FORUM STATISTICUM SLOVACUM Vol XIX (2023) No 1 pp 50-61

Jak stáhnout data z webové stránky pomocí programového systému SAS
How to scrape data from a web page using the SAS software system environment

Roman Pavelka
Štatistický úrad Slovenskej republiky, Odbor metód štatistických zisťovaní, Lamačská cesta 3, 840 05 Bratislava, Slovenská republika
Statistical Office of the Slovak Republic, Statistical Surveys and Methodology Department, Lamačská cesta 3, 840 05 Bratislava, Slovak Republic

roman.pavelka@statistics.sk

Abstrakt: *Internet je bohatý na data a informace. Velká část těchto dat existuje pouze na webových stránkách, které jsou navrženy tak, aby je lidé procházeli a četli. Když výzkumníci-analytici chtějí použít techniky datové vědy k analýze informací z Internetu, chtějí v mnoha případech shromažďovat a analyzovat tato data. A jedinou efektivní cestou k získání dat z Internetu je přejít k tzv. webscrapingu. Příspěvek ukazuje použití programového systému SAS k webscrapingu pro zefektivnění analýzy informací jednodušším propojením s analytickým vybavením systému SAS.*

Abstract: *The Internet is rich in data and information. Much of this data only exists on websites that are designed for people to browse and read. When research analysts want to use data science techniques to analyse information from the Internet, they often want to collect and analyse that data. In addition, the only effective way to get data from the Internet is to go to the so-called web scraping. The paper demonstrates usability of the SAS software system for web scraping to make analysis of information more efficient by simpler connection with the analytical equipment of the SAS system.*

Klíčové slova: *internet, programový systém SAS, syntaktická analýza, webscraping.*

Key words: *internet, parsing, program system SAS, web scraping.*

1 Úvod

Webscraping je zpravidla založen na použití programu k načtení obsahu webové stránky, "prosévání" tohoto načteného obsahu pomocí funkcí syntaktické analýzy textu, tzv. parsování a ukládání informací a dat do strukturovaných datových polí.[5] Výsledkem webscrapingu jsou tedy data uložená do struktury, která usnadňuje jejich analýzu. Webscraping je možný nejen použitím tzv. klasických programů jako je Python nebo R, ale také i programovým systémem SAS. Pro stahování informací z webových stránek uživatelé programů

[5] Parsování (nebo také parsing) je slangový výraz pro syntaktickou analýzu textu. V informatice a v lingvistice se tak nazývá proces analýzy posloupnosti formálních (textových) prvků s cílem určit jejich gramatickou strukturu vůči předem dané (byť ne nutně explicitně vyjádřené) formální (textové) gramatice.

Python a R mohou využívat specializované balíčky jako součásti uvedených programů. Obdobnými funkcionality pro webscraping je vybaven také i programový systém SAS, o čemž mnozí jeho uživatelé často ani neví. Pro účely webscrapingu je již dobře vhodně vybaven základní programový modul SAS (označovaný jako SAS/BASE) bez nutnosti dalších speciálních programových doplňků.

Pro stahování informací z webových stránek programový modul SAS/BASE je vybaven následujícími funkcionalitami (Hemendinger, 2017):

- načtení obsahu zájmové webové stránky s požadovanými daty,
- syntaktická analýza zdrojového obsahu načtené webové stránky (tzv. parsing) a uložení informací do strukturované podoby,
- možnost opakovaného načítaní webových stránek nutných pro sesbírání kompletní sady dat.

Jednotlivé funkcionality programovacího jazyka SAS pro výše uvedené kroky stahování dat z webových stránek jsou uvedeny v tabulce 1.

