

Řešení domácího úkolu 4

4ST525 Pravděpodobnostní a statistické metody v neživotním pojištění

Lubomír Štěpánek

20. dubna 2017

Obsah

Zadání úlohy	1
Řešení úlohy	1
Reference	3
Apendix	3

Zadání úlohy

Vytvořme v R nástroj, který na základě vloženého aktuárského trojúhelníku odhadne metodou *chain ladder* budoucí rezervy.

Vstupem je aktuárský trojúhelník uložený jako `trojuhelnik.csv` a vycházející ze článku Taylor, G. a Ashe, F. R. (1983) *Second moments of estimates of outstanding claims*, Journal of Econometrics, 23, 37-61 [1]. Výstupem by měl být opět nějaký vhodný spustitelný formát typu `.csv`, `.txt` či `.xlsx`, jež bude obsahovat budoucí rezervy v matici s rozměry odpovídajícími vstupnímu trojúhelníku.

Vliv měnící se inflace během let vzniku závazků, které jsou pokryty trojúhelníkem, lze nyní zanedbat.

Řešení úlohy

Úloha byla vyřešena kompletně v prostředí a jazyce R. Řešení bylo rozděleno do několika na sebe navazujících fází.

- (i) Pomocí funkce `read.csv()` byl načten aktuárský trojúhelník, jehož data vychází z práce Taylora a Ashe [1]. Prostřednictvím uživatelsky definované funkce `asTriangle()` byl aktuárský trojúhelník formátově upraven pro další výpočty. Výsledkem je podoba trojúhelníku tak, jak ji ukazuje tabulka 1.
- (ii) Poté byla za použití vlastní funkce `getCumulativeTriangle()` spočtena kumulativní podoba trojúhelníku, jak ji ukazuje tabulka 2.
- (iii) Následně byly pro jednotlivé dvojice sousedních vývojových roků vypočítány vývojové faktory, tj podíly součtů sousedních sloupců v kumulativním trojúhelníku tak, že v rámci dvojice sousedních sloupců byl ten, který je o jeden prvek delší, o daný prvek zkrácen. K výpočtu byla

naimplementována funkce `getDevelopmentFactors()`. Výstupem pak je přehled vývojových faktorů tak, jak je ukazuje tabulka 3.

- (iv) Nakonec byly odhadnuty budoucí hodnoty závazků (rezerv) pomocí již zmíněných vývojových faktorů a funkce `getOutstandingClaimsReserves`, a to tak, že pro každou danou buňku dolního pravého trojúhelníkového pole byl odhad spočítán jako kumulativní součin vedlejšího diagonálního prvku řádku buňky a všech následujících odpovídajících vývojových faktorů konče faktorem odpovídajícím vývojovému roku buňky, viz tabulka 4.

Pro účely prezentace všech trojúhelníků v tomto dokumentu byly použity funkce R-kového balíčku `xtable`, které vrací R-kové objekty v \LaTeX -ovém kódu. Rovněž byly vytvořeny identické výstupy `aktuarske_trojuhelniky.xlsx` pro tabulkový procesor MS Excel[®], a sice pomocí funkcionalit R-kového balíčku `openxlsx`¹. Skript řešící celý problém je možné jednoduše inicializovat a exektovat v R pomocí následujícího příkazu

```
source(
  "https://raw.githubusercontent.com/LStepanek/chain_ladder/master/script.R"
)
```

kdy skript i vzorová vstupní data jsou uložena online v příslušném autorově repozitáři na GitHubu.

počáteční rok \ vývojový rok	1	2	3	4	5	6	7	8	9	10
2003	357848	766940	610542	482940	527326	574398	146342	139950	227229	67948
2004	352118	884021	933894	1183289	445745	320996	527804	266172	425046	
2005	290507	1001799	926219	1016654	750816	146923	495992	280405		
2006	310608	1108250	776189	1562400	272482	352053	206286			
2007	443160	693190	991983	769488	504851	470639				
2008	396132	937085	847498	805037	705960					
2009	440832	847631	1131398	1063269						
2010	359480	1061648	1443370							
2011	376686	986608								
2012	344014									

Tabulka 1: Zadaný aktuárský trojúhelník, částky jsou v dolarech

počáteční rok \ vývojový rok	1	2	3	4	5	6	7	8	9	10
2003	357848	1124788	1735330	2218270	2745596	3319994	3466336	3606286	3833515	3901463
2004	352118	1236139	2170033	3353322	3799067	4120063	4647867	4914039	5339085	
2005	290507	1292306	2218525	3235179	3985995	4132918	4628910	4909315		
2006	310608	1418858	2195047	3757447	4029929	4381982	4588268			
2007	443160	1136350	2128333	2897821	3402672	3873311				
2008	396132	1333217	2180715	2985752	3691712					
2009	440832	1288463	2419861	3483130						
2010	359480	1421128	2864498							
2011	376686	1363294								
2012	344014									

