FORUM STATISTICUM SLOVACUM Vol XVI (2020) No 2 pp 41-51

Statistické programování v jazyku R v prostředí modulu BASE softwarového systému SAS
Statistical programming in the R language in the BASE module environment of the SAS software system

Roman Pavelka

Štatistický úrad Slovenskej republiky, Odbor metód štatistických zisťovaní, Miletičova 3, 824 67 Bratislava, Slovenská republika

Statistical Office of the Slovak Republic, Statistical Surveys and Methodology Department, Miletičova 3, 824 67 Bratislava, Slovak Republic

roman.pavelka@statistics.sk

Abstrakt: *Předkládaný příspěvek popisuje jednoduché programové rozhraní vytvořené pomocí příkazů SAS, které umožňuje vkládat a spouštět příkazy jazyka R v rámci základního modulu programu SAS v operačním systému Windows. Vytvořené programové rozhraní vykoná uživatelem definovanou posloupnost příkazů jazyka R v dávkovém režimu v základním modulu programového systému SAS. Vytvořená syntaxe příkazů také dokáže automaticky převádět data mezi datovými tabulkami systému SAS a datovými rámci programu R tak, že data a výsledky z jednoho statistického prostředí mohou být využita druhým statistickým prostředím. Cílem příspěvku je ukázat čtenářům, že lze několikerými způsoby využívat schopností programovacího jazyka R v rámci programového systému SAS, a to už na úrovni základních modulů prostředí SAS. Navíc toto rozhraní také pomáhá statistickým programátorům naučit se nový statistický jazyk při pobytu ve známém prostředí systému SAS. Článek obsahuje několik alternativních možností, jakým způsobem lze spouštět příkazy jazyka R v prostředí systému SAS, a ukázkový příklad příkazů programovacího jazyka R, kterými budou reprezentovány základní schopnosti vytvořeného programového rozhraní.*

Abstract: *This paper describes a simple program interface created using SAS statements that allows you to insert and execute R language commands within the basic modules of the SAS program in the Windows operating system. The created program interface executes the user-defined sequence of R language commands in batch mode in the basic module of the SAS program system. These created macros can also automatically convert data between SAS data tables and R data frames so that data and results from one statistical environment can be used by another statistical environment. The aim of this paper is to show that it is possible to use the capabilities of the R programming language within the SAS programming system in several ways, already at the level of the basic modules of the SAS environment. In addition, this interface also helps statistical programmers learn a new statistical language while staying in the familiar SAS environment. The article also contains some sample example of R programming language commands running in SAS environment, which will represent basic abilities of created programming interface.*

Kľúčové slová: *dávkový režim, HTML, makro, ODS, R, SAS, R.*

Key words: *batch mode, HTML, macro, ODS, R, SAS, R.*

1 Úvod

Jelikož počet programovacích jazyků a platforem pro analýzu dat rapidně vzrostl, je mnohdy pro programátory-analytiky dat stále více žádoucí volat určitý programovací jazyk z jiného softwarového prostředí. Četné úsilí v tomto směru umožnilo statistickým analytikům používat svůj oblíbený programovací jazyk v jiném softwarovém prostředí. Například projekt R (R Development Core Team, 2008), přední open-source statistický výpočetní jazyk a prostředí, poskytuje moduly pro integraci nativního C ++ (Eddelbuettel a Francois, 2011) a kódu SQL (Groethendieck, 2017). Programátoři Pythonu vyvinuli různé metody, které umožňují přístup k bohatým statistickým balíčkům projektu R v prostředí Pythonu (Xia et al., 2010). V komerční sféře je široce používaný systém pro správu dat a statistickou analýzu, to programový systém SAS. V rámci produktu SAS umožňuje procedura SQL programátorům účinně provádět manipulaci s daty pomocí strukturovaného dotazovacího jazyka v rámci platformy SAS (SAS Institute Inc., 2016a).

