M5201 – 1. CVIČENÍ: *Práce s daty v R*

1 Načítání dat

Pro načítání dat máme v programovacím prostředí R celou řadu možností. Uveďme postupně některé z nich.

1.1 Načítání souboru ze schránky pomocí funkce `scan`

PŘÍKLAD 1

Mezi českými domácnostmi se zjišťovalo, jak jsou vybaveny předměty pro dlouhodobé užívání. My máme k dispozici údaje o tom, jaký podíl domácností (v procentech) vlastnil v letech 1993–2007 automatickou pračku a osobní počítač:

pračka 63,7; 65,6; 69,7; 72,5; 78,1; 80,1; 81,0; 84,1; 86,4; 88,7; 89,9; 91,2; 91,7; 91,9; 93,4
počítač 5,2; 5,3; 7,2; 8,5; 11,5; 13,6; 17,5; 21,4; 25,2; 28,4; 34,2; 39,7; 41,9; 48,6; 55,6

Data vždy nejprve označíme myší a zkopírujeme do schránky a následně je načteme pomocí funkce `scan`. Nejprve zkopírujeme data týkající se pračky a s využitím funkce `scan` je vložíme do proměnné `pracka`.

```
> pracka <- scan(file="clipboard", sep=";", dec=",")
```

Argumentem `sep=";"` specifikujeme, že jednotlivé hodnoty jsou odděleny středníkem a argumentem `dec=","` sdělujeme, že používáme desetinnou čárku (a nikoli desetinnou tečku, která se předpokládá implicitně).

Úplně stejně postupujeme i v případě dat týkajících se osobního počítače, opět použijeme funkci `scan` se stejnými argumenty a vytvoříme proměnnou `pocitac`:

```
> pocitac <- scan(file="clipboard", sep=";", dec=",")
>
```

Obě proměnné `pracka` a `pocitac` nyní spojíme do datového rámce (datové tabulky, datového souboru) pomocí funkce `data.frame`. Do prvního sloupce `rok` vložíme údaje o letech příslušejících jednotlivým údajům, do zbývajících dvou sloupců vložíme proměnné `pracka` a `pocitac`.

```
> vybaveni <- data.frame(rok = 1993:2007, pracka, pocitac)
```

Strukturu nově vytvořené proměnné `vybaveni` zjistíme příkazem `str`.

```
> str(vybaveni)
```

```
'data.frame':        15 obs. of  3 variables:
 $ rok    : int  1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 ...
 $ pracka : num  63.7 65.6 69.7 72.5 78.1 80.1 81 84.1 86.4 88.7 ...
 $ pocitac: num  5.2 5.3 7.2 8.5 11.5 13.6 17.5 21.4 25.2 28.4 ...


>
```

Začátek a konec proměnné `vybaveni` si prohlédneme pomocí příkazu `head`, resp. `tail`.

```
> head(vybaveni)
```

```
  rok pracka pocitac
1 1993   63.7     5.2
2 1994   65.6     5.3
3 1995   69.7     7.2
4 1996   72.5     8.5
5 1997   78.1    11.5
6 1998   80.1    13.6
```

```
> tail(vybaveni)
```

```
   rok pracka pocitac
10 2002   88.7    28.4
11 2003   89.9    34.2
12 2004   91.2    39.7
13 2005   91.7    41.9
14 2006   91.9    48.6
15 2007   93.4    55.6
```

V případě časových řad bývá často výhodnější nepracovat v R s datovými rámci (datový typ *data frame*), ale s datovým typem *time series*, protože mnoho funkcí pro práci časovými řadami vyžaduje právě tuto strukturu dat. K vytvoření datového typu *time series* je určena funkce `ts`.

Protože součástí datového typu time series je automaticky proměnná zachycující čas pozorování, použijeme z datového rámce `vybaveni` všechny proměnné mimo proměnnou `rok`. Čas pozorování zadáme pomocí argumentů `start = 1993` (čas prvního pozorování) a `frequency = 1` (udává, že se jedná o roční data; pro čtvrtletní bychom zadali `frequency = 4` a pro měsíční `frequency = 12`).

```
> vybaveniTS <- ts(vybaveni[,-1], start = 1993, frequency = 1)
```

Strukturu proměnné `vybaveniTS` opět zjistíme příkazem `str`.

```
> str(vybaveniTS)
```

```
 mts [1:15, 1:2] 63.7 65.6 69.7 72.5 78.1 80.1 81 84.1 86.4 88.7 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:2] "pracka" "pocitac"
 - attr(*, "tsp")= num [1:3] 1993 2007 1
 - attr(*, "class")= chr [1:2] "mts" "ts"
```

1.2 Čtení dat z datového souboru pomocí funkce `read.table`

PŘÍKLAD 2

Nyní pomocí funkce **read.table** načteme datový soubor **airtransport.txt**. V něm jsou uvedeny měsíční údaje o počtu osob přepravených leteckou dopravou v letech 2003–2009 v těchto čtyřech zemích: Rakousko, Mad'arsko, Slovensko a Česká republika.

