UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Data mining a project R

Martina Kršíková

Bakalářská práce
2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2013/2014**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Martina Kršíková**
Osobní číslo:	**I10104**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Datamining a project R**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Dolování dat je v poslední době velmi žádaná disciplína na rozhraní mnohorozměrné statistické analýzy a informatiky. K nejpoužívanějším nástrojům patří:
1) regresní metody (lineární regresní analýza, nelineární regresní analýza, neuronové sítě),
2) klasifikace (diskriminační analýza, logistická regresní analýza, rozhodovací stromy, neuronové sítě),
3) segmentace - shlukování (shluková analýza, genetické algoritmy, neuronové shlukování - Kohonenovy mapy),
4) analýza vztahů (asociační algoritmus pro odvozování pravidel typu "if X then Y"),
5) predikce v časových řadách (Boxova-Jenkinsonova metoda, neuronové sítě, autoregresní modely, ARIMA),
6) detekce outlierů.
Cílem práce je zmapovat možnosti programu R pro řešení popsaných úloh.
V aplikační části autor provede analýzu údajů z kontrol užitkovosti několika desítek tisíc dojnic. Cílem bude vysvětlit závislost dojivosti na dni laktačního cyklu, pořadí laktace, ročním období a dalších faktorech a také predikce dojivosti dojnice v případné další laktaci.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Řezánková, H., Húsek, D., Snášel, V.: Shluková analýza dat, Kamil Mařík-Profesional Publishing, Praha, 2007.
Hebák, P., Hustopecký, J., Malá, I.: Vicerozměrné statistické metody 3, Informatorium, Praha, 2005. Stowell, S.: Instant R: An Introduction to R for Statistical Analysis. Jotunheim Publishing, 2012. ISBN 978-0-957-46490-2.

Vedoucí bakalářské práce: **Mgr. Jaroslav Marek, Ph.D.**
Katedra matematiky a fyziky

Datum zadání bakalářské práce: **20. prosince 2013**
Termín odevzdání bakalářské práce: **9. května 2014**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 1. 5. 2014 Martina Kršíková

Poděkování

Tímto bych chtěla poděkovat vedoucímu mé bakalářské práce Mgr. Jaroslavu Markovi, Ph.D. za velkou ochotu a trpělivost, poskytnutí informací i cenných rad. Děkuji také všem, kteří mě podporovali při zpracování bakalářské práce i při studiu, především rodičům.

Anotace

Hlavním cílem bakalářské práce je popsat a předvést statistické metody a analýzy v programu R, které data mining využívá. Především má práce zmapovat možnosti programu R pro regresní a shlukovou analýzu. Tyto analýzy jsou demonstrovány na databázi kontrol užitkovosti dojnic. Dále byly odhadovány laktační křivky pomocí dvou různých modelů a získané výsledky graficky ilustrovány.

Klíčová slova

project R, popisná statistika, číselné charakteristiky náhodného výběru, statistická grafika, nelineární regrese, aproximace laktační křivky Gainesovou funkcí, Nelderovou funkcí, index determinace, mnohorozměrná statistická analýza, shluková analýza, časové řady

Title

Data mining and project R

Annotation

The main goal of this bachelor's thesis is describe and demonstrate statistical methods and analysis in the program R, which uses data mining. To map the possibilities of the program R for regression and cluster analyzes. These analyzes demonstrate on the database checks of dairy cattle. Find and estimate lactation curves for two different models and plot this curves.

Keywords

project R, descriptive statistics, numerical characteristics of random sampling, statistical graphics, non-linear regression, approximation of the lactation curve by Gaines function and Nelder function, index of determination, multivariate statistical analysis, cluster analysis, time series

OBSAH

SEZNAM ZKRATEK

GNU	General Public Licence
LaTeX	Lamport *τέχνη* (starořecké slovo znamenající umění nebo dovednost)

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Data mining je v poslední době velmi žádaná disciplína na rozhraní mnohorozměrné statistické analýzy a informatiky. Slouží k získávání užitečných informací z dat. Mezi nejpoužívanější nástroje dolování dat patří především regresní analýzy, rozhodovací stromy, neuronové sítě, shlukování, predikce v časových řadách a detekce outlierů. [1] Existuje spousta komerčního i nekomerčního softwaru určených pro data mining. Mezi nekomerční software patří například program RapidMiner, Weka, Orange nebo R.

Cílem této práce je: zmapovat možnosti data miningu v programu R, provádět statistické analýzy nad daty z kontrol užitkovosti dojnic, najít odhad laktační křivky a vysvětlit závislost dojivosti na dni laktačního cyklu. Práce slouží jako návod k programu R a jeho využívání ke statistickým analýzám. Neslouží však k podrobnému popisu všech datových typů a funkcí, které program R obsahuje. Pokud by měl čtenář zájem naučit se s tímto programem pracovat v celém rozsahu a zjistit vše, co program nabízí, tak může využít podrobný manuál, který je dostupný na oficiálních webových stránkách projektu R. Všechny použité obrázky, které jsou v této práci použity, jsou dílem autorky práce.

1 ZÁKLADNÍ CHARAKTERISTIKA

1.1 Project R

R je programovací jazyk a prostředí, ve kterém je možné provádět statistické výpočty a grafiku. Jedná se o volně šiřitelný software s GNU licencí. R je velmi podobný jazyku a prostředí S, který byl vyvinut v Bell Laboratories. Většina kódu napsaném v jazyce S by měla fungovat i v R, ovšem existuje zde i několik významných rozdílů. R poskytuje spoustu statistických a grafických technik a je snadno rozšiřitelný o metody vlastní. Snadno v něm lze vytvářet obrázky a grafy v profesionální kvalitě, do kterým jde jednoduše přidávat matematické vzorce a symboly.

K analýze dat nabízí prostředí R prostředky pro manipulaci a ukládání dat, operátory pro výpočty na polích a maticích, konzistentní a integrované prostředky pro analýzu dat. Dále jazyk R nabízí prostředky pro vstup a výstup dat, podmínky, cykly a uživatelem definované funkce. Výpočetně náročné operace je možné programovat např. v jazyce C nebo C++ a za běhu je připojit k R. Dále má R svůj vlastní formát pro tvorbu dokumentace, který je podobný LaTeXu.

1.2 Vstupní soubory

Pro svojí práci jsem se rozhodla používat dva databázové soubory, které jsem dostala k dispozici pro tuto práci, které obsahují data o užitkovosti krav. První soubor soubor_1.txt obsahuje informace o krávách, které byly vyřazeny v roce 2013. Druhý soubor soubor_2.txt obsahuje data o živých krávách. V souborech jsou data od roku 2006 až 2013.

Na následujícím obrázku (obrázek 1) je náhled souboru soubor_1.txt.

Soubor krav vyřazených v roce 2013.

Kráva	Kód	Stáj	Dat.nar.	Vyřazena	Otelena	Porl	Dat.kontr.	Litry	Tuk	Bílk	PSB
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	25.01.06	32.00	3.65	3.38	516.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	22.02.06	26.20	4.01	3.30	151.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	23.03.06	29.20	3.92	3.37	98.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	24.04.06	28.80	4.58	3.23	88.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	24.05.06	23.20	4.75	3.57	28.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	26.06.06	27.60	4.13	3.22	172.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	25.07.06	28.20	3.72	3.21	123.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	23.08.06	23.40	4.08	3.36	220.00
36257	544	1	01.12.1996	02.11.2006	17.09.2005	7	25.09.06	19.40	3.45	3.74	284.00

Obrázek 1 - Náhled souboru soubor_1.txt

2 STAŽENÍ A INSTALACE PROGRAMU R

Program R je možno nainstalovat na unixové operační systémy, MacOS a systémy Windows. Uvedeme postup pro instalaci v operačním systému Windows. Program je možné stáhnout na internetových stránkách projektu http://www.R-project.org. V levé části stránek je v sekci Download, packages odkaz CRAN. Po kliknutí na tento odkaz se nám zobrazí seznam oblastí, kde vybereme nejbližší např. jednu z Germany. Poté vybereme, ve kterém operačním systému chceme program R využívat. V našem případě použijeme odkaz Download R for Windows a následně Install R for first time a stáhneme soubor R-3.0.3-win.exe. Po spuštění tohoto souboru pokračujeme podle instrukcí instalace, dokud se nám program nenainstaluje a na ploše se neobjeví ikona s modrým R.

2.1 Balíky

Do programu je možné popřípadě doinstalovávat různé přídavné balíky. Ty se instalují příkazem:

```
> install.packages("Název balíku").
```

Spuštění nainstalovaného balíku se poté provádí tímto příkazem:

```
> library(Název balíku).
```

Seznam a popis všech dostupných balíků je umístěn na stejné internetové adrese, na které lze stáhnout program R.

3 IMPORT DAT

3.1 Import z txt formátu

Pro načítání obdélníkových dat z textových souborů se používá funkce *read.table*. Já pro import souboru soubor_1.txt použiji tento příkaz :

```
> soubor <- read.table("D:/Bakalarskaprace/soubor_1.txt",skip = 1,
header = TRUE).
```

Při zadávání cesty k souboru je třeba vždy používat normální lomítka, nikoli zpětná. Argument *skip* udává počet řádků na začátku souboru, jenž neobsahují žádnou datovou informaci, a které chceme přeskočit. Argument *header* udává, že před samotnými daty je řádek, který obsahuje jména jednotlivých proměnných. Stejným způsobem si importuji i soubor_2.txt.