Tab. 1 Postup pro stahování webových stránek a odpovídající funkcionality jazyka SAS *(Zdroj: vlastní zpracování)*

Etapy stahování obsahu webových stránek	Vhodná funkcionalita jazyka SAS
Získání obsahu webové stránky	Globální příkaz FILENAME a procedura HTTP
Syntaktická analýza (parsing) obsahu stránky	DATA krok s funkcemi syntaktické analýzy textu (např. FIND, SCAN a regulárními výrazy)
Možnost opakovaného načítání webových stránek	Jazyk maker SAS (např. cykly %DO %UNTIL) nebo DATA krok pro generování více iterací načítání/analýzy načteného obsahu

V následujících částech tohoto článku budou jednotlivé postupy při stahování informací z webových stránek a jejich uložení do strukturované podoby vysvětleny podrobněji včetně jednoduchých názorných příkladů programové syntaxe jazyka SAS.

2 Jak stahovat obsah WWW stránek pomocí globálního příkazu FILENAME

Příkaz FILENAME má velmi jednoduchý účel – vytvoření symbolického odkazu na externí soubor nebo zařízení. Samotný příkaz nezpracovává žádná data, nespecifikuje formát nebo tvar datové sady ani přímo nevytváří výstup určitého typu. Přesto je tento jednoduchý příkaz významným konstruktem, který umožňuje programům SAS spolupracovat s okolním prostředím mimo programový systém SAS. Příkaz FILENAME prostřednictvím specifikace vhodného typu zařízení umožňuje symbolicky odkazovat na soubory na externím disku, komunikovat s FTP servery, odesílat e-mailové zprávy a integrovat data

z externích programů a procesů – včetně místního operačního systému a vzdálených webových služeb (Schacherer, 2012).

Podle dokumentace programového systému (SAS Institute, 2017a, s. 101-106) je syntaxe globálního příkazu FILENAME následující:

FILENAME *fileref* <zařízení-typ> <'externí-soubor'> <parametry>.

Toto je základní syntaxe k asociaci externího souboru nebo zařízení k symbolickému odkazu *fileref*. Příkaz také umožňuje výpis nebo vymazání již existujících symbolických odkazů. Úplnou syntaxi globálního příkazu FILENAME včetně volitelných parametrů lze dohledat v originální dokumentaci k programovému systému SAS.

Jakmile je vytvořeno spojení mezi fyzickým souborem nebo zařízením a symbolickým odkazem *fileref*, může být přístup v DATA kroku nebo procedurách k připojenému souboru nebo zařízení prováděn přes symbolický odkaz *fileref* podle potřeby, aniž by bylo nutné znovu spouštět příkaz FILENAME.

V rámci programového systému SAS se pro přístup na internet používá příkaz FILENAME pro typ zařízení URL[6] a SOCKET[7]. Zvládne přenosový protokol HTTP i HTTPS (Hypertext Transfer Protocol Secure) a také webové stránky, které vyžadují ověření[8] (například uživatelské jméno/heslo). Základním typem zařízení v přístupu na webové stránky je zařízení typu URL. Použití zařízení typu URL v globálním příkaze FILENAME je základním pro přístup k obsahu webové stránky, a to výlučně pomocí metody GET[9]. Dotazovací metodu POST[10] umožňuje

[6] URL, zkratka pro Uniform Resource Locator („jednotný lokátor zdroje“), běžně webová adresa je řetězec znaků, který slouží k přesné specifikaci umístění zdrojů informací na Internetu. Nejběžnějším zdrojem je webová stránka (protokol http/https), ale používá se i řada dalších, například sdílené úložiště souborů (ftp) nebo e-mailová schránka (mailto). V internetovém prohlížeči se zadává a zobrazuje v adresním řádku.

[7] Síťový socket (anglicky network socket) je v informatice koncový bod připojený přes počítačovou síť. S rozvojem internetu většina komunikace mezi počítači používá rodinu protokolů TCP/IP. Vlastní přenos zajišťuje IP protokol, a proto je používáno i označení internetový socket. Socket je odkaz, který může program použít při volání síťového rozhraní pro programování aplikací (API), například ve funkci „odeslat tato data na tento socket“.

