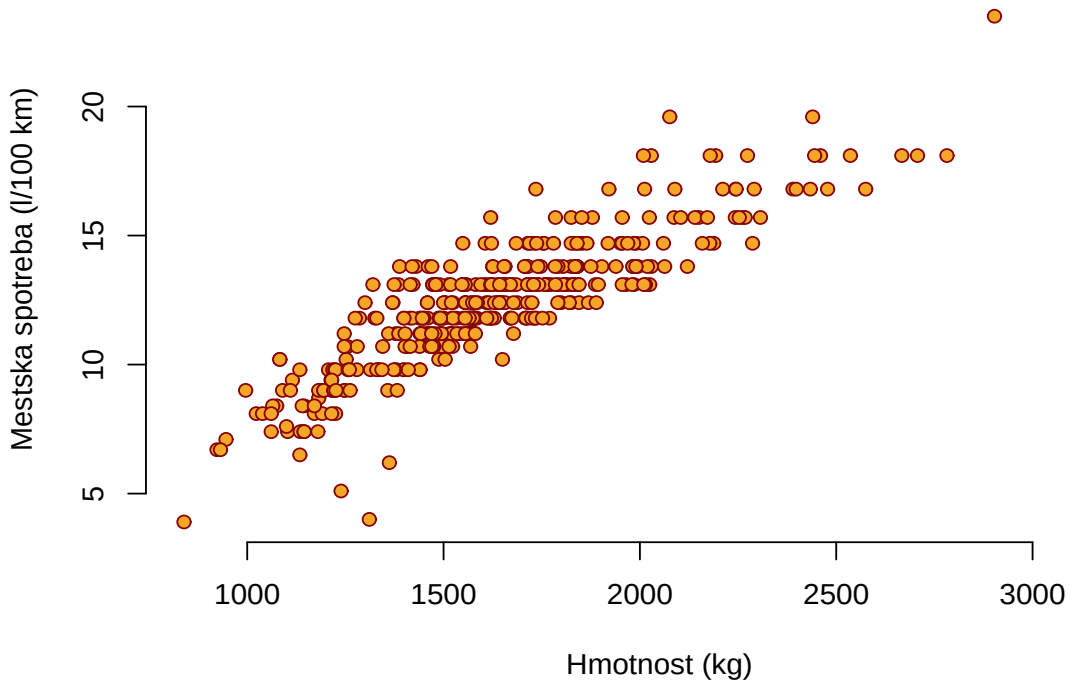

Rel. cetnost
0.0
0.2
0.4
0.6
0.8
predni
zadni
4x4
osobni
combi
SUV
pickup
sport
minivan
Rel. cetnost
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
osobni
combi
SUV
pickup
sport
minivan
predni
zadni
4x4

Obrázek 14:

6.10 Obrázky při zkoumání vztahu mezi dvěma kvantitativními proměnnými

Při zkoumání vztahu mezi dvěma kvantitativními proměnnými se mohou hodit následující obrázek (co z něj vyčtete?) (viz obr. 15):

```
par(mfrow=c(1, 1), bty="n")
plot(auta2004$hmotnost, auta2004$spotreba.mesto,
     pch=21, col="red4", bg="orange",
     xlab="Hmotnost (kg)", ylab="Mestska spotreba (l/100 km)")
```