Než provedeme samotné načtení dat ze souboru, připravíme si do proměnné **fileDat** řetězec popisující cestu k souboru. Předpokládejme, že v proměnné **data.library** je uložen řetězec popisující cestu k pracovnímu adresáři, v němž je soubor **airtransport.txt** uložen. Novou proměnnou **fileDat** vytvoříme spojením řetězce **data.library** a názvu souboru pomocí funkce **paste**.

```
> fileDat <- paste(data.library, "airtransport.txt" ,sep = "")
```

Před načtením dat je důležité zjistit, zda jsou na prvním řádku datového souboru uvedeny názvy proměnných či nikoliv. Náš soubor takovou hlavičku na prvním řádku obsahuje, a proto musíme jako argument funkce **read.table** uvést **header = TRUE**.

```
> airtrans <- read.table(fileDat, header = TRUE)
```

Strukturu načtených dat zjistíme příkazem **str** a začátek a konec datového souboru zobrazíme pomocí funkcí **head** a **tail**.

```
> str(airtrans)
```

```
'data.frame':        48 obs. of  5 variables:
 $ time          : Factor w/ 48 levels "2006M01","2006M02",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Hungary       : int  507429 466268 585366 674487 736358 796190 927454 945472 842612 741992 ...
 $ Austria       : int  1440668 1411654 1614514 1694636 1815870 1997481 2164619 2094164 2052305 1778166 ...
 $ Slovakia      : int  115749 115777 140641 152780 162739 225513 288783 294070 231163 162642 ...
 $ Czech.Republic: int  662733 627400 811225 971111 1053513 1252057 1387088 1423442 1324527 1098371 ...
```

```
> head(airtrans)
```

```
     time Hungary Austria Slovakia Czech.Republic
1 2006M01  507429 1440668   115749         662733
2 2006M02  466268 1411654   115777         627400
3 2006M03  585366 1614514   140641         811225
4 2006M04  674487 1694636   152780         971111
5 2006M05  736358 1815870   162739        1053513
6 2006M06  796190 1997481   225513        1252057
```

```
> tail(airtrans)
```

```
      time Hungary Austria Slovakia Czech.Republic
43 2009M07  868608 2203186   278036        1487917
44 2009M08  918750 2149481   309010        1508034
45 2009M09  829149 2052873   177093        1325999
46 2009M10  766245 1891020   127247        1068580
47 2009M11  577937 1584254   107888         859954
48 2009M12  548905 1632022   111194         854167
```

Z výpisu o struktuře objektu `airtrans` vidíme, že se jedná o datový rámec. Z něj opět pomocí funkce `ts` vytvoříme vícerozměrnou časovou řadu a znovu se podíváme na její strukturu. Protože data jsou měsíční, zvolíme možnost `frequency = 12`.

```
> airtransTS<-ts(airtrans[,-1], start = 2006, frequency = 12)
> str(airtransTS)
```

```
 int [1:48, 1:4] 507429 466268 585366 674487 736358 796190 927454 945472 842612 741992 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:4] "Hungary" "Austria" "Slovakia" "Czech.Republic"
 - attr(*, "tsp")= num [1:3] 2006 2010 12
 - attr(*, "class")= chr [1:2] "mts" "ts"
```

1.3 Čtení textových souborů pomocí funkce `readLines`

Občas se můžeme setkat s tím, že popis dat je uložen v jiném souboru než samotná data. V této situaci bývá nejvhodnější soubor obsahující popis dat načíst pomocí funkce `readLines`.

Příklad 3

Budeme načítat data ze souboru `tourism_israel.txt`. Popis dat je uložen v souboru `tourism_israel.inf`, což je textový soubor, který můžeme načíst funkcí `readLines`. Podobně jako v předchozíme případě vytvoříme řetězec `fileTxt`, v němž bude uložena cesta k souboru `tourism_israel.inf`. Tentokrát ale navíc použijeme funkci `file`, pomocí níž vytvoříme jakési spojení na tento soubor a tím ho zpřístupníme. Poté soubor načteme pomocí funkce `readLines`. Jestliže jakýkoli příkaz napíšeme do závorek, výsledek se automaticky zobrazí. Spojení se souborem `tourism_israel.inf` uzavřeme příkazem `close`.

```
> fileTxt <- paste(data.library, "tourism_israel.inf", sep = "")
> con <- file(fileTxt)
> (popis <- readLines(con))
```

```
 [1] "Ubytovaci sluzby v hotelech v Izraeli"
 [2] "======================================================="
 [3] "Datovy soubor obsahuje udaje o poctu ubytovanych turistu "
 [4] "v hotelech v Izraeli v obdobi leden 1991 -- brezen 2011."
 [5] "Udaje jsou uvedeny v tisicich."
 [6] "Soubor obsahuje tyto promenne (ve sloupcich):"
 [7] "[1] time -- cas pozorovani ve tvaru rok_mesic"
 [8] "[2] America -- pocet ubytovanych turistu z Ameriky"
 [9] "[3] Europe -- pocet ubytovanych turistu z Evropy"
[10] "[4] Other -- pocet ubytovanych turistu z ostatnich zemi"
```

```
> close(con)
```

Datový soubor `tourism_israel.txt` načteme funkcí `read.table`. Protože soubor neobsahuje v prvním řádku názvy proměnných, použijeme argument `header = FALSE`. Po načtení si prohlédneme strukturu dat.

```
> fileDat <- paste(data.library, "tourism_israel.txt", sep = "")
> turiste <- read.table(fileDat, header = FALSE)
> str(turiste)
```

```
'data.frame':        243 obs. of  4 variables:
 $ V1: Factor w/ 243 levels "1991-01","1991-02",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ V2: num  11.4 4.5 10.6 17.5 27.6 36.1 49.3 43.3 26.1 64.5 ...
 $ V3: num  13 8 24.2 46.1 58 ...
 $ V4: num  4.2 1.4 4.5 8.2 10.3 12.1 15.3 17.6 13.9 27.6 ...
```

