

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra geoinformatiky

**TESTOVÁNÍ PROPOJENÍ A PROSTOROVÉ
VIZUALIZACE V PROSTŘEDÍ THE R PROJECT
FOR STATISTICAL COMPUTING S ESRI
ARCGIS FOR DESKTOP**

Bakalářská práce

Miloš NEKUŽA

Vedoucí práce: doc. Mgr. Pavel Tuček, Ph.D.

Olomouc 2017

Geoinformatika a geografie

ANOTACE

Bakalářská práce se zaměřuje na testování propojení mezi softwary ESRI ArcGIS for Desktop a The R Project for statistical computing. Testování je prováděno pomocí vytváření skriptů. Každý skript je průběžně testován, zda se nevyskytla chyba.

Cílem práce je vytipovat a vytvořit soubor úloh, které demonstrují propojení mezi ArcGIS for Desktop s The R Project. Druhým cílem práce je vytvořit a strukturovat obecný návod, jak vytvořit vlastní úlohu.

Vytvořené úlohy jsou strukturovány do dvou částí. Jsou jimi vytvoření R-Skriptu a vytvoření toolboxu. R-Skript je dále strukturován na použité knihovny, načtení dat, úprava dat a samostatné řešení příkladu.

KLÍČOVÁ SLOVA

R-Bridge, ArcGIS, R Project, Propojení

Počet stran práce: 52

Počet příloh: 8 (z toho 2 volné a 1 elektronická)

ANOTATION

The bachelor thesis focuses on testing the links between ESRI ArcGIS for Desktop and The R Project for statistical computing. Testing is done by creating scripts. Each script is continuously tested to see if there was an error.

The aim of the thesis is to identify and create a set of tasks that demonstrate the connection between ArcGIS for Desktop and The R Project. The second objective of the thesis is to create and structure general instructions on how to create a role.

The tasks created are structured into two parts. They are creating an R-script and creating a toolbox. The R-script is further structured on the used libraries, data retrieval, data editing, and stand-alone solution.

KEYWORDS

R-Bridge, ArcGIS, R Project

Number of pages 52

Number of appendixes 8

Prohlašuji, že

- bakalářskou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.

- jsem si vědom, že na moji bakalářskou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou/diplomovou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk bakalářské práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé bakalářské práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne (15.8.2017)

Miloš Nekuža

Děkuji vedoucímu práce doc. Mgr. Pavlovi Tučkovi, Ph.D. za podněty, rady a připomínky při vypracování práce.

Za konzultace a trpělivost při tvorbě skriptů děkuji Romanu Hittlovi.

zadání

zadáni

OBSAH

SEZNAM POUŽITÝCH ZKRATEK	9
ÚVOD	10
1 CÍLE PRÁCE.....	11
2 METODY A POSTUPY ZPRACOVÁNÍ.....	12
2.1 Postup zpracování.....	12
2.2 Použité programy a data	13
2.2.1 Použité programy	13
2.2.2 Použítá data.....	14
3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	15
3.1 R-Bridge	15
3.2 Balíček arcgisbinding.....	16
3.2.1 Základní funkce pro čtení a zápis	16
3.2.2 Funkce pro konverzi dat.....	17
3.2.3 Funkce nadstavbových prvků R skriptu.....	18
3.3 Referenční příklad	19
4 R -> ARCGIS.....	22
4.1 Načtení rasteru.....	22
4.2 Načtení SHP	22
5 ARCGIS -> R.....	24
5.1 Příčný profil.....	24
5.2 3D Grafy.....	29
5.3 Kriging	34
5.4 Wordcloud.....	39
6 VÝSLEDKY	44
6.1 Obecné vytvoření příkladu	44
6.2.1 R-Skript.....	44
6.2.2 Toolbox.....	45
7 DISKUZE.....	49
8 ZÁVĚR	51
POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE	
PŘÍLOHY	

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
ESRI	Environmental System Research Institute
SW	software
PNG	Portable Network Graphic
3D	Three dimensional
SHP	Shapefile
RAM	Random access memory
DMR	Digitální model reliéfu
WKT	Well-know text
HTML	Hyper Text Markup Language

ÚVOD

ArcGIS for Desktop je nástroj GIS pro tvorbu, správu a editaci dat přes prostorové analýzy až ke tvorbě map. Na druhou stranu R Project for statistical computing je software, který slouží výhradně pro statistické a grafické účely. Tedy dva softwary, které mají některé funkce stejné, ale každý z nich se zabývá jinou problematikou.

Tato práce se zaměřuje na možnosti propojení mezi těmito dvěma softwary a jeho následný popis, jak propojení realizovat. Funkčnost propojení je následně testována několika příklady, které byly vymyšleny tak, aby zahrnovali propojení jak z ArcGIS for Desktop do The R Project for Statistical Computing, ale také obráceně.

K realizaci propojení, je využito knihovny R-Bridge. R-Bridge je knihovna, která po spuštění instalačních skriptů z prostředí toolboxu v ArcGIS for Desktop nainstaluje funkce z ArcGIS for Desktop do The R Project for statistical Computing a obráceně. V podstatě uživatel má možnost využívat funkce z ArcGIS for Desktop v aplikaci The R Project for Statistical Computing a funkce z The R Project for Statistical Computing v aplikaci ArcGIS for Desktop. Pomocí tohoto propojení má uživatel možnost využít v ArcGIS for Desktop statistických a vizualizačních nástrojů a v The R Project for Statistical Computing má uživatel možnost využít tvorbu správu a editaci dat až po tvorbu map.

Důležitým faktorem tohoto propojení je, že The R Project for Statistical Computing je volně dostupný software. Pomocí tohoto propojení se otevírá možnost pro vytvoření funkcí, které obsahují oba softwary, ale v ArcGIS for Desktop je tato funkce placenou extenzí.

Bylo vybráno celkem sedm úloh, na kterých je testována funkčnost propojení pomocí knihovny R-Bridge. Tyto úlohy obsahují jeden příklad referenční, tedy příklad, který již byl zpracován a je z něj vycházeno jako z předlohy. Dále obsahují čtyři úlohy, které využívají propojení z aplikace The R Project for Statistical Computing do aplikace ArcGIS for Desktop a dvě úlohy využívající propojení z aplikace ArcGIS for desktop do aplikace The R Project for Statistical Computing.

1 CÍLE PRÁCE

Cílem práce je vytvořit soubor cvičných úloh demonstrujících propojení dvou GIS systémů, a to konkrétně The R project a Esri ArcGIS for Desktop. Úkolem je vytipovat názorné úlohy a na vhodných datových sadách demonstrovat možnosti takového propojení. Výsledkem práce pak bude soubor tematicky řazených úloh, kde student uživateli demonstruje jednak sílu takového propojení a jednak snadnost takového řešení. Veškeré shromážděné úlohy budou rovněž publikovány vhodnou interaktivní formou.