Tabulka 2: Aktuárský trojúhelník v kumulativní podobě, částky jsou v dolarech

¹Dlužno říci, že vytvoření plnohodnotného formátovaného dokumentu `.xlsx` jen pomocí R vyžaduje mimo jiné i instalaci vývojářských balíčků `Rtools`.

sousední vývojové roky	2/1	3/2	4/3	5/4	6/5	7/6	8/7	9/8	10/9
vývojový faktor	3.491	1.747	1.457	1.174	1.104	1.086	1.054	1.077	1.018

Tabulka 3: Vývojové faktory pro sousední vývojové roky

počáteční rok \ vývojový rok	1	2	3	4	5	6	7	8	9	10
2003										
2004										5433719
2005									5285148	5378826
2006								4835458	5205637	5297906
2007							4207459	4434133	4773589	4858200
2008						4074999	4426546	4665023	5022155	5111171
2009					4088678	4513179	4902528	5166649	5562182	5660771
2010				4174756	4900545	5409337	5875997	6192562	6666635	6784799
2011			2382128	3471744	4075313	4498426	4886502	5149760	5544000	5642266
2012	1200818	2098228	3057984	3589620	3962307	4304132	4536015	4883270	4969825	

Tabulka 4: Odhady budoucích nesplacených rezerv, částky jsou v dolarech

Reference

- [1] Taylor, G., Ashe, F. R. (1983) *Second moments of estimates of outstanding claims*, Journal of Econometrics, 23, 37-61.

Appendix

Zde je uveden kód v jazyce R, ve kterém byly zpracovávány veškeré výpočty a generovány tabulky.

```
#####
#####
#####

## instaluji a inicializuji balíčky -----

for(package in c(
  "openxlsx",
  "xtable"
)){

  if(!(package %in% rownames(installed.packages()))){

    install.packages(
      package,
      dependencies = TRUE,
```

```

        repos = "http://cran.us.r-project.org"
    )

}

library(package, character.only = TRUE)

}

## -----

#####

## nastavuji handling se zipováním v R -----

Sys.setenv(R_ZIPCMD = "C:/Rtools/bin/zip")

## -----

#####

## nastavuji pracovní složku -----

while(!"domaci_ukol_4.R" %in% dir()){
    setwd(choose.dir())
}

mother_working_directory <- getwd()

## -----

#####
#####
#####

## helper funkce -----

isTriangle <- function(

    triangle_data

){

    # '''

```

```

# Kontroluje, zda čtvercová matice "triangle_data" může být
# smysluplným aktuárským trojúhelníkem.
#
# Čtvercovou matici "triangle_data" považujeme za smysluplný
# aktuárský trojúhelník, pokud má v horním levém trojúhelníkovém
# poli závazky tak, že v řádcích jsou závazky vzniklé v roce,
# který je uveden řádkem, a vyřešeném v roce, který
# uvozuje sloupec; dolní pravé trojúhelníkové pole je tvořeno
# nedostupnými hodnotami (NA).
#
# Zároveň je požadováno, aby měla data "triangle_data" smysluplné
# popisky.
# '''

if(dim(triangle_data)[1] != dim(triangle_data)[2]){

    return(FALSE)

}

if(any(
    is.na(
        as.numeric(
            triangle_data[upper.tri(
                as.matrix(
                    triangle_data,
                    nrow = triangle_data[1]
                ),
                diag = TRUE
            )[, dim(triangle_data)[2]:1]]
        )
    )
)){

    return(FALSE)

}

if(!all(
    is.na(
        as.numeric(
            triangle_data[lower.tri(
                as.matrix(
                    triangle_data,
                    nrow = triangle_data[1]
                ),
                diag = FALSE
            )
        )
    )
))

```

```

        )[, dim(triangle_data)[2]:1]]
    )
  )
)) {

  return(FALSE)

}

if(is.null(colnames(triangle_data))) {

  return(FALSE)

}

if(is.null(rownames(triangle_data))) {

  return(FALSE)

}

if(any(is.na(
  as.integer(
    gsub(
      "(.*?)([0-9]+)(.*?)",
      "\\2",
      colnames(triangle_data)
    )
  )
))) {

  return(FALSE)

}

if(any(is.na(
  as.integer(
    gsub(
      "(.*?)([0-9]+)(.*?)",
      "\\2",
      rownames(triangle_data)
    )
  )
))) {

  return(FALSE)