Pokud by datové položky nebyly odděleny mezerou, ale nějakým jiným oddělovačem např. středníkem, pak by se za argument *header* přidal další argument *sep = ";"*. V případě, že bychom v souboru neměli hlavičku s názvy proměnných a chtěli bychom je dodatečně pojmenovat použijeme následující příkaz:

```
> colnames(soubor1) <- c("Kráva", "Kód", "Stáj", "Dat.nar",
"Vyřazena", "Otelena", "Porl", "Dat.kontr.", "Litry", "Tuk",
"Bílk", "PSB").
```

Pokud bychom v souboru měli nějaká chybějící data, tak bychom do funkce *read.table* přidali argument *na.strings(" ").*Na prázdná místa by se přiřadila hodnota *NA.* Jestliže by chybějící data byla nějak označena např.znakem *x,* tak by argument vypadal takto: *na.strings("x").*

3.2 Import z Excelu

Pro import z Excelu existuje několik variant. První varianta je přes clipboard. Označí se matice dat z jednoho listu a uloží se do clipboardu (Ctrl + C). V programu R se data načtou pomocí funkce *read.delim*, která je určená pro data oddělená tabulátory:

```
> data<- read.delim(file = 'clipboard', head = TRUE, row.names = 1).
```

Argument *row.names* v tomto případě znamená, že první sloupec obsahuje názvy řádků. Další variantou načítání dat z Excelu je z formátu csv. Tento formát závisí na jazykové lokalizaci počítače. Pokud je nastavena anglická lokalizace, tak se jednotlivé proměnné oddělují čárkami

a desetinná čísla tečkami. Při české lokalizaci se proměnné oddělují středníky a desetinná místa čárkami. Import se provede pomocí funkce *read.table*:

```
> data <- read.table(file = 'D:/Dokumenty/soubor.csv', head =
TRUE, sep = ';',  dec = ',').
```

Argument *sep* označuje oddělovač a argument *dec* desetinnou čárku. Do argumentu *file* je možné zadat i internetový odkaz, který odkazuje na soubor csv. Dále je možné každý list z Excelu uložit do formátu txt a načíst data opět funkcí *read.table*. Pokud chceme importovat data ze souboru xls, tak je nutné nejprve si nainstalovat balíček XLConnect:

```
> install.packages("XLConnect"),

> library(XLConnect).
```

A následně pomocí funkce *readWorksheetFromFile* načíst pracovní list souboru.

```
> data <- readWorksheetFromFile(file = 'D:/Dokumenty/soubor.xls',
sheet = 1, header = TRUE)
```

Argument *sheet* značí číslo listu, který chceme načíst.

4 INFORMACE O NAČTENÝCH DATECH

Nyní máme načtená data v datových tabulkách *soubor* a *soubor2*. Pokud chceme zjistit počet řádků nebo sloupců použijeme funkce *nrow()* nebo *ncol()*. Obdobnou funkcí je funkce *dim()*, která nám vrátí dimenzi tabulky nebo matice. Pro vypsání názvů sloupců je možné použít příkaz *names()*.

Na následujícím obrázku (obrázek 2) jsou ukázky jednotlivých příkazů.

```
> ncol(soubor)
[1] 12
> nrow(soubor)
[1] 54168
> dim(soubor)
[1] 54168    12
> names(soubor)
 [1] "Kráva"      "Kód"        "Stáj"       "Dat.nar."   "Vyřazena"   "Otelena"    "Porl"
```

Obrázek 2 - Ukázka příkazů *ncol()*, *nrow()*, *dim()* a *names()*

Dalším užitečným příkazem je příkaz *summary()*, který nám zobrazí informace o jednotlivých sloupcích datové tabulky. Především zobrazí minimální a maximální hodnotu, aritmetický průměr hodnot, medián, první a třetí kvantil.

Na následujícím obrázku (obrázek 3) je ukázka příkazu *summary()* nad maticí soubor.

```
> summary(soubor)
     Kráva             Kód             Stáj            Dat.nar.          Vyřazena
 Min.   : 36257   Min.   :205.0   Min.   :1.000   01.11.2003:  167   01.06.2010:  774
 1st Qu.:125073   1st Qu.:931.0   1st Qu.:1.000   08.11.2004:  161   13.03.2012:  766
 Median :182083   Median :931.0   Median :1.000   03.05.2004:  149   08.11.2011:  674
 Mean   :203169   Mean   :758.8   Mean   :1.352   01.04.2004:  137   09.04.2013:  665
 3rd Qu.:260504   3rd Qu.:931.0   3rd Qu.:2.000   18.03.2005:  132   05.12.2011:  633
 Max.   :470269   Max.   :931.0   Max.   :2.000   11.11.2004:  131   03.09.2009:  621
                                                  (Other)   :53291   (Other)   :50035
      Otelena           Porl          Dat.kontr.        Litry            Tuk
 11.08.2009:  108   Min.   : 0.000   26.06.06:  476   Min.   : 0.20   Min.   :0.000
 01.05.2007:  101   1st Qu.: 1.000   23.03.06:  475   1st Qu.:26.50   1st Qu.:2.830
 15.03.2008:   91   Median : 2.000   25.07.06:  473   Median :33.40   Median :3.480
 02.12.2009:   89   Mean   : 2.161   23.08.06:  469   Mean   :33.18   Mean   :3.075
 11.07.2006:   81   3rd Qu.: 3.000   20.03.08:  466   3rd Qu.:40.00   3rd Qu.:4.030
 14.09.2011:   81   Max.   :10.000   25.01.06:  466   Max.   :70.90   Max.   :8.290
 (Other)   :53617                    (Other) :51343
     Bílk             PSB
 Min.   :0.000   Min.   :    0.0
 1st Qu.:2.830   1st Qu.:   25.0
 Median :3.140   Median :   87.0
 Mean   :2.669   Mean   :  377.6
 3rd Qu.:3.380   3rd Qu.:  268.0
 Max.   :5.940   Max.   :20101.0
```

Obrázek 3 - Ukázka příkazu *summary()*

Pokud chceme zjistit datové typy jednotlivých sloupců, tak je vhodné použít příkaz *contents()*. Tento příkaz nám kromě datových typů vypíše i jednotlivé levely sloupců.

Na následujícím obrázku (obrázek 4) je ukázka příkazu *contents()*.

```
> contents(soubor)

Data frame:soubor       54168 observations and 12 variables    Maximum # NAs:0

           Levels Storage
Kráva              integer
Kód                integer
Stáj               integer
Dat.nar.     1760 integer
Vyřazena      725 integer
Otelena      2407 integer
Porl               integer
Dat.kontr.    195 integer
Litry               double
Tuk                 double
Bílk                double
PSB                 double

+----------+-------------------------------------------------------------------+
|Variable  |Levels                                                             |
+----------+-------------------------------------------------------------------+
|Dat.nar.  |01.01.2002,01.01.2003,01.01.2004,01.01.2006,01.01.2008,01.01.2009|
|          |01.01.2010,01.02.2007,01.03.2003,01.03.2005,01.03.2006,01.03.2008|
|          |01.04.2004,01.04.2006,01.04.2007,01.04.2011,01.05.2002,01.05.2004|
```

Obrázek 4 - Ukázka příkazu *contents()*

5 POPISNÁ STATISTIKA

Pomocí popisné statistiky se zpracovávají data ve formě grafů a tabulek a vypočítávají se jejich číselné charakteristiky jako např. aritmetický průměr. Popisná statistika se často věnuje určováním charakteristik polohy a variability, viz [3].

5.1 Charakteristiky (míry) polohy

5.1.1 Aritmetický průměr

Aritmetický průměr z hodnot $\{x_1, x_2, \dots, x_n\}$ se vypočítá podle následujícího vztahu:

$$\bar{x} = \frac{1}{n}\sum_{k-1}^{n} x_k.$$

Pokud máme různé hodnoty $\{z_1, z_k, \dots, z_m\}$, u kterých je známá četnost n_j, $j = 1, 2, \dots, m$ a relativní četnost p_j, tak se aritmetický průměr vypočítá podle následujícího vztahu:

$$\bar{x} = \frac{1}{n}\sum_{j-1}^{m} z_j n_j = \sum_{j-1}^{m} z_j p_j.$$

V programu R se aritmetický průměr vypočte pomocí funkce *mean()*. V následující ukázce (obrázek 5) je výpočet aritmetického průměru všech nadojených litrů z datové tabulky soubor. Výpočet provedeme od prvního řádku devátého sloupce, který obsahuje počet nadojených litrů, až po poslední řádek. Pro hodnotu posledního řádku využijeme funkci *nrow()*, která vrací počet řádků.

```
> mean(soubor[1,9]:soubor[nrow(soubor),9])
[1] 23
```

Obrázek 5 - Ukázka funkce *mean()*

5.1.2 Modus a medián

Modus je hodnota, která má v souboru dat největší četnost, tj. vyskytuje se nejčastěji. Nemusí být vždy jednoznačně určen jednou hodnotou, ale může nabývat i hodnot několik. Abychom data rozdělili do intervalů a určili interval s největší četností (modální interval), tak bychom modus stanovili pomocí interpolace sousedních intervalů podle následujícího vzorce:

$$\hat{x} = x_D + h\frac{n_2}{n_1+n_2},$$

kde: h - je délka modálního intervalu,

x_D - je dolní hranice modálního intervalu,

n_1 - je četnost předchozího intervalu,

n_2 - je četnost následujícího intervalu.