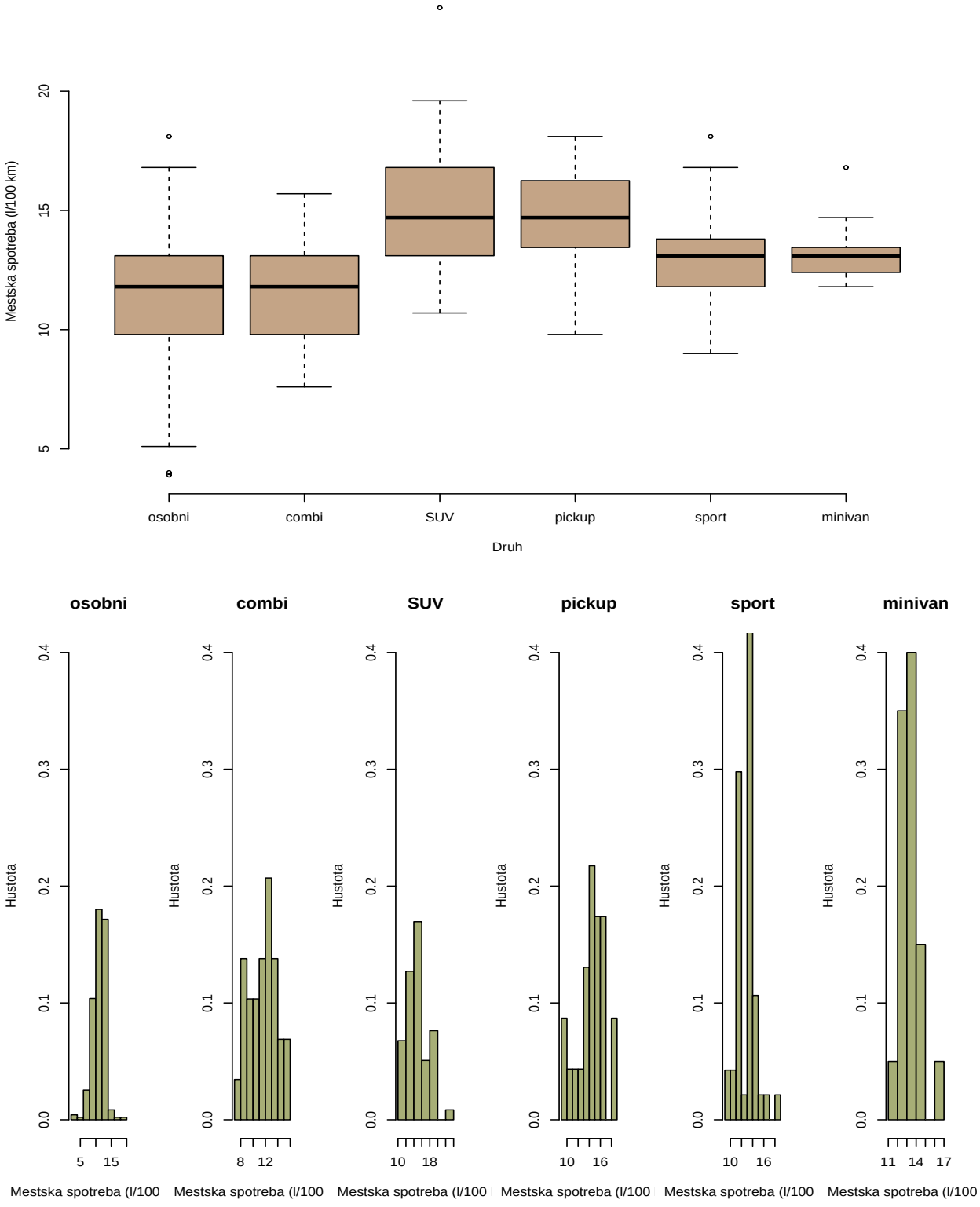


Obrázek 15:

6.11 Obrázky při zkoumání vztahu mezi kvalitativní a kvantitativní proměnnou

Při zkoumání vztahu mezi kvalitativní a kvantitativní proměnnou se mohou hodit následující obrázky (opět, víte co z těchto obrázků vyčtete?) (viz obr. 16):

```
par(bty="n")
layout(matrix(c(1,1,1,1,1,1, 2,3,4,5,6,7), nrow=2, byrow=TRUE))
plot(auta2004$fdruh, auta2004$spotreba.mesto, col=rainbow_hcl(1, start=50),
     xlab="Druh", ylab="Mestska spotreba (l/100 km)")
for (dd in levels(auta2004$fdruh)){
  hist(auta2004$spotreba.mesto[auta2004$fdruh==dd], prob=TRUE, ylim=c(0, 0.4),
       xlab="Mestska spotreba (l/100 km)", ylab="Hustota", main=dd,
       col=rainbow_hcl(1, start=80))
}
```


Mestska spotreba (l/100 km)
5
10
15
20
osobni
combi
SUV
pickup
sport
minivan
Druh
osobni
combi
SUV
pickup
sport
minivan
Hustota
0.0
0.1
0.2
0.3
0.4
5
15
8
12
10
18
10
16
10
16
11
14
17
Mestska spotreba (l/100

Obrázek 16:

7 Základní metody matematické statistiky

7.1 Jednovýběrové problémy s kvantitativními daty

Ukážeme si několik postupů, vesměs založených na předpokladu normality (**měli byste se vždy nejprve zamyslet/ověřit, zda je tento předpoklad reálný, respektive zda jeho nesplnění lze ignorovat díky dostatečnému rozsahu dat a platnosti centrální limitní věty**). V R jsou všechny zde uvedené metody implementovány jako testy. Ve výstupu však lze nalézt též související interval spolehlivosti.