```

```

}

if(length(
  unique(
    as.integer(
      gsub(
        "(.*?)([0-9]+)(.*?)",
        "\\2",
        colnames(triangle_data)
      )
    )[2:length(colnames(triangle_data))] - as.integer(
      gsub(
        "(.*?)([0-9]+)(.*?)",
        "\\2",
        colnames(triangle_data)
      )
    )[1:(length(colnames(triangle_data)) - 1)]
  )
) != 1){

  return(FALSE)

}

if(length(
  unique(
    as.integer(
      gsub(
        "(.*?)([0-9]+)(.*?)",
        "\\2",
        rownames(triangle_data)
      )
    )[2:length(rownames(triangle_data))] - as.integer(
      gsub(
        "(.*?)([0-9]+)(.*?)",
        "\\2",
        rownames(triangle_data)
      )
    )[1:(length(rownames(triangle_data)) - 1)]
  )
) != 1){

  return(FALSE)

}

return(TRUE)

```

```

}

## -----

asTriangle <- function(

  triangle_data

){

  # '''
  # Čtvercovou matici "triangle_data", která je smysluplným
  # aktuárským trojúhelníkem, vrací ve formátu, kde horní levé
  # trojúhelníkové pole jsou závazky tak, že v řádcích jsou závazky
  # vzniklé v roce, který je uveden řádkem, a vyřešeném v roce, který
  # uvozuje sloupec; dolní pravé trojúhelníkové pole je tvořeno
  # nedostupnými hodnotami (NA). Zároveň jsou přidány popisky popisků.
  # '''

  if(isTriangle(triangle_data)){

    return(
      matrix(
        apply(triangle_data, 2, as.numeric),
        nrow = dim(triangle_data)[1],
        dimnames = list(
          "origin" = gsub(
            "(.*?)([0-9]+)(.*?)",
            "\\2",
            rownames(triangle_data)
          ),
          "development" = gsub(
            "(.*?)([0-9]+)(.*?)",
            "\\2",
            colnames(triangle_data)
          )
        )
      )
    )
  }

}

## -----

```



```

getCumulativeTriangle <- function(

  my_triangle

){

  # '''
  # Vrací pro objekt "my_triangle", který je aktuárským trojúhelníkem,
  # kumulativní závazky pro každý rok vzniku.
  # '''

  if(isTriangle(my_triangle)){

    return(
      asTriangle(
        t(apply(
          my_triangle,
          1,
          cumsum
        ))
      )
    )

  }

}

```

```

getDevelopmentFactors <- function(

  cumulative_triangle

){

  # '''
  # Vrací pro objekt "cumulative_triangle", který je kumulativním
  # aktuárským trojúhelníkem, vývojové faktory pro všechny
  # sousední dvojice development let.
  # '''

  if(isTriangle(cumulative_triangle)){

    my_sums_wi_diag <- apply(
      cumulative_triangle,
      2,

```

```

        sum,
        na.rm = TRUE
    )

    my_sums_wo_diag <- apply(
        cumulative_triangle,
        2,
        sum,
        na.rm = TRUE
    ) - rev(
        diag(
            cumulative_triangle[,
                dim(cumulative_triangle)[2]:1
            ]
        )
    )

    return(

        setNames(
            my_sums_wi_diag[2:length(my_sums_wi_diag)] /
            my_sums_wo_diag[1:(length(my_sums_wo_diag) - 1)],
            paste(
                colnames(cumulative_triangle)[
                    2:length(colnames(cumulative_triangle))
                ],
                colnames(cumulative_triangle)[
                    1:(length(colnames(cumulative_triangle)) - 1)
                ],
                sep = "/"
            )
        )
    )

}

}

## -----

getOutstandingClaimsReserves <- function(

    cumulative_triangle

){

```

```

# '''
# Vrací pro objekt "cumulative_triangle", který je kumulativním
# aktuárským trojúhelníkem, odhady budoucích (nesplacených) rezerv.
# '''

if(isTriangle(cumulative_triangle)){

  my_matrix <- matrix(
    data = diag(rev(
      diag(
        cumulative_triangle[,
          dim(cumulative_triangle)[2]:1
        ]
      )
    ))[,
      dim(cumulative_triangle)[1]:1
    ],
    nrow = dim(cumulative_triangle)[1],
    dimnames = list(
      "origin" = rownames(cumulative_triangle),
      "development" = colnames(cumulative_triangle)
    )
  )

  my_matrix[upper.tri(my_matrix)[, dim(my_matrix)[2]:1]] <- 1
  my_matrix[lower.tri(my_matrix)[, dim(my_matrix)[2]:1]] <-
  t(
    replicate(
      dim(cumulative_triangle)[1],
      c("1/0" = 1, getDevelopmentFactors(cumulative_triangle))
    )
  )[lower.tri(my_matrix)[, dim(my_matrix)[2]:1]]

  output <- t(
    apply(
      my_matrix,
      1,
      cumprod
    )
  )

  output[upper.tri(output, diag = TRUE)[, dim(output)[2]:1]] <- NA

  return(
    output
  )
}