Medián je prostřední hodnota dat. Pokud bychom měli data seřazena vzestupně, tak medián bude hodnota, která by data rozdělila na dvě stejně velké skupiny. V lichém počtu dat by byl medián prostředním z nich. V sudém počtu dat by byl medián průměr ze dvou prostředních hodnot. V programu R se medián vypočte pomocí funkce *median()*. V následující ukázce (obrázek 6) je vypočtený medián z devátého sloupce datové tabulky soubor, který obsahuje počty nadojených litrů.

```
> median(soubor[1,9]:soubor[nrow(soubor),9])
[1] 23
```

Obrázek 6 - Ukázka funkce *median()*

5.1.3 Kvantily

Kvantil rozděluje hodnoty na dvě části. Používá se rozdělení pomocí percentilů. Medián je $\tilde{x}_{50}$ percentil, kde 50% hodnot je menších a 50% hodnot větších. Dále se používá dolní kvartil $\tilde{x}_{25}$, kde je 25% menších a 75% větších, nebo horní kvartil $\tilde{x}_{75}$. V programu R se kvantily vybraných hodnot zobrazí pomocí funkce *quantile()*. V následující ukázce (obrázek 7) je použita funkce *quantile()* pro devátý sloupec z datové tabulky soubor.

```
> quantile(soubor[1,9]:soubor[nrow(soubor),9])
  0%  25%  50%  75% 100%
14.0 18.5 23.0 27.5 32.0
```

Obrázek 7 - Ukázka funkce *quantile()*

5.2 Charakteristiky (míry) variability

5.2.1 Rozpětí

Variační rozpětí se počítá jako rozdíl mezi největší a nejmenší hodnotou:

$$R = x_{max} - x_{min}.$$

Výsledek tohoto výpočtu může být často ovlivněn extrémními hodnotami a může poskytnout zavádějící informace. Proto se někdy používá hodnot $\tilde{x}_{90} - \tilde{x}_{10}$, kdy se vynechá 10% nejmenších a největších hodnot. Další možností je vypočtení mezikvartilového rozpětí:

$$IQR = \tilde{x}_{75} - \tilde{x}_{25},$$

kde se pracuje jen s 50% středních hodnot. V programu R funkce *range()* vrací minimum a maximum hodnot. V následující ukázce (obrázek 8) je funkce *range()* nad hodnotami z devátého sloupce datové tabulky soubor.

```
> range(soubor[1,9]:soubor[nrow(soubor),9])
[1] 14 32
```

Obrázek 8 - Ukázka funkce *range()*

5.2.2 Směrodatná odchylka

Střední kvadratická odchylka je průměr čtverců odchylek od průměru. Čím je jeho hodnota vyšší, tím se údaje více odchylují od průměru. Vypočítá se následovně:

$$s^2 = \frac{1}{n}\sum_{i-1}^{n}(x_i - \bar{x})^2.$$

Výběrový rozptyl je možné vypočítat takto:

$$s^2 = \frac{1}{n-1}\sum_{i-1}^{n}(x_i - \bar{x})^2.$$

Směrodatná odchylka je odmocninou výběrového rozptylu. Pro výpočet směrodatné odchylky v programu R slouží funkce *sd()*. V následující ukázce (obrázek 9) je výpočet směrodatné odchylky devátého sloupce datové tabulky soubor.

```
> sd(soubor[1,9]:soubor[nrow(soubor),9])
[1] 5.627314
```

Obrázek 9 - Ukázka funkce *sd()*

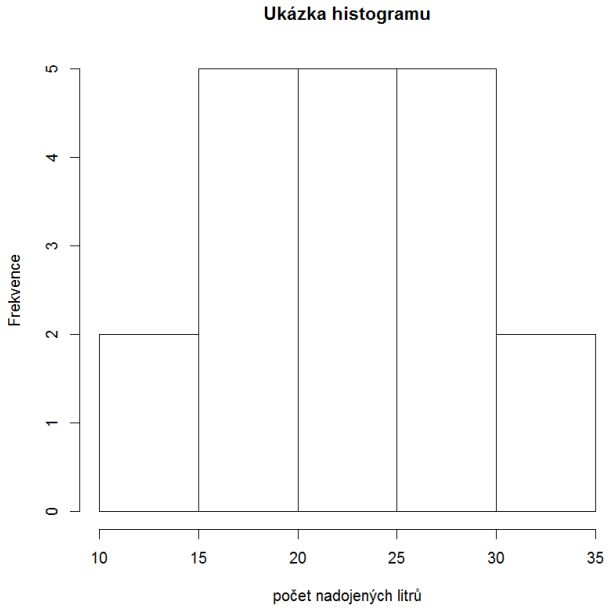
6 STATISTICKÁ GRAFIKA

V této kapitole bude představeno několik základních grafických nástrojů a jejich implementace v programu R, které se využívají v data miningu.

6.1 Histogram

Histogram je sloupcový graf, který zachycuje četnost sledované veličiny v daném intervalu. Program R poskytuje pro vykreslení histogramu funkci *hist()*.Následující příkaz představuje ukázku vytvoření histogramu a následující obrázek (obrázek 10) výsledný histogram.

```
> hist(soubor[1,9]:soubor[nrow(soubor),9], main = "Ukázka
histogramu", xlab = "početnadojenýchlitrů", ylab = "Frekvence")
```



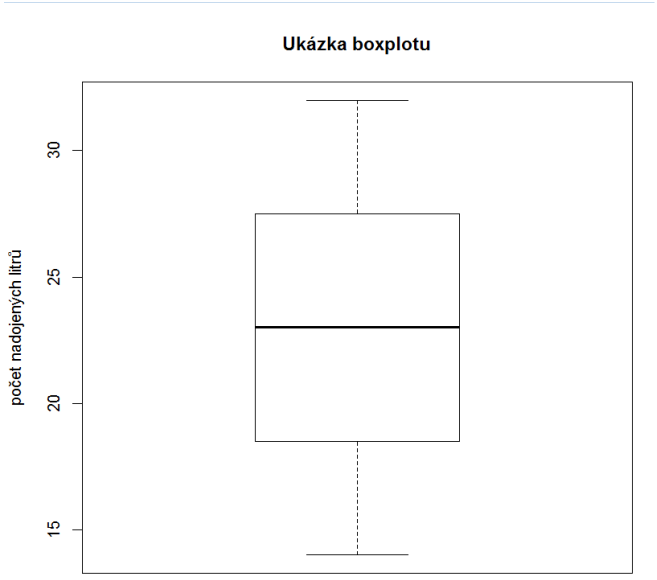
Obrázek 10 - Histogram průměru nadojených litrů

6.2 Boxplot

Boxplot je krabicový graf, který zobrazuje data pomocí jejich kvartilů. Střední část vymezuje medián, který je zespoda ohraničen prvním a z vrchu třetím kvartilem. Variabilita dat pod prvním a třetím kvartilem je vyjádřena liniemi, které vycházejí ze střední části. Pro vykreslení

boxplotu v programu R slouží funkce *boxplot()*. Následující příkaz představuje ukázku pro vytvoření boxplotu a následující obrázek (obrázek 11) výsledný boxplot.

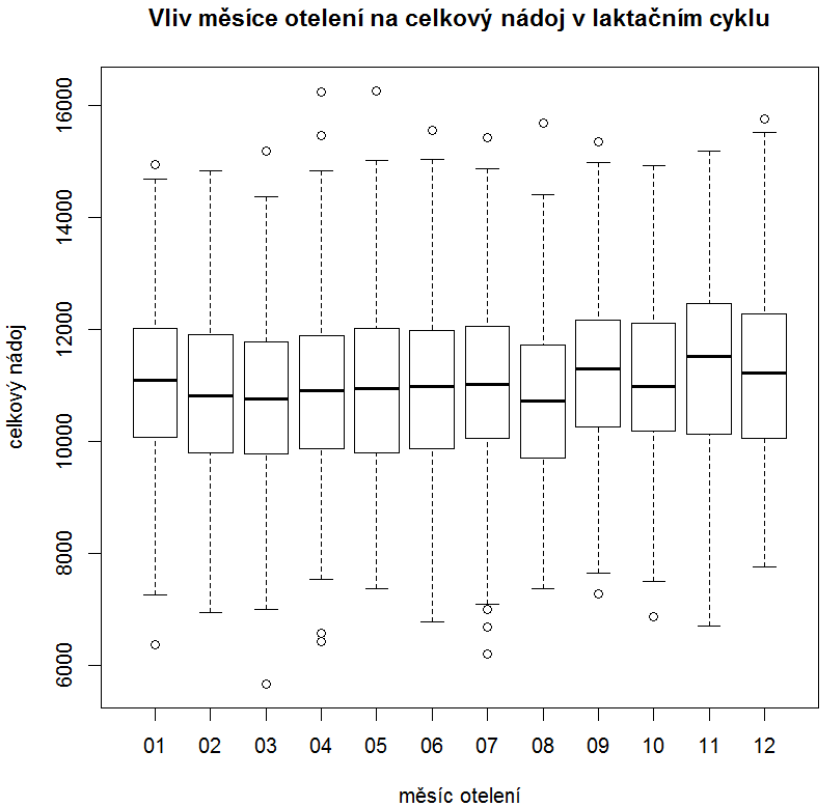
```
> boxplot(soubor[1,9]:soubor[nrow(soubor),9], main = "Ukázka
boxplotu", ylab = "početnadojenýchlitrů")
```



Obrázek 11 - Boxplot nadojených litrů

V programu R je datový typ factor, který vypadá jako vektor, a který navíc obsahuje levely jednotlivých položek. Pokud by jsme například měli factor obsahující měsíce otelení všech dojnic, tak by levely tohoto factoru byla čísla měsíců. Kdybychom takový factor chtěli vykreslit pomocí funkce *plot()*, tak by se vykreslil pomocí boxplotů, kde by hodnoty na ose *x* byli položky levelů. Pokud tedy máme factor obsahující měsíce otelení dojnic a vektor celkových nádojů v laktačním cyklu po otelení, tak pomocí následujícího příkazu vykreslíme boxploty (obrázek 12).

```
> plot(factorOteleni ,celkNadoje , main = "Vliv měsíce otelení na
celkový nádoj v laktačním cyklu", xlab = "měsíc otelení", ylab =
"celkový nádoj")
```



Obrázek 12 - Boxploty vlivu celkového nádoje na měsíci otelení

Z obrázku (obrázek 12) je vidět, že měsíc otelení nemá na výslednou dojivost velký vliv, protože výsledky jsou hodně podobné.