[8] Při ověření přístupu neboli autentizaci (v překladu basic access authentication) webový server vyzve pomocí protokolu HTTP přistupujícího klienta (typicky webový prohlížeč), aby poslal v rámci požadavku na stránku také autentizační informace (tj. jméno a heslo). Jméno a heslo je zasláno jako jeden textový řetězec. Výsledný řetězec je poté zakódován metodou Base64 a odeslán v rámci HTTP požadavku.

[9] Metoda GET je v informatice jedna z dotazovacích metod HTTP protokolu, kterou webový prohlížeč (klient) získává webovou stránku (nebo jiný objekt – například obrázek) z webového serveru. Označuje též jednu z dvou metod předávání metaproměnných z webového formuláře na webový server, kde mohou být informace zpracovány (například PHP skriptem).

[10] Metoda POST odesílá uživatelská data na server. Používá se například při odesílání formuláře na webu. S předaným objektem se pak zachází podobně jako při metodě GET. Data může odesílat i metoda

příkaz FILENAME výlučně jen s typem zařízení SOCKET (SAS Institute, 2017b, s. 106-111). Nevýhodou tohoto přístupu na internet je však podrobnější znalost sítových internetových protokolů (Allen, 2017).

Příkladem použití příkazu FILENAME pro získání informací z Internetu ilustruje následující sekvence příkazů programovacího jazyka SAS:

```
%let dataCube = as1001rs;     /*Obyvateľstvo a charakteristiky veku*/
%let Obdobi = 2021;           /*Rok*/
%let Ukazatel = UKAZ02,UKAZ03,UKAZ04;     /*Sledovaný ukazatel*/
%let Sex = /*POH0,*/POH1,POH2;      /*Pohlaví – Muži Ženy*/

filename SUSR url
"https://data.statistics.sk/api/v2/dataset/%superq(dataCube)/%superq(Obdobi
)/%superq(Ukazatel)/%superq(Sex)?lang=sk%nrstr(&type)=csv";
proc format ;
      value $pohlavi
      'POH1'='Muž'
      'POH2'='Žena';
      value $VekKat
      'UKAZ02'='do 14 let'
      'UKAZ03'='15 až 64 let'
      'UKAZ04'='nad 65 let';
data work.Tab;
      length Rok $4 Kod1 $8 Kod2 $8 Value 8;
      infile SUSR dlm=";" firstobs=8 dsd;
      input Rok$ Kod1$ Kod2$ value;
      if not missing(Rok) then output;
      format Kod2 $pohlavi.;
run;
filename SUSR clear;
options locale=sk_sk;
proc tabulate data=work.Tab f=nlnum10.0;
      class Rok Kod1 Kod2;
      var Value;
      table
            Rok=''*(Kod2='' all='SPOLU'),
            Kod1=''*Value=''*sum='' / nocellmerge box="&dataCube.";
      format Kod1 $VekKat.;
run;
options locale=en_us;
```

Příkazem FILENAME jsou z webových stránek Statistického úřadu SR načteny vybrané údaje z databáze DATAcube do znakového řetězce pojmenovaného symbolickým jménem SUSR. Znakový řetězec obsahuje údaje věkových charakteristik obyvatelstva z datové struktury (tzv. datové kostky) označené *as1001rs* za období roku 2021 v rozdělení podle pohlaví na muže a ženy. Specifikace požadovaných údajů se předává dotazovací metodou GET pomocí řetězce URL protokolem *https*. Odezvou z webových stránek Statistického úřadu

GET, metoda POST se však používá pro příliš velký objem dat (víc než 512 bajtů, což je velikost požadavku GET), nebo pokud není vhodné přenášená data zobrazit jako součást URL (data předávaná metodou POST jsou obsažena v HTTP požadavku).

SR jsou požadované informace předané ke zpracování do DATA kroku. Pro komunikaci je na straně serveru využíváno *aplikační programové rozhraní* (ve zkratce API), což umožňuje přenášet data ve strukturované podobě a syntaktická analýza přenášeného textu v DATA kroku není potřebná. Tímto způsobem přes API je možné získávat do programového systému SAS strukturovaná data i složitějších rozměrů pro sofistikovanější datové analýzy. Výsledkem činnosti výše uvedené syntaxe je tabulka 2 s rozdělením obyvatelstva podle věkové struktury.