7.1.1 Jednovýběrový t-test

X značí náhodnou veličinu, jež reprezentuje městskou spotřebu náhodně vybraného auta.

- ✧ $H_0 : E(X) = 12{,}5$ (l/100 km) proti $H_1 : E(X) \neq 12{,}5$ (l/100 km).
  - ⋆ čím se liší ty dva výstupy?
  - ⋆ Najdete interval spolehlivosti pro $E(X)$?
  - ⋆ Uměli byste si „ručně“ spočítat P-hodnotu?

```
t.test(auta2004$spotreba.mesto, mu=12.5)
t.test(auta2004$spotreba.mesto, mu=12.5, conf.level=0.99)
```

- ✧ $H_0 : E(X) \leq 12{,}5$ (l/100 km) proti $H_1 : E(X) > 12{,}5$ (l/100 km).

```
t.test(auta2004$spotreba.mesto, mu=12.5, alt="greater")
```

- ✧ $H_0 : E(X) \geq 12{,}5$ (l/100 km) proti $H_1 : E(X) < 12{,}5$ (l/100 km).
  - ⋆ Jak interpretujete výsledek testu na 5% hladině významnosti?
  - ⋆ Jak interpretujete výsledek testu na 1% hladině významnosti?
  - ⋆ Uměli byste si „ručně“ spočítat P-hodnotu?

```
t.test(auta2004$spotreba.mesto, mu=12.5, alt="less")
```

7.2 Dvouvýběrové problémy s kvantitativními daty

7.2.1 Dvouvýběrový t-test

X značí náhodnou veličinu, jež reprezentuje městskou spotřebu náhodně vybraného auta druhu combi a Y náhodnou veličinu, jež reprezentuje městskou spotřebu náhodně vybraného auta druhu pickup.

✧ $H_0 : E(X) = E(Y)$ proti $H_1 : E(X) \neq E(Y)$.

⋆ První t-test je ten „učebnicový“, který předpokládá shodné rozptyly.

⋆ Druhý t-test nepožaduje shodu rozptylů.

⋆ V jakém intervalu se s 95% spolehlivostí pohybuje rozdíl mezi průměrnou spotřebou combi a pickup aut?

⋆ Dovedli byste tento interval upravit tak, aby dával 99% spolehlivost?

```
ms.combi <- auta2004$spotreba.mesto[auta2004$fdruh == "combi"]
ms.pickup <- auta2004$spotreba.mesto[auta2004$fdruh == "pickup"]
t.test(ms.combi, ms.pickup, var.equal=TRUE)
t.test(ms.combi, ms.pickup)
```

✧ $H_0 : E(Y) - E(X) \leq 3$ (l/100 km) proti $H_1 : E(Y) - E(X) > 3$ (l/100 km).

⋆ Oba níže uvedené příkazy testují totéž, proč?

⋆ Jak se od sebe liší interpretace spočtených intervalů spolehlivosti?

⋆ Jak interpretujete na 5% hladině významnosti výsledek testu?

⋆ Jakou minimální hodnotu rozdílu jste na 5% hladině významnosti schopni statisticky prokázat?

```
t.test(ms.pickup, ms.combi, mu=3, alt="greater")
t.test(ms.combi, ms.pickup, mu=-3, alt="less")
```

7.2.2 Dvouvýběrový F-test

X značí náhodnou veličinu, jež reprezentuje městskou spotřebu náhodně vybraného auta druhu combi a Y náhodnou veličinu, jež reprezentuje městskou spotřebu náhodně vybraného auta druhu pickup.