Proměnné v datovém rámci pojmenujeme a podíváme se na začátek a konec datového souboru.

```
> names(turiste) <- c("time", "America", "Europe", "Other countries")
> head(turiste)
```

```
     time America Europe Other countries
1 1991-01    11.4   13.0             4.2
2 1991-02     4.5    8.0             1.4
3 1991-03    10.6   24.2             4.5
4 1991-04    17.5   46.1             8.2
5 1991-05    27.6   58.0            10.3
6 1991-06    36.1   40.5            12.1
```

```
> tail(turiste)
```

```
       time America Europe Other countries
238 2010-10   139.5  215.3            65.0
239 2010-11   111.1  148.4            60.2
240 2010-12    80.6   94.0            63.7
241 2011-01    87.0   92.9            57.0
242 2011-02    76.0  122.7            53.5
243 2011-03   111.9  169.1            45.8
```

Datový rámec opět převedeme na objekt se strukturou časové řady.

```
> turisteTS<-ts(turiste[,-1], start = 1991, frequency = 12)
> str(airtransTS)
```

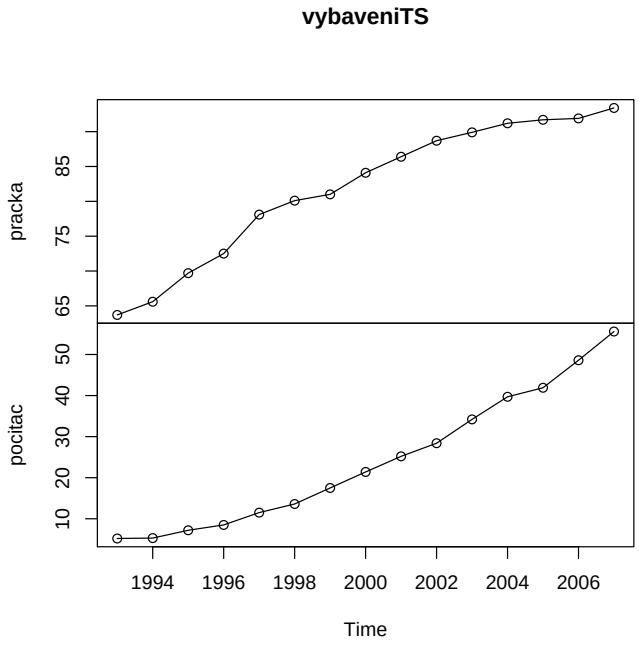
```
 int [1:48, 1:4] 507429 466268 585366 674487 736358 796190 927454 945472 842612 741992 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:4] "Hungary" "Austria" "Slovakia" "Czech.Republic"
 - attr(*, "tsp")= num [1:3] 2006 2010 12
 - attr(*, "class")= chr [1:2] "mts" "ts"
```

2 Grafické znázornění dat a jednoduchá průzkumová analýza dat

Příklad 4

Nyní se vrátíme k datům z prvního příkladu. Data znázorníme graficky, k čemuž použijeme funkci `plot`. Volba `type = "o"` způsobí, že se vykreslí jednotlivé body, které budou spojeny čárou. (Protože vykreslujeme objekt typu časová řada, ve skutečnosti se provádí funkce `plot.ts`).

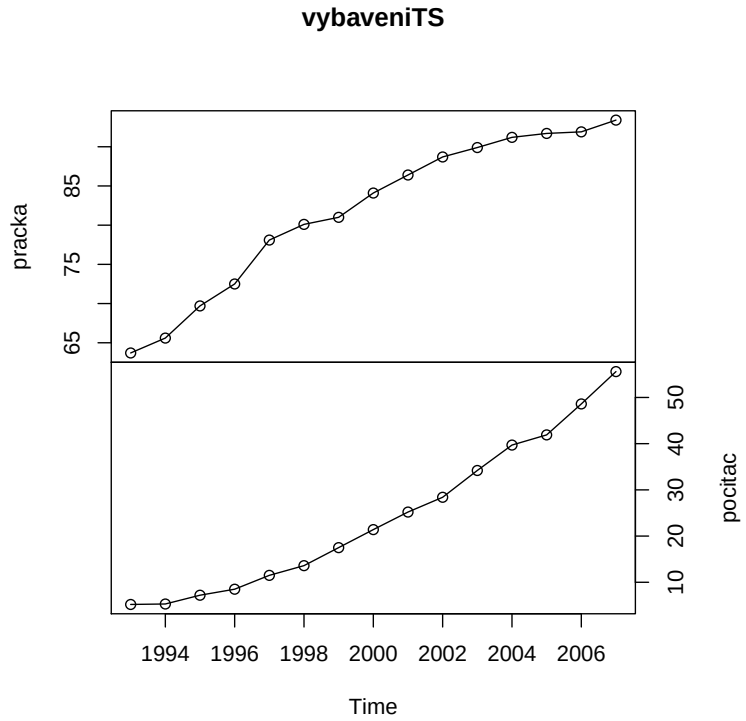
```
> plot(vybaveniTS, type = "o")
```



Obrázek 1: Vykreslení časových řad: *Podíl domácností vlastnících automatickou pračku a osobní počítač v letech 1993–2007* – více panelů

Pro zajímavost se podívejme, co se stane, přidáme-li parametr `yax.flip=TRUE`.

```
> plot(vybaveniTS,type="o",yax.flip=TRUE)
```



Obrázek 2: Vykreslení časových řad: *Podíl domácností vlastnících automatickou pračku a osobní počítač v letech 1993–2007*: volba `yax.flip=TRUE`

Četnosti naměřených hodnot, minimální a maximální hodnoty, kvantily a další základní informace o časové řadě zjistíme pomocí funkce `describe` z balíčku `Hmisc`. Vzhledem k tomu, že první sloupec datového rámce je rok, nemá smysl jej zahrnout mezi vybrané proměnné.