Tab. 2 Věková struktura obyvatelstva získaná příkazem FILENAME z databáze Statistického úřadu Slovenské republiky *(Zdroj: vlastní zpracování)*

as1001rs		do 14 let	15 až 64 let	nad 65 let
2021	Muž	446 781	1 831 411	379 711
	Žena	426 015	1 785 547	565 247
	SPOLU	872 796	3 616 958	944 958

Posloupnost příkazů programového jazyka SAS na další straně je typickým příkladem stahování obsahu webové stránky, sémantická analýza staženého textu a vytvoření strukturovaných dat z obsahu webové stránky o důvěře spotřebitelů *Consumer Confidence*[11].

Globálním příkazem FILENAME se načte obsah webové stránky, jejíž adresa je uvedena za parametrem URL, do textového řetězce a přidruží se k symbolickému názvu *output*. Pomocí tohoto názvu může systém SAS programově přistupovat a manipulovat s načteným obsahem webové stránky v podobě řetězce. Příkazem INFILE se získaný textový řetězec začíná zpracovávat operacemi v DATA kroku. Pro správné strukturování získaného obsahu webové stránky je nutná syntaktická analýza textu. K tomuto účelu slouží DATA krok, jehož funkce na základě značek a orientačních bodů (tzv. tagů) HTML kódu vyhledají požadované informace a vytvářejí jejich požadovanou strukturu. V uvedeném příkladu úlohou DATA kroku je vyhledat tabulku (pokud existuje), sloupce a řádky tabulky a hodnoty v jednotlivých buňkách tabulky. Hledání se provádí s využitím HTML značek - např. značky pro tabulku a značek pro sloupce a řádky tabulky. Pokud je tabulka s hodnotami v řetězci nalezena, úlohou DATA kroku je správně vybrat tyto hodnoty a vytvořit potřebnou datovou strukturu. V případě potřeby – tak, jak je tomu i v uvedeném příkladu – se může vykonat nad staženými i další operace, např. transpozice. Po vykonání výše uvedených činností je vybraný obsah webové stránky uložen jako datový soubor ve formátu SAS do knihovny WORK a je připraven k dalšímu zpracování programovým systémem SAS. Obrázek 1

[11] Ku dnu 28. 03. 2023, citováno 02.04.2023, http://hosting.briefing.com/cschwab/Calendars/EconomicReleases/conf.htm

znázorňuje výsek (tabulku) z dotčené webové stránky, jejíž hodnoty byly výše uvedeným způsobem získány do programu SAS.

```
filename output url
"http://hosting.briefing.com/cschwab/Calendars/EconomicReleases/conf.htm";

data Table_Cells (keep=row_num name value);
      length name value $64;
      infile output _infile_=line;
      input;
      retain landmark_found 0 table_found 0 naming 1 in_row 0 row_num -1;
      if not landmark_found then landmark_found = prxmatch('/Highlights/',
line);
      if not landmark_found then delete;
      if not table_found then table_found = prxmatch('/<table /', line);
      if not table_found then delete;
      array names(20) $8 _temporary_;
      if not in_row then
            if prxmatch('/<tr /', line) then
            do;
                  col_index = 0;
                  in_row = 1;
                  row_num + 1;
                  return;
            end;
      if not in_row then delete;
td:
      rxtd = prxparse('/<td .*?>(.*)<\/td>/');
      if prxmatch(rxtd, line) then
            do;
                  col_index + 1;
                  if naming then
                        do;
                              names(col_index) = prxposn(rxtd, 1, line);
                        end;
                  else
                        do;
                              name = names(col_index);
                              value = prxposn(rxtd, 1, line);
                              value = transtrn(value, ' ', trimn(''));
                              OUTPUT;
                        end;

                  return;
            end;
      in_row = not prxmatch('/<\/tr/', line);
      if naming then if not in_row then naming = 0;
      if prxmatch('/<\/table>/', line) then stop;
run;

filename output clear;

proc transpose data=Table_Cells out=SASData(drop=_NAME_);
      by row_num;
      id name;
      var value;
run;
```