7 REGRESE - LAKTAČNÍ KŘIVKA

U každé dojnice je možné popsat průběh laktace počtem nadojených litrů v závislosti na čase. Laktační proces probíhá 305 dní do doby, kdy se dojnice dostane do fáze zasušení a přestane dojit. Poté se organismus dojnice nechá v klidu a připravuje se na další laktační cyklus. Na počátku cyklu je fáze rozdoje, kdy lze pozorovat výrazný nárůst produkce mléka. Tato fáze trvá přibližně 50 dní, poté produkce začíná klesat. [2] V této kapitole budeme hledat odhad laktační křivky pomocí nelineární regrese.

7.1 Gainesova funkce - exponenciální model

Gainesova funkce má tento tvar: [4]

$$y = f_1(\boldsymbol{\beta}, t) = \beta_1 \cdot e^{-\beta_2 t},$$

kde: t - je v našem případě číslo dne, kdy bylo prováděno měření nadojených litrů, $0 < t \leq 305$

β_1^0, β_2^0 - je počáteční řešení.

Pro Gainesovu funkci můžeme v programu R vytvořit následující funkci (obrázek 13):

```
> gaines_function
function(beta,t){
n<-length(t)
y<-(1:n)*0
for (i in 1:n) {
y[i]<-beta[1]*(2.71828182846^(-beta[2]*t[i]))
}
y
}
```

Obrázek 13 - Gainesova funkce

Vstupními parametry jsou dva vektory. První parametr *beta* obsahuje počáteční řešení, které pro tuto funkci zvolíme hodnotami 45 a 0.002. Druhý parametr *t* obsahuje čísla dnů, kdy bylo prováděno měření nadojených litrů. Lokální proměnná *n* slouží k počtu měření, které se zjistí funkcí *length(t)*. Dále se vytvoří vektor *y*, který slouží i jako návratová hodnota funkce. Tento vektor zatím vyplníme hodnotami 0. Následně přes cyklus *for* od jedné do *n* tento vektor *y* vyplníme hodnotami vypočtenými pomocí vzorce Gainesovy funkce.

Dalším krokem je vytvoření matice ***X***, která obsahuje parciální derivace Gainesovi funkce.

$$\boldsymbol{X} = \frac{\partial f}{\partial \boldsymbol{\beta}'},$$

$$\boldsymbol{X} = \begin{pmatrix} e^{-\beta_2 t_1} & \beta_1 e^{-\beta_2 t_1} \cdot (-t_1) \\ \vdots & \vdots \\ e^{-\beta_2 t_n} & \beta_1 e^{-\beta_2 t_n} \cdot (-t_n) \end{pmatrix}.$$

Funkce pro vytvoření této matice $\boldsymbol{X}$ vypadá následovně (obrázek 14):

```
> maticeX_gainess
function(beta,t){
n=length(t)
P=c(2.71828182846^(-beta[2]*t),
 beta[1]*(2.71828182846^(-beta[2]*t))*(-t))
X <- array(P,c(n,2))
}
```

Obrázek 14 - Funkce pro vytvoření matice *X* Gainesovy funkce

V této funkci nejprve vytvoříme vektor, který naplníme vypočtenými hodnotami a následně ho pomocí funkce *array()* převedeme na matici o dvou sloupcích. Výsledná matice vypadá takto (obrázek 15):

```
> X
             [,1]       [,2]
 [1,] 0.9627129  -823.1196
 [2,] 0.9030296 -2072.4528
 [3,] 0.8504412 -3099.8582
 [4,] 0.7961243 -4084.1175
 [5,] 0.7512626 -4834.3749
 [6,] 0.7089289 -5487.1099
 [7,] 0.6636503 -6122.1736
 [8,] 0.6518114 -6276.9439
 [9,] 0.6138529 -6740.1045
[10,] 0.5792622 -7116.2365
[11,] 0.5433509 -7457.4907
```

Obrázek 15 - Matice plánu *X* u zvolené dojnice a pro Gainesovu funkce

Dále je nutné vypočítat parametry $\delta\boldsymbol{\beta}$ podle následujícího vzorce:

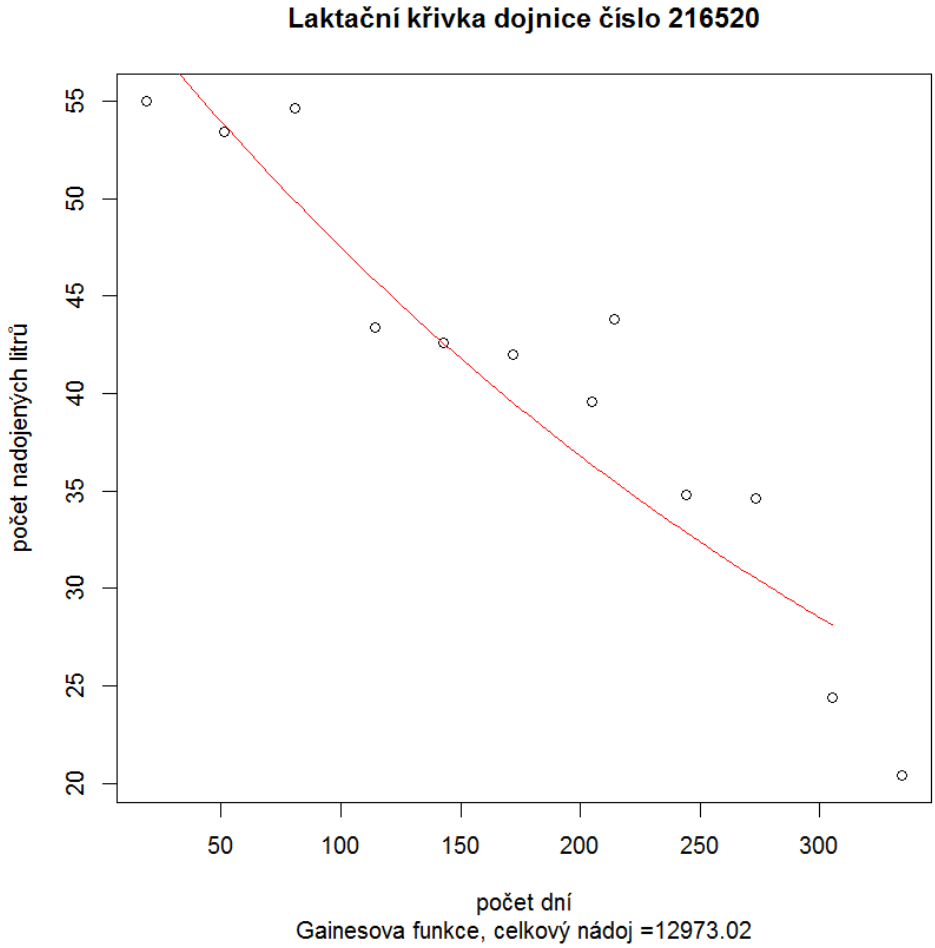
$$\delta\boldsymbol{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}.$$

Příkaz pro tento výpočet vypadá následovně:

```
> solve((t(X)%*%X))%*%t(X)%*%(Y1-Y0).
```

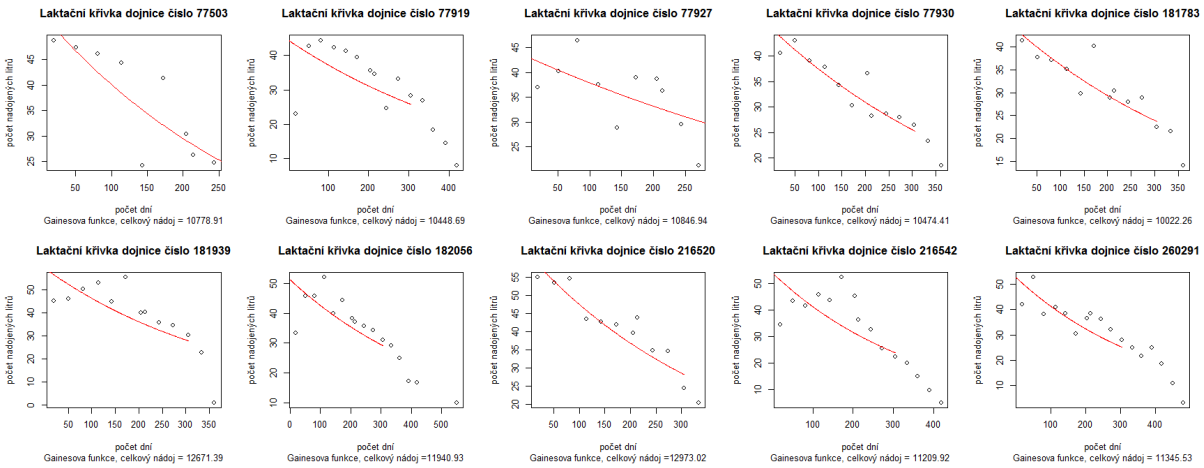
Pro maticové násobení se v programu R používá operátor %*% a pro transponovanou matici funkce *t()*.