S ohledem na zvyšující se význam projektu R ve statistických výpočtech firma SAS před několika lety vytvořila softwarové rozhraní mezi R označované jako SAS/IML (SAS Institute Inc., 2016c). V programovém systému SAS je modul SAS/IML (Wicklin, 2010) zaměřený na výpočet matic (SAS Institute Inc., 2018). Bohužel základní modul BASE systému SAS, který je základem všech programových modulů SAS a má mnohem větší komunitu uživatelů než modul SAS/IML, postrádá uživatelsky přívětivé a dobře zdokumentované programové rozhraní pro vkládání příkazů pro programování v jazyku R. Z tohoto důvodu je integrace programovacího rozhraní základního modulu SAS/BASE a R je velmi žádoucí, protože uživatelé základních modů SAS často potřebují přístup k novým statistickým, resp. vizualizačním nástrojům, které byly vyvinuty v projektu open source R a dosud nebyly implementovány v programovém systému SAS.

V předkládaném článku je představováno jednoduché programovací rozhraní SAS, ve kterém lze v rámci relace programu SAS psát a vykonávat příkazy jazyka R a výsledný textový a grafický výstup jazyka R je přístupný za pomoci příkazů SAS. Toto jednoduché programovací rozhraní SAS je tvořeno funkční sekvencí příkazů pro programovací jazyk R založenou na využívání příkazů v dávkovém režimu jazyka R a uživatelsky definovaném programovém kódu. Po spuštění programu SAS jsou příkazy jazyka R vykonány prostřednictvím relace dávkového režimu pomocí příkazového řádku. Textový a grafický výstup programu R lze odeslat do vhodného výstupu SAS. Toto se realizuje pomocí tzv. výstupního systému (anglicky „Output Destination system“- ve zkratce ODS).

2 Nejdůležitější principy propojení jazyka R a programového systému SAS

Cílem tohoto článku není popis programovacího jazyka R. Pro další text je však důležité přiblížit některé možnosti programového prostředí jazyka R, které jsou důležité pro spolupráci s programovým systémem SAS. Programovací jazyk R je možné používat s využitím některého z několika grafických rozhraní[2] nebo v režimu tzv. dávkového zpracování.

Programovací jazyk R pracuje s několika typy objektů, jako jsou například datové rámce (dataframe), matice a seznamy, které využívá k ukládání dat a manipulaci s nimi. Programový systém SAS využívá k ukládání dat objekty jako jsou datové soubory (dataset), které jsou velmi podobné datovým rámcům jazyka R. Zatímco jazyk R pro uložená data používá pojmy „sloupce“ a „řádky“, programový systém SAS „proměnné“ a „pozorování“. Na rozdíl od programovacího systému SAS jazyk R rozlišuje v názvech proměnných, funkcí a dalších objektů velká a malá písmena. I formátování proměnných je v programovacím prostředí R a systému SAS odlišné. Z tohoto důvodu je pro export datových souborů SAS ke zpracování jazykem R a importu zpracovaných výsledků do systému SAS vhodné použití textových souborů, nejlépe ve formátu CSV[3] nebo TXT[4]. V současné době programové prostředí jazyka R již obsahuje některé specializované programové moduly, které umožňují v jazyku R přímé načtení databází programového systému SAS[5]. Grafické výstupy jazyka R lze importovat do programu SAS nejlépe jako grafické soubory.

3 Dávkový režim jazyka R

Dávkový režim neboli dávkové zpracování (anglicky „batch processing“), představuje zpracování jednoho nebo více příkazů programu najednou a nasměrování výsledků tohoto zpracování do jiného souboru, případně souborů. Posloupnost příkazů, která realizuje požadovanou činnost a může současně využívat již definovaných objektů z pracovního prostoru programovacího jazyka, se nazývá dávka příkazů. Příkazy se mohou vytvořit v libovolném textovém editoru, pouze název souboru, do něhož se příkazy ukládají, musí mít příponu R (např. davka.R). Programovací jazyk R lze

[2] Mezi nejznámější grafická rozhraní pro jazyk R jsou například *RStudio* nebo *RCommander*.

[3] Jedná se o označení *comma separated values* (česky čárkou oddělené údaje), podle normy IETF RFC4180, což je jednoduchý a standardizovaný textový formát pro reprezentaci tabulkových dat. Díky jednoduchosti, nenáročnosti a čitelnosti i bez specializovaného software se tento formát používá pro výměnu informací mezi různými systémy.