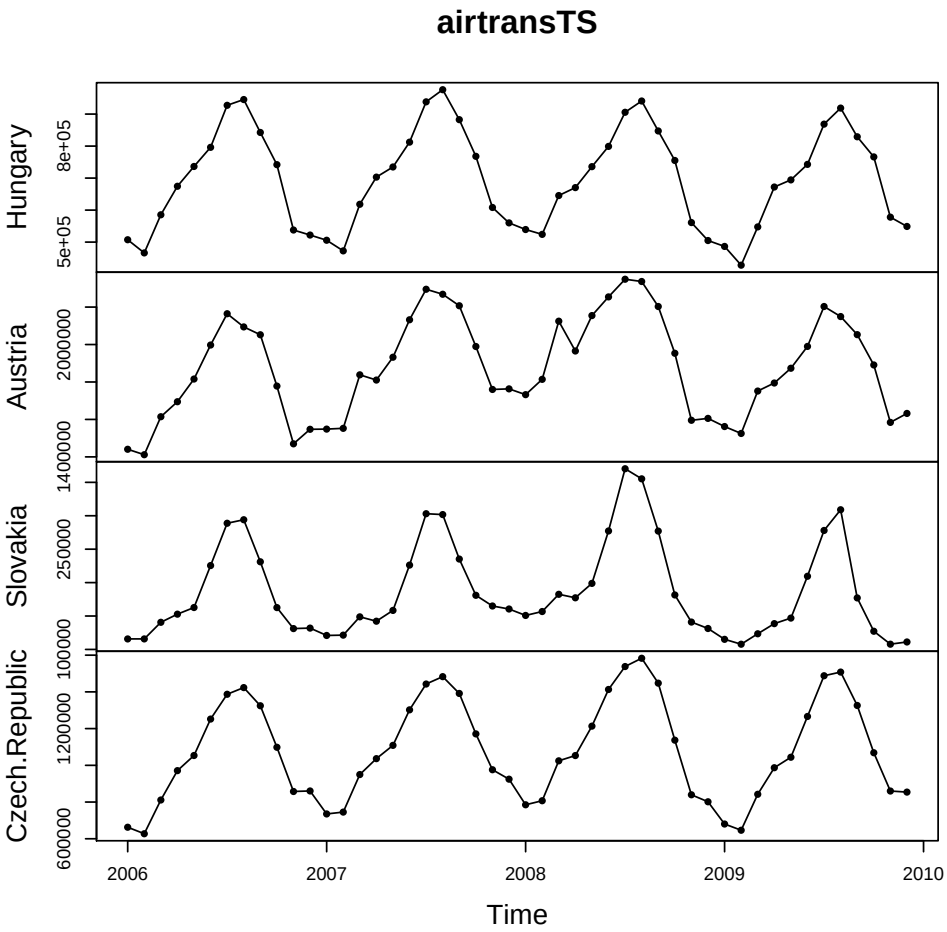
```
> library(Hmisc)
> describe(vybaveni[,2:3])
```

```
vybaveni[, 2:3]

 2  Variables      15  Observations
---------------------------------------------------------------------------
pracka
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
     15       0      15   81.87   65.03   67.24   75.30   84.10   90.55   91.82   92.35

          63.7 65.6 69.7 72.5 78.1 80.1 81 84.1 86.4 88.7 89.9 91.2 91.7 91.9 93.4
Frequency    1    1    1    1    1    1  1    1    1    1    1    1    1    1    1
%            7    7    7    7    7    7  7    7    7    7    7    7    7    7    7
---------------------------------------------------------------------------
pocitac
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
     15       0      15   24.25    5.27    6.06   10.00   21.40   36.95   45.92   50.70

          5.2 5.3 7.2 8.5 11.5 13.6 17.5 21.4 25.2 28.4 34.2 39.7 41.9 48.6 55.6
Frequency   1   1   1   1    1    1    1    1    1    1    1    1    1    1    1
%           7   7   7   7    7    7    7    7    7    7    7    7    7    7    7
---------------------------------------------------------------------------
```

Příklad 5

Nyní se podrobněji podíváme na data z druhého příkladu. Data opět vykreslíme pomocí funkce `plot`. Přitom použijeme následující volby argumentů: pro graf, v němž budou znázorněny jednotlivé body a ty budou navzájem spojeny čarou, zadáme argument `type = "o"`; argument `pch = 20` udává, jakým symbolem budou znázorněny body (zde to bude plné kolečko) a poslední argument `cex = 3` udává velikost těchto bodů.

