Programové prostředí R

pro biology

Karel Zvára

Pomůcka pro studenty biologie, kteří si v letním semestru akademického roku 2008/09 zapsali Základy biostatistiky

Verze ze dne 4. března 2009

Děkuji řadě svých kolegů, zejména Arnoštu Komárkovi a Janě Rubešové

Obsah

1 Úvod

R je prostředí pro statistické výpočty. Je to volně šiřitelná implementace jazyka S, čímž se liší od jiné jeho implementace, od komerčního S Plus. V následujícím se pokusíme čtenáře seznámit s používáním erka, poradíme, kde tento program najde, aby si jej mohl instalovat. Text je určen především studentům biologie, kteří si zapsali předmět Základy biostatistiky, takže příklady budou inspirovány touto přednáškou a cvičením k ní. Další zvláštností je ohled na práci s Commanderem, který je obsažen v erkové knihovně `Rcmdr`, což poněkud ovlivní i doporučenou instalaci.

Děkuji svým kolegům, kteří se mnou uvedenou výuku vedou a kteří s přípravou textu velice pomohli. Zejména byly užitečné podobné pomůcky, které připravil Arnošt Komárek, které jsou však určeny studentům, jež mají k počítačům a matematice poněkud blíže.

2 Instalace

Popíši zde instalaci na osobním počítači, který běží pod Windows XP, nemám zkušenost s Windows Vista. Naštěstí hned výchozí stránka CRAN (viz dále) obsahuje odkaz na příslušná doporučení i pro tuto nejnovější verzi Windows. Uživatelé Linuxu se musí spokojit s tím, co najdou na erkových stránkách na internetu.

2.1 Stažení

Program se stáhne z některého ze zrcadel CRAN dostupných na internetu na adrese `http://cran.r-project.org/`. V odstavečku **Download and Install R** klepneme na Windows, v příštím kroku zvolíme **Base** a pak stáhneme (download) nejnovější verzi programu (dnes `R-2.8.1`) do svého počítače. Číselné označení se s novými verzemi postupně mění. Na okraj poznamenejme, že na začátku letního semestru ak. roku 2008/09 je v počítačových učebnách PřF verze R 2.7.2. Rozdíl však nebude patrný. Po spuštění staženého programu proběhne běžná instalace. Ve všech případech, kdy se instalační program na něco táže, je možno odklepnout přednastavenou odpověď. Po skončení by se na obrazovce měla objevit nová ikona s velkým modrým symbolem R.

2.2 Úprava volání programu

Program R má dvě možnosti, jak si postupně otevírat jednotlivá okna. Při standardním nastavení (MDI – multiple-document interface) erko svoje okénka (kromě windowsovské nápovědy) otevírá uvnitř jednoho velkého okna. Commander však předpokládá SDI (single-document interface), ale za běhu erka není přepínání možné. Je několik možností, jak tomuto požadavku vyhovět. Uvedu jednu z nich, snad tu nejjednodušší. Najdeme na panelu ikonku R, jemným klepnutím si ji označíme a pomocí Ctrl+C a Ctrl+V umístíme na panel její kopii, kterou následně upravíme. Najedeme na kopii myší, klepneme pravým tlačítkem, což umožní upravit název (např. na `SDI`). Použijeme pravé tlačítko myši znovu a upravíme `Vlastnosti`: upravíme **Cíl** tak, že za poslední uvozovky uděláme mezeru a napíšeme `--sdi`. Doporučuji při té příležitosti upravit také výchozí adresář v položce **Spustit** na takový adresář nebo aspoň logický disk, kam budou soubory pro práci s erkem ukládány, i když si tento adresář můžeme v erku měnit i za jeho běhu.

Učebnová instalace obsahuje dvě ikony zmíněné verze `R 2.7.2`. Režim SDI, který pro běh commanderu potřebujeme, spustí ikona označená `R 2.7.2 SDI`.

2.3 První spuštění

Dříve, než budeme opravdu pracovat, doporučuji zvolit vhodné místo na práci, tedy vhodnou pracovní složku (adresář). Vždycky se nejlépe pracuje „doma“. K tomu slouží posloupnost příkazů z horní nabídky **File, Change dir...**. Doporučuji mít na svém pracovním disku `J:` speciální složku k našemu předmětu, např. `J:\biostat` a tuto složku používat jako pracovní. Data je užitečné ukládat do zvláštní složky, např. do `J:\biostat\data`.

Erko budeme spouštět z upravené ikony R na ploše. Otevře se okno s úvodním textem obsahujícím například informaci o verzi programu. V posledním řádku se objeví červený vyzývací symbol >. Abychom mohli spustit commander, musíme nejprve instalovat příslušnou knihovnu.

2.4 Doplnění knihoven

Erko dnes obsahuje obrovskou spoustu nejrůznějších statistických postupů, modelů atd. Je jich tolik, že není možné mít všechny současně aktivně použitelné, připravené v paměti počítače. Vždy jsou aktivní jen některé balíčky (package), všechny instalované balíčky jsou uloženy na disku. Na síti lze nalézt více než tisícovku knihoven. Při nahoře popsané instalaci se do počítače, na jeho disk, dostanou jen některé z nich, ale například knihovna Rcmdr nikoliv. Ve chvíli, kdy jsme připojeni na internet, je však velice snadné instalovat další knihovny.

Ve spuštěném erku zvolíme z horní nabídky po řadě `Packages | Install packages`. Objeví se dlouhý seznam internetových zrcadel CRAN, z nichž si jedno vybereme. Doporučuji zvolit `Austria`.

V dalším seznamu, který se za chvíli objeví, označíme knihovny, které chceme stáhnout. Pro fungování commanderu je nutné označit knihovnu `Rcmdr`. Doporučuji zvolit také `RcmdrPlugin.TeachingDemos`. Knihovna Rcmdr vyžaduje ke svému běhu řadu dalších knihoven. Některé se automaticky stáhnou spolu s Rcmdr, o některé si řekne Commander až při svém běhu. Proto doporučuji hned po instalaci hned si commander vyzkoušet. Příště už budou potřebné knihovny připraveny na pevném disku počítače.

Poznámka Někdy se až při prvním spuštění commanderu vyskytl problém, že chyběla knihovna tkrplot. Pokud se to stane, stačí ji dodatečně pomocí `Packages | Install packages` instalovat z některého internetového zrcadla CRAN.

3 Commander 1

Na cvičení budeme většinou pracovat v R prostřednictvím commanderu, několik poznámek o práci bez této knihovny bude uvedeno na konci tohoto textu. Také přednáškové slajdy obsahují informace o funkcích prostředí Rpoužitelných přímo, bez commanderu.