- $\mathrm{H}_0 : \mathrm{var}(X) = \mathrm{var}(Y)$ proti $\mathrm{H}_1 : \mathrm{var}(X) \neq \mathrm{var}(Y)$.

```
var.test(ms.combi, ms.pickup)
```

- $\mathrm{H}_0 : \mathrm{var}(X)/\mathrm{var}(Y) \leq 0{,}5$ proti $\mathrm{H}_1 : \mathrm{var}(X)/\mathrm{var}(Y) > 0{,}5$.

```
var.test(ms.combi, ms.pickup, ratio=0.5, alt="greater")
```

7.3 Párové problémy s kvantitativními daty

7.3.1 Párový t-test

X značí náhodnou veličinu, jež reprezentuje městskou spotřebu náhodně vybraného auta a Y náhodnou veličinu, jež reprezentuje dálniční spotřebu stejného auta.

✧ $H_0 : E(X) = E(Y)$ proti $H_1 : E(X) \neq E(Y)$.

⋆ Párový t-test lze též provést „jednovýběrovým“ způsobem. Víte proč?

```
t.test(auta2004$spotreba.mesto, auta2004$spotreba.dalnice, paired=TRUE)
t.test(auta2004$spotreba.mesto - auta2004$spotreba.dalnice)
```

✧ $H_0 : E(X) - E(Y) \geq 3{,}2$ (l/100 km) proti $H_1 : E(X) - E(Y) < 3{,}2$ (l/100 km).

⋆ Jak interpretujete na 5% hladině významnosti výsledek testu?

```
t.test(auta2004$spotreba.mesto, auta2004$spotreba.dalnice, mu=3.2,
       paired=TRUE, alt="less")
t.test(auta2004$spotreba.dalnice, auta2004$spotreba.mesto, mu=-3.2,
       paired=TRUE, alt="greater")
```

8 Souhrnný přehled nejdůležitějších příkazů

8.1 Základní elementy

Příkazy

`ls()` nebo `objects()`	vypiš seznam objektů definovaných a dostupných na pracovní ploše
`rm(object)`	vymaž `object` z pracovní plochy
`search()`	vypiš co všechno je prohledáváno a v jakém pořadí, když se hledá nějaký objekt

Jména proměnných

Kombinace písmen, číslic a teček. Nesmí začínat číslicí. Nedoporučuje se začínat jméno proměnné tečkou. Rozlišují se velká a malá písmena, tj. objekt pojmenovaný krabicka je něco jiného než objekt pojmenovaný Krabicka.

Přiřazovací příkazy

$<-$ nebo $=$	přiřad' hodnotu proměnné
$->$	přiřazení „doprava“
$<<-$	globální přiřazení (ve funkcích)

8.2 Operátory

Aritmetické operátory

$+$	sčítání
$-$	odčítání
$*$	násobení
$/$	dělení
$\wedge$	umocňování
$\%/\%$	celočíselné dělení (div)
$\%\%$	zbytek po celočíselném dělení (mod)

Logické operátory a operátory vztahu

Výsledkem těchto operátorů je vždy logická hodnota TRUE anebo FALSE.

==	je rovno?
!=	není rovno?
<	je menší než?
>	je větší než?
<=	je menší než nebo rovno?
>=	je větší než nebo rovno?
is.na(x)	je x chybějící hodnota?
&	logické A SOUčASNě (AND)
|	logické NEBO (OR)
!	logická negace (NOT)

8.3 Vektory a datové typy

Generování vektorů s nějakou strukturou

numeric(25)	vektor o 25 nulách
character(25)	vektor s 25 prázdnými znaky, tj. ""
logical(25)	logický vektor s 25 elementy rovnými FALSE
seq(-4, 4, 0.1)	vektor aritmetické posloupnosti -4, $-3{,}9$, $-3{,}8, \ldots, 3{,}9, 4$
1:10	vektor aritmetické posloupnosti 1, 2, ..., 10, to samé jako příkaz seq(1, 10, 1)
c(5, 7, 9, 13, 1:5)	vytvoř vektor spojením složek uvnitř c (*concatenation*), zde vektor 5, 7, 9, 13, 1, 2, 3, 4, 5
rep(1, 10)	vektor, kde se 1 opakuje 10×
gl(3, 2, 12)	faktorový vektor o 3 úrovních, opakuj každou úroveň v blocích o velikosti 2, a to až do celkové délky vektoru 12, zde tedy 1, 1, 2, 2, 3, 3, 1, 1, 2, 2, 3, 3