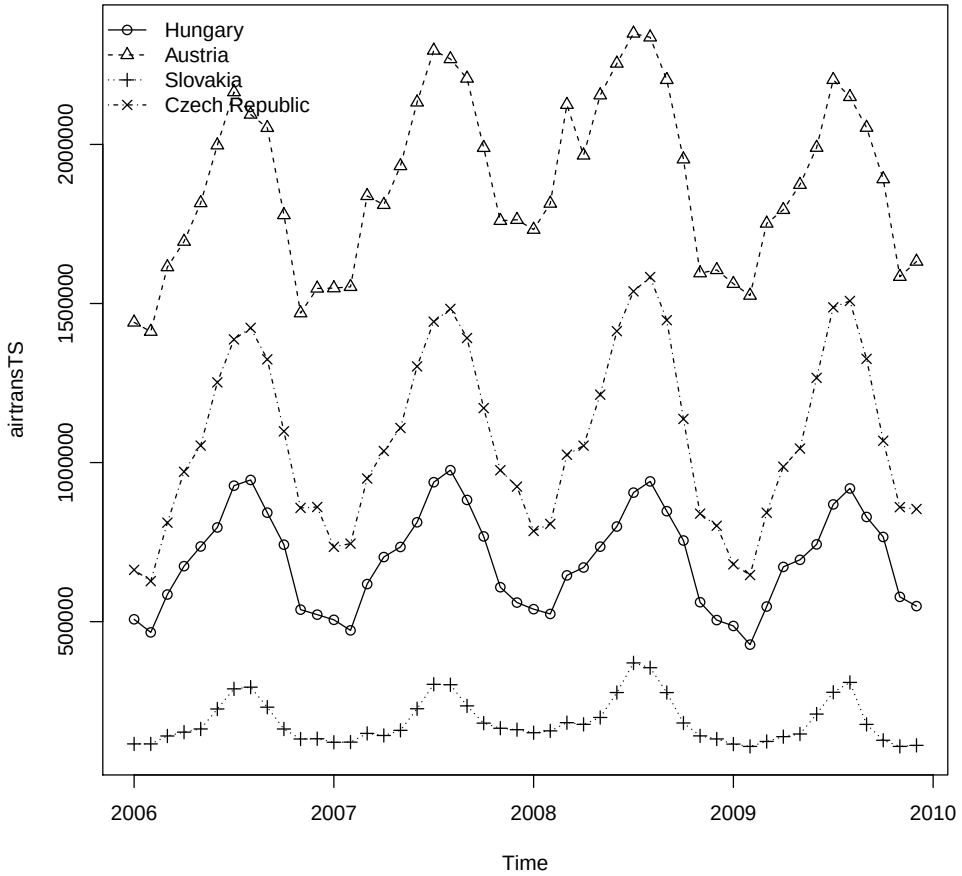
```
> plot(airtransTS, type = "o", pch = 20, cex = 3)
```



Obrázek 3: *Počet osob přepravených leteckou dopravou v letech 2006–2009* : více panelů

Pokud bychom chtěli data za všechny čtyři země zobrazit do jediného obrázku, použijeme následující příkazy (argument `lty =` *číslo* udává typ použité čáry v grafu)

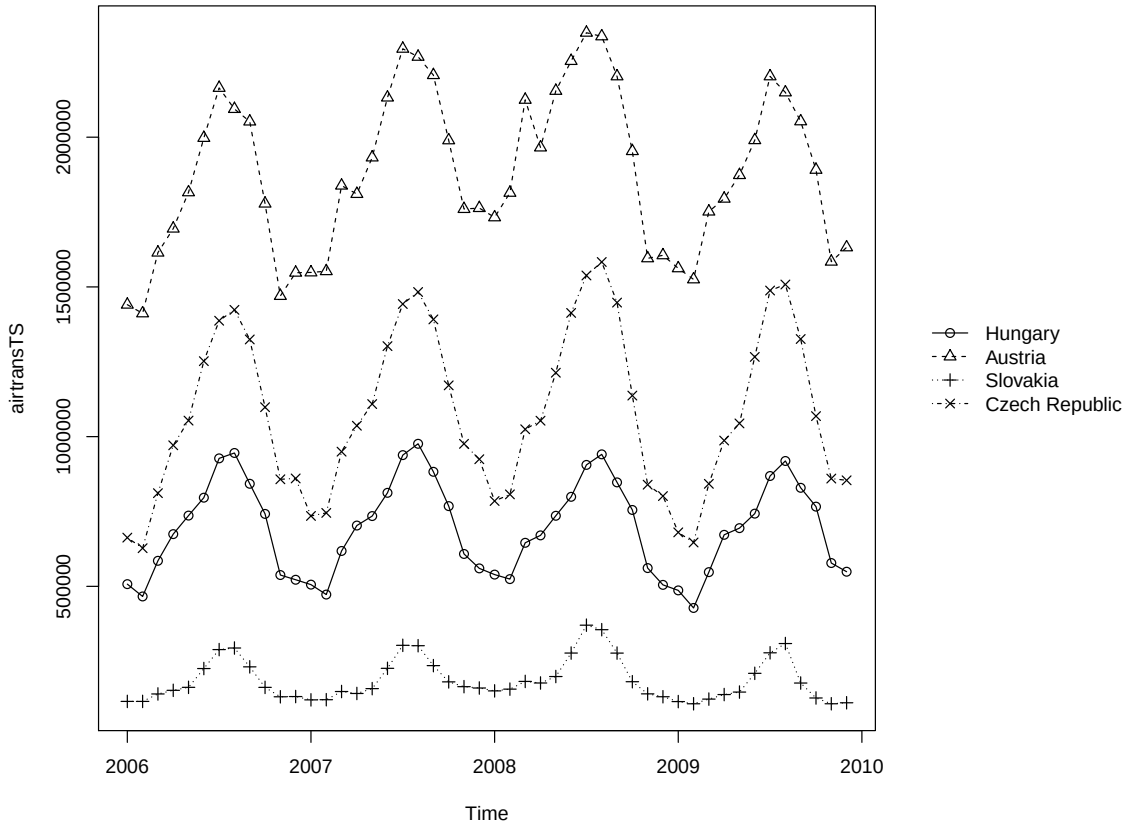
```
> k <- dim(airtransTS)[2]
> plot(airtransTS,type = "o", plot.type = "single", lty = 1:k,pch=1:k)
> legend("topleft",
 legend=c("Hungary","Austria","Slovakia","Czech Republic"),
 lty = 1:k,pch=1:k, bty = "n")
```



Obrázek 4: *Počty přepravených osob v letecké dopravě v letech 2006–2009* – jediný panel (legenda uvnitř panelu)

Z grafu je patrné, že legenda a průběh jedné časové řady se překrývají. Pokud bychom chtěli umístit legendu mimo grafické okno, mohli bychom to provést například takto

```
> layout(matrix(c(1,2), nrow = 1), widths = c(0.8, 0.22))
> par(mar = c(5, 4, 4, 2) + 0.1)
> plot(airtransTS,type = "o", plot.type = "single", lty = 1:k,pch=1:k)
> par(mar = c(5, 0, 4, 2) + 0.1)
> plot(1:3, rnorm(3), pch = 1, lty = 1, ylim=c(-2,2), type = "n", axes = FALSE, ann = FALSE)
> legend("left",
 legend=c("Hungary","Austria","Slovakia","Czech Republic"),
 lty = 1:k,pch=1:k, bty = "n")
```



Obrázek 5: *Počty přepravených osob v letecké dopravě v letech 2006–2009* – jediný panel (legenda vně panelu)

Dále se budeme zabývat pouze jednou časovou řadou, a to údaji pro Českou republiku. Základní popisné statistiky a informace o časové řadě zjistíme příkazem `describe`.