Fáze práce:

- 1) Studium propojení Esri ArcGIS Desktop a The R project
- 2) Shromáždit dostupná data podkladových vrstev a případových studií
- 3) Vytipovat konkrétní úlohy na kterých bude vše demonstrováno
- 4) Interaktivní prezentace úloh
- 5) Vlastní tvorba práce

Student vyplní údaje o všech datových sadách, které vytvořila nebo získala v rámci práce, do Metainformačního systému Katedry geoinformatiky UP a současně provede zálohu údajů ve formě validovaného XML souboru. Celá práce (text, přílohy, výstupy, zdrojová a vytvořená data, XML soubor) bude odevzdána v digitální podobě na DVD a text práce s vybranými přílohami bude odevzdán ve dvou svázaných výtiscích na sekretariát katedry ve stanoveném termínu. O diplomové práci student vytvoří webovou stránku v souladu s pravidly dostupnými na stránkách katedry. Student vytvoří také poster informující o výsledcích práce v anglickém jazyce na formátu A1. Práce bude zpracována podle zásad dle šablony dostupné na webových stránkách katedry.

2 METODY A POSTUPY ZPRACOVÁNÍ

2.1 Postup zpracování

Nastudování literatury a možnosti propojení

V této části práce bylo nutné zjistit a nastudovat vše k propojení mezi R Project a ArcGIS for Desktop. Hlavním úkolem bylo nalézt existující propojení, aby mohly být realizovány další části práce.

Pro tento účel byla vybrána knihovna R-Bridge, která pomocí instalace skriptů propojí aplikaci R project s aplikací ArcGIS for Desktop a vytvoří možnost využívat funkce z aplikace R Project v ArcGIS for Desktop a obráceně.

Vytipování úloh

V další fázi práce bylo nutné vybrat úlohy, které by se hodily pro daný účel testování propojení. Tedy úlohy, které se v jedné či druhé aplikaci nevyskytují, nejsou zpracované na takové úrovni, nebo je zde možnost porovnání těchto úloh. Úlohy se dělí na dvě části, a to na úlohy, které pracují s propojením z ArcGIS for Desktop do R Project a na úlohy, které pracují s propojením z R Project do ArcGIS for Desktop.

Jako první úloha, která využívá propojení z aplikace R Project do aplikace ArcGIS for Desktop byl vybrán Profil. Tedy funkce, kde ze vstupních dat, kterými jsou raster a liniová vrstva, je získán výsledný graf profilu. Tato úloha se od všech ostatních liší ještě tím, že první část funkcí je použita z ArcGIS for Desktop a druhá část funkcí je pomocí propojení použita z R Project. Tímto se stává tato úloha komplexní.

Druhou úlohou, která využívá stejného propojení jako u Profilu je úloha 3D grafy. Tato úloha reprezentuje vizualizační typ příkladu. Z toho důvodu lze výsledný 3D graf použít jako nadstavbový prvek například k mapě, nebo vložit přímo do mapy.

Třetí úlohou, vybranou, aby reprezentovala propojení mezi R Project a ArcGIS for Desktop je interpolační úloha Kriging. Tuto interpolační metodu obsahuje jak aplikace R Project, tak aplikace ArcGIS for Desktop, z toho důvodu patří úloha mezi porovnávací příklad.

Čtvrtou úlohou, tedy poslední, která využívá propojení z R Project do ArcGIS for Desktop je úloha Wordcloud. Zmíněný příklad by měl sloužit jako vizualizační nadstavbový kompoziční prvek.

Pátá úloha se již zabývá implementací funkcí z aplikace ArcGIS for Desktop do R Project. Touto úlohou je načtení rasteru do R Project, aby mohl uživatel dále pracovat s rasterem, či daty rasteru.

V šesté úloze je stejně jako u páté využita implementace funkcí z ArcGIS for Desktop do R Project. V tomto případě je do R Project načítán shapefile, se kterým se následně mohou provádět další operace, jako například analýzy nad daty v atributové tabulce, zapisování nových atributů do SHP a mnoho dalších operací.

Referenční příklad

Pro vytvoření vytipovaných úloh bylo nutné nejdříve rozkódovat již vytvořený příklad, a zjistit potřebné funkce, ze kterých byl příklad vytvořen. V této části práce byl také nastudován balíček `arcgisbinding`, který obsahuje funkce důležité při realizaci

propojení. Pro tento účel byla vybrána již vytvořená úloha „Combining ArcGIS and R – Clustering Toolbox“, tedy úloha, která se zabývá vytvořením clusterů (Combining ArcGIS and R - Clustering Toolbox, 2016).

Vytvoření vlastních úloh

V tomto kroku práce již byla nastudována veškerá dokumentace a byl pochopen referenční příklad. Ještě, než bylo možné se pustit do vytváření úloh, bylo potřeba nastudovat, jak se dané úlohy řeší v aplikaci R Project. Zde bylo nutné zjistit nejen funkce, které se v dané úloze vyskytují, ale také knihovny, které bylo nutné načíst, a popřípadě i stáhnout, v aplikaci R Project.

Úlohy byly vytvářeny postupně, každá část vytvořeného R-Skriptu byla ihned otestována, zda funguje. R-Skript byl nahrán do skriptu, který byl součástí toolboxu v aplikaci ArcGIS for Desktop. Ještě, než bylo započato vytváření R-Skriptu, tak bylo nutné stanovit vstupní a výstupní proměnné a tím také poupravit kód, který danou problematiku řešil v aplikaci R Project. Dalším krokem při vytváření R-Skriptu bylo načtení balíčků, poté definování proměnných, načtení dat, jejich úprava a v poslední řadě samotné řešení příkladu. Po dokončení R-Skriptu, byly jednotlivé funkce popsány přímo v R-Skriptu pomocí komentářů. Posledním krokem byla vizualizace prostředí skriptu v toolboxu aplikace ArcGIS for Desktop. Pokud v průběhu vytváření R-Skriptu došlo k problému, tak dokud nebyl vyřešen, nebylo postupováno dále ve vytváření R-Skriptu. Po každé změně kódu v R-Skriptu byl R-Skript aktualizován a otestován novým spuštěním.

Vytvoření obecného návodu

Poslední fází a zároveň tou nejdůležitější je vytvoření obecného návodu. Obecný postup, jak si uživatel může vytvořit vlastní úlohu bylo od začátku hlavním cílem této práce. Bez vytvořených vlastních různých druhů úloh by nebylo možné zpracovat objektivně tuto část práce.