Výsledná aproximace vybrané dojnice ve třetím laktačním cyklu pomocí Gainesovy funkce je znázorněna na obr. 16.



Obrázek 16 - Aproximace Gainesovovou funkcí

Pro 10 vybraných dojnic ve třetím laktačním cyklu vypadají křivky následovně (obrázek 17):



Obrázek 17 - Aproximace Gainesovovou funkcí pro 10 dojnic

7.2 Nelderova funkce - inverzní polynomiální křivka

Nelderova funkce má tento tvar: [4]

$$y = f_2(\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, t) = \frac{t}{\beta_0 + \beta_1 t + \beta_2 t^2},$$

kde: t - je v našem případě číslo dne, kdy bylo provedeno měření nadojených litrů, $0 < t \leq 305$

$\beta_0^0, \beta_1^0, \beta_2^0$ - je počáteční řešení.

Nelderovu funkci můžeme v programu R napsat takto (obrázek 18):

```
> nelder_function
function(beta,t){
n<-length(t)
y<-(1:n)*0
for (i in 1:n) {y[i]<-t[i]*(beta[1]+beta[2]*t[i]+beta[3]*t[i]*t[i])^(-1)
}
y

}
```

Obrázek 18 - Nelderova funkce

Parametry funkce jsou dva vektory. První vektor obsahuje tři proměnné *Beta*, které jsou nejprve vlastním odhadem řešení, jenž se v průběhu výpočtu zpřesní. Pro odhad řešení $\beta_0$zvolíme hodnoty 0.0949, 0.0174 a 0.0001. Druhý vektor t obsahuje čísla dnů, kdy bylo prováděno měření nadojených litrů. Stejně jako u Gainesovy funkce slouží lokální proměnná *n* k počtu měření, které se zjistí příkazem *length(t).* Proměnná y, který slouží jako návratová hodnota funkce. Tento vektor zatím vyplníme hodnotami 0. Následně přes cyklus *for* od jedné do *n*, vektor y vyplníme hodnotami vypočtenými pomocí Nelderovy funkce.

Dalším krokem je vytvoření matice $\boldsymbol{X}$. Ta by měla vypadat následovně:

$$\boldsymbol{X} = \frac{\partial f}{\partial \boldsymbol{\beta}},$$

$$\boldsymbol{X} = \begin{pmatrix} \frac{-t_1}{(\beta_0+\beta_1 t_1+\beta_2 {t_1}^2)^2} & \frac{-{t_1}^2}{(\beta_0+\beta_1 t_1+\beta_2 {t_1}^2)^2} & \frac{-{t_1}^3}{(\beta_0+\beta_1 t_1+\beta_2 {t_1}^2)^2} \\ \vdots & \ddots & \vdots \\ \frac{-t_n}{(\beta_0+\beta_1 t_n+\beta_2 {t_n}^2)^2} & \frac{-{t_n}^2}{(\beta_0+\beta_1 t_n+\beta_2 {t_n}^2)^2} & \frac{-{t_n}^3}{(\beta_0+\beta_1 t_n+\beta_2 {t_n}^2)^2} \end{pmatrix}.$$

Pro vytvoření této matice můžeme v programu R napsat tuto funkci (obrázek 19):

```
> maticeX_function
function(beta,t){
n=length(t)
P=c(-t/(beta[1]+beta[2]*t+beta[3]*t*t)^2,
-t^2/(beta[1]+beta[2]*t+beta[3]*t^2)^2,
-t^3/(beta[1]+beta[2]*t+beta[3]*t^2)^2)
X <- array(P,c(n,3))
}
```

Obrázek 19 - Funkce pro vytvoření matice *X*

V této funkci nejprve vytvoříme vektor *P*, který naplníme příslušnými hodnotami. Ten pak následně pomocí funkce *array()* převedeme na matici o *n* řádkách a 3 sloupcích. Výsledná matice vypadá takto (obrázek 20):

```
> X
             [,1]       [,2]       [,3]
 [1,] -89.170664 -1694.2426  -32190.61
 [2,] -33.040552 -1685.0681  -85938.48
 [3,] -17.354683 -1405.7293 -113864.07
 [4,]  -9.989870 -1138.8452 -129828.35
 [5,]  -6.676507  -954.7406 -136527.90
 [6,]  -4.705197  -809.2939 -139198.54
 [7,]  -3.314535  -679.4798 -139293.35
 [8,]  -3.034252  -649.3300 -138956.62
 [9,]  -2.302571  -561.8274 -137085.88
[10,]  -1.834435  -497.1318 -134722.71
[11,]  -1.221371  -396.9456 -129007.32
```

Obrázek 20 - Matice *X*

Parametry $\boldsymbol{\beta}$ se pomocí metody nejmenších čtverců vypočtou takto:

$$\boldsymbol{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y},$$

kde: $\boldsymbol{X}$ - je naše matice,

$\boldsymbol{Y}$ - je vektor s naměřenými hodnotami.

Příkaz pro tento výpočet vypadá následovně:

```
> solve((t(X)%*%X))%*%t(X)%*%(Y1-Y0).
```

Pro maticové násobení se v programu R používá operátor %*% a pro transponovanou matici funkce *t()*. Toto vypočtené β se pro další použití přičte k původnímu odhadu β_0.

Pro vykreslení funkce se použije následujících příkazů:

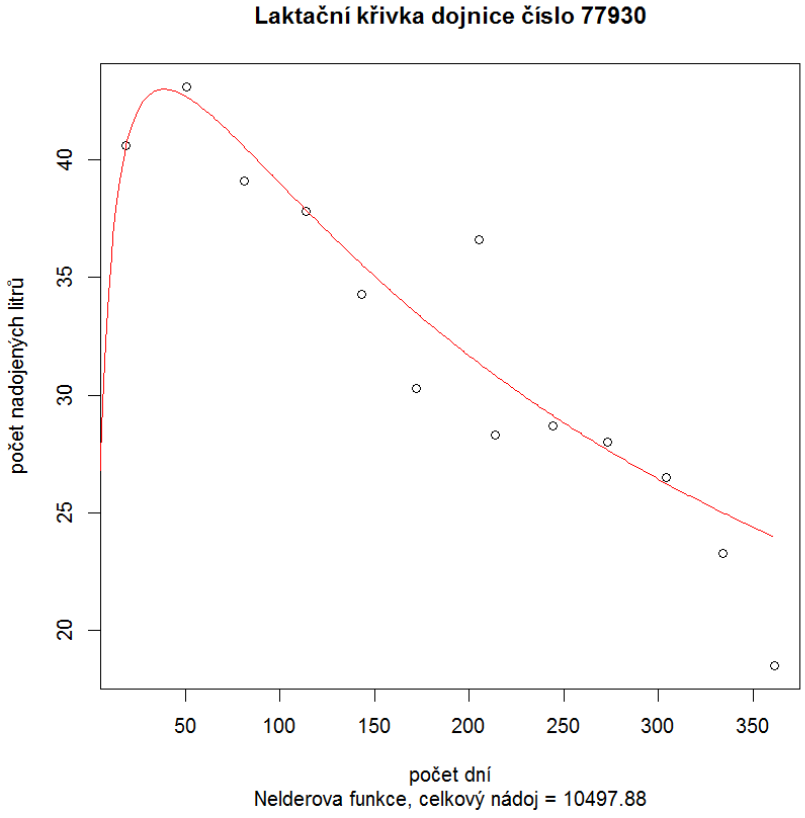
```
> Ygraf<- nelder_function(beta,c(1:305)),

> plot(t, Y, type = "p", main = "Nelder function", sub = "Laktační
křivka 305 dní", xlab = "početdní", ylab = "početnadojených
litrů"),

> lines(t, Ygraf, type = "l", col = "red").
```

Příkaz *plot* s parametrem *type = "p"* vykreslí body znázorňující výsledky měření nadojených litrů vybrané dojnice. Parametr *main* slouží k hlavnímu nadpisu grafu a *sub* k podnadpisu. Parametry *xlab* a *ylab* slouží k popiskům os grafu. Příkazem *lines* se do grafu přikreslí linie funkce. Parametr *type = "l"* slouží k vykreslení jednoduché čáry a parametr *col* k barvě.

Výsledný graf vypadá takto (obrázek 21):



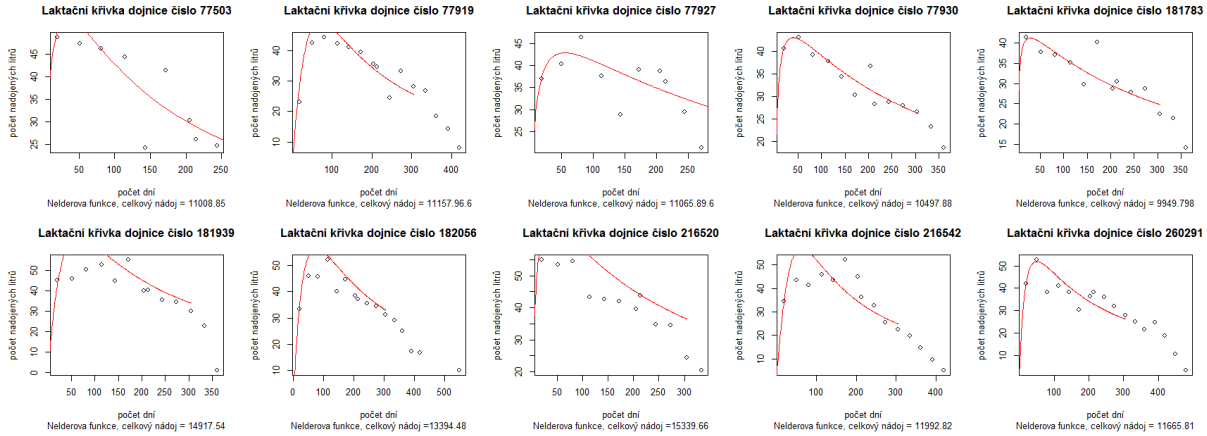
Obrázek 21 - Aproximace Nelderovou funkcí

Pro vytvoření více grafů do jednoho obrázku slouží v programu R funkce *par()*, která má jako argument rozměr matice výsledných grafů. Příklad příkazu této funkce vypadá takto:

```
> par(mfrow=c(2, 5)).
```

Pokud za tento příkaz budeme vytvářet jednotlivé grafy pomocí příkazu *plot()*, tak se nám budou vykreslovat do matice o dvou řádkách a pěti sloupcích.