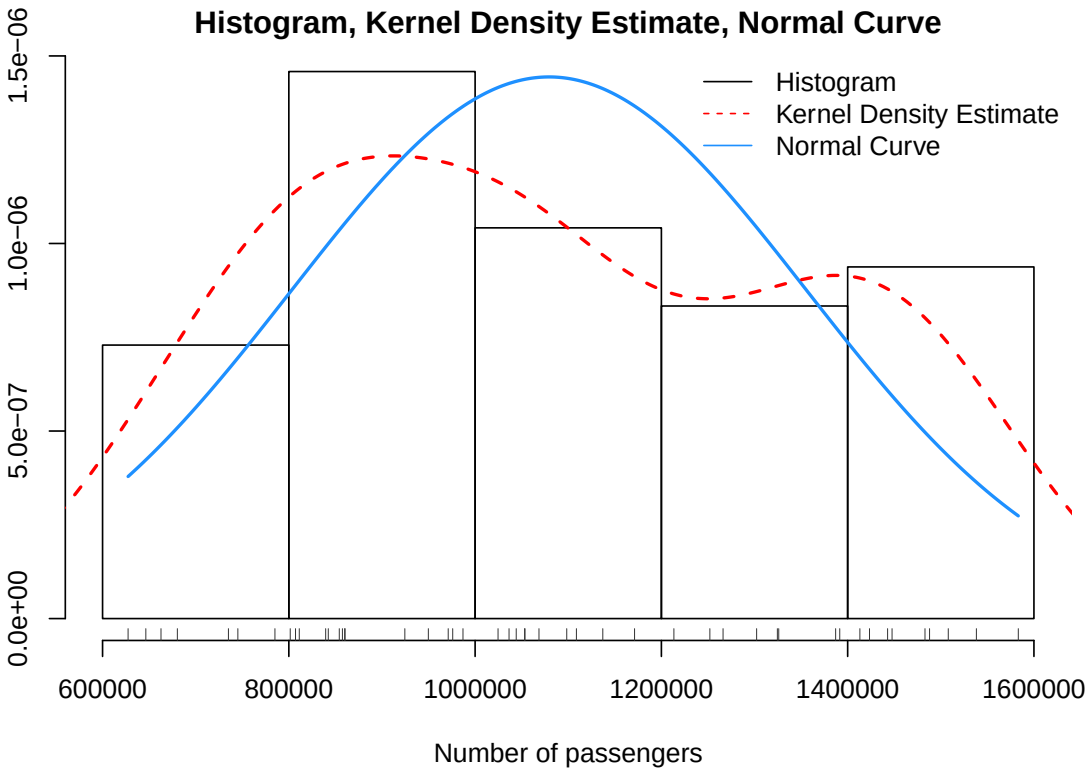
```
> describe(airtrans$Czech.Republic)
```

```
airtrans$Czech.Republic
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
     48       0      48 1079396  668883  742229  851215 1048664 1324895 1458268 1500993

lowest :  627400  646452  662733  680304  735146
highest: 1483032 1487917 1508034 1538203 1583219
```

Nyní vykreslíme histogram, doplníme do něj jádrový odhad hustoty a hustotu normálního rozdělení, jehož střední hodnota a rozptyl odpovídá výběrové střední hodnotě a výběrovému rozptylu dat.

```
> x <- airtransTS[,4]
> par(mar = c(4,2,1,0) + 0.75)
> h <- hist(x, probability = TRUE, breaks = "FD",
          xlab = "Number of passengers",
          main = "Histogram, Kernel Density Estimate, Normal Curve")
```

```
> xfit <- seq(min(x), max(x), length=512)
> yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
> lines(xfit, yfit, col = "dodgerblue", lwd = 2)
> lines(density(x,n=512), lwd = 2, col = "red", lty = 2)
> rug(x, side = 1, ticksize = 0.02, col = "grey20")
> legend("topright",
 legend=c("Histogram","Kernel Density Estimate","Normal Curve"),
 col = c("black","red","dodgerblue"), lty = c(1,2,1), bty = "n")
```