K obecnému návodu bylo dospěno pomocí všech vytvořených úloh, referenčního příkladu a nastudovaných materiálů. Ve výsledku by tento návod měl podchytit každou úlohu, kterou by uživatel chtěl vytvořit.

Obecný návod je rozdělen do dvou částí. Je to vytvoření R-Skriptu a vytvoření toolboxu.

2.2 Použité programy a data

Do kategorie programů patří Esri ArcGIS for Desktop, The R Project for Statistical Computing a PSPad editor pro vytvoření webových stránek. Do kategorie použitých dat spadají všechna data, která byla použita pro testovací účely při vytváření vlastních úloh.

2.2.1 Použité programy

ESRI ArcGIS for Desktop

Konkrétně byla využita verze ArcGIS for Desktop 10.4.1 a to s podporou Python. U tohoto programu byl hlavně využito prostředí toolboxu, do kterého byly R-Skripty nahrány a poté spuštěny při testování funkčnosti. Při vytváření prvního příkladu, tedy

Profilu, byla také využita funkce Stack Profile, aby mohla být získána tabulka s hodnotami ze které je následně vytvořen Profil.

The R Project for Statistical Computing

Byla využita verze R 3.3.3 ve 32 bitové verzi a také prostředí R studio. V R Project byly vytvořeny R-Skripty, které byly následně načteny do skriptu toolboxu v aplikaci ArcGIS for Desktop.

PSPad 4.6.2

Freeware editor, který byl použit při vytváření Hyper Text Markup Language (HTML) kódu webových stránek, na kterých je umístěna bakalářská práce.

2.2.2 Použitá data

Pro tuto práci byla použita datová sada ArcČR 500. Data byla použita na testování jednotlivých částí R-Skriptu při vytváření požadovaných úloh. Nejčastěji byla použita data krajů, okresů a obcí a jejich obyvatelstvem, dále pak také digitální model reliéfu (DMR) pro možnost vytvoření Profilu.

3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

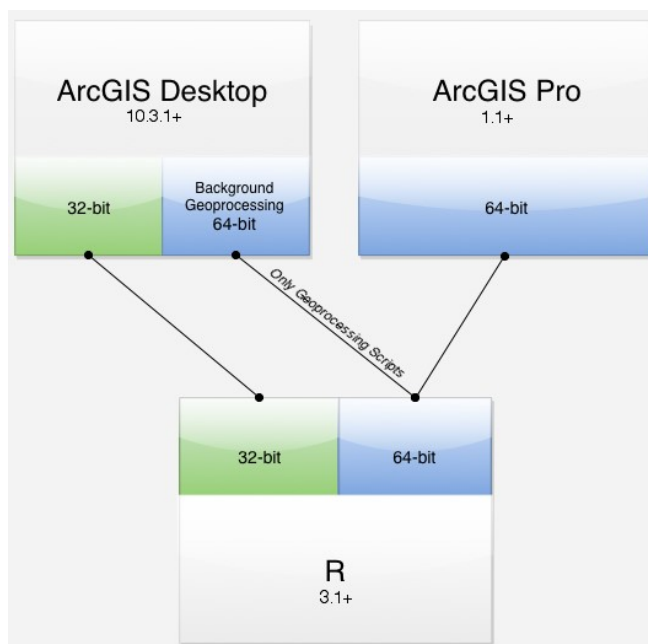
3.1 R-Bridge

Nástroj byl vyvinut na začátku roku 2016, kdy ESRI vytvořila R knihovnu, která je schopna komunikovat a vyměňovat data mezi aplikacemi ArcGIS for Desktop a R Project. Tím pádem můžeme využívat funkcí R Project v ArcGIS for Desktop a obráceně.

Verze ArcGIS for Desktop a R Project for Statistical Computing

Pro ArcGIS desktop je nutné mít nainstalovanou verzi 10.3.1 nebo novější a pro ArcGIS Pro verzi 1.1 nebo novější. Pro statistický SW R Project je nutné mít nainstalovanou verzi 3.1.0 nebo novější (Install the R-ArcGIS Bridge, 2016).

Propojení ArcGIS for Desktop a R Project ve verzích, které byly uvedeny již dříve, probíhá výhradně na 32-bitové verzi. Na 64-bitové verzi probíhají pouze geoprocessingové skripty. Na druhou stranu ArcGIS Pro běží pouze na 64-bitové verzi (Install the R-ArcGIS Bridge, 2016).



Obr. 1: Verze ArcGIS for Desktop a ArcGIS Pro
(Zdroj: <https://github.com/R-ArcGIS/r-bridge-install>)

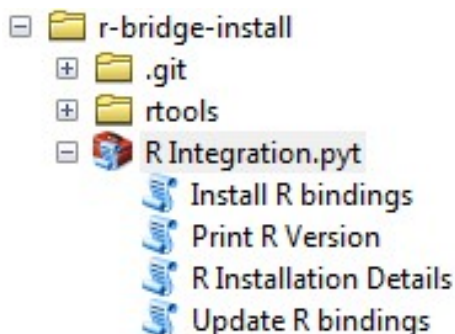
Instalace

Instalace je možná dvěma způsoby. Prvním způsobem je instalace online, ve které je stažen zip soubor s toolboxem a přes ArcCatalog je otevřen daný toolbox. Druhou možností, jak nainstalovat R-bridge je offline instalace, tedy instalace, pokud není uživatel připojen k síti. Při této instalaci je nutné mít v počítači stažený balíček s instalačními daty, který je dostupný stejně jako toolbox na adrese <https://github.com/R-ArcGIS/r-bridge-install>. Je nutností, aby byl ArcGIS for Desktop při instalaci R-Bridge spuštěn jako správce (Install the R-ArcGIS Bridge, 2016).

Toolbox obsahuje čtyři skripty. Pro instalaci R-Bridge nejprve spustíme script s názvem Install R Bindings a poté script R Instalations Details. Tyto skripty do ArcGIS

for Desktop doinstalují knihovny z R project, tím pádem můžeme využívat funkce z R project v ArcGIS for Desktop (Install the R-ArcGIS Bridge, 2016).

Celá instalace je pomocí videa dostupná na stránce, na které se stahuje toolbox.