Přetypování vektorů

as.numeric(x)	přetypuj x na numerický vektor
as.character(x)	přetypuj x na znakový vektor
as.logical(x)	přetypuj x na logický vektor (obsahující pouze TRUE a FALSE)
factor(x)	vytvoř faktor (kategoriální veličinu) z x

8.4 Datové soubory (data frames)

`data.frame(height=c(165, 185), weight=c(90, 65))`	vytvoř data frame se dvěma pojmenovanými veličinami
`data.frame(height, weight)`	ulož dříve vytvořené vektory jako dva sloupce v data framu
`dfr$var`	vyber proměnnou (sloupec) var z data framu dfr
`attach(dfr)`	polož data frame dfr do vyhledávací cesty, k jednotlivým proměnným lze potom přistupovat i bez $
`detach(dfr)`	odstraň data frame z vyhledávací cesty

8.5 Numerické funkce

Matematické

`log(x)`	přirozený logaritmus x
`log10(x)`	dekadický logaritmus x
`exp(x)`	exponenciální funkce e^x
`sin(x)`	sinus x
`cos(x)`	kosinus x
`tan(x)`	tangens x
`asin(x)`	arcus-sinus x
`acos(x)`	arcus-kosinus x
`atan(x)`	arcus-tangens x
`min(x)`	minimum z vektoru x
`min(x1, x2, ...)`	minimum z několika vektorů, výsledkem je jedno číslo
`max(x)`	maximum z vektoru x
`max(x1, x2, ...)`	maximum z několika vektorů, výsledkem je jedno číslo
`range(x)`	to samé jako `c(min(x), max(x))`
`pmin(x1, x2, ...)`	paralelní (po složkách) minimum z několika stejně dlouhých vektorů
`pmax(x1, x2, ...)`	paralelní (po složkách) maximum z několika stejně dlouhých vektorů
`length(x)`	počet složek vektoru
`sum(complete.cases(x))`	počet nechybějících složek ve vektoru

Statistické

`mean(x)`	aritmetický průměr
`sd(x)`	směrodatná odchylka
`var(x)`	rozptyl
`median(x)`	medián
`quantile(x, p)`	kvantily
`cor(x, y)`	korelace

8.6 Indexace/vybírání

`x[1]`	první element
`x[1:5]`	podvektor obsahující prvních pět elementů
`x[c(2,3,5,7,11)]`	podvektor obsahující 2., 3., 5., 7. a 11. element
`x[y <= 30]`	výběr podvektoru pomocí logického výrazu
`x[sex=="male"]`	výběr podvektoru pomocí faktorové proměnné
`i <- c(2,3,5,7,11); x[i]`	výběr podvektoru pomocí číselné proměnné
`l <- (y<=30); x[l]`	výběr podvektoru pomocí logické proměnné

Matice a datové soubory

`m[4,]`	čtvrtý řádek
`m[,3]`	třetí sloupec
`dfr[dfr$promenna<=30]`	částečný datový soubor
`subset(dfr, subset=(promenna<=30))`	to samé jako předcházející příkaz

8.7 Pravděpodobnostní rozdělení

Normální rozdělení

`dnorm(x)`	hustota $\mathcal{N}(0, 1)$
`pnorm(x)`	distribuční funkce $\mathcal{N}(0, 1)$, $\mathrm{P}(X \leq x)$
`qnorn(p)`	p-kvantil $\mathcal{N}(0, 1)$, x: $\mathrm{P}(X \leq x) = p$
`rnorm(n)`	n pseudonáhodných standardně normálně rozdělených hodnot

Diskrétní rozdělení – pravděpodobnostní funkce

`dbinom(x, n, p)`	binomické rozdělení s `n` pokusy a pravděpodobností úspěchu `p`
`dgeom(x, prob)`	geometrické rozdělení s pravděpodobností úspěchu `p`
`dnbinom(x, size, prob)`	negativně binomické rozdělení s pravděpodobností úspěchu `p` a počtem `size` úspěchů na které čekáme
`dhyper(x, m, n, k)`	hypergeometrické rozdělení (výběr bez vracení), kde `m` je počet bílých koulí v urně, `n` počet černých koulí v urně a `k` počet koulí tažených z urny
`dpois(x, lambda)`	Poissonovo rozdělení se střední hodnotou `lambda`
`dmultinom(x, size, prob)`	multinomické rozdělení
`dsignrank(x, n)`	rozdělení Wilcoxonovy jednovýběrové statistiky ve výběru o rozsahu `n`
`dwilcox(x, m, n)`	rozdělení Wilcoxonovy dvouvýběrové statistiky pro výběry o rozsahu `m` a `n`

