

Praktická časť diplomovej práce

Sprievodca ekfmetriou

Sprievodca ekonometriou má za úlohu priblížiť Vám ekonometriu, a pomôcť Vám jej porozumieť. Sprievodcu píšem ako študent, ktorý sa ekonometriu začal učiť sám, a sám si prešiel zdĺhavým procesom bádania a usmerňovania. Sprievodca je zostrojený ako-tak súbežne s osnovou a zadaniami, ktoré obdržíte na hodine. Nebudeme sa konkrétne držať vypracovania zadanií, ale skôr princípmi, z ktorých zadania ťažia. Mnoho študentov tento predmet nezaujíma, a zadania vypracujú okopírovaním postupov starších spolužiakov, nuž, pochopiť ekonometriu a jej postupy nie je vôbec jednoduché, a dokážem pochopiť, keď si študenti hľadajú skratky. Na druhú stranu, ekonometria predstavuje skvelú vstupnú bránu do sveta analytiky. Človek je zavalený Machine Learningom, Data Sciencom a AI-čkom, nuž až po sfúknutí pozlátko zistí, že je to zmes matematiky, štatistiky a počítačovej vedy (CS – computer science). Ekonometria je teda skvelou výhovorkou, ako oprášiť matematiku, doučiť sa štatistiku, a naučiť sa troška programovania. R-ko sa môže zdať ako jazyk, ktorý žije v tieni Pythonu, avšak, akoby nejedna Dominika vedela povedať, netreba sa nechať voviesť do omylu. R-ko je najvhodnejší programovací jazyk pre štatistikov, Google ho zahrnul do najnovších kurzov Google Analytics. Mojou úlohou je pomôcť Vám prekonať problém, ktorý som na začiatku svojej cesty ekonometriou vôbec nepovažoval za problém, a to množstvo materiálov, ktoré zavalí študenta. Pomôžem Vám postupne zložiť skladačku konceptov a teórií, na ktorých ekonometria stojí. Náročnosť prezentovania konceptov bude prispôbená. Nemá zmysel vysvetľovať odvodzovanie každého estimátora. Cieľom je poskytnúť všeobecný náhľad ekonometrie, a pomôcť Vám pochopiť, a teda príjemnejšie zvládnuť predmet Ekonometria.

Čo nás čaká:

- Základy programovania v R
- Intuitívny prehľad štatistických konceptov
- Ekonometrické techniky

Základy programovania v R

Sprievodca je interaktívny, teda začneme stiahnutím a inštaláciou R a RStudio. R má samo o sebe programovacie prostredie, avšak dnešným štandardom je používanie integrovaného vývojového prostredia (IDE) v podobe RStudio.

Aritmetické operátory

Podme teda rovno na vec. Začneme základnými funkciami. R môžeme používať ako kalkulačku, teda za pomoci klasických aritmetických operátorov môžeme sčítat, odčítat, násobiť, deliť či umocňovať:

```
5 + 5
```

```
## [1] 10
```

```
5 - 5
```

```
## [1] 0
```

```
5 * 5
```

```
## [1] 25
```

```
5 / 5
```

```
## [1] 1
```

```
5^2
```

```
## [1] 25
```

R-ko skrýva mnoho ďalších operátorov:

```
# zobrazí zvyšok z delenia
```

```
5 %% 2
```

```
## [1] 1
```

My sa budeme zapodievať len tým, s čím sa na cvičeniach stretneme. Našou úlohou nie je naučiť sa dokonalo ovládať R, ale naučiť sa používať ho v dostatočnej miere, aby sme s ním zvládli to, čo v najbližšej dobe budeme potrebovať.

Balíky

Okrem základných operátorov budeme využívať aj funkcie:

```
mean(2, 4, 6)
```

```
## [1] 2
```

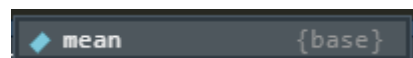
```
abs(-5)
```

```
## [1] 5
```

```
sqrt(8)
```

```
## [1] 2.828427
```

Tieto funkcie sa nachádzajú v balíkoch, ktoré si môžeme predstaviť ako také Addony. R figuruje balíkmi, ktoré sú predinštalované, a zahŕňajú najpoužívanejšie a najzákladnejšie funkcie. Vyššie použité funkcie sa nachádzajú v balíku `base`. To, v akom balíku sa funkcia nachádza zistíte po napísaní funkcie



to však len za predpokladu, že už máte balík nainštalovaný. Ak narazíte na názov funkcie, ktorú chcete použiť, avšak nemáte nainštalovaný balík a chcete zistiť jeho názov, buď si funkciu zadajte do Google, alebo do konzoly napíšete:

```
# ?meno funkcie
```

```
?base
```

```
# ??meno funkcie
```

```
??base
```

```
# help(menofunkcie)
```

```
help(base)
```

My budeme často využívať predinštalované balíky **base** a **stats**, avšak za pochodu si budeme inštalovať aj ďalšie balíky, s ktorými sa na cvičeniach stretnete. Nový balík je potrebné prv nainštalovať a potom ho načítať do prostredia.

```
# stiahneme a naštálujeme pomocou R konzoly a funkcie install.packages()
```

```
# ! názov balíka je citlivý na veľkosť písma
```

```
# ! názov musí byť v úvodzovkách
```

```
install.packages("fBasics")
```

```
# po nainštalovaní máme balík stiahnutý v našom PC, a tento príkaz už viac nepoužívame
```

```
# ak však chceme funkcie z balíka použiť, musíme po zapnutí R-ka balík načítať príkazom
```

```
library(fBasics)
```

```
# ! tu už úvodzovky nie sú potrebné
```

Pri inštalovaní názov balíka zabalíme do úvodzoviek. Pri jeho načítaní pomocou `library()` už úvodzovky nepíšeme. Na pohovor si vezmeme oblek (úvodzovky), ale po prijatí už chodíme do práce bez obleku.

Objekty a vektory

Často chceme vyrátané výsledky znova použiť, a preto by bolo vhodné si ich niekde uložiť. Na ukladanie a uskladnenie výsledkov slúžia objekty. Každý objekt má meno a obsah. Meno si môžeme zadať akékoľvek, musí však:

- začínať malým alebo veľkým písmenom a nie číslom
- obsahovať iba čísla, písmená alebo niektoré špeciálne znaky ako "." či "_".

Nezabúdajme, že R je case sensitive (rozlišuje veľké a malé písmená).

Povedzme že chceme vytvoriť objekt **a** a priradiť mu hodnotu $2 + 2$. Na priradenie obsahu je možné použiť "=" , avšak štandardom je používanie "<-".

*Znak <- nie je nutné písať dvoma znakmi, používa sa na to skratka “**ľavý Alt**” a “-”. Na SK klávesnici nájdeme znak naľavo od pravého Shiftu, na EN klávesnici zvyčajne naľavo od Backspacu. Osobne programujem s EN klávesnicou, nech som použiteľný v akomkoľvek štáte bez potreby inštalovať SK klávesnicu.*

```
# Priradíme teda objektu "a" výsledok "2 + 2".
```

```
a <- 2 + 2
```

```
# Po napísaní názvu objektu do konzoly nám konzola ukáže už len výsledok.
```

```
a
```

```
## [1] 4
```

```
# Nové priradenie hodnoty starému objektu prepíše starú hodnotu.
```

```
a <- 5 + 5
```

```
a
```

```
## [1] 10
```

```
#nainštalovať len jeden balík a potom keď prejdeme vektory tak c(viac balíkov)
```

Import údajov

Table Header

->	First Header
Content Cell	Content Cell
Content Cell	Content Cell