Category	MAR	FEB	JAN	DEC	NOV
Conference Board	104.2	103.4	106.0	109.0	101.4
Expectations	73.0	70.4	76.0	83.4	76.7
Present Situation	151.1	153.0	151.1	147.4	138.3
Employment ('plentiful' less 'hard to get')	38.8	40.7	37.0	34.5	31.5
1 yr inflation expectations	6.3%	6.2%	6.7%	6.6%	7.1%

Obrázek 1 Tabulka, jejíž údaje byly staženy do programu SAS *(Zdroj: vlastní zpracování)*

Datový soubor po webscrapingu ve formátu SAS je znázorněn v tabulce 3.

Tab. 3 Struktura souboru dat obsahující údaje ze staženého obsahu webové stránky *(Zdroj: vlastní zpracování)*

row_num	Category	MAR	FEB	JAN	DEC	NOV
1	Conference Board	104.2	103.4	106.0	109.0	101.4
2	Expectations	73.0	70.4	76.0	83.4	76.7
3	Present Situation	151.1	153.0	151.1	147.4	138.3
4	Employment ('plentiful' less 'hard to get')	38.8	40.7	37.0	34.5	31.5
5	1 yr inflation expectations	6.3%	6.2%	6.7%	6.6%	7.1%

3 Jak stahovat obsah webových stránek využitím procedury HTTP?

Procedura HTTP zabezpečuje realizaci požadavku protokolu HTTP i protokolu umožňující zabezpečenou komunikaci HTTPS. Procedura umožňuje realizovat většinu dotazovacích metod přenosového protokolu. Kromě standardních HTTP metod akceptuje procedura HTTP jakoukoli metodu, která odpovídá normě HTTP/1.1 a která je rozpoznána cílovým webovým serverem. Procedura HTTP také implementuje funkce HTTP/1.1, jako jsou trvalá připojení, ukládání souborů cookies[12], podpora stavových kódů a umožňuje specifikaci typu ověření přístupu. Procedura HTTP dokáže také přistupovat na webové stránky přes server sloužící jako prostředník mezi klientem a cílovým serverem (přes tzv. proxy server) a je schopna zvládnout i případné šifrování internetovské komunikace.

Nejjednodušší syntaxe příkazu pro proceduru HTTP je použití dotazovací metody GET s implicitními parametry:

```
filename response temp;
proc http
      url="http://url.to/web-service-endpoint '<url-options>"
      method=GET
      out=response;
run;
```

[12] Jako cookie (anglicky koláček, oplatka, sušenka) se v protokolu HTTP označuje malé množství dat, která WWW server pošle prohlížeči, který je uloží na počítači uživatele. Při každé další návštěvě téhož serveru pak prohlížeč tato data posílá zpět serveru. Cookies běžně slouží k rozlišování jednotlivých uživatelů, ukládají se do nich uživatelské předvolby, apod.

Argumenty předávané do procedury HTTP jsou:

- **URL**: Koncový bod webové služby, na kterou se dotazuje. Řetězec URL je pro proceduru jediný povinný argument, s případnými parametry pro API,
- **Method**: Metoda dotazu použitá v požadavku. Metoda GET je výchozí hodnota a tento argument lze v tomto případě vynechat a
- **Out**: Odkaz na výstupní řetězec. Odkaz je vytvořen příkazem FILENAME na soubor například v dočasném umístění.

Kompletní popis syntaxe procedury HTTP včetně jednoduchých příkladů použití pro různé dotazovací metody je uveden v originální dokumentaci programového systému SAS (SAS Institute, 2023, s. 1237-1278).