Jak již bylo řečeno, commander je obsažen v erkové knihovně Rcmdr. Seznam knihoven, které jsou v dané chvíli dostupné (jsou zavedeny do operační paměti, program do nich „vidí“), poskytne příkaz `search()`, který můžeme napsat za vyzývací symbol do okénka `R Console`. Druhou možností je v horní nabídce zvolit postupně `Misc | List search path`, což napíše příkaz za nás. Odpovědí je **vyhledávací cesta**, tedy seznam míst, kde program hledá význam slov (identifikátorů, názvů objektů, funkcí atd), když na ně narazí. Názvy uvozené slovem `package` ukazují na knihovny v dané chvíli zavedené do operační paměti.

Při spuštěném erku knihovnu `Rcmdr` zavedeme a současně commander spustíme příkazem `library(Rcmdr)`. Při prvním spuštění commanderu systém může nabídnout, že (nejspíš z

internetu) nainstaluje případně chybějící knihovny. Nelekejte se, jej jich poměrně dost, takže instalace chvíli potrvá. Pokud bychom commander někdy opustili, ale v operační paměti nechali knihovnu `Rcmdr`, commander opětovně spustíme příkazem `Commander()`.

Na cvičení budeme využívat některé didaktické pomůcky z knihovny `TeachingDemos`. V commanderu budou dostupné přímo v nabídce, pokud zavedeme knihovnu `RcmdrPlugin.TeachingDemos`. Proto budeme na prvních cvičeních commander i s didaktickými pomůckami zavádět pomocí příkazu `library(RcmdrPlugin.TeachingDemos)`. Ten vedle pomůcek zavede i samotný commander.

3.1 Okna commanderu

Bez ohledu na spuštěný commander bychom mohli svoje příkazy psát přímo do erkového okna nazvaného **R Console**. Příkazy se píší červeně, erko odpovídá modře. Pro nás bude vhodnější psát příkazy a jejich posloupnosti – scripty – do okna commanderu nazvaného **Script Window**. Příkaz se provede, když umístíme kurzor do řádku, v němž je příkaz napsán a stiskneme kombinaci kláves `Ctrl+R` resp. klepneme na tlačítko `Submit`.

Několik takových příkazů uvedených v po sobě jdoucích řádcích se provede, když je windowsovským způsobem označíme a stiskneme známou kombinaci kláves `Ctrl+R` resp. klepneme na tlačítko `Submit`. Skripty můžeme pomocí `File | Save script...` ukládat do souboru v paměti počítače a naopak z paměti nahrávat pomocí `File | Open script file`.

Práce se souborem skriptů má v porovnání s přímým zápisem příkazů do konzole několik výhod. Zejména si můžeme několik příkazů připravit předem, můžeme je upravovat pomocí běžných edičních příkazů. Do tohoto souboru můžeme psát také svoje poznámky a komentáře, které bychom ovšem neměli spouštět jako příkazy. Pokusu spustit komentář jako příkaz zabráníme, když jej uvedeme symbolem #, který platí až do konce řádku. Ještě jedno poučení. Chceme-li zapsat do jednoho řádku více než jeden příkaz, musíme příkazy oddělit středníkem. Doporučuji čas od času soubor se skripty uložit, např. kombinací kláves `Ctrl+S`.

Při běžném nastavení commanderu se náš odeslaný příkaz červeně zkopíruje do okna nazvaného **Output Window**. Na dalších řádcích tohoto výstupního okna pak zpravidla následuje modře zbarvená odpověď.

Zcela dole je umístěno malé okénko označené **Messages** určené pro hlášení chyb, upozornění, někdy také oznámení úspěšného vytvoření nebo rozšíření objektu obsahujícího data (databáze, data.frame).

3.2 Čeština

V erku se zhusta používají nealfanumerické znaky, které jsou na české klávesnici poněkud skryty, lze je vytvořit jen vhodnou kombinací kláves. V tabulce 1 jsou soustředěny některé kombinace kláves potřebné při psaní těchto symbolů, když máme nastavenu českou klávesnici. Všimněme si, že vždy jde o kombinaci pravé klávesy `Alt` s některou další klávesou. Stejný význam jako pravý Alt má kombinace kláves `Ctrl+Alt` (levý).

4 Data

4.1 Načtení dat

Dříve, než začneme vůbec počítat, ukážeme si, jak dostat do erka větší množství dat. Nejprve popíšeme jak načíst data předem připravená ve formě textového souboru. Zde uvedeme pouze několik prvních několik řádků souboru `Deti23.csv`

Tabulka 1: Kombinace kláves, kterými na české klávesnici lze napsat některé nealfanumerické znaky. Znaky za symbolem + se vztahuje k anglickému popisu kláves.

symbol	klávesy	použití
#	pravý Alt + x	uvozuje komentář
~	pravý Alt + 1	vlnka v zápisu příkazů (něco závisí na něčem)
|	pravý Alt + w	popis modelu v některých příkazech též logické sjednocení (nebo)
<	pravý Alt + ,	nerovnost, součást přiřazovacího příkazu
>	pravý Alt + .	nerovnost
[	pravý Alt + f	levá hranatá závorka, zápis indexů
]	pravý Alt + g	pravá hranatá závorka, zápis indexů
{	pravý Alt + b	levá složená závorka, začátek bloku příkazů
}	pravý Alt + n	pravá složená závorka, konec bloku příkazů
^	pravý Alt + 3	symbol mocniny, objeví se až po dalším znaku
$	pravý Alt + ;	paragraf, spojí název databáze a název proměnné
&	pravý Alt + c	logický průnik (současně platí)

```
hoch;poradi;vekMatky;vekOtce;vaha;delka;hcd;Gender
1;1;20;23;11,95;83;0;M
1;1;34;36;10,2;72;0;M
```

V prvním řádku máme názvy proměnných, každý další řádek obsahuje informaci o jenom dítěti. Jde o export z excelovské tabulky provozované v českém prostředí (u váhy je desetinná čárka) do souboru typu CSV, kde je použit středník jako oddělovač jednotlivých hodnot. Čárka je tu oddělovačem desetinným. Další možnosti načtení dat si ukážeme později.

Protože pracujeme pomocí commanderu, použijeme jeho nabídku. Zvolíme `Data | Import data | from text file, clipboard or URL`.
V oknu, které se otevře, zvolíme název budoucí databáze (`Deti23`), ponecháme zaškrtnuté okénko indikující, že v prvním řádku našeho souboru jsou uvedeny názvy jednotlivých proměnných, ponecháme označení NA pro chybějící hodnoty, ponecháme volbu zdroje dat na `Local file system`, určíme středník jako oddělovač záznamů tak, že označíme volbu `Other` a do následujícího čtverečku umístíme středník a nakonec jako oddělovač desetinných míst označíme čárku. Po odklepnutí tlačítka OK budeme mít možnost běžným windowsovským způsobem zvolit nahrávaný soubor `Deti23.csv`. Při doporučené konfiguraci pracovních složek jej najdeme ve složce nazvané `data`. Commander vygeneruje příslušný příkaz, napíše jej do skriptového okna commanderu, příkaz také zkopíruje do výstupního okna commanderu. Jméno zavedené databáze se objeví i na tlačítku. Současně se v informačním okénku objeví sdělení o počtu řádků (počet jedinců, zde dětí) a počtu sloupců (proměnných) v nové databázi.