[4] Soubory s příponou TXT jsou označovány jako prostý text, protože pro formátování textu používají pouze znaky konce řádku, mezery a tabulátory.

[5] Jedná se zejména o doplňkové programové moduly jazyka R *haven* a *sas7bdat*, které umožňují načítat databáze ve formátu sas7bdat (nativní formát SAS) a speciální exportní formát ve tvaru SAS XPORT (jedná se o soubory s koncovkou XPT).

v dávkovém režimu spustit na příkazovém řádku operačního systému (například Windows nebo jiných operačních systémů) pomocí příkazu ve tvaru „R [parametry] [<davka.R] [>vystup.out]“. Symbol „R“ představuje spustitelný soubor („R.exe“) jazyka R, „davka.R“ tvoří soubor posloupnosti příkazů a soubor „vystup.out“ je souborem výstupním, do něhož se ukládají výsledky dávkového zpracování. V prostředí operačního systému Windows lze dávkové zpracování příkazů jazyka R na příkazové řádce lze spustit pomocí syntaxe:

R CMD BATCH "--args arg1 arg2" davka.R output.out (1)

Parametr CMD vyjadřuje použití příkazové řádky operačního systému. Parametr --args uvozuje sadu vstupních parametrů arg1 arg2, které vstupují do vstupního vektoru parametrů. Cestu ke spustitelnému souboru „R.exe“ i k souborům vstupu a výstupu zpracování posloupnosti příkazů je nutno nastavit podle aktuální struktury adresářů. Příkladem příkazu k dávkovému zpracování jazykem R může být výraz:

"C:\Program Files\R\R-4.0.3\bin\x64\R.exe" CMD BATCH --no-save --quiet < myprog.r > myprog.out. (2)

Po spuštění výše uvedeného příkazu se v prostředí jazyka R vykoná sekvence příkazů uložených v souboru „myprog.r“ a výsledky dávkového zpracování jazykem R se zaznamenají do výstupního souboru „myprog.out“. Parametry --no-save --quiet slouží k nastavení módu dávkového zpracování jazyka R. Parametr „no save“ určuje, že se na konci dávkového zpracování jazykem R datové soubory ukládat nemají, a parametre „quiet“ znamená, že se nebude zobrazit počáteční copyright a uvítací hlášení v okamžiku spuštění jazyka R.

4 Spouštění jazyka R v programovém systému SAS

Základním principem zpracování příkazů programovacího jazyka R v rámci programového systému SAS je spouštění jazyka R na úrovni příkazového řádku. Funkcionalitu příkazového řádku v prostředí SAS lze vyvolat více způsoby, které jsou zabudovány již v základním modulu BASE systému SAS. V rámci tohoto modulu existuje několik příkazů, jejichž pomocí lze programovací jazyk R volat, předat požadované příkazy i převzít výsledky operací jazyka R.

V prostředí SAS lze pomocí systémových parametrů explicitně nastavit externí proces běhu jazyka R tak, aby se ukončoval zpracováním předávané dávky příkazů anebo aby jeho ukončení a návrat do systému SAS proběhlo zásahem uživatele. Je-li aktivní možnost automatického ukončení činnosti jazyka R a návratu do systému SAS, vykonáním dávky příkazů se programové prostředí R uzavře. V opačném případě je prostředí jazyka R ukončeno až zásahem uživatele. Při dávkovém zpracování příkazů je zpravidla vhodné nastavit automatické ukončování běhu jazyka R.

Jinými parametry systému SAS lze dosáhnout toho, aby program SAS čekal na dokončení dávky příkazů zpracovávaných programem R. To znamená, že řízení běhu programu v systému SAS není vráceno, dokud nejsou zpracovány všechny externí příkazy jazyka R, přičemž se jedná o tzv. synchronní režim. V asynchronním režimu jsou programové příkazy v systému SAS zpracovávány bez ohledu na vykonání všech externích příkazů jazyka R.