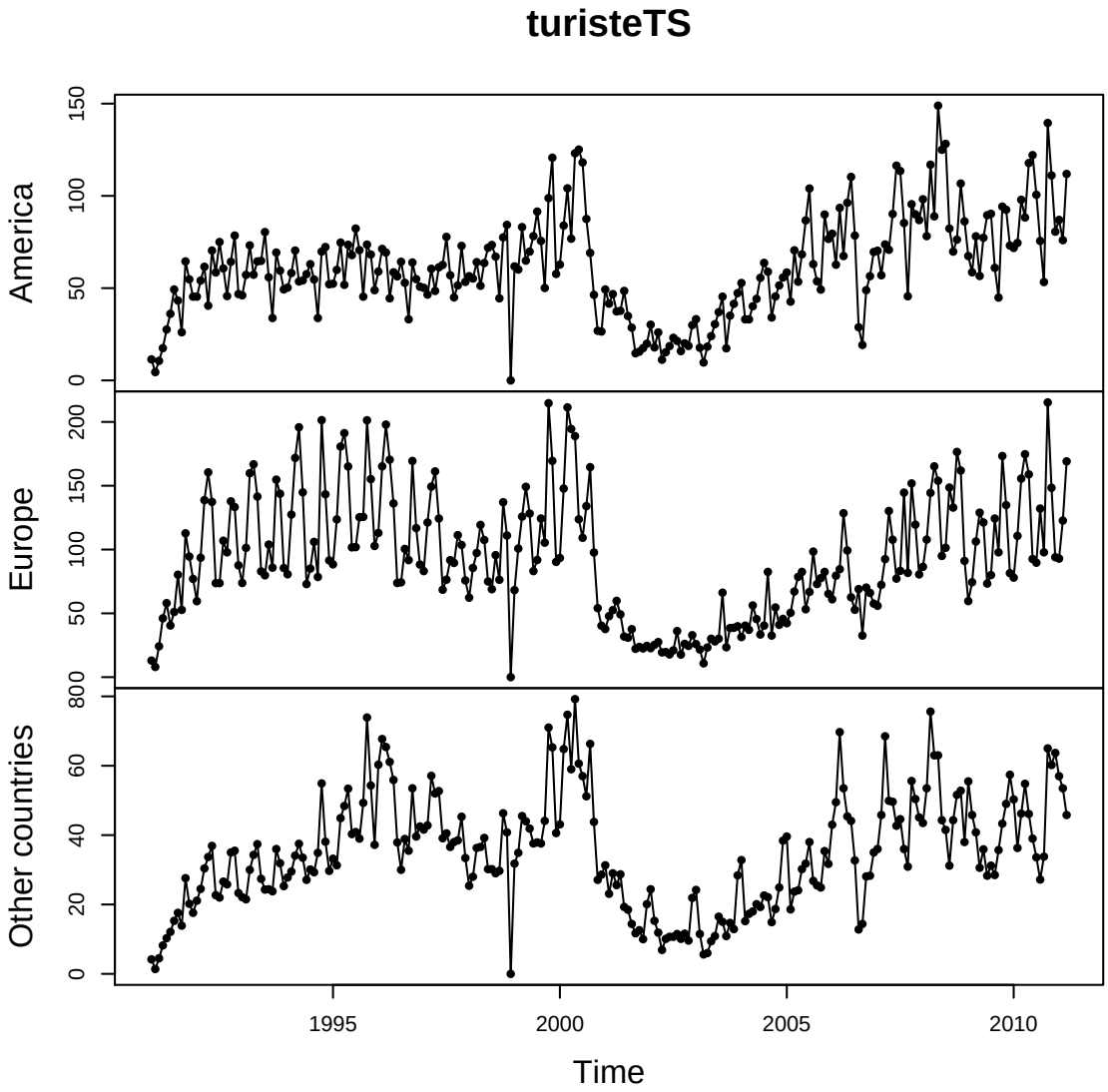


Obrázek 6: Histogram, jádrový odhad hustoty, normální hustota pro počet přepravených osob v letecké dopravě v ČR

Příklad 6

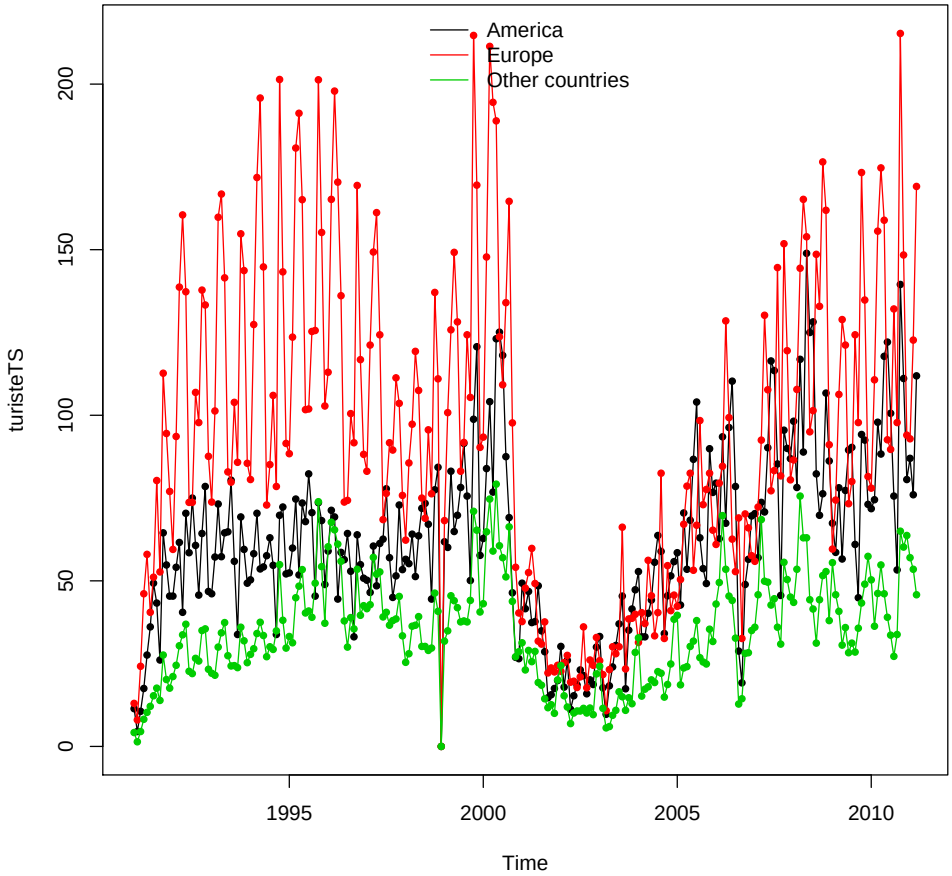
Na závěr se budeme blíž zabývat daty ze třetího příkladu. Data vykreslíme pomocí funkce `plot` obdobně jako v předchozím případě, nejprve každou časovou řadu zvlášť a poté všechny řady do jediného panelu.

```
> plot(turisteTS, type = "o", pch = 20, cex = 3)
```