Podkladem pro ukázkový příklad použití procedury HTTP je sociologický průzkum z knihy Psychologie zpravodajské analýzy (Heuer, 1999), kde autor předkládá výsledky experimentu s 23 vojenskými důstojníky NATO zvyklých číst zpravodajské zprávy. Bylo jim předloženo několik vět, jako např.: "Je velmi nepravděpodobné, že ...". Všechny věty byly stejné až na to, že se změnilo slovní vyjádření pravděpodobnosti. Důstojníci byli dotázáni, jakou procentuální pravděpodobnost by přisoudili každé z těchto vět, kdyby si ji přečetli ve zpravodajské zprávě.

Získaná experimentální data jsou uloženy v souboru ve formátu CSV a jsou dostupná na https://github.com/zonination/perceptions/blob/master/probly.csv. Většina proměnných v CSV souboru je z pohledu programového systému SAS nestandardní – například obsahují mezery anebo speciální znaky, které nejsou pro uložení do knihovny SAS vhodné. Proto je nutné pro zpracování systémem SAS názvy proměnných z původního CSV souboru upravit. Upravená data stažená z internetu budou uložena do datového souboru v knihovně WORK a dále vizualizována do krabicového grafu pro jednotlivá vyjádření pravděpodobnosti.

Ukázkový příklad stahování CSV souboru a jeho následného zpracování v systému SAS lze rozdělit do několika kroků:

- Stažení souboru CSV z webové stránky.

```
filename probly temp;

proc http url="https://raw.githubusercontent.com/zonination/perceptions/
master/probly.csv"
method="GET"
out=probly;
run;
```

Stažení CSV souboru je realizováno procedurou HTTP. Příkazem FILENAME je ke staženému souboru v dočasném úložišti asociován odkaz *probly*. S pomocí tohoto přiřazeného odkazu mohou být realizovány další operace v systému SAS.

- Import stažených dat pomocí procedury IMPORT.

  Aby bylo možné importovat soubor, jehož názvy proměnných nesplňují pravidla pro standardní SAS proměnné, je nutné nastavit systémový parametr:

```
options validvarname=any;
```

  Toto nastavení tohoto parametru systému dovoluje importovat soubor s nestandardnímu názvy proměnných z dočasného úložiště do knihovny WORK:

```
proc import file=probly out= work.Perception replace dbms=csv;
run;
```

- Generování nových názvů v souladu s pravidly SAS.

  Originální názvy proměnných stahovaného CSV obsahují mezery a speciální znaky, které nejsou v systému SAS standardně povoleny. Například název proměnné v původním souboru je "Almost Certainly". Proto je nutné z názvu odstranit mezeru a přejmenovat proměnné do podoby bez mezer, např. "AlmostCertainly". Toto přejmenování původních názvů se vykoná pomocí procedury SQL s využitím metadat ze systémové tabulky *sashelp.vcolumn*. Ke zjištění, které proměnné vyžadují úpravu je využita funkce NVALID. Sekvence příkazů nejen, že přejmenuje všechny proměnné, ale z originálních názvů vytvoří štítky (popisky) proměnných. Nové názvy proměnných, resp. štítky proměnných jsou uloženy do makroproměnných *renameStmt*, resp. *labelStmt*. Generování nových názvů a příslušných štítků proměnných zajistí následující sekvence příkazů:

```
proc sql noprint;
  /* zachování originálních názvů jako štítky */
select   cat("'",   trim(name),   "'n",   "=",   "'",   trim(name),   "'")
into    :labelStmt    separated    by    '    '    from    sashelp.vcolumn
where memname="PERCEPTION" and libname="WORK";

  /* přejmenování originálních názvů proměnných */
select   cat("'",   trim(name),   "'n",   "=",   compress(name,   ,   'kn'))
into    :renameStmt    separated    by    '    '    from    sashelp.vcolumn
where memname="PERCEPTION" and libname="WORK"
  /* s vyjímkou těch proměnných, jejichž názvy jsou standardní */
  and not NVALID(trim(name), 'V7');
quit;
```