Funkcionalitu příkazové řádky ke spouštění jazyka R v rámci základního modulu BASE programového systému SAS poskytují následující příkazy, funkce[6] a programové rutiny[7], tj. příkazy začínající slovem CALL:

- Příkaz X
- Funkce SYSTEM
- Rutina CALL SYSTEM
- Příkaz %SYSEXEC
- Příkaz SYSTASK COMMAND
- Příkaz FILENAME s parametrem na zařízení PIPE

Všechny tyto vyjmenované příkazy, funkce a programová rutina jako vstupní parametr obsahují příkaz operačního systému, jehož odesláním z příkazové řádky operačního systému zajistí jeho vykonání. Příkazy X[8], SYSTASK COMMAND a FILENAME patří k tzv. globálním příkazům, což znamená, že tyto příkaz lze spustit kdekoliv v programu SAS. Návratová hodnota po provedení příkazu SYSTASK COMMAND je uložena do systémové makroproměnné SYSRC. Je-li vykonání příkazu SYSTASK COMMAND úspěšné, je hodnota SYSRC rovna 0. Návratová hodnota po provedení příkazu FILENAME je 0, pokud je operace úspěšná. Příkaz %SYSEXEC patří do skupiny příkazů jazyka maker SAS. Použití tohoto příkazu je povoleno v definici makra anebo i mimo něj. O vykonání příkazu příkazové řádky operační systém informuje programový systém návratovou hodnotou, kterou je naplněna systémová makroproměnná SYSRC. Zbývající funkci SYSTEM a programovou rutinu CALL SYSTEM se musí použít v rámci DATA kroku. Je-li vykonání příkazu v operačním systému úspěšné, je návratová hodnota funkce SYSTEM rovna 0.

[6] Funkce SAS provádí výpočet nad vstupními daty nebo vykoná změny systému s využitím vkládaných argumentů a vrací hodnotu.

[7] Rutina CALL může měnit hodnoty proměnných nebo hodnoty některých nastavení systému. Rutiny CALL jsou podobné funkcím. Od funkcí SAS se liší tím, že je nelze používat v přiřazovacích příkazech, tj. nevrací hodnotu.

[8] Z důvodu bezpečnosti není globální příkaz X pro uživatele SAS Enterprise Guide implicitně povolen. K povolení činnosti tohoto příkazu je potřebné nastavit systémový parametr ALLOW XCMD na SAS serveru a restartovat SAS Metadata server.

Přesnou syntaxi příkazů, funkcí a programové rutiny lze dohledat v nápovědě a manuálech k programovému systému SAS (SAS Institute Inc., 2016b).