Stejného výsledku jsme mohli dosáhnout tak, že bychom příkaz k načtení napsali do skriptového okna sami. Protože víme předem, jak jsou data připravena, mohl by příkaz být jednodušší, stačilo by použít funkci `read.csv2()` a napsat `Deti23 <- read.csv2("data/Deti23.csv")`.

Je užitečné všimnout si toho, jak je vyjádřeno, že funkce `read.csv2()` resp. `read.table()` ukládá načtená data do objektu nazvaného `Deti23`. Je to pomocí dvojice znaků `<-`.

Dvojka na konci označení funkce indikuje, že jde o variantu funkce `read.csv()`, která předpokládá jako oddělovač čísel či názvů proměnných středník a za desetinný oddělovač po-

važuje čárku. Kdybychom se chtěli ujistit, jaké parametry má funkce `read.csv2()`, zeptáme se. Příkaz `?read.csv2` otevře okénko s nápovědou. Ekvivalentem je `help(read.csv2)`.

Takto se můžeme ptát na všechny funkce z knihoven, které jsou aktivovány (jsou na vyhledávací cestě). Přístup k nápovědě o všech funkcích instalovaných v počítači, byť v této chvíli třeba neaktivních (podrobněji bude vysvětleno dále), bychom získali vyvoláním helpu ve formátu html v horní nabídce pomocí `Help | Html help`.

Možná, že význam většiny parametrů commanderem připraveného dlouhého příkazu uhodnete, když si vzpomenete na formulář, který jsme při přípravě tohoto příkazu vyplňovali. Pomůckou může být také `?read.table`. Námi „ručně“ použitá funkce `read.csv2` je pouhým zkráceným vyvoláním funkce `read.table()`, v němž jsou jinak než standardně nastaveny některé parametry.

4.2 Zápis dat kvantitativních, kvalitativních a dalších

Dost však řečí, prohlédněme si načtená data. Poklepeme-li na tlačítko nazvané `View dataset`, otevře se nové okénko (může se schovávat pod jiným okénkem!) s daty. Opět zde vypíšeme jen několik prvních řádků:

```
  hoch poradi vekMatky vekOtce  vaha delka hcd Gender
1    1      1       20      23 11.95    83   0      M
2    1      1       34      36 10.20    72   0      M
3    1      4       36      55 11.10    80   1      M
4    1      2       31      38 12.80    77   2      M
5    1      1       20      36 12.60    78   0      M
6    1      3       24      29 10.60    75   2      M
```

Vidíme, že desetinná čárka byla nahrazena tečkou, že řádky nemají speciální pojmenování, takže jsou označeny pouze pořadovými čísly, kdežto sloupce databáze jsou označeny jmény příslušných proměnných. Zajímavý je poslední sloupec, nazvaný `Gender`, jehož hodnotami jsou nečíselné znaky. Jsou tam sice jen jednotlivá písmena, ale mohly to být i delší řetězce. Budeme mít možnost se přesvědčit, že tato proměnná nese stejnou informaci, jako číselná proměnná `hoch`. Proměnná `Gender` nese informaci o kvalitativním znaku. V erku se takové proměnné říká `faktor`. Lze ji použít například při třídění, takže nebude problém spočítat průměrnou váhu hochů a průměrnou váhu dívek. Když má faktor vyjadřovat úrovně kvalitativního znaku v ordinálním měřítku, pak tento faktor může dostat příznak `ordinal`. Hodnoty takového znaku (ordinálního faktoru) jsou uspořádány.

Ukažme si, jak z číselné proměnné lze vytvořit faktor. V okénku s daty si můžeme ověřit, že počet onemocnění horních cest dýchacích (`hcd`) nabývá hodnot od nuly do devíti, což je pro další zkoumání příliš podrobné rozlišování. Vyrobme novou veličinu, která rozliší pouze tři úrovně: bez onemocnění, jediné a opakované onemocnění. V horní nabídce commanderu postupně zvolíme `Data | Manage variables in active data set | Recode variables...`. Označíme `hcd` jako veličinu určenou k překódování, zaškrtneme, že chceme výsledek (novou veličinu) jako faktor a do největšího okénka napíšeme do jednotlivých řádků definice nových hodnot ve tvaru původní hodnota = nová hodnota. Protože výsledek má být faktor, pro nové hodnoty zvolíme slovní vyjádření. Potřebné tři řádky mohou mít tvar:

```
0 = "bez onemocnení"
1 = "jediný výskyt"
else = "opakovaný výskyt"
```

Nezapomeneme zvolit označení pro novou veličinu, např. `HCD`, teprve pak klepneme na `OK`.

5 Popisné statistiky

5.1 Míry polohy

Když už víme, že máme databázi v pracovním prostoru, mohli bychom si zjistit kolik sloupců (proměnných) obsahuje. Snadná pomoc. V horní nabídce zvolíme `Data | Active data set | variables in active data set`. Následuje odpověď ve výstupním oknu:

```
[1] "hoch"     "poradi"   "vekMatky" "vekOtce"  "vaha"
[6] "delka"    "hcd"      "Gender"
```

Nás však zajímají míry polohy. Mnohou informaci o všech proměnných databáze získáme aplikací funkce `summary()` použité na databázi, tedy
`summary(Deti23)`. Stejný příkaz připraví postupná volba v horní nabídce:
`Statistics | Summaries | Active data set`

```
     hoch            poradi         vekMatky        vekOtce
 Min.   :0.0000   Min.   :1.000   Min.   :17.00   Min.   :19.00
 1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:21.50   1st Qu.:24.00
 Median :0.0000   Median :1.000   Median :24.00   Median :27.00
 Mean   :0.4783   Mean   :1.739   Mean   :25.83   Mean   :30.22
 3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:31.00   3rd Qu.:36.00
 Max.   :1.0000   Max.   :4.000   Max.   :41.00   Max.   :55.00
      vaha            delka            hcd        Gender
 Min.   : 8.40   Min.   :68.00   Min.   :0.000   F:12
 1st Qu.: 9.55   1st Qu.:74.00   1st Qu.:0.000   M:11
 Median :10.20   Median :77.00   Median :1.000
 Mean   :10.50   Mean   :76.65   Mean   :1.739
 3rd Qu.:11.22   3rd Qu.:79.00   3rd Qu.:2.000
 Max.   :13.00   Max.   :83.00   Max.   :9.000
```

S výjimkou proměnné `Gender` poskytl nám výstup z erka pro každou proměnnou její aritmetický průměr (`Mean`), medián (`Median`), dolní a horní kvartil (`1st Qu., 3rd Qu.`), minimum (`Min.`) a maximum (`Max.`).