Obr. 2: R-Bridge instalační toolbox

(Zdroj: <https://github.com/R-ArcGIS/r-bridge-install>)

3.2 Balíček `arcgisbinding`

Balíček byl vytvořen 11.7.2016 a nejnovější verze je 1.0.0.121. Poskytuje funkce, které uživatel může využít k načtení, konverzi a exportu datových formátů z aplikace ArcGIS for Desktop do R Project. Knihovna celkem obsahuje šestnáct příkazů. Jsou to příkazy `arc.check_product`, `arc.data2sp`, `arc.dataset-class`, `arc.env`, `arc.fromP4ToWkt`, `arc.fromWktToP4`, `arc.open`, `arc.progress_pos`, `arc.progres_label`, `arc.select`, `arc.shape`, `arc.shape-class`, `arc.shapeinfo`, `arc.sp2data` a funkci `arc.write`. Tyto příkazy lze rozdělit celkem do tří kategorií. Základní funkce pro čtení a zápis, funkce pro konverzi dat, a funkce nadstavbových prvků R skriptu (Package ‘`arcgisbinding`’, 2016).

3.2.1 Základní funkce pro čtení a zápis

Do této kategorie patří funkce, které uživatel nejvíce využívá, pokud začne pracovat s propojením R-bridge, neboť patří mezi základní stavební kameny mostu mezi aplikacemi ArcGIS for Desktop a R Project.

`arc.check_product`

Inicializuje připojení k ArcGIS for Desktop. V každém R-skriptu by měla být zavolána nejprve funkce `arc.check_product`, aby bylo vytvořeno spojení. Funkce `arc.check_product` poskytuje podrobnosti o nainstalované verzi aplikace ArcGIS se kterou komunikuje balíček `arcgisbinding`.

Údaje, které funkce `arc.check_product` vrací je za prvé zda je propojení realizováno na ArcGIS Desktop, nebo na ArcGIS Pro. Za druhé licenční úroveň produktu ArcGIS (Základní, standartní či pokročilá). Třetím údajem, který funkce `arc.check_product` vrátí je číslo sestavy použité verze. Tento údaj je nápomocný při ladění a vytváření chybových hlášení. Posledním údajem je název knihovny, pomocí které je ArcGIS for Desktop propojen s R Project (Package ‘`arcgisbinding`’, 2016).

`arc.open`

Pomocí této funkce je uživatel schopen otevřít datové sady, tabulky a vrstvy z aplikace ArcGIS for Desktop. Funkce vrací nový objekt, který obsahuje podrobnosti

o prostorových informacích a informacích o atributech. Argumentem funkce `arc.open` je pouze cesta k datům, které chce uživatel načíst (Package 'arcgisbinding', 2016).

U datových sad lze otevřít pouze prvky se stejným typem geometrie, tedy body, linie nebo polygony, a prostorovou složku v kombinaci s atributovou tabulkou. Datové sady mohou být uloženy v různých datových formátech jako například shapefile, geodatabáze či feature dataset. Všechny tyto datové typy jsou přístupné pomocí úplné cesty k souboru (Package 'arcgisbinding', 2016).

Tabulky jsou v podstatě stejné jako datové rámce. Jsou uspořádané do řad a sloupců. Do sloupců jsou ukládány proměnné. Možné datové typy tabulek jsou textové soubory, tabulky aplikace MS excel, dBASE tabulky a INFO tabulky (Package 'arcgisbinding', 2016).

arc.select

Pomocí této funkce je uživateli umožněno ze vstupních dat vybrat data, která jsou potřeba pro danou analýzu. Data musí být nejprve načtena pomocí funkce `arc.open`, aby s nimi mohlo být dále pracováno (Package 'arcgisbinding', 2016).

arc.shapeinfo a arc.shape

Tyto funkce pomáhají uživateli zjistit podrobnosti o geometrii dat, nebo například slot WKT obsahuje údaje o referenčním systému (Package 'arcgisbinding', 2016).

arc.write

Funkce `arc.write` umožňuje ukládání dat do shapefilu, geodatabáze nebo do tabulky. Je použita v poslední části R-skriptu, kdy je potřeba zapsat výsledky do výstupního souboru. Funkce má dva povinné argumenty, za prvé cesta k umístění souboru, kde má být vytvořen výstupní soubor, a za druhé objekt, proměnnou, ve které je výsledek uložen v R-skriptu (Using the R-ArcGIS Bridge: the arcgisbinding Package, 2016).

3.2.2 Funkce pro konverzi dat

Druhá skupina funkcí, kterou obsahuje balíček `arcgisbinding` je zaměřena na konverzi dat. Jelikož má R Project obrovský počet balíčků a metod, je důležité, aby pro následnou manipulaci s daty byly data ve správných formátech.

arc.data2sp

Balíček `sp` má vlastní formát, ve kterém pracuje s body, liniemi a polygony. Tato funkce převádí bodovou vrstvu na `SpatialPointsDataFrame`, liniiovou vrstvu na `SpatialLinesDataFrame` a polygonovou vrstvu na `SpatialPolygonDataFrame`. Touto funkcí je zaručeno, že data budou ve správném formátu, který uživatel požaduje (Using the R-ArcGIS Bridge: the arcgisbinding Package, 2016).

arc.sp2data

Tato funkce je opakem funkce `arc.data2sp`. Po dokončení potřebných analýz převede data zpět na body, linie či polygony (Using the R-ArcGIS Bridge: the `arcgisbinding` Package, 2016).

arc.shape2sp

Alternativou převedení dat je převedení bodů na `SpatialPoints`, linií na `SpatialLines` a polygonů na `SpatialPolygons`. Toto převedení je dosaženo pomocí funkce `arc.shape2sp` (Using the R-ArcGIS Bridge: the `arcgisbinding` Package, 2016).

arc.fromP4ToWkt a arc.fromWktToP4

Funkce `arc.fromP4ToWkt` převádí projekci ze systému souřadnic PROJ 4 do textového řetězce (WKT). Zato funkce `arc.fromWktToP4` je opačného rázu, tedy převádí projekci z textového řetězce (WKT) do souřadnicového systému PROJ 4 (Using the R-ArcGIS Bridge: the `arcgisbinding` Package, 2016).

arc.select

Tato funkce spadá kromě první kategorie také do této. Je to kvůli tomu, že funkce `arc.select` obsahuje volitelný argument pro zadání prostorové reference. Z toho důvodu je velmi zjednodušené reprojektování dat a získání dat ve formátu, který uživatel potřebuje (Package 'arcgisbinding', 2016).