- Nové názvy a nové štítky v souboru se upraví pomocí procedury DATASETS

Procedura DATASET představuje funkcionalitu, jejíž prostřednictvím lze měnit názvy proměnných, štítky (popisky) proměnných i jejich formáty, aniž by bylo nutné data kompletně přepisovat. Pro názvy proměnných, resp. jejich popisky se uplatní makroproměnné *renameStmt*, resp. *labelStmt*, v nichž jsou uloženy nově vygenerované názvy a štítky (popisky) z přechozího kroku. Příkazová syntaxe pro modifikace staženého souboru je následující:

```
proc datasets lib=work nolist;
  modify Perception / memtype=data;
  label &labelStmt.;
  rename &renameStmt.;
  /* volitelný parametr: report jmen a popisek proměnných */
  contents data= work.Perception nodetails;
quit;
```

Pro ukončení zpracování souboru s nestandardními názvy proměnných je nutné zpětně nastavit systémový parametr na konfiguraci opětovného zpracování standardních názvů[13]:

```
options validvarname=V7;
```

Po uvedených několika krocích je stažený soubor se standardními názvy uložen v knihovně WORK připravený k dalšímu zpracování.

- Vizualizace informace z upraveného CSV souboru proměnných (Wicklin, 2017)

Následující příkazy SAS slouží k vizualizaci obsahu upraveného CSV souboru:

```
/* Výběr sledované statistiky */
%let Stat = Median; /* nebo mean, stddev, qrange, skew, atd */

proc means data=work.Perception &Stat STACKODSOUTPUT;
  ods output Summary=StatOut;
run;

/* uložení názvů proměnných do makroproměnné VarList */
proc sql noprint;
  select Variable into :VarList separated by ' '
        from StatOut order by &Stat;
quit;

/* výpočet kvantilů (Q1, medián, Q3) pro všechny proměnné */
proc means data=work.Perception Q1 median Q3 nolabels;
  var &varList;
run;

/* úpravy vypočítaných statistik do tvaru vhodného pro vizualizaci */
data Wide / view=Wide;
  retain &VarList;        /* nastavení pořadí proměnných */
  set work.Perception;
```

[13] Hodnota parametru V7 označuje, že názvy proměnných datového souboru odpovídají názvům podle pravidel SAS. Toto je výchozí hodnota pro SAS 7 a novější (povoleno je až 32 alfanumerických znaků, názvy proměnných musí začínat abecedním znakem nebo podtržítkem, případné neplatné znaky se změní na podtržítko a každý název sloupce musí být jedinečný).