Na dalším obrázku (obrázek 22) jsou vykresleny laktační křivky po třetím otelení 10 různých dojnic.



Obrázek 22 - Aproximace Nelderovou funkcí pro 10 dojnic

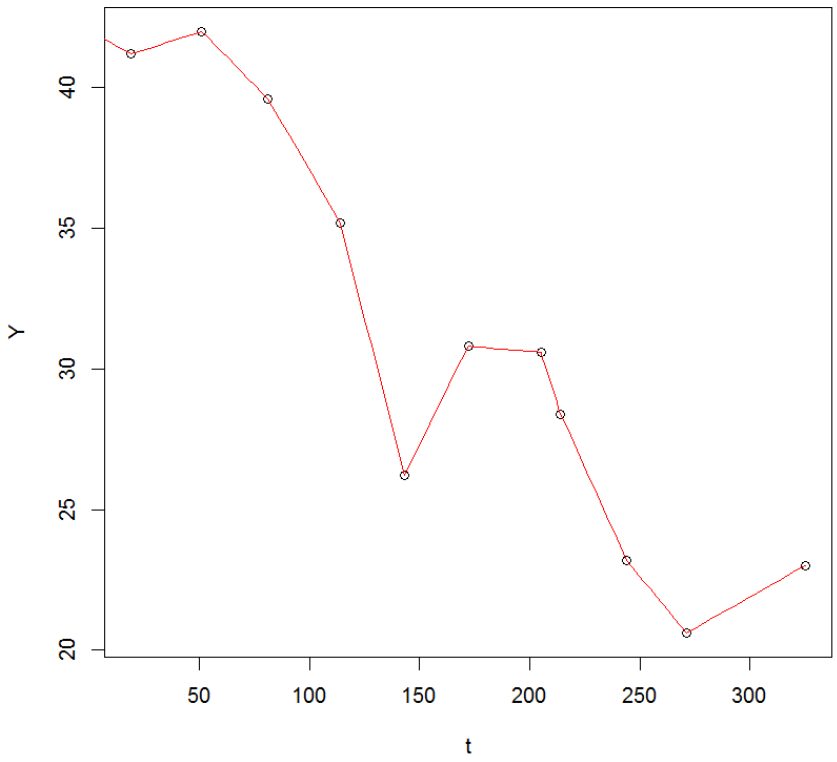
7.3 Polygon

Průběh laktační křivky je možné vyjádřit polygonem tzn. spojením rovných úseček mezi body jednotlivých měření. Na následujícím obrázku (obrázek 23) je funkce, která vykreslí jednotlivé úsečky mezi naměřenými body a vypočítá celkový nádoj laktačního cyklu.

```
> polynom_function
function(Y,t){
soucet<-0
n<-length(t)-1
plot(t,Y,"p")
for(i in 1:n){
s<-seq(from=Y[i],to=Y[i+1],length=(t[i+1]-t[i]))
soucet<-soucet+sum(s)
Cas<-seq(from=t[i],to=t[i+1],length=11)
usecka<-seq(from=Y[i],to=Y[i+1],length=11)
lines(Cas,usecka, type="l", col="red")
}
s<-seq(from=Y[2],to=Y[1],length=t[1])
soucet<-soucet+sum(s)
Cas<-seq(from=0,to=t[1],length=11)
usecka<-seq(from=Y[2],to=Y[1],length=11)
lines(Cas,usecka, type="l", col="red")
soucet
}
```

Obrázek 23 - Funkce pro vykreslení polygonu

Vstupními parametry funkce jsou dva vektory Y a t. První vektor obsahuje výsledky měření jedné dojnice v laktačním období a druhým vektor obsahuje čísla dnů, kdy bylo měření prováděno. Pomocí funkce *plot()* se nejprve vykreslí body s hodnotami jednotlivých měření. Následně se v cyklu provede výpočet spojující úsečky a vykreslí je. Zároveň se vypočítává celkový nádoj litrů za laktační cyklus, který se ukládá do proměnné *soucet*. Tato proměnná je i návratovou hodnotou funkce. Výsledný polygon vypadá následovně (obrázek 24)



Obrázek 24 - Polygon

8 INDEX DETERMINACE

Index determinace je číslo v intervalu $\langle 0,1 \rangle$, které udává kvalitu regresního modelu. Většinou se udává v procentech a značí kolik procent rozptylu vysvětlované proměnné je vysvětleno a kolik zůstalo nevysvětleno. Hodnoty blízké nule říkají, že má regresní model špatnou kvalitu a naopak hodnoty blízké jedné říkají, že je model kvalitní. Index determinace je možné vypočítat dle následujícího vzorce:

$$I = \sqrt{\frac{\sum_{i=0}^{n}(\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}},$$

kde: Y_i - jsou naměřené hodnoty,

$\hat{Y}_i$ - jsou hodnoty vypočtené pomocí funkce např. Nelderovy,

$\bar{Y}$ - je průměr naměřených dat.

Pro výpočet indexu determinace si v programu R můžeme vytvořit následující funkci (obrázek 25):

```
> index_determinace
function(Y,Y0,P){
n<-length(Y)
a<-0
b<-0
for(i in 1:n){
a<-a+(Y0[i]-P)^2
b<-b+(Y[i]-P)^2
}
I<-sqrt(a/b)
I
}
```

Obrázek 25 - Funkce pro výpočet indexu determinace

Funkce má tři vstupní parametry potřebné k výpočtu. Obsahuje tři lokální proměnné n, a, b. Parametr n obsahuje počet hodnot vstupního vektoru Y a slouží k počtu opakování cyklu, ve kterém se následně vypočtou hodnoty parametrů a a b dle předchozího vzorce. Výstupní proměnnou je proměnná I, jenž obsahuje odmocninu parametrů a a b.

Následující tabulka (tabulka 1) znázorňuje indexy determinace dvou použitých regresních funkcích u 10 dojnic.

Tabulka 1 - indexy determinace jednotlivých funkcí

Číslo krávy	Nelderova funkce	Gainessova funkce
77503	0,5686	0,4994
77919	0,7070	0,6660
77927	0,8218	0,7765
77930	0,9653	0,9290
181783	0,9626	0,8548
181939	0,4415	0,4175
182056	0,6196	0,6386
216520	0,4694	0,4544
216542	0,5423	0,5078
260291	0,6491	0,6423

9 LINEÁRNÍ REGRESE - VÝVOJ DOJIVOSTI

Vývoj dojivosti je možné vykreslit pomocí přímky. K vykreslení je třeba vektor obsahující data narození všech dojnic a příslušný vektor, který obsahuje celkový nádoj v prvním laktačním cyklu. Pro výpočet lineární regrese musíme vypočítat matici ***X***, která bude obsahovat parciální derivace lineární funkce. V tomto případě bude matice vypadat tak, že v první sloupec bude obsahovat 1 a ve druhém sloupci budou jednotlivé roky narození dojnic. Dále je třeba pomocí matice ***X*** vypočítat parametr β, který je možné vypočítat následovně:

```
> beta <- solve((t(X) %*% X)) %*% t(X) %*% nadoje,
```

kde parametr *nadoje* je vektor, který obsahuje všechny nádoje, operátor *%*%* slouží pro maticové násobení a funkce *t()* pro transponovanou matici. Protože v grafu budeme chtít vykreslit i horní a dolní mez tak je musíme vypočítat společně s přímkou. Pro výpočty budeme potřebovat vypočet sumy, pro který si můžeme napsat následující funkci (obrázek 26).

```
> mojesuma
function(rada){

n<-length(rada)
soucet<-0;
for (i in 1:n){
soucet<-soucet+rada[i]
}
soucet

}
```

Obrázek 26 - Funkce pro výpočet sumy

Pro výpočet přímky, horní a dolní meze použijeme následujících příkazů [3]:

```
> Se <- mojesuma(((X%*%beta) - nadoje)^2),

> SY <-mojesuma((nadoje - mean(nadoje))^2),

> skvadrat <- Se/( length(nadoje) - 2),

> s <- sqrt(skvadrat),

> cislo <- mojesuma((dataNarozeni - mean(dataNarozeni))^2),

> for(k in 1:length(datumNarozeni)){,
```

```
> Td[k] <- beta[1] + beta[2] * cas - s * 1.9609 * sqrt(1 /
length(nadoje) + (cas[k] - mean(nadoje))^2 / cislo),

> Th[k] <- beta[1] + beta[2] * cas + s * 1.9609 * sqrt(1 /
length(nadoje) + (cas[k] - mean(nadoje))^2 / cislo),

> primka[k] <- beta[1] + beta[2] * dataNarozeni},
```

kde pomocí směrodatné odchylky vypočteme jednotlivé meze a následně pomocí funkcí *plot()* a *lines()* vše vykreslíme.