3.2.3 Funkce nadstavbových prvků R skriptu

Tyto funkce by měli uživatelé přinášet přehled nad chodem skriptu. Knihovna `arcgisbinding` dále obsahuje několik funkcí, pomocí kterých lze navrhnout a přizpůsobit nástroj, který si autor sám vytvoří.

arc.env

Tato funkce umožňuje získat místní prostředí nástrojů pro geoprocessing v prostředí aplikace ArcGIS a také umožňuje zkontrolovat, zda jsou nastaveny odpovídajícím způsobem. Zahrnuje například nastavení výstupního souřadnicového systému či definování rozsahu zpracování (Using the R-ArcGIS Bridge: the `arcgisbinding` Package, 2016).

arc.progress_pos

Tato funkce umožňuje zobrazení aktuálního kroku analýzy. Zobrazuje celočíselné procento od 0 do 100, kde 100 značí, že skript je dokončen (Using the R-ArcGIS Bridge: the `arcgisbinding` Package, 2016).

arc.progress_label

Geoprocessingový nástroj, který uživatelé zobrazí pokrok při čtení R skriptu. Funkce umožňuje kontrolu nad průběhem skriptu. V dialogovém okně v aplikaci ArcGIS zobrazí stav chodu skriptu.

3.3 Referenční příklad

Jako referenční příklad byla vybrána úloha Combining ArcGIS and R – Clustering toolbox. Tato úlohu vytvořil Fabio Veronesi pro školní účely seznámení se s nástrojem R-Bridge (Combining ArcGIS and R - Clustering Toolbox, 2016).

Celý R-skript je zabalen do funkce `tool_exec` s argumenty `in_params` a `out_params`. Jsou to seznamy vstupních a výstupních parametrů, které jsou následně předány ArcGISu, když R-skript nahrán do skriptu.

```
tool_exec <- function(in_params, out_params)
{
```

Obr. 3: `tool_exec`

Tato část skriptu uživateli stáhne a nainstaluje daný balíček, pokud jej uživatel již nemá a poté jej načte. V tomto skriptu je dále používána pouze funkce `kmeans`, která je obsažena ve standardním balíčku R Project.

```
if (!requireNamespace("sp", quietly = TRUE))
  install.packages("sp")
require(sp)
```

Obr. 4: Použité balíčky v referenčním příkladu

Další řádky skriptu obsahují volání, které při spuštění v ArcGIS v dialogovém okně vypíše řetězec, který je obsažen v závorce funkce `print`. V tomto případě to je název a jméno autora.

```
print("K-Means Clustering of Shapefiles")
print("Author: Fabio Veronesi")
```

Obr. 5: Volání v referenčním příkladu

Dalším, již nutným, krokem je vytvoření objektů pro každý vstupní a výstupní parametr. Tyto parametry jsou později při vytváření skriptu v ArcGIS specifikovány a je jim přiřazen datový typ. Prvním vstupním parametrem je `shapefile`, pojmenovaný `source_dataset`. Druhým vstupním parametrem je počet clusterů, který autor nazval `nclust`. Posledním vstupním parametrem je výběr proměnné, která bude využita pro seskupování. Jediným výstupním parametrem je výsledný SHP, který obsahuje body a jejich cluster a je pojmenován `out_shape`.

```

source_dataset = in_params[[1]]
nclust = in_params[[2]]
variable = in_params[[3]]

out_shape = out_params[[1]]

```

Obr. 6: Parametry v referenčním příkladu

V dalších krocích začíná samotný výpočet. Funkce `arc.progress_label` pouze vypisuje do dialogového okna Loading Dataset. V dalším řádku je již použita funkce `arc.open`, která otevře daný soubor a následně jej uloží do proměnné `d`.

```

### Read Data
arc.progress_label("Loading Dataset")
d <- arc.open(source_dataset)

```

Obr. 7: Otevírání souboru v referenčním příkladu

Dále je využito funkce `arc.select`. Pomocí této funkce získáme ze souboru, pouze proměnné, které si uživatel zvolí v toolboxu. V dalším kroku jsou vybraná data uložena do tabulky, která je pojmenována `data_clust`.

```

### Create a Data.Frame with the variables to cluster
data <- arc.select(d, variable)
data_clust <- data.frame(data[,variable[1]])

```

Obr. 8: Vybrání dat v referenčním příkladu

Dalším krokem R skriptu je cyklus pro vytvoření dalších sloupců v tabulce `data_clust`. Přidané sloupce jsou následně přejmenovány podle názvů proměnných.

```

if(length(variable)>1){
for(i in 2:length(variable)){
data_clust <- cbind(data_clust,data[,variable[i]])
}
}

names(data_clust) <- variable

```

Obr. 9: Vytvoření sloupců v referenčním příkladu

V další fázi R skriptu jsou vytvářeny histogramy, které si uživatel navolil před spuštěním toolboxu. Pomocí funkce `dev.new()` probíhá každé vytvoření histogramu odděleně.

```

for(i in 1:length(variable)){
dev.new()
plot(hist(data_clust[,i]),main=paste0("Histogram of ",variable[i]),xlab=variable[i])
}

```

Obr. 10: Vytváření histogramů v referenčním příkladu

V dalším kroku je volána funkce `kmeans` ke shlukování dat. Pomocí funkce `arc.write` jsou výsledky zapsány do shapefilu `out_shape`. Funkcí `arc.shape` jsou každému výslednému bodu přiřazeny souřadnice.

```

clusters <- kmeans(data_clust, nclust)

result <- data.frame(cluster=clusters$cluster)

arc.write(out_shape, result, coords = arc.shape(data))

```

Obr. 11: Shlukování v referenčním příkladu

Na konci R skriptu je ještě nepovinný prvek, funkce `print`, která do dialogového okna vypíše „Done!!“, neboli Hotovo. Nakonec jsou ještě vráceny výstupní parametry a zobrazeny uživateli.

```

print("Done!!")
return(out_params)
}

```

Obr. 12: Výstupní parametry v referenčním příkladu

4 R -> ARCGIS

4.1 Načtení rasteru

Balíčky

Knihovny neboli balíčky obsahují funkce, které nejsou v základním programu a rozšiřují možnosti R Project. R Skript nejdříve prohledá, zda se nutný balíček nachází v databázi balíčků R Project. Pokud se nenachází, tak jej stáhne. V této úloze jsou využity celkem tři balíčky. Jsou jimi balíčky "raster", "sp", a balíček "rgdal". Druhým krokem po instalaci je aktivace těchto balíčků.

```
# nacteni balicku
if (!requireNamespace("raster", quietly = TRUE))
  install.packages("raster")
require(raster)

if (!requireNamespace("sp", quietly = TRUE))
  install.packages("sp")
require(sp)

if (!requireNamespace("rgdal", quietly = TRUE))
  install.packages("rgdal")
require(rgdal)

# aktivace balicku
library(raster)
library(sp)
library(rgdal)
```