Číselné charakteristiky proměnné `vekOtce` získáme postupnou volbou `Statistics | Summaries | Numerical summaries ...`, po které následuje možnost zvolit si jednu či několik proměnných. Zvolíme-li (pomocí `Ctrl`) současně proměnné `vekOtce, vekMatky`, dostaneme tabulku

```
             mean       sd 0%  25% 50% 75% 100%  n
vekMatky 25.82609 6.436064 17 21.5  24  31   41 23
vekOtce  30.21739 9.019960 19 24.0  27  36   55 23
```

Každou z uvedených statistik můžeme spočítat zvlášť, když do skriptového okénka commanderu sami napíšeme příslušnou funkci. Abychom nemuseli opakovaně používat dlouhé názvy proměnných (včetně názvu databáze, například `Deti23$vekMatky`), zpřístupníme si kopii databáze pomocí příkazu `attach(Deti23)`. Ten zapíšeme do `Script Window` a pomocí `Ctrl+R` nebo klepnutím na `Submit` necháme provést.

Měli bychom si ověřit, že rozumíme vzorečkům pro výpočet nejznámějších měr polohy a variability. Použijeme proměnnou `vekMatky` a necháme spočítat průměr a medián:

```
c(mean(vekMatky),median(vekMatky))
[1] 25.826087 24.000000
```

Poněkud šikovnější by byl příkaz
`c(průměr=mean(vekMatky),medián=median(vekMatky))` s odpovědí obsahující právě zvolené označení jednotlivých statistik:

```
  průměr   medián
30.21739 27.00000
```

Kvartily (a nejen kvaritily) nám spočítá funkce `quantile()`:
`quantile(vekMatky,probs=(0:4)/4)`

```
  0%  25%  50%  75% 100%
17.0 21.5 24.0 31.0 41.0
```

Abychom vysvětlili význam parametru `probs` funkce `quantile()`, připomeňme, že například dolní kvartil je číslo, které odděluje 25 % nejmenších hodnot od ostatních. Je to 25% parcentil, tedy 25% výběrový kvantil. Minimum lze chápat jako 0% percentil, maximum jako 100% percentil. Když si uvědomíme, že `0:4` znamená posloupnost čísel $0, 1, 2, 3, 4$, je zřejmé, že `(0:4)/4` dá posloupnost čísel 0, 0,25, 0,5, 0,75, 1, tedy pomocí procent vyjádřenou posloupnost 0 %, 25 %, 50 %, 75 %, 100 %. K vlastnímu výpočtu percentilů se vrátíme za chvíli.

5.2 Variační řada, pořadí

Když řadu čísel uspořádáme do neklesající posloupnosti, dostaneme variační řadu. K tomu slouží funkce `sort()`:
`sort(vekMatky)`

```
 [1] 17 19 19 20 20 21 22 22 23 23 23 24 24 25 25 26 31 31
[19] 34 34 34 36 41
```

Medián je tedy roven věku maminky v pořadí dvanácté podle věku. Že právě dvanácté, plyne ze skutečnosti, že máme právě 23 hodnot proměnné `vekMatky`, jak víme z nedávného výpočtu číselných statistiky nebo jak jsme mohli zjistit jednoduchým příkazem
`length(vekMatky)`

```
[1] 23
```

Stačilo tedy říci si o dvanáctou hodnotu variační řady:
`sort(vekMatky)[12]`

```
[1] 24
```

S funkcí `sort()` souvisí funkce `order()`. První složka vektoru, který je výsledkem této funkce nám říká, kde hledat nejmenší hodnotu, druhá složka říká, kde hledat druhou nejmenší hodnotu atd.
`order(vekMatky)`

```
 [1]  7 10 21  1  5 16  8 12  9 13 20  6 17 14 18 19  4 15
[19]  2 11 22  3 23
```

Pro zajímavost, variační řadu bychom tedy mohli dostat také pomocí
`vekMatky[order(vekMatky)]`

```
 [1] 17 19 19 20 20 21 22 22 23 23 23 24 24 25 25 26 31 31
[19] 34 34 34 36 41
```

Takto složitě jistě variační řadu počítat nebudeme, ale funkce `order()` nám umožní uspořádat řádky databáze tak, aby ve zvolené proměnné byly hodnoty neklesající, např. v našem případě od nejmladší maminky k nejstarší:
`Deti23[order(vekMatky),]`

```
   hoch poradi vekMatky vekOtce  vaha delka hcd Gender
7     1      1       17      19  9.40    78   1      M
10    1      1       19      24  9.60    73   3      M
                     ...
6     1      3       24      29 10.60    75   2      M
                     ...
22    0      1       34      44 10.15    82   0      F
3     1      4       36      55 11.10    80   1      M
23    0      4       41      47 10.70    78   0      F
```

Mnohé statistické postupy jsou založeny na pořadí (angl. rank). Ta určí funkce `rank()`:

```
rank(vekMatky)
```

```
 [1]  4.5 20.0 22.0 17.5  4.5 12.5  1.0  7.5 10.0  2.5 20.0
[12]  7.5 10.0 14.5 17.5  6.0 12.5 14.5 16.0 10.0  2.5 20.0
[23] 23.0
```

Z předchozí tabulky víme, že nejmladší matka byla původně v 7. řádku, takže 7. prvek vektoru `rank(vekMatky)` má opravdu pořadí 1. Dále, 2. a 3. člen variační řady proměnné `vekMatky` jsou, jak víme, rovny 19. Proto mají obě hodnoty 23 průměrné pořadí 2,5 atd.

Nyní máme příležitost ověřit si výpočet kvartilů. Dolní kvartil má oddělovat čtvrtinu nejmenších hodnot. Při výpočtu parcentilů se používá řada postupů, z nichž každý je v nějakém smyslu nejlepší. V erku je možno volit celkem devět různých postupů. Popíšeme ten, který je nastaven standardně.