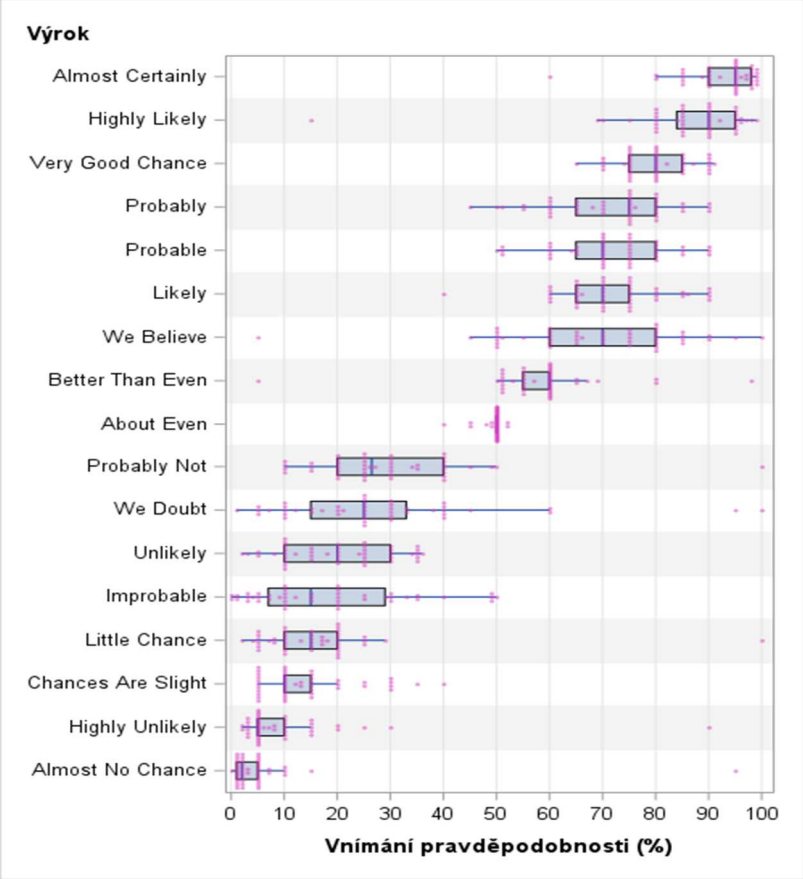
```
    obsNum = _N_;            /* připojení ID ke každému pozorování */
    keep obsNum &VarList;
 run;

/* transpozice původního formátu dat */
proc transpose data=Wide name=VarName
    out=Long(rename=(Col1=_Value_));
    by obsNum;
run;

/* Vytvoření krabicového grafu */
ods graphics / height=700px width=500px subpixel;
title "Vnímání pravděpodobnosti";

proc sgplot data=Long noautolegend;
hbox _Value_ / category=_Label_ nooutliers nomean nocaps;
scatter     x=_Value_     y=_Label_     /     jitter     transparency=0.5
markerattrs=GraphData2(symbol=circlefilled size=4 color=bippk);
yaxis  reverse  discreteorder=data  labelpos=top  labelattrs=(weight=bold)
colorbands=even colorbandsattrs=(color=gray transparency=0.9)
  offsetmin=0.0294 offsetmax=0.0294;
  xaxis grid values=(0 to 100 by 10) labelattrs=(weight=bold);
  label _Value_ = "Vnímání pravděpodobnosti (%)" _label_="Statement";
run;
```

Výsledkem vizualizace údajů ze staženého CSV souboru ilustruje obrázek 2.



Obrázek 2 Vizualizace údajů (graf) ze staženého CSV souboru *(Zdroj: vlastní zpracování)*

4 Závěr

Cílem článku bylo ukázat, že systém SAS již ve svém základním programovém modulu SAS/BASE je vybaven funkcionalitami, které umožňují stahování obsahu webových stránek (webscraping). Ve spojení s analytickými možnostmi tohoto systému tak vzniká efektivní nástroj k analýzám dat z místních i externích zdrojů.

5 Literatura

Allen, K. S. (2017). What every web eeveloper should know about HTTP. 2. vyd. Huntingtown (MD, US): OdeToCode LLC.

Hemendinger, C. (2017) The SAS dummy: How to scrape data from a web page using SAS. [cit. 31.03.2023], https://blogs.sas.com/content/sasdummy/2017/12/04/scrape-web-page-data/.

Heuer, R. J. (1999). Psychology of intelligence analysis. Langley (VA, US): Center for the study of intelligence, CIA.

SAS Institute Inc. (2023). Base SAS® 9.4 procedures guide. "HTTP Procedure". 7. vyd. Cary (NC, USA): SAS Institute Inc.

SAS Institute Inc. (2017a). SAS® 9.4 Global statements: Reference. "FILENAME statement, SOCKET access method". Cary (NC, USA): SAS Institute Inc.

SAS Institute Inc. (2017b). SAS® 9.4 Global statements: Reference. "FILENAME statement, URL access method". Cary (NC, USA): SAS Institute Inc.

Schacherer, C. (2012). The FILENAME Statement: Interacting with the world outside of SAS®. Proceedings of the SAS Global Forum 2012 Meeting. Cary (NC, USA): SAS Institute Inc.

Wicklin, R. (2017) The DO loop: Perceptions of probability. [cit. 10. 04. 2023], https://blogs.sas.com/content/iml/2017/05/03/perceptions-of-probability.html.