Obr. 13: Použité balíčky v úloze načtení rasteru

Načtení dat

V této části R Skriptu jsou rastrová data nahrávána do R Project. Prvním krokem je nastavení adresáře. Je využito funkce `setwd`, kde uvnitř funkce je úplná cesta ke složce. Druhým krokem je načtení rasteru a to je provedeno pomocí funkce `raster`. Uvnitř funkce `raster` je buď název souboru, nebo cesta z nastaveného adresáře k souboru včetně souboru. Posledním krokem je zobrazení rasteru v R Project, a to je provedeno pomocí funkce `plot`.

```
# nastaveni adresare
setwd("D:/toolbox")

# nacteni rasteru ze slozky
DEM <- raster("bo/raster_zk1.tif")

# zobrazeni rasteru
plot(DEM)
```

Obr. 14: Načtení dat v úloze načtení rasteru

4.2 Načtení SHP

Balíčky

Dříve než jsou knihovny aktivovány, je nutné zjistit, zda je počítač uživatele obsahuje. Pokud R Skript zjistí, že nejsou přítomny, stáhne je a nainstaluje a poté aktivuje. Pro tento R Skript je potřeba jen jeden balíček. Tím balíčkem je knihovna

"arctgisbinding", která obsahuje funkci arc.open pro otevření shapefilů. Následuje aktivace balíčku funkcí library.

```
# instalace balicku
if (!requireNamespace("arctgisbinding", quietly = TRUE))
  install.packages("arctgisbinding")
require(arctgisbinding)

#aktivace balicku
library(arctgisbinding)
```

Obr. 15: Použité balíčky v úloze načtení shapefilu

Načtení dat

V této části R Skriptu je shapefile načítán do R Project. Shapefile je otevřen pomocí funkce arc.open, kde v těle funkce je úplná cesta k souboru i s názvem souboru. Tato funkce je hned uložena do proměnné, v tomto případě je název proměnné tab. Pro zjištění, které pole obsahuje shapefile je použita funkce tab@fields, kde tab je načtený shapefile. Po vypsání těchto polí si uživatel může vybrat pomocí funkce arc.select pole, která potřebuje a následně je uložít do proměnné. A s vybranými daty může následně dále pracovat dle svého uvážení.

```
#nacteni dat
tab <- arc.open("D:/toolbox/bod_final.shp")

# pole shapefilu
tab@fields

#vytvoreni datasetu a vybrani prislusnych sloupcu
dataset<- arc.select(object = tab, fields = c("GRID_CODE"))
```

Obr. 16: Načtení dat v úloze načtení shapefilu

5 ARCGIS -> R

5.1 Příčný profil

Příklad, který nese název Příčný profil, byl vybrán, protože v desktopové verzi ArcGIS není možnost vytvořit graf ze vstupní linie a rasteru. ArcGIS for Desktop má dvě možnosti, jak vytvořit graf příčného profilu. První variantou je nástroj obsahující extenze *3D Analyst*, který z dočasné linie a rasteru, na kterém je linie vytvořena, zobrazí příčný profil. Nevýhodou této funkce je, že již nikdy se uživateli nepovede zadat linii úplně stejně. Druhou variantou, kterou nabízí ArcGIS for Desktop je funkce *Stack Profile*. Tato funkce z existující linie a rasteru, TINu či digitálního modelu terénu vytvoří tabulku, kde je také možnost zobrazit graf příčného profilu. Nevýhodou této funkce je právě vytvořený graf příčného profilu, který nemá mnoho vizualizačních parametrů na výběr. Z důvodu široké nabídky možností vizualizace v programu R Project bylo rozhodnuto využít funkce *Stack Profile*. Těto funkce bude využito pouze do doby, než se vytvoří tabulka s hodnotami vzdáleností a nadmořských výšek. Daná tabulka je následně skrz propojení načtena do R Project, podle zadaných parametrů je vytvořen graf a zobrazen v ArcGIS for Desktop, aniž by uživatel musel R Project spustit.

Stack Profile

Stack profile je nástroj, který vytvoří tabulku s hodnotami nadmořských výšek a vzdáleností. Vstupními parametry jsou liniová vrstva a rasteru, TINu nebo digitální model povrchu. V ArcGIS toolboxu je k nalezení v sekci 3D Analyst Tool a v záložce Functional Surface (Stack Profile, 2017).

1) R-Skript

Balíčky

Pro tuto úlohu je nutné nainstalovat či jen aktivovat celkem dvě knihovny. První knihovnou je balíček *readr*, který slouží pro úspěšné načtení dat. Druhou knihovnou je balíček *arcgisbinding*, který slouží pro načtení shapefilu do R project. Další funkcí balíčku, kterou využívá tento skript je vybrání určitých dat z atributové tabulky shapefilu. Při spuštění skriptu, jsou prohledány všechny knihovny R project, které jsou na počítači nainstalované. Pokud R project neobsahuje knihovnu, potřebnou pro tento skript, je tato knihovna nainstalována. Po prohledání a instalaci všech potřebných knihoven se dané knihovny zapnou, a to příkazem *library*, kde se následně v kulatých závorkách uvede název balíčku.

```
if (!requireNamespace("readr", quietly = TRUE))
  install.packages("readr")
require(readr)

if (!requireNamespace("arcgisbinding", quietly = TRUE))
  install.packages("arcgisbinding")
require(arcgisbinding)

library(arcgisbinding)
library(readr)
```

Obr. 17: Použité balíčky v příkladu příčný profil

Proměnné

Proměnné jsou vstupními parametry skriptu. Dělí se na parametry povinné, kde musíme zadat hodnotu a na parametry nepovinné. Povinným parametrem v tomto skriptu je výstupní složka, jako parametr nazvaná `output_filename` a název výstupního souboru, s názvem `output_path`. Výstupní složka s názvem souboru je použita k uložení obrázku profilu v typu `.png`. Další povinné vstupní parametry vstupují již do vytvořeného skriptu *Stack Profile*, těmito povinnými vstupními parametry jsou raster a liniová vrstva. Ze skriptu *Stack Profile* do R-skriptu vstupuje tabulka s hodnotami a je uložena jako vstupní parametr `in_table`. Nepovinných parametrů je celkem šest. Patří mezi ně popis grafu, který je umístěn pod grafem, dále pak titul, umístěný nad grafem, dalšími nepovinnými parametry jsou popisky osy x a osy y. Posledními dvěma parametry je barva pozadí grafu a barva linie výsledného grafu. Všechny nepovinné parametry jsou předvyplněny, tudíž pokud uživatel nemá zájem je zadávat, tak nemusí.

```
# vstupní parametry
input_variables = 0
in_table = in_params[[input_variables+1]]

output_variables = input_variables+1
output_path = in_params[[output_variables+1]]
output_filename = in_params[[output_variables+2]]

visual_variables = output_variables+2
graph_color = in_params[[visual_variables+1]]
titul = in_params[[visual_variables+2]]
popisek_osa_x = in_params[[visual_variables+3]]
popisek_osa_y = in_params[[visual_variables+4]]
popisek_grafu = in_params[[visual_variables+5]]

color_variables = visual_variables+5
bg_color = in_params [[color_variables+1]]
```