Počítáme p-tý percentil, kde p je číslo mezi nulou a jedničkou. Jak je zvykem, členy variační řady označíme indexy v závorce, tedy

$$x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)}.$$

Postupně určíme čísla

$$\begin{aligned} k &= \lfloor 1 + (n-1)\cdot p \rfloor && \text{celá část, zaokrouhlíme dolů} \\ q &= \{1 + (n-1)\cdot p\} = (1 + (n-1)\cdot p) - k && \text{zlomková část} \\ x_p &= (1-q)\cdot x_{(k)} + q \cdot x_{(k+1)} && p\text{-tý percentil} \end{aligned}$$

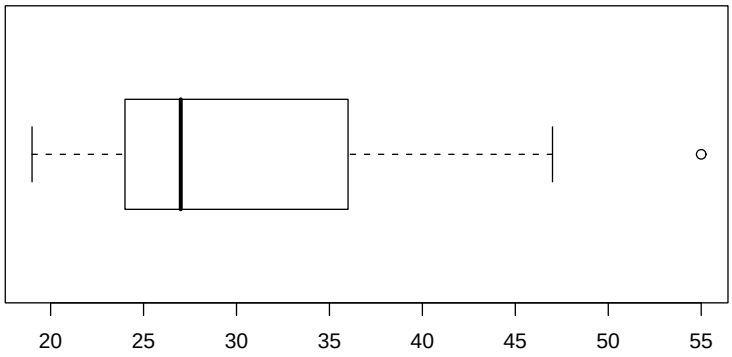
Počítáme-li dolní kvartil věku matek, je postupně ($p = 0{,}25, n = 23$)

$$\begin{aligned} k &= \lfloor 1 + (23-1)\cdot 0{,}25 \rfloor = \lfloor 6{,}5 \rfloor = 6 \\ q &= 6{,}5 - 6 = 0{,}5 \\ Q_1 = x_{0{,}25} &= (1 - 0{,}5)\cdot x_{(6)} + 0{,}5 \cdot x_{(7)} = 0{,}5 \cdot 21 + 0{,}5 \cdot 22 = 21{,}5, \end{aligned}$$

což přesně odpovídá dosavadním zjištěním

5.3 Krabicový diagram

Je velice užitečné znázorňovat data graficky. K nejjednodušším grafům, patří krabicový diagram. Vytvořme krabicový diagram proměnné `vekOtce` použitím příkazu `boxplot(vekOtce, horizontal=TRUE)` nebo snáze postupnou volbou `Graphs | Boxplot ...`, načež následuje volba proměnné. Otevře se grafické okno (někdy je třeba jej na ploše chvíli hledat). Abychom graf otočili volbou parametru `horizontal`, upravme ve skriptovém oknu poslední příkaz `boxplot(Deti23$vekOtce, ylab="vekOtce")` na
`boxplot(Deti23$vekOtce, horizontal=TRUE)` a ten znovu odešleme. Dostaneme tak



Uveďme percentily, které lze bez námahy identifikovat na našem grafu.
`quantile(vekOtce)`

```
  0%  25%  50%  75% 100%
  19   24   27   36   55
```

Na pravé straně je osamocený bod, který odpovídá pětapadesátiletému muži. Je označen jako odlehlý bod, protože je od bližšího (horního) kvartilu 36 vzdálen dál, než je $(Q_3 - Q_1) * 3/2 = 18$.

Na místě je však malé upozornění. Ne vždy odpovídá krabicový diagram přesně tomu, co spočítáme pomocí funkce `quantile()`, protože při konstrukci krabicového diagramu používá erko (nejspíš z historických důvodů) poněkud jiné odhady populačních kvartilů – tzv. hinges. Používá při tom funkci `fivenum()`, která při sudém n může dát poněkud jiné odhady populačních kvartilů, než dává `quantile()`.

Ukažme si ještě, nakolik dostupné míry polohy vyhoví známým dvěma požadavkům, jak reagují na posunutí počátku (přičtení stejné konstanty ke všem napozorovaným hodnotám), a na změnu měřítka (vynásobení stejnou kladnou konstantou). Ukážeme si přitom, jak můžeme z našich proměnných vytvořit nové.

Nejprve zaveďme proměnnou `delkaM`, která délku dítěte vyjádří nikoliv v centimetrech, ale v metrech. Zvolme postupně
`Data | Manage variables in active data set | Compute new variable`. Otevře se okno, v něm zvolíme název nové proměnné a v pravém okénku uvedeme příslušný vzoreček `delka/100`. Název proměnné `delka` můžeme do toho okénka nakopírovat poklepáním na příslušný název. Po vyplnění pochopitelně klepneme na tlačítko OK.

Posun si ukážeme na věku matek, když si spočítáme, jak dlouho je ta která matka plnoletá a tuto hodnotu (`vekMatky - 18`) uložíme jako `plnoletost` do databáze Deti23. Kontrola zmíněných vlastností je snadná, použijeme opět volbu
`Statistics | Summaries | Numerical summaries ...`, na čtveřici proměnných. Zrušíme přitom zaškrtnutí výpočtu směrodatné odchylky. Dostaneme tabulku, v níž snadno ověříme, že všechny uvedené míra polohy oběma požadavkům vyhověly

```
                mean    0%   25%   50%   75%  100%  n
delka      76.6521739 68.00 74.00 77.00 79.00 83.00 23
delkaM      0.7665217  0.68  0.74  0.77  0.79  0.83 23
plnoletost  7.8260870 -1.00  3.50  6.00 13.00 23.00 23
vekMatky   25.8260870 17.00 21.50 24.00 31.00 41.00 23
```

5.4 Míry variability (rozptýlení)

Nejpoužívanějšími mírami variability jsou asi směrodatná odchylka a rozptyl. Spolu s jejich výpočtem si připomeňme vztah mezi nimi:

```
c(var=var(vekOtce),sd=sd(vekOtce),sd2=sd(vekOtce)^2)
```

```
     var       sd      sd2
81.35968  9.01996 81.35968
```

Další používanou mírou variability je kvartilové rozpětí

```
quantile(vekOtce,3/4)-quantile(vekOtce,1/4)
[1] 36
```

Kvartilové rozpětí určuje mimo jiné délku strany krabicového diagramu, která je rovnoběžná s číselnou osou.

V nabídce commanderu je snadno dostupný pouze výpočet směrodatné odchylky. Zvolíme tedy `Statistics | Numerical summaries` , nastavíme naše čtyři proměnné a necháme počítat jen směrodatnou odchylku.

```
                   sd  n
delka       3.93825868 23
delkaM      0.03938259 23
plnoletost 6.43606440 23
vekMatky    6.43606440 23
```

Ověření dvou základních vlastností dobré míry variability (na posun nereaguje, kdežto na násobení konstantou ano) u rozptylu provedme jinako, přímým výpočtem. Ukážeme si přitom, proč jsme u příkazu `attach()` hovořili o **kopii** databáze. Když zkusíme spočítat například rozptyl proměnné `plnoletost` pomocí funkce `var()`, tedy zvolíme-li ve skriptovém oknu `var(plnoletost)`, dostane se nám v dolním šedivém okénku commanderu hlášení chyby: `ERROR: object "plnoletost" not found`. Erko naši novou proměnnou v databázi `nevidí`? Pomocí `Deti23` jsme totiž zpřístupnili kopii databáze tak, jak v té chvíli vypadala. Proměnné `plnoletost, delkaM` jsme vytvořili později. Řešení? Kopii odstranit a udělat ji znovu pomocí příkazů `detach(Deti23)` a `attach(Deti23)`. Nyní již nové proměnné budou přímo dostupné. Pro úsporu místa však provedeme výpočet odkazem přímo na databázi.

```
apply(Deti23[,c("delka","delkaM","vekMatky","plnoletost")],2,var)
```

dá odpověď

```
       delka        delkaM     vekMatky   plnoletost
15.509881423  0.001550988 41.422924901 41.422924901
```

Vidíme, že ani jedna z vyšetřovaných měr variability nereagovala na posunutí o 18 let u věku matky. Směrodatná odchylka reagovala přesně, pro délku v metrech vyšla stokrát menší, kdežto původní rozptyl délky dítěte bychom museli vydělit duhou mocninou čísla 100, abychom dostali rozptyl délky vyjádřené v metrech.