Spojitá rozdělení s oborem hodnot $\mathbb{R}$ – hustoty

`dnorm(x, mean, sd)`	normální rozdělení se střední hodnotou `mean` a směrodatnou odchylkou `sd`
`dt(x, df)`	Studentovo t rozdělení s `df` stupni volnosti
`dcauchy(x, location, scale)`	Cauchyho rozdělení (zobecnění t_1 rozdělení)
`dlogis(x, location, scale)`	logistické rozdělení

Spojitá rozdělení s oborem hodnot $\mathbb{R}^+$ – hustoty

`dexp(x, rate)`	exponenciální rozdělení se střední hodnotou `1/rate`
`df(x, n1, n2)`	Fisherovo-Snedeckorovo F rozdělení se stupni volnosti `n1` a `n2`
`dchisq(x, df)`	χ^2 rozdělení s `df` stupni volnosti
`dlnorm(x, mean, sd)`	log-normální rozdělení, tj. $\log(X) \sim \mathcal{N}(\texttt{mean}, \texttt{sd}^2)$
`dweibull(x, shape, scale)`	Weibullovo rozdělení
`dgamma(x, shape, rate)`	gamma rozdělení se střední hodnotou `shape/rate`

Spojitá rozdělení s oborem hodnot rovným intervalu v $\mathbb{R}$ – hustoty

`dunif(x, min, max)`	rovnoměrné rozdělení na intervalu (`min`, `max`)
`dbeta(x, a, b)`	beta rozdělení na intervalu (`a`, `b`)

Stejné značení jako u normálního rozdělení, tj. p-q-r, platí pro hustoty, kvantilové funkce a funkce generující pseudonáhodná čísla.

8.8 Standardní statistické metody

Kvantitativní (spojitá) odezva

`t.test`	jedno a dvouvýběrový t test
`pairwise.t.test`	párový t test, resp. mnohonásobné porovnávání
`cor.test`	test o korelačním koeficientu pro normálně rozdělená data
`var.test`	porovnání dvou rozptylů pro normálně rozdělená data (F test)
`lm(y ~ x)`	jednoduchá regrese `y` na `x`
`lm(y ~ f)`	je-li `f` faktor, jednoduchá analýza rozptylu (*one-way ANOVA*)
`lm(y ~ f1 + f2)`	jsou-li `f1` a `f2` faktory, analýza rozptylu se 2 faktory (*two-way ANOVA*)
`lm(y ~ f + x)`	je-li `f` faktor, analýza kovariance (*ANCOVA*)
`lm(y ~ x1 + x2 + x3)`	vícerozměrná regrese
`bartlett.test`	Bartlettův test na porovnání k rozptylů

Kvantitativní (spojitá) odezva – neparametrické metody

`wilcox.test`	jedno a dvouvýběrový Wilcoxonův test
`kruskal.test`	Kruskalův-Wallisův test (neparametrická jednoduchá ANOVA)
`friedman.test`	Friedmanova neparametrická ANOVA s dvěma faktory
`cor.test(method = "kendall")`	test o nulovosti Kendallova τ
`cor.test(method = "spearman")`	test o nulovosti Spearmanova ρ

Diskrétní odezva

`binom.test`	binomický test (včetně znaménkového testu)
`prop.test`	test porovnávající pravděpodobnosti úspěchu (*proportions*) ve dvou výběrech
`prop.trend.test`	test pro trend v pravděpodobnostech úspěchu
`fisher.test`	Fisherův exaktní faktoriálový test v malých kontingenčních tabulkách
`chisq.test`	χ^2 test nezávislosti v kontingenční tabulce
`glm(y ~ x1 + x2 + x3, binomial)`	logistická regrese
`glm(count ~ f1 + f2 + f3, poisson)`	poissonovská regrese (log-lineární model)