Obr. 18: Použití proměnné v příkladu příčný profil

Načtení dat

V této části R-skriptu se aplikuje balíček *arcgisbinding*, pomocí kterého se načítají data tabulek či shapefilu. Funkcí `arc.open` je tabulka, která pochází ze skriptu *Stack Profile* načtena a následně uložena do proměnné `tab`. V dalším kroku jsou z tabulky pomocí funkce `arc.select` vybrány sloupce, které obsahují nadmořské výšky a vzdálenosti. Uvnitř funkce `arc.select` je zadán objekt odkud chceme vybírat, tedy tabulka (`tab`) a poté pole `FIRST_DIST` a `FIRST_Z`, která v tabulce symbolizují právě nadmořské výšky a vzdálenosti.

```
# otevreni souboru
tab <- arc.open(in_table)

#print(tab@fields) #seznam poli v tabulce

# vyber poli ze vstupniho souboru a ulozeni do promenne
dataset<- arc.select(object = tab, fields = c("FIRST_DIST", "FIRST_Z"))
```

Obr. 19: Načtení dat v příkladu příčný profil

Úprava dat

Pro vytvoření grafu je potřeba, aby byly hodnoty nadmořských výšek a vzdáleností převedeny na numerické hodnoty. Toho je docíleno pomocí funkce `as.numeric`, která hodnoty ve sloupci převede na numerické, aby s nimi mohla dále pracovat. Druhou věcí, která je tímto krokem také udělána, je že u desetinných čísel

převědeme čárky na tečky. Do těla funkce jsou v uvozovkách uvedeny nejdříve čárka a poté tečka, a tím je docíleno, že se čárky převedou na tečky. Dalším parametrem, který `as.numeric` vyžaduje je sloupec jehož hodnoty chceme převést na numerické hodnoty. Toto je vyřešeno zadáním názvu souboru a názvu sloupce. Mezi název souboru a název sloupce je následně vložena značka dolaru. Poté zápis vypadá následovně: `dataset$FIRST_DIST` pro vzdálenosti a `dataset$FIRST_Z` pro nadmořské výšky. Funkci `as.numeric` je nakonec uložena do proměnné `xcol` pro vzdálenost a `ycol` pro nadmořskou výšku.

```
# převedení hodnot osy x na numerické hodnoty
xcol= as.numeric(sub(",", ".", dataset$FIRST_DIST, fixed = TRUE));

# převedení hodnot osy y na numerické hodnoty
ycol= as.numeric(sub(",", ".", dataset$FIRST_Z, fixed = TRUE));
```

Obr. 20: Úprava dat v příkladu příčný profil

Samotné řešení příkladu

V této fázi R-skriptu je řešena již daná úloha. Ještě před samotným vykreslením grafu se pomocí funkce `par` nastaví barva pozadí.

Poté následuje funkce `plot`, která již vykreslí graf. Uvnitř kulatých závorek funkce `plot` pak zadáváme hodnoty vzdáleností, tedy `xcol` a poté hodnoty nadmořských výšek, tedy `ycol`. Následně zadáme limity osy x (`xlim`) a osy y (`ylim`), aby se graf vykreslil úplný. Tyto limity získáme funkcemi `min` a `max`. Pro osu x pak bude příkaz vypadat `xlim=c(min(xcol),max(xcol))` a pro osu y pak bude vypadat `ylim=c(min(ycol),max(ycol))`. Poté zadáme titul grafu pomocí funkce `main` a pomocí funkce `sub` popisek grafu. Popisky os x a y jsou vyřešeny funkcemi `xlab` a `ylab`. Typ grafu je zvolen `type "l"`, který vykreslí spojitou linii. Do funkce `col` nakonec ještě zadáme barvu, kterou chceme, aby linie grafu měla.

K samostatnému řešení příkladu jsou ještě přidány funkce, které uloží vytvořený obrázek profilu v typu `.png` do vybrané složky, která byla ve vstupních parametrech vybrána. První částí je vytvoření adresáře, kam se má obrázek uložit. Toho je docíleno pomocí funkce `setwd`. Následně se funkcí `png` vytvoří ve složce, kterou zadáme do vstupního parametru `output_path` s názvem souboru, který zadáme do proměnné `output_filename`. V kulatých závorkách funkce `png` je nutné spojit řetězec názvu výstupního souboru a přípony `.png`. Po vytvoření grafu příčného profilu, je nutné ukončit zapisování do souboru `output_filename` pomocí funkce `dev.off()`.

```
setwd(output_path)

png(filename = paste(output_filename, ".png", sep=""))

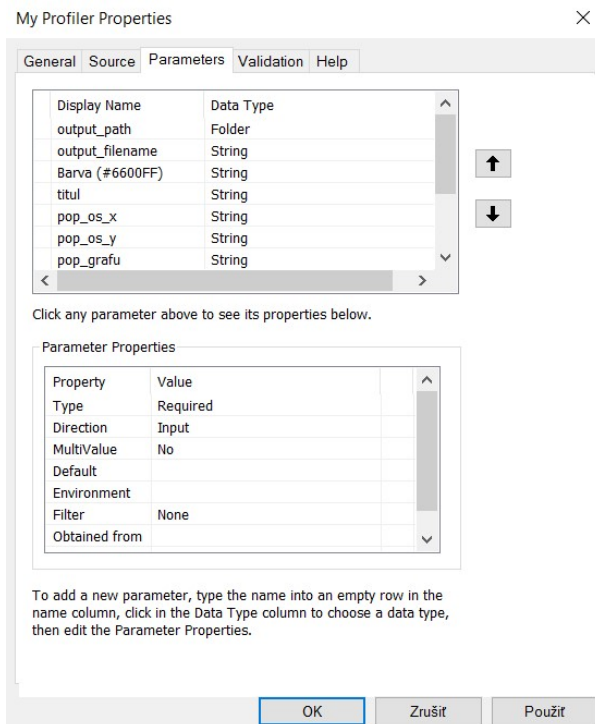
par(bg = bg_color)# barva pozadí grafu
plot(xcol,# osa x, vzdálenosti
     ycol,# osa y, nadmořská výška
     xlim=c(min(xcol),max(xcol)),# limita osy x
     ylim=c(min(ycol),max(ycol)),# limita osy y
     main=titul,# titul grafu
     sub = popisek_grafu,# popisek grafu
     xlab= popisek_osa_x,# popisek osy x
     ylab= popisek_osa_y,# popisek osy y
     type="l",# typ grafu (spojitá linie)
     asp=0,
     col=graph_color)# barva linie grafu

dev.off()
```

Obr. 21: Samotné řešení v příkladu příčný profil

2) Toolbox v ArcGIS

V ArcGIS se vytvoří toolbox podle návodu a v něm je vytvořen skript. Dále se do skriptu nahraje R-Skript. Při vytváření skriptu je nutné zadat stejné parametry jako jsou v R-Skriptu, aby se R-Skript se skriptem spojili a fungovali tak jak mají.