5.5 Další popisné statistiky

Existují situace, kdy se zajímáme o jiné vlastnosti kvantitativního znaku, než jsou míry polohy a variability. Tehdy je výhodné upravit data tak, aby již nenesla žádnou informaci o poloze resp. variabilitě. K tomu slouží tzv. z-skóry definované (musí být smaozřejmě $s_x > 0$) jako

$$z_i = \frac{x_i - \bar{x}}{s_x}.$$

Tyto nové hodnoty mají vždy stejný průměr $\bar{z} = 0$ a stejnou směrodatnou odchylku (a tedy stejný rozptyl) $s_z = 1$. O symtetrii rozdělení vypovídá (výběrová) **šikmost** definovaná jako

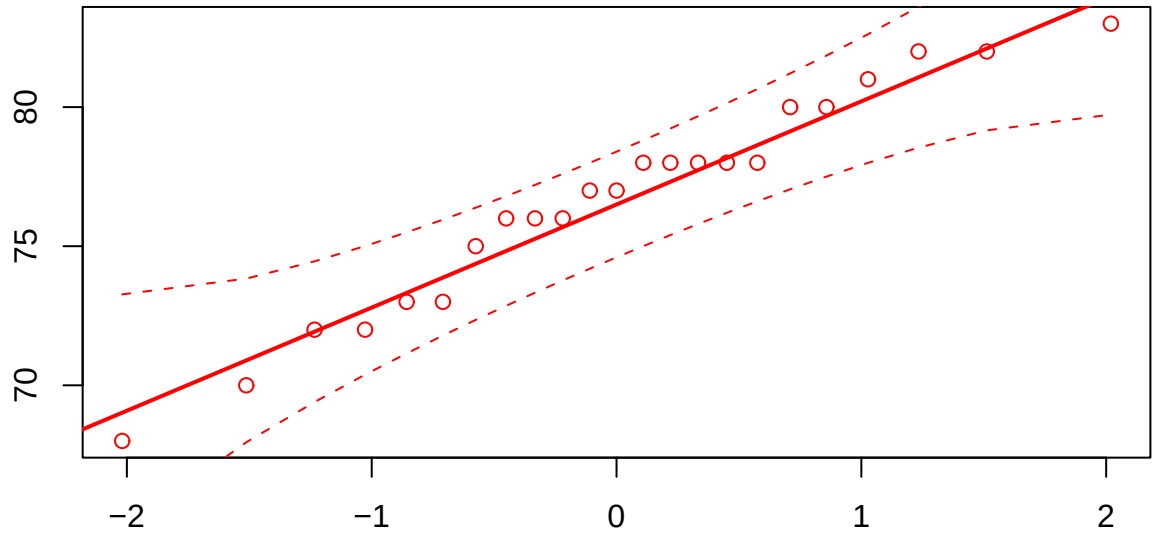
$$g_1 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3/s_x^3 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)^3,$$

tedy průměr z třetích mocnin z-skórů. Podobně se definuje (výběrová) **špičatost**

$$g_2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4/s_x^4 - 3 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)^4 - 3,$$

tedy jako průměr ze čtvrtých mocnin z-skórů zmenšený o 3. Vzdálenost těchto statistik od nuly do jisté míry vypovídá o představě, že data pocházejí z normálního rozdělení. Šikmost proměnné `delka` snadno spočítáme pomocí `delka-mean(delka))/sd(delka))^3` resp. pomocí jednoduššího (ale méně průzračného) příkazu `mean(scale(delka)^3`. V obou případech dostaneme stejnou šikmost −0,3665634. Po malé úpravě příkazů pro výpočet šikmosti spočítáme špičatost. Příkaz `mean(scale(delka)^4)-3` dá −0,6960152. V tomto případě, kdy máme jen 23 pozorování, nemůžeme na základě výběrové šikmosti či špičatosti o normalitě dat rozhodovat.

O předpokladu o normálním rozdělení vypovídá také **normální diagram** ((normal) probability plot, quantile-comparison plot), ke kterému nás v commanderu přivede posloupnost voleb `Graphs | Quantile-comparison plot ...`, když v posedním kroku necháme originální nastavení (`Normal`) a zvolíme proměnnou `delka`. Výsledek je na obrázku 1. V



Obrázek 1: Normální diagram veličiny `delka`

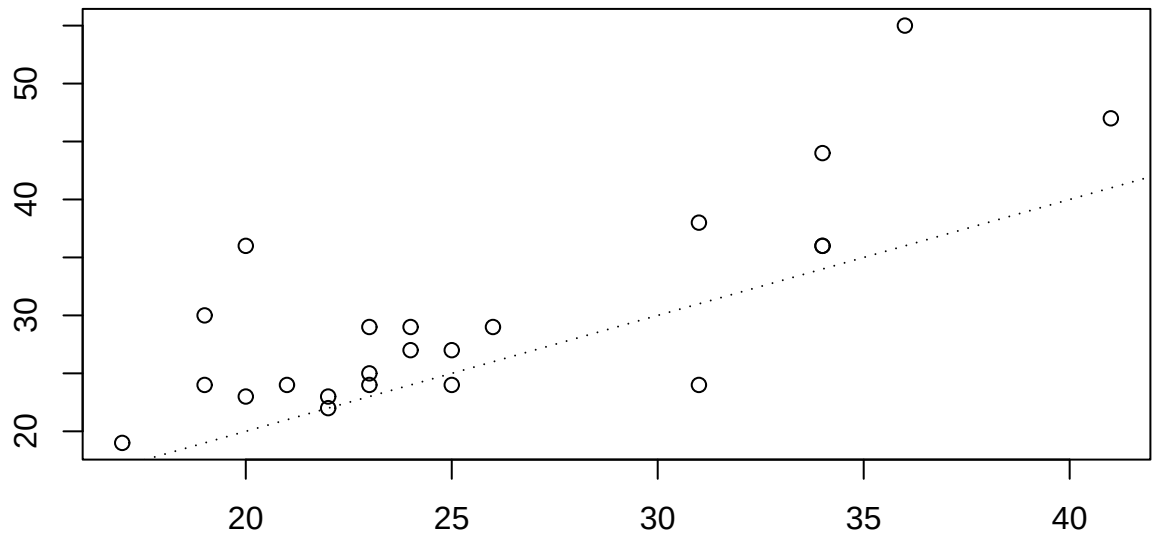
ideálním případě jsou body soustředěny kolem tučné přímky zhruba v pásu vymezeném dvěma čárkovanými křivkami. Systematický odklon od lineárního průběhu řady bodů by ukazoval na problém se šikmostí či špičatostí.