5 Jednoduchý příklad volání jazyka R v rámci programového systému SAS

Jako velmi názorná ukázka toho, jak lze spouštět příkazy programovacího jazyka R z prostředí softvérového systému SAS, může sloužit následující program SAS. Cílem tohoto ukázkového programu je demonstrovat možnosti programového systému SAS v základním modulu BASE spouštět, předávat příkazy a přebírat výstupy z programovacího jazyka R. Jedná se o fungující program složený z následující sekvence příkazů:

```
/* XXXXXXXXXXXXXXXXXXXXXXXX Nastavení proměnných XXXXXXXXXXXXXXXXXXXXXXX */
options nonumber pagesize=max formdlim=' ';
%let SASDataPath = D:\SASData\Odbor320\O_PAVELKA\Integrace_R_do_SAS\R Programming\Do_Textu;
%let RExecFile = D:\SASData\odbor320\O_PAVELKA\R\R-4.0.3\bin\x64\R.exe;
%let RScriptFile = &SASDataPath.\program.r;
%let RLogFile = &SASDataPath.\program.log;
%let RDataPath = %sysfunc(translate(&SASDataPath., %str(/), %str(\)));
%let SAStoRFile = DataFromR;
%let SASDataFile = &SAStoRFile..txt;
%let SASCodeFile = &SAStoRFile..sas;
%let ROutputFile = vystupy.txt;
%let ImageFile1 = Grafika1.png;
%let ImageFile2 = Grafika2.png;
/* export-import knihovna ze SASu do R a z R do SASu */
libname RData "&SASDataPath.";

/* XXXXXXXXXXXXXXXXXXXXXXXXX Výkonný program XXXXXXXXXXXXXXXXXXXXXXXXX */
/* Vytváření dávky příkazů v programovacím jazyku R a uložení do dávkového souboru */
data _null_;
        file "&RScriptFile.";
        infile cards;
        input;
        put _infile_;
        cards4;

#začátek posloupnosti příkazů jazyka R
#sink()
#stažení a instalace nutných knihoven programovacího jazyka R
if(!require(ggplot2)){install.packages("ggplot2", repos = "http://cran.us.r-project.org")}
if(!require(gridExtra)){install.packages("gridExtra", repos = "http://cran.us.r-project.org")}
if(!require(plyr)){install.packages("plyr", repos = "http://cran.us.r-project.org")}
if(!require(foreign)){install.packages("foreign", repos = "http://cran.us.r-project.org")}
if(!require(scales)){install.packages("scales", repos = "http://cran.us.r-project.org")}
if(!require(zoo)){install.packages("zoo", repos = "http://cran.us.r-project.org")}
# připojení nutných knihoven programovacího jazyka R
library(ggplot2)
library(gridExtra)
library(plyr)
library(foreign)
library(scales)
library(zoo)
#stažení dat a jejich úpravy
df <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/yahoo.csv")
df$date <- as.Date(df$date)  # format date
df <- df[df$year >= 2010, ]  # filter reqd years
# Create Month Week
df$yearmonth <- as.yearmon(df$date)
df$yearmonthf <- factor(df$yearmonth)
df <- ddply(df,.(yearmonthf), transform, monthweek=1+week-min(week))  # compute week number of
month
df <- df[, c("date", "year", "yearmonthf", "monthf", "week", "monthweek", "weekdayf",
"VIX.Close")]
# Or specify the factor levels in the order you want
df$weekdayf <- factor(df$weekdayf, levels = c("Fri", "Thu", "Wed", "Tue", "Mon"))
```

```
df$monthf <- factor(df$monthf, levels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec"))
#nastavení pracovního adresáře
setwd("D:/SASData/Odbor320/O_PAVELKA/Integrace_R_do_SAS/R Programming/Do_Textu")
#otevření zařízení PNG (vytvoří obrázek formátu PNG pro první graf)
png("Grafika1.png", width = 1024, height = 768, units = "px", pointsize = 12, bg = "white",
res = 150, family = "symbol")
ggplot(data = df, aes(x = date, y = VIX.Close)) + geom_line(color = "#00AFBB", size = 0.25) +
geom_point(colour = "yellow", size = 0.5) +
        stat_smooth(color = "#FC4E07", fill = "#FC4E07", size = 0.5, method = lm, formula = y ~
splines::bs(x, 10), se = FALSE) +
        labs(x="Doba", y="Uzavírací cena [USD]", title = "Denní uzavírací ceny akcií Yahoo
[USD] a jejich trend", subtitle="") +
        scale_y_continuous(limits = c(10, 50)) +
        scale_x_date(name = "Období", breaks=date_breaks("1 year"), labels = date_format("%b
%Y"))
#uzavření zařízení PNG
dev.off
#otevření zařízení PNG (vytvoří obrázek formátu PNG pro první graf)
png("Grafika2.png", width = 800, height = 600, units = "px", pointsize = 12, bg = "white", res
= 150)
ggplot(df, aes(monthweek, weekdayf, fill = VIX.Close)) + geom_tile(colour = "white") +
facet_grid(year~monthf) +
        scale_fill_gradient(low="red", high="green") +
        labs(x="Týden", y="", title = "Mapa denních uzavíracích cen akcií Yahoo", subtitle="",
fill="Cena\n [USD]") +
        theme(axis.text.x = element_text(face="bold", color="black", size=5, angle=0),
                axis.text.y = element_text(face="bold", color="black", size=5, angle=0))
#uzavření zařízení PNG
dev.off
#od tohoto místa ukládám výstupy z procedur R. Textové výsledky můžeme namísto zobrazení do
konzoly ukládat do externího souboru pomocí příkazu
sink("vystupy.txt")
#print(unname(as.data.frame("================================ Zahájení výpisu výstupů z R-ka
================================")),quote = FALSE, row.names = FALSE)
print(unname(as.data.frame("======================== Dataframe s daty časové řady (prvních
několik záznamů) ==================")),quote = FALSE, row.names = FALSE)
head(df)
print(unname(as.data.frame("============================== Struktura dataframe s daty časové
řady ==============================")),quote = FALSE, row.names = FALSE)
str(df)
print(unname(as.data.frame("================================ Ukončení výpisu výstupů z R-ka
================================")),quote = FALSE, row.names = FALSE)
#ukládání ukončíme pomocí
sink()
# uložení vygenerovaných dat pro import do SAS
Pokus <- write.foreign(df,
        datafile = "D:/SASData/Odbor320/O_PAVELKA/Integrace_R_do_SAS/R
Programming/Do_Textu/DataFromR.txt",
        codefile = "D:/SASData/Odbor320/O_PAVELKA/Integrace_R_do_SAS/R
Programming/Do_Textu/DataFromR.sas",
        package  = "SAS", dataname = "DataFromR")
#ukončení prostředí R
q()
#konec posloupnosti příkazů jazyka R

;;;;
run;
/*Použití příkazu X:*/
options noxwait xsync;
X ""&RExecFile." CMD BATCH --no-save --quiet "&RScriptFile." "&RLogFile."";
/* výpis výstupu z prostředí R do systému SAS*/
data _null_;
        title 'Výpis souboru s výstupy z programu R do SAS Reportu:';
        infile "&SASDataPath.\&ROutputFile." lrecl=1024;
        file log;
        file print ls=1024 titles line=lineno ll=lineleft ps=32767;
        input;
        put ' ' _infile_;
run;
/*tvorba SAS dat z prostředí R*/
%include "&SASDataPath.\&SASCodeFile.";
data RData.&SAStoRFile.;
        set work.&SAStoRFile.;
run;
proc delete data = work.&SAStoRFile.;
```

```
run;
/* vykreslení grafických výstupů */
ods graphics on / reset border=on width=800px height=600px;
goptions reset=all cback="yellow" border hsize=800pt vsize=600pt;
proc gslide iframe="&SASDataPath.\&ImageFile1." imagestyle=FIT;
       title "Časová řada denních uzavíracích cen akcií Yahoo [USD]";
run;
goptions reset=all cback="yellow" border hsize=800pt vsize=600pt;
proc gslide iframe= "&SASDataPath.\&ImageFile2." imagestyle=FIT;
       title "Mapa denních uzavíracích cen akcií Yahoo [USD]";
run;
quit;
ods _ALL_ close;
```