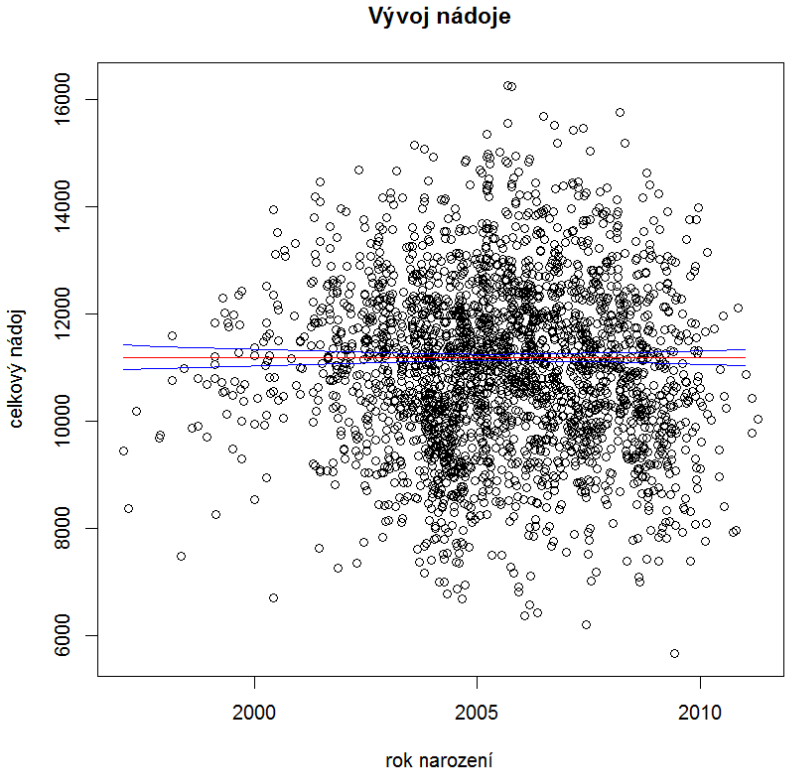
```
> plot(dataNarozeni, nadoje, xlab = "rok narození", ylab = "celkový
nádoj", main = "Vývoj nádoje"),

> lines(dataNarozeni, Td, type = "l", col = "blue"),

> lines(dataNarozeni, Th, type = "l", col = "blue"),

> lines(dataNarozeni, primka, type = "l", col = "red").
```

Na následujícím obrázku (obrázek 27) je výsledný graf, na kterém je vidět, že vývoj je téměř konstantní.



Obrázek 27 - Vývoj nádoje

10 SHLUKOVÁ ANALÝZA

Pro shlukovou analýzu využijeme balíček clustergas, který obsahuje shlukové metody na bázi genetických algoritmů. Instalace balíčku je popsaná v kapitole 2. Nejprve si vytvoříme matici, která obsahuje tyto položky: číslo krávy, celkový nádoj, pořadí laktace a kvartál otelení. Tuto matici je třeba převést na matici vzdáleností dle následujícího příkazu:

```
> D <- as.matrix(daisy(Mat, metric = "euclidean", stand = TRUE)).
```

Funkce *as.matrix(daisy())* vytvoří matici vzdáleností dle následujícího vzorce:

$$D_{i,j} = \sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2.$$

V dalším kroku použijeme funkci *agnes.gas()* z balíčku clustergas. Tato funkce spustí genetický algoritmus pro vytváření shluků a vytvoří dendrogramy [6]. Příkaz pro použití této funkce vypadá následovně:

```
> dendro<- agnes.gas(D, 10, 4).
```

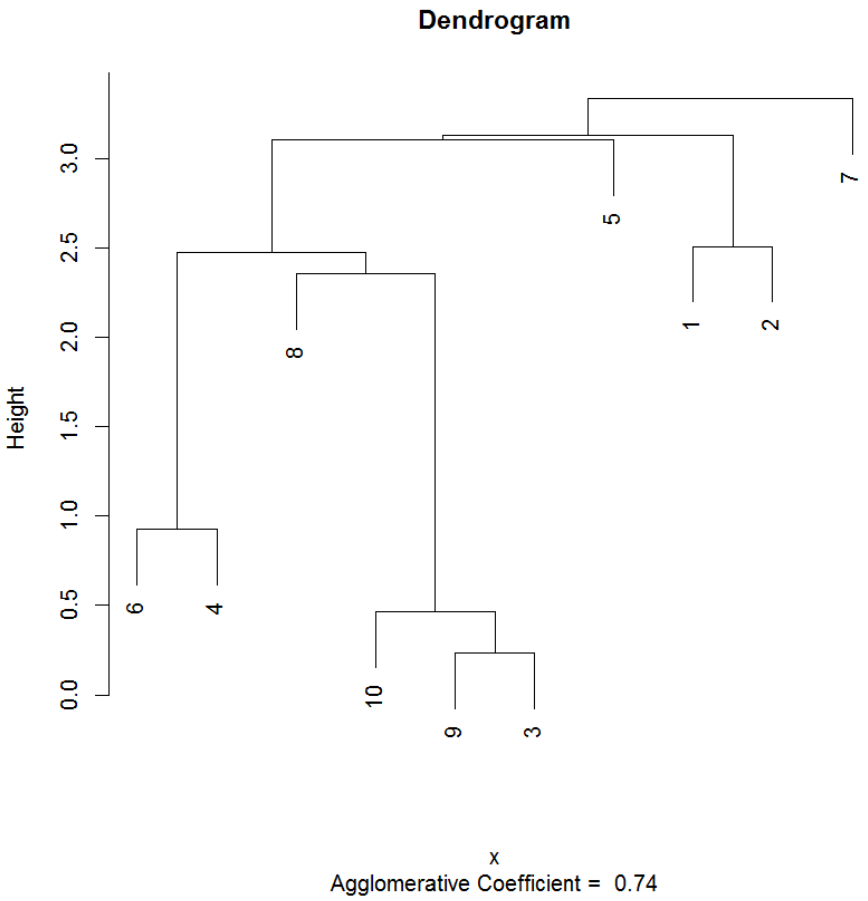
Parametr s hodnotou 10 udává počet shluků, které chceme vytvořit a parametr s hodnotou 4 udává kolik iterací chceme použít. Dále z dendrogramu vytvoříme graf pomocí funkce *dendogram.graph()* využitím následujícího příkazu:

```
> agn<- dendrogram.graph(complete.tree(dendro), Mat).
```

Výsledný graf vykreslíme pomocí funkce *plot():*

```
> plot(agn,main="Dendrogram").
```

Na následujícím obrázku (obrázek 28) je vykreslený dendrogram.



Obrázek 28 - Dendrogram

Na výsledném dendrogramu je vidět, že nejvíce si jsou podobné 3. a 9. dojnice, 6. a 4. dojnice a 1. a 2. dojnice. Podobné dojnice nadojili podobné množství litrů za stejné laktační období.

11 LOGISTICKÁ REGRESNÍ ANALÝZA

Tato analýza se zabývá problematikou odhadu pravděpodobnosti závislé proměnné na základě určitých událostí ovlivňujících výskyt jevu. Modeluje se náhodná veličina, která nabývá hodnoty 1, pokud jev nastal a hodnoty 0, pokud jev nenastal. [5] Pro tuto analýzu máme k dispozici seznam operovaných dojnic a informaci po kolika dnech od operace byly jednotlivé dojnice vyřazeny. Z celkového počtu 110 operovaných dojnic bylo již 100 vyřazeno. Nejprve je třeba načíst počty dnů od operace do vyřazení do jednoho vektoru. A následně je vzestupně seřadit. Pro seřazení je možné využít následující příkaz:

```
> Vyrazeni<- sort(vyrazeni).
```

Dále vytvoříme vektor, který bude obsahovat pro každý den po operaci počet nevyřazených krav. Tento vektor bude obsahovat tolik hodnot, kolik je nejvyšší hodnota z vektoru vyřazení. Na následujícím obrázku (obrázek 29) je funkce, která vytvoří požadovaný vektor.

```
> pocet_function
function(vyrazeni){
m<-max(vyrazeni)
pocet<-(1:m)*0
p<-110
j<-1
for (i in 1:m){
pocet[i]=p
if (i>=Vyrazeni[j]){
j<-j+1
p=p-1
pocet[i]=p
}
}
pocet
}
```

Obrázek 29 - Funkce pro vytvoření vektoru, obsahujícího počty nevyřazených dojnic

Do proměnné *m* se pomocí funkce *max()* načte nejvyšší hodnota. Poté se vytvoří požadovaný vektor o délce *m* a prozatím se vyplní hodnotami 0. Dále se v cyklu od 1 do *m* tento vektor naplní příslušnými hodnotami.

Dále potřebujeme vytvořit ještě jeden vektor, který bude mít stejnou délku jako vektor předešlý, a který bude obsahovat hodnoty počtu dní kdy byli dojnice vyřazeny. Pro vytvoření tohoto vektoru je možné použít funkci která je na následujícím obrázku (obrázek 30).

```
> vyrazeneDny_function
function(Vyrazeni){
m<-max(Vyrazeni)
v<-(1:m)*0
j<-1
for (i in 1:m){
if (i>=Vyrazeni[j])
{
j<-j+1
v[i]=Vyrazeni[j]
}else{
v[i]=Vyrazeni[j]
}
}
v
}
```

Obrázek 30 - Funkce pro vytvoření vektoru, obsahujícího počty dnů od vyřazení

Stejně jako v předešlé funkci se pomocí funkce *max()* načte nejvyšší hodnota dnu do proměnné *m*. Následně se vytvoří vektor *v* s délkou m naplněný hodnotami 0. V cyklu od jedné do *m* se tento vektor následně naplní příslušnými hodnotami.