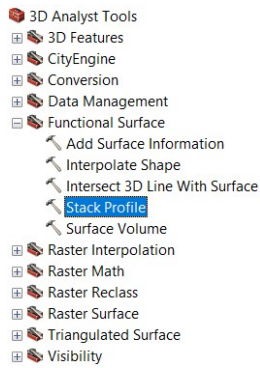
Obr. 22: Parametry skriptu v příkladu příčný profil

Parametry skriptu lze upravovat i později dle vlastního uvážení. Dalším krokem je vytvoření modelu v prostoru toolboxu.



Obr. 23: Toolbox příkladu příčný profil

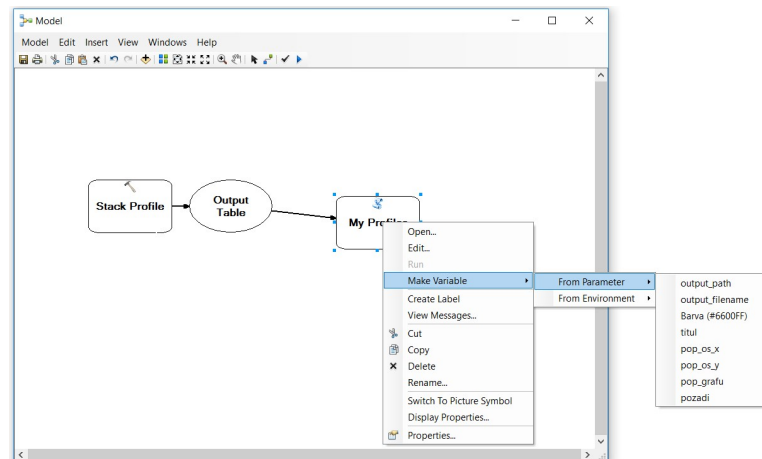
Otevře se editační okno modelu a skript se přetáhne do editačního okna. Tímto způsobem je zajištěno, že se dále můžou nabalovat další funkce na tento skript. V dalším kroku se přidá z toolboxu aplikace ArcGIS for Desktop funkce Stack profile. Tato funkce se nachází v sekci *Toolbox - Functional Surface - Stack Profile*.



Obr. 24: Umístění funkce Stack Profile v toolboxu 3D Analyst tool

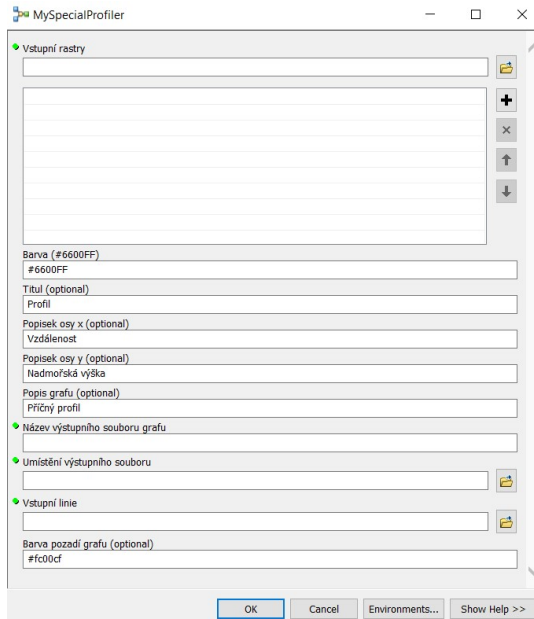
Výstupní tabulka se v následujícím kroku propojí se skriptem pomocí funkce *Connect*. Tím, že byli při vytváření skriptu zadány parametry, je dalším krokem přidání parametrů do modelu. Tento krok se provede tak, že se klikne pravým tlačítkem myši na skript v editačním okně modelu a pomocí funkce *Make Variable - From Parameter* se parametry přidají do editačního okna modelu. Podle názvů v editačním okně modelu, jsou pak parametry nazvány i v rozhraní, kde se zadávají vstupní parametry pro spuštění modelu.

p



Obr. 25: Model příkladu příčný profil

Dále je nutné nastavit parametry, u kterých má být možnost zadat je uživatelem. Tohoto kroku je docíleno tím způsobem, že na daný parametr klikneme pravým tlačítkem myši a zadáme *Model Parameter*. Tento model má všechny vybrané parametry nastavené tak, že si je může nastavit samotný uživatel. Pokud má být parametr předdefinovaný, musí být daný parametr vyplněný příslušnou hodnotou. Toto je použito především u parametrů, které nejsou pro uživatele povinné na zadávání. Patrné jsou parametry, které musí uživatel vyplnit, aby mohl skript spustit (viz obr.26).



Obr. 26: Rozhraní příkladu příčný profil

5.2 3D Grafy

1) R-Skript

Balíčky

Knihovny neboli balíčky, umožňují uživateli využívat v R Project funkce, které základní verze nenabízí. R-Skript nejdříve projde všechny knihovny, které jsou na počítači uživatele nainstalovány. Pokud není daný balíček přítomný, R-Skript jej stáhne a nainstaluje. Dalším krokem, je aktivace balíčků, které jsou pro úlohu nutné. Pro tuto úlohu jsou využity dvě knihovny. První knihovnou je balíček scatterplot3d, který zajišťuje funkce a vytvoření 3D grafu. Druhou knihovnou pro tento R-Skript je balíček arcgisbinding, který obsahuje funkce, které otevrou shapefile a následně s ním umožní další práci.

```
# instalace balicku
if (!requireNamespace("scatterplot3d", quietly = TRUE))
  install.packages("scatterplot3d")
require(scatterplot3d)

if (!requireNamespace("arcgisbinding", quietly = TRUE))
  install.packages("arcgisbinding")
require(arcgisbinding)

# aktivace