Výše uvedený program z prostředí programového systému SAS je tvořen několika relativně samostatnými částmi. Data, která program využívá, jsou časové řady uzavíracích cen akcií a jsou stažena z Internetu[9]. V programovacím prostředí jazyka R jsou pomocí programových instrukcí instalovány další softvérové balíčky, které jsou nutné při zpracování syntaxe příkazů.

Úvodní část programu slouží k definici tzv. makroproměnných[10], které se využívají v dalších částech programu. Dochází zde k definování knihovny SAS, pracovních adresářů, souborů vstupu: souboru dávky příkazů a souboru programovacího jazyka R (tj. R.exe), a výstupu: jedná se o soubory výstupů z programovacího jazyka R, tj. výstupních textových souborů s výsledky činnosti jazyka R a obrazových souborů, v nichž jsou uloženy výstupní grafy. Výstupní grafy nejsou vytvořeny v softvérovém systému SAS, ale k jejich tvorbě byl použit příkaz programovacího jazyka R.

Následující příkazy již náleží k výkonné části programu. Základem uvedených příkazů je příkaz DATA (v systému SAS nazývaný DATA krok), který zajišťuje vytvoření vstupního textového souboru s posloupností příkazů pro programovací jazyk R. Příkazy jazyka R jsou vkládány již přímo v syntaxi, která zodpovídá syntaxi příkazů jazyka R. Napsané příkazy jsou uloženy do souboru příkazů jazyka R, který slouží jako vstupní soubor pro činnost jazyka R.