Obrázek 7: *Počet turistů ubytovaných v izraelských hotelech v letech 1991–2010* – více panelů

```
> k <- dim(turisteTS)[2]
> plot(turisteTS,type = "o", plot.type = "single", col = 1:k, pch = 20, cex = 3)
> legend("top",
 legend=c("America","Europe","Other countries"), lty = c(1,1,1), col = 1:k, bty = "n")
```



Obrázek 8: *Počet turistů ubytovaných v izraelských hotelech v letech 1991–2010* – jeden panel

Dále se budeme zabývat opět pouze jednou časovou řadou, a to počtem ubytovaných turistů z Evropy. Použijeme funkci `describe`, abychom zjistili základní informace a popisné statistiky.

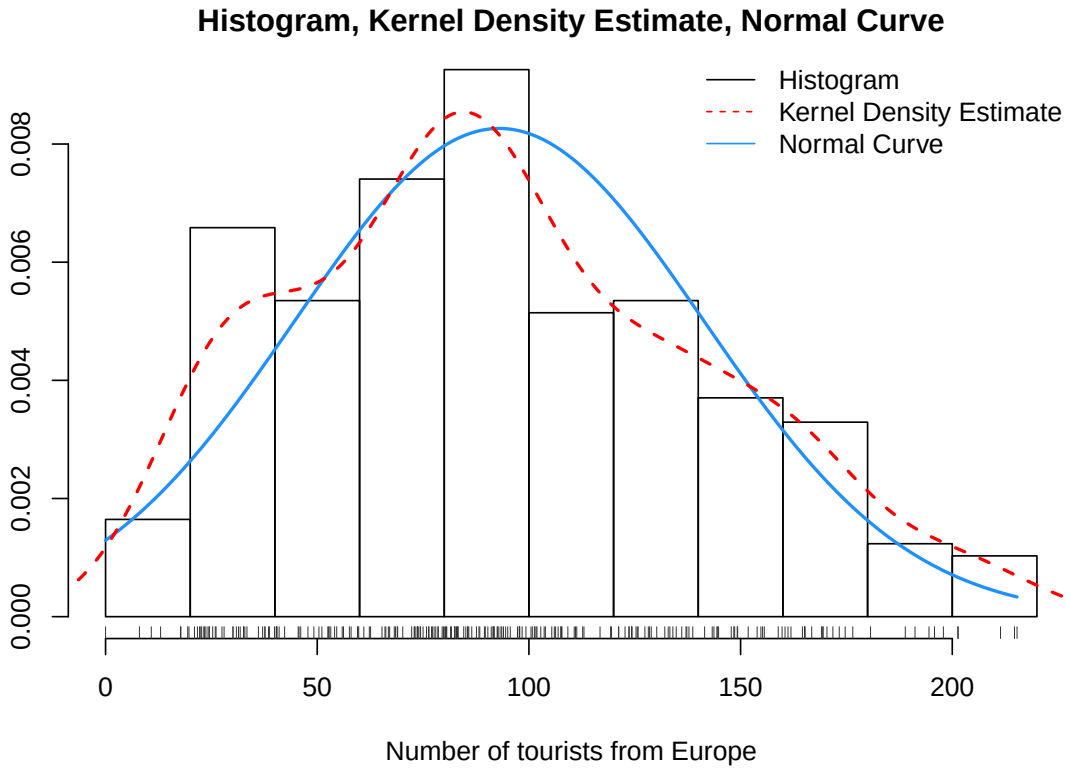
```
> describe(turiste$Europe)
```

```
turiste$Europe
      n missing  unique    Mean     .05     .10     .25     .50
    243       0     227      93   22.75   30.10   56.90   88.20
    .75     .90     .95
 125.70  161.76  174.56

lowest :   0.0   8.0  10.8  13.0  17.7
highest: 201.3 201.4 211.4 214.7 215.3
```

A na závěr vykreslíme histogram s jádrovým odhadem hustoty a hustotou normálního rozdělení s parametry danými výběrovými odhady.

```
> x <- turisteTS[,2]
> par(mar = c(4,2,1,0) + 0.75)
```

```
> h <- hist(x, probability = TRUE, breaks = "FD",
         xlab = "Number of tourists from Europe",
         main = "Histogram, Kernel Density Estimate, Normal Curve")
> xfit <- seq(min(x), max(x), length=512)
> yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
> lines(xfit, yfit, col = "dodgerblue", lwd = 2)
> lines(density(x,n=512), lwd = 2, col = "red", lty = 2)
> rug(x, side = 1, ticksize = 0.02, col = "grey20")
> legend("topright",
 legend=c("Histogram","Kernel Density Estimate","Normal Curve"),
 col = c("black","red","dodgerblue"), lty = c(1,2,1), bty = "n")
```



Obrázek 9: Histogram, jádrový odhad hustoty, normální hustota pro počet přepravených osob v letecké dopravě v ČR

3 Úkol

Na internetu najděte a stáhněte zajímavé časové řady, jak roční, tak čtvrtletní nebo měsíční. Načtěte je do systému R, data vykreslete, zjistěte základní popisné statistiky a zobrazte histogram.