5.6 Závislost dvou znaků

Zabývejme se nyní vyšetřováním závislosti dvou znaků. Záležet bude na tom, zda jde o znaky kvantitativní nebo kvalitativní. Začněme dvojicí kvantitativních znaků a zabývejme se možnou závislostí věku rodičů.

5.6.1 Dvojice znaků kvantitativní – kvantitativní

Závislost věku otce na věku matky znázorníme graficky pomocí bodového diagramu (scatter plot), když v commanderu zvolíme `Graphs | Scatterplot` a vybereme si příslušné proměnné. Chceme-li zkoumat závislost věku otce na věku matky, umístíme na vodorovnou osu x věk matky a na svislou osu y věk otce. Necháme nezaškrtnuté zatím jen jedno políčko, totiž u nápisu `Least-squares line`. Každý z 23 bodů znázorňuje údaje o jedněch rodičích. Abychom si usnadnili hledání případů, kdy je matka starší než otec, sami přidáme příkaz `abline(0,1,lty=3)`, který do grafu doplní tečkovaně (`lty=3`) přímku s rovnicí $y = 0 + 1 \cdot x$. Pokud některý bod leží na této přímce, mají oba rodiče stejný věk. Pokud některý bod leží pod uvedenou přímkou, je matka starší než otec, leží-li nad přímkou, je tomu naopak. Je



Obrázek 2: Závislost věku otce na věku matky

patrné, že otec je mladší ve dvou případech, v jednom případě mají rodiče věk stejný.

Číselné hodnocení síly závislosti poskytne korelační koeficient, který získáme pomocí

```
> cor(vekOtce,vekMatky)
[1] 0.7938426
```

Hodnota $r = 0{,}79$ ukazuje na poměrně těsnou lineární závislost.

5.6.2 Dvojice znaků kvalitativni – kvalitativní

Možnosti si ukážeme na dvojici znaků `HCD` (výskyt zánětu horních cest dýchacích) a `Gender` (pohlaví). Spíše, než hovořit o závislosti `HCD` na `Gender` by bylo vhodnější porovnávat výskyt `HCD` u chlapců a u děvčat.

Četnosti jednotlivých kombinací hodnot znaků spočítáme pomocí funkce `xtabs()`, kterou nám připraví commander. V jeho horní nabídce zvolíme postupně `Statistics | Contingency tables | Two-way table` a pak zvolíme veličinu, jejíž hodnoty určí řádky tabulky (`Row variable` – zvolíme `Gender`) a veličinu určující sloupce (`Column variable` – zvolíme `HCD`). Ještě zrušíme zaškrtnutí v řádku `Chi-square test of independence` a můžeme odklepnout `OK`. Výsledná tabulka má tvar

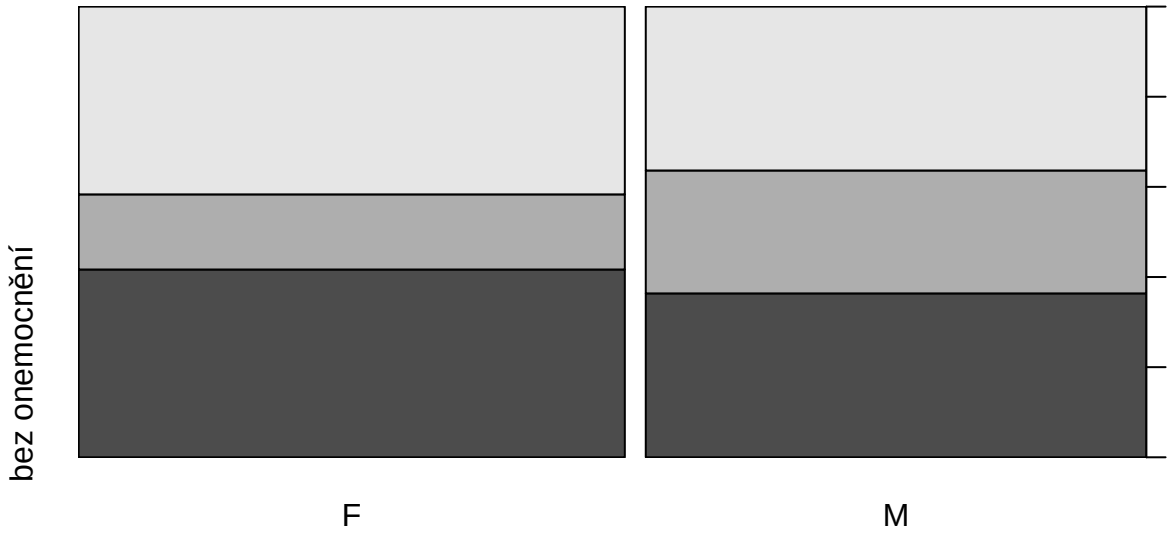
```
      HCD
Gender bez onemocnění jediný výskyt opakovaný výskyt
     F              5             2                5
     M              4             3                4
```

Pokud nás zajímají procenta tří hodnot `HCD` zvlášť mezi chlapci a mezi děvčaty, použijeme stejnou nabídku znovu, ale označíme volbu `Row percentages`. Dostaneme

```
      HCD
Gender bez onemocnění jediný výskyt opakovaný výskyt Total Count
     F           41.7          16.7             41.7 100.1    12
     M           36.4          27.3             36.4 100.1    11
```

Doporučuji vyzkoušet si i další možnosti zadání výsledné tabulky.

Commander nám pro dvojici kvalitativních znaků nenabízí žádné grafické znázornění, proto potřebnou funkci `plot(HCD ~ Gender)` zapíšeme do okénka `Script Window` sami. Výsledek ukazuje obrázek 3. Drobnou vadou na kráse je skutečnost, že se nám v popisu svislé osy nevešly do grafu zvolené dlouhé názvy jednotlivých hodnot. Je to způsobenou snahou o úsporu místa. Pokud necháte graf namalovat na svém počítači, zmíněný problém nejspíš nenastane.