```

```

    }
}

## -----

#####
#####
#####

## nahrávám data -----

setwd(mother_working_directory)

my_triangle <- asTriangle(read.table(
  file = "trojuhelnik.csv",
  header = TRUE,
  sep = ",",
  row.names = 1
))

my_codebook <- read.table(
  file = "codebook.txt",
  header = TRUE,
  sep = ";",
  colClasses = "character"
)

## -----

#####

## počítám kumulativní formu trojúhelníku -----

cumulative_triangle <- getCumulativeTriangle(my_triangle)

## počítám vývojové faktory -----

development_factors <- t(
  as.matrix(getDevelopmentFactors(cumulative_triangle))
)

## počítám budoucí nesplacené závazky (rezervy) -----

```

```

outstanding_claims_reserves <- getOutstandingClaimsReserves(
  cumulative_triangle
)

## -----

#####

## TeXový výstup -----

for(my_table_name in c(
  "my_triangle",
  "cumulative_triangle",
  "development_factors",
  "outstanding_claims_reserves"
)){

  # '''
  # Vrací TeXový kód postupně pro všechny zadané tabulky.
  # '''

  my_table <- get(my_table_name)

  print(xtable(my_table, align = rep("", ncol(my_table) + 1),
    digits = if(
      my_table_name == "development_factors"
    ){3}else{0}),
    floating = FALSE, tabular.environment = "tabular",
    hline.after = NULL, include.rownames = TRUE,
    include.colnames = TRUE
  )

}

## -----

#####

## vytvářím i excelový výstup -----

#### vytvářím sešit -----

actuary_triangles <- createWorkbook()

```

```

for(my_table_name in c(
  "my_triangle",
  "cumulative_triangle",
  "development_factors",
  "outstanding_claims_reserves"
)){

  # '''
  # Vrací excelový sešit s listy obsahujícími jednotlivé tabulky.
  # '''

  my_table <- get(my_table_name)

  ## vytvářím list -----

  addWorksheet(
    wb = actuarial_triangles,
    sheetName = my_codebook[
      my_codebook$variable_name == my_table_name,
      "variable_czech_name"
    ]
  )

  ## ukládám do sešitu data -----

  writeData(
    wb = actuarial_triangles,
    sheet = my_codebook[
      my_codebook$variable_name == my_table_name,
      "variable_czech_name"
    ],
    rowNames = TRUE,
    colNames = TRUE,
    x = my_table
  )

  ## nastavuji automatickou šířku sloupce -----

  setColWidths(
    wb = actuarial_triangles,
    sheet = my_codebook[
      my_codebook$variable_name == my_table_name,
      "variable_czech_name"
    ],
    cols = 1:(dim(my_table)[2] + 1),

```

```

        widths = "auto"
    )

    ## vytvářím dva své styly - jednak tučné písmo, jednak písmo zarovnané
    ## doprava v rámci buňky -----

    my_bold_style <- createStyle(textDecoration = "bold")
    right_halign_cells <- createStyle(halign = "right")

    addStyle(
        wb = actuarial_triangles,
        sheet = my_codebook[
            my_codebook$variable_name == my_table_name,
            "variable_czech_name"
        ],
        style = my_bold_style,
        rows = c(1:(dim(my_table)[1] + 1), rep(1, dim(my_table)[2] + 1)),
        cols = c(rep(1, dim(my_table)[1] + 1), 1:(dim(my_table)[2] + 1))
    )

    addStyle(
        wb = actuarial_triangles,
        sheet = my_codebook[
            my_codebook$variable_name == my_table_name,
            "variable_czech_name"
        ],
        style = right_halign_cells,
        rows = 2:(dim(my_table)[1] + 1),
        cols = 2:(dim(my_table)[2] + 1),
        gridExpand = TRUE
    )
}

## -----

#### ukládám workbook -----

setwd(mother_working_directory)

saveWorkbook(
    wb = actuarial_triangles,
    file = "aktuarske_trojuhelniky.xlsx",
    overwrite = TRUE
)

```

```
## -----  
  
#####  
#####  
#####
```