Následuje příkaz X (byl popisován v předcházejících kapitolách), který představuje funkcionalitu příkazové řádky ke spouštění jazyka R v rámci základního modulu BASE programového systému SAS. Jedná se o jediný příkaz v programu SAS ke spuštění programovacího jazyka R, který vykoná vstupní sekvenci příkazů, a zajistí uložení výstupů do výstupního souboru.

Výstupy z prostředí jazyka R jsou po ukončení zpracování vstupních příkazů uloženy opět do textového souboru, který je vyhodnocován dalšími úseky programu z uvedeného příkladu. Základ této části programu představuje další

[9] Dostupné k 2020-10-12 na adrese: <https://raw.githubusercontent.com/selva86/datasets/master/yahoo.csv>.

[10] Podle nápovědy k programovému systému SAS makroproměnné jsou proměnné, u nichž lze dynamicky modifikovat jejich hodnotu v programu SAS pomocí symbolické substituce.

příkaz DATA, jehož úlohou je výpis výstupů jazyka R do výstupního reportu v programovém systému SAS. Výpis výstupů z prostředí R je možné formátovat do požadované podoby, případně uložit do SAS knihovny. Následuje skupina příkazů, která umožní data vzniklá v programovém prostředí jazyka R importovat do knihovny na serveru SAS. Ve výše uvedeném příkladu byla data načtena ze vzdáleného WWW serveru a zpracována programovými příkazy jazyka R a následně uložena do SAS knihovny RData. Grafické výstupy (grafy časových řad) byly vytvořeny v prostředí jazyka R a ve formě obrazových souborů jsou zobrazeny ve výstupním reportu programového systému SAS.

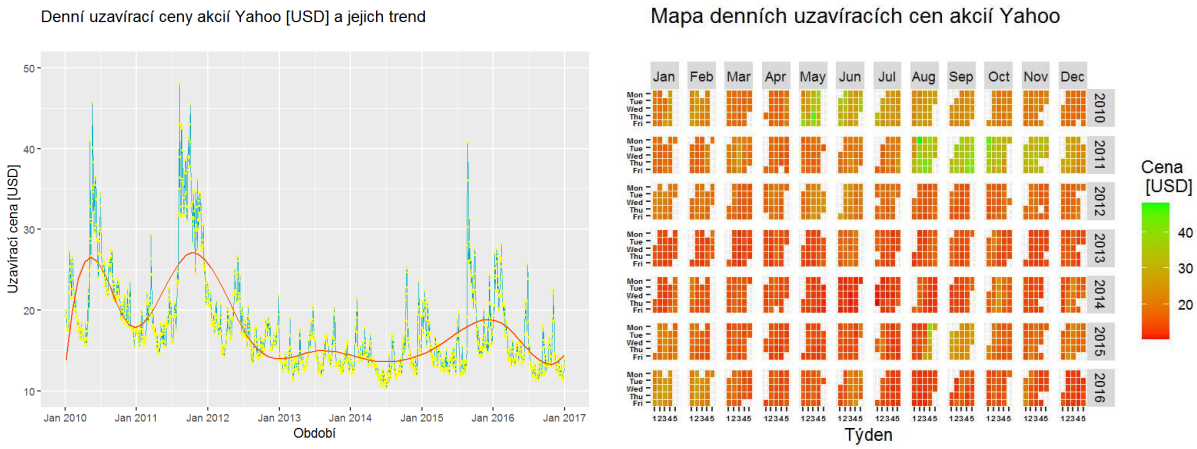
Po spuštění a úspěšném proběhnutí programu, který obsahuje výše uvedené instrukce jazyka R, se ve výstupním SAS reportu objeví výpis datového rámce (dataframe) s daty časové řady z prostředí jazyka R. Ukázka tohoto výpisu je znázorněna na obrázku 1. Grafy časové řady, které byly vytvořeny pomocí jazyka R a byly vloženy do programu v systému SAS, jsou ilustrovány na obrázku 2. V prostředí jazyka R vznikl čárový graf časové řady denních uzavíracích cen akcií včetně trendu a mapa denních hodnot stejné časové řady v rámci jednotlivých týdnů a let, po které byly tyto uzavírací ceny sledovány.

```
====================== Dataframe s daty časové řady (prvních několik záznamů) =================
       date year yearmonthf monthf week monthweek weekdayf VIX.Close
1 2010-01-04 2010   Jan 2010    Jan    1         1      Mon     20.04
2 2010-01-05 2010   Jan 2010    Jan    1         1      Tue     19.35
3 2010-01-06 2010   Jan 2010    Jan    1         1      Wed     19.16
4 2010-01-07 2010   Jan 2010    Jan    1         1      Thu     19.06
5 2010-01-08 2010   Jan 2010    Jan    1         1      Fri     18.13
6 2010-01-11 2010   Jan 2010    Jan    2         2      Mon     17.55

============================ Struktura dataframe s daty časové řady ============================
'data.frame':1762 obs. of  8 variables:
 $ date      : Date, format: "2010-01-04" "2010-01-05" ...
 $ year      : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
 $ yearmonthf: Factor w/ 84 levels "Jan 2010","Feb 2010",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ monthf    : Factor w/ 12 levels "Jan","Feb","Mar",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ week      : int  1 1 1 1 1 2 2 2 2 2 ...
 $ monthweek : num  1 1 1 1 1 2 2 2 2 2 ...
 $ weekdayf  : Factor w/ 5 levels "Fri","Thu","Wed",..: 5 4 3 2 1 5 4 3 2 1 ...
 $ VIX.Close : num  20 19.4 19.2 19.1 18.1 ...

================================ Ukončení výpisu výstupů z R-ka ================================
```