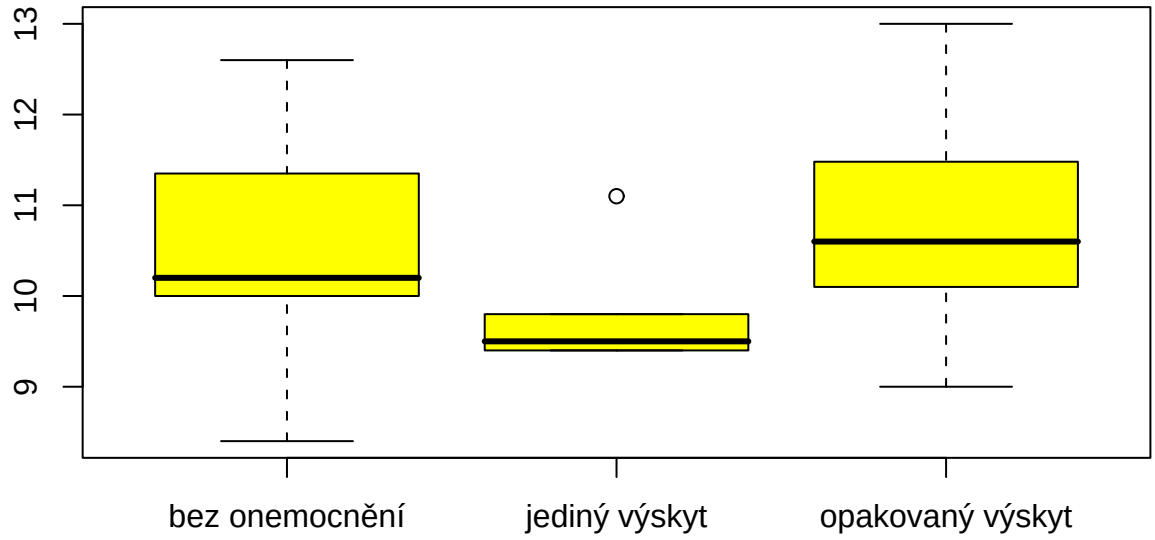


Obrázek 3: Porovnání nemocnosti na HCD pro děvčata a chlapce

5.6.3 Dvojice znaků kvantitativní – kvalitativní

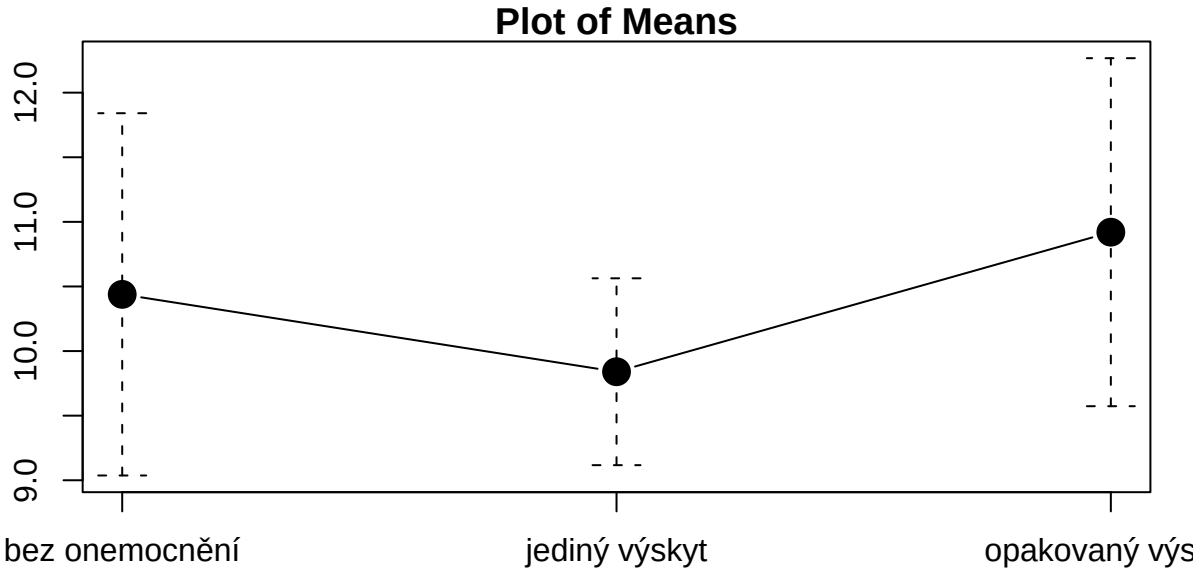
Podobně jako v případě dvojice kvalitativních znaků zde můžeme místo o vzájemné závislosti hovořit také o porovnání kvantitativního znaku pro jednotlivé hodnoty znaku kvalitativního. Ukažme si to na příkladu znaků `vaha` a `Gender`. Začneme krabicovým diagramem. V horní nabídce commanderu zvolíme `Graphs | Boxplot`. Pak zvolíme hodnocený kvantitativní znak `vaha` a kvalitativní znaku (`Plot by groups...`) pro třídění, např. `HCD`. Pokud navíc zaškrtneme možnost `Identify outliers with mose`, budeme mít možnost získat informace o případných odlehlých hodnotách. K tomu je třeba po odklepnutí `OK` odklepnout ještě malé okénko, které popisuje, jak případné odlehlé pozorování označit (levým tlačítkem myši) a jak proces označování ukončit (pravým tlačítkem myši). Pokud opravdu najdeme

odlehé pozorování a do jeho blízkosti klepneme levým tlačítkem myši, objeví se na grafu pořadové číslo pozorování (číslo řádku databáze). Pokud by jednotlivé řádky databáze měly svoje pojmenování, objevilo by se tu pojmenování příslušného řádku.



Obrázek 4: Porovnání váhy podle výskytu HCD

Další používanou možnost, jak graficky porovnat několik skupin číselných hodnot, poskytne volba `Graphs | Plot of means`, kdy pak musíme zvolit třídicí faktor (`HCD`) a hodnocenou spojitou veličinu (`vaha`). Přednastavena je ještě volba `Standard errors`, což je pojem zatím nám neznámý. Zvolíme tedy směrodatné odchylky (`Stadard deviations`) a můžeme odklepnout `OK`.



Obrázek 5: Porovnání váhy v jednom roce mezi děvčaty a chlapci

Při popisu rozdílu mezi třemi skupinami podle četnosti onemocnění je užitečné zjistit pro ednotlivé skupiny běžné popisné statistiky. K tomu účelu je vhodné zvolit v horní nabídce commanderu (Statistics — Summaries — Numerical summaries…) a tam zvolit hodnoce-

nou číselnou proměnnou (`vaha`) a třídicí faktor (`Summarize by: HCD`). Po odklepnutí `OK` dostaneme tabulku, která nám umožní ověřit obě grafická vyjádření, pomocí krabicových diagramů i pomocí tzv. `bar plot`.

```
                     mean        sd  0%  25%  50%   75% 100% n
bez onemocnění   10.43889 1.4019580 8.4 10.0 10.2 11.35 12.6 9
jediný výskyt     9.84000 0.7231874 9.4  9.4  9.5  9.80 11.1 5
opakovaný výskyt 10.92000 1.3468853 9.0 10.1 10.6 11.48 13.0 9
```