Pro výpočet logistické regrese použijeme následující příkaz:

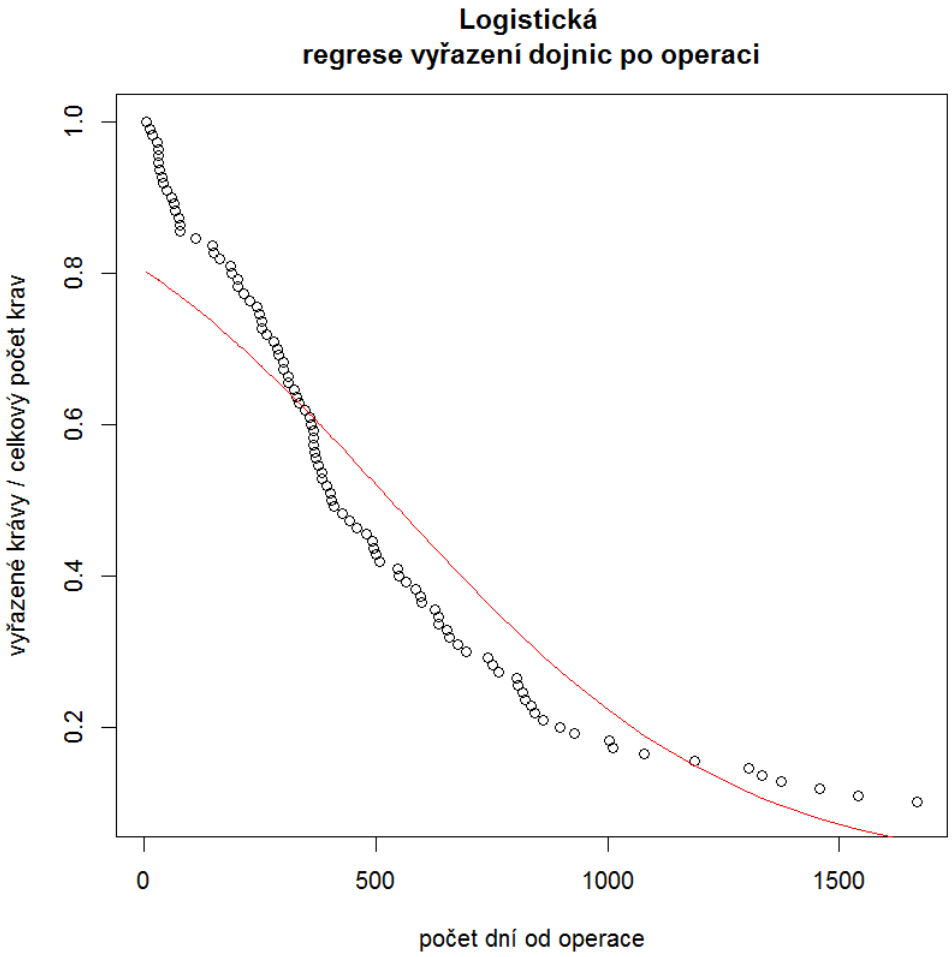
```
> glm.out<-glm(cbind(pocet, 110-pocet) ~ v,    family=
binomial(logit)),
```

kde do proměnné *glm.out* uložíme výsledek funkce *glm()*, která pokud obsahuje parametr *family = binomia(logit)* vypočítá logistickou regresi. Pro výsledné vykreslení použijeme následujících dvou příkazů:

```
> plot(pocet/110 ~ v, xlab="početdní od operace", ylab="vyřazené
krávy / celkovýpočetkrav", main="Logistickáregresevyřazení
dojnicpooperaci"),

> lines(v, glm.out$fitted, type="l", col="red").
```

Výsledná logistická regrese je vykreslena na následujícím obrázku (obrázek 31).



Obrázek 31 - logistická regrese

Výsledkem této analýzy je křivka vyjadřující pravděpodobnost vyřazení dojnice po operaci.

12 ČASOVÉ ŘADY

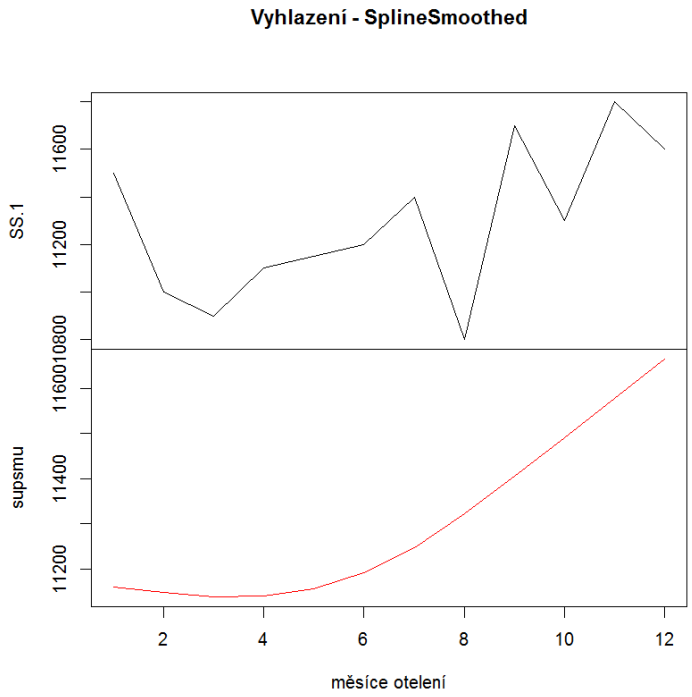
Existuje spousta metod a funkcí pro práci s časovými řadami. V této kapitole bude předvedeno vyhlazování funkcí. Pro tento účel využijeme balíček timeSeries, jehož instalace je popsaná v kapitole 2. Tento balíček obsahuje tři funkce pro vyhlazování. Jsou to funkce *smoothLowess(), smoothSpline()* a *smoothSupsm()*. Každá tato funkce používá trochu jiný vyhlazovací algoritmus. Použijeme zde funkci *smoothSupsm*(). V kapitole 6.2 jsme vykreslovaly boxploty dle měsíce otelení a nádoje. Pro vyhlazení zde použijeme hodnoty mediánů jednotlivých boxplotů, aby jsme viděli jakou křivku tvoří. Mediány si nejprve načteme do vektoru *mediany* a podle následujícího příkazu použijeme příslušnou funkci:

```
> i <- smoothSupsmu(as.timeSeries(mediany)).
```

Protože parametr funkce musí být objekt časové řady tak před proměnnou s načtenými hodnotami přidáme funkci *as.timeSeries*, která nám data příslušně upraví. Následně pomocí funkce *plot()* vykreslíme:

```
> plot(i, main = "Vyhlazení - SplineSmoothed")
```

Na následujícím obrázku (obrázek 32) je výsledné vyhlazení mediánů, ze kterého můžeme říci, že dojnice dojí více pokud jsou oteleny v pozdějším měsíci.



Obrázek 32 -Vyhlazení

ZÁVĚR

V práci byly popsány a předvedeny funkce z popisné statistiky a statistické grafiky. Byl proveden odhad laktační křivky pomocí Gainesovy a Nelderovy funkce, který splňoval předpokládaný průběh. Dále bylo poukázáno na průběžný vývoj v laktaci za posledních 18 let. Tento vývoj byl pomocí lineární regrese graficky znázorněn a ve výsledku nebyl tak výrazný jak se předpokládalo. Byla provedena shluková analýza, pro kterou byly využity funkce z přídavného balíčku clustergas, a která některé dojnice spojila podle podobnosti do shluků. Dále se provedla logistická regresní analýza, která znázorňuje pravděpodobnost vyřazení dojnic po operaci.

Práce v programu R je velmi příjemná a doporučila bych jeho využití těm, kdo by rád provedl statistické analýzy nad velkým množstvím dat. Je k dispozici velké množství přídavných balíčků, které obsahují spoustu funkcí pro data mining. Ke každému balíčku je možné stáhnout rozsáhlý manuál s popisem všech funkcí a ukázkových příkladů. Díky tomu, že program je poměrně populární, tak je možné na internetu získat mnoho cenných rad a shlédnout spoustu video tutoriálů.

SEZNAM POUŽITÉ LITERATURY

[1] BERKA, Petr. *Dobývání znalostí z databází*. Praha: Academia, 2003, 366 s. ISBN 80-200-1062 9.

[2] ŠIMONOVÁ, Jitka a Vojtěch ZINK. Mléčná žláza, průběh laktace a laktační křivka. *Agropress* [online]. 2014 [cit. 2014-05-01]. Dostupné z: http://www.agropress.cz/mlecna_zlaza_laktace.php.

[3] NEUBAUER, Jiří, Marek SEDLAČÍK a Oldřich KŘÍŽ. *Základy statistiky: Aplikace v technických a ekonomických oborech*. 1. vyd. Praha: Grada Publishing, a.s., 2012, 240 s. ISBN 978-80-247-4273-1.

[4] LEON - VELARDE, C. U., I. MCMILLAN, GENTRY a WILTON. *Journal of Animal Breeding and Genetics: Models for estimating typical lactation curves in dairy cattle*. Wiley, 1995. ISBN 112: 333–340 DOI: 10.1111/j.1439-0388.1995.tb00575.

[5] HEBÁK, Petr a Jiří HUSTOPECKÝ. *Vícerozměrné statistické metody 1*. Informatorium, 2006, 253 s. ISBN 978-80-7333-056-9.

[6] ŘEZANKOVÁ, Hana, Dušan HÚSEK a Václav SNÁŠEL. *Shluková analýza dat*. Praha: Professional Publishing, 2007, 196 s. ISBN 978-80-86946-26-9.