Obr. 1 Výpis dat z prostředí jazyka R ve výstupním reportu SAS *(Zdroj: vlastní zpracování)*



Obr. 2 Grafy časové řady z programu R v programu SAS *(Zdroj: vlastní zpracování)*

6 Závěr

Cílem článku bylo ukázat na možné řešení, které by dovolilo integrovat programovací jazyk R do základního modulu BASE softvérové platformy SAS. Programování v jazyku R je tedy možné uskutečnit i v rámci programového systému SAS bez nutnosti využívání dodatečně pořizovaného modulu SAS/IML. Pro spolupráci s jazykem R tedy stačí využít funkcionalitu příkazové řádky, programově ošetřit tvorbu příkazového souboru, vyhodnotit výstupní soubor, případně importovat grafické výstupy nebo datové vstupy či výstupy. Podobné programové rozhraní, které bylo naprogramováno pro spouštění jazyka R, lze v rámci platformy SAS realizovat i pro jiné softvérové nástroje, jako jsou například Python, C++, MatLab a další.

7 Literatura

Eddelbuettel, D., Francois, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8), 1-18.

Grothendieck, G. (2016) *sqldf: Manipulate R data frames using SQL.* R package version 0.4-11. Dostupné na:<https://CRAN.R-project.org/package=sqldf>.

R Development Core Team (2008). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2008.

SAS Institute Inc. (2016a). *9.4 SQL procedure user's guide.* 4. vyd. Cary, NC: SAS Institute Inc., 2016, 470 s. ISBN 978-1-62960-816-7.

SAS Institute Inc. (2016b). *SAS® 9.4 companion for Windows*. 5. vyd. Cary, NC: SAS Institute Inc., 2016, 726 s. Bez ISBN.

SAS Institute Inc. (2016c). *SAS® 9.4 output delivery system: procedures guide.* 3. vyd. Cary, NC: SAS Institute Inc., 2016, 952 s. ISBN 978-1-62960-836-5

SAS Institute Inc. (2018). *SAS/IML 15.1 user's guide.* 5. vyd. Cary, NC: SAS Institute Inc., 2018, 1154 s. Bez ISBN.

Wicklin, R. (2010) *Rediscovering SAS/IML® software: Modern data analysis for the practicing Statistician*. SAS Global Forum 2010, paper 329-2010. Cary, NC: SAS Institute Inc., 2010.

Xia, X.-Q., McClelland, M., Wang, Y. (2010). PypeR, a Python package for using R in Python. *Journal of Statistical Software*, 35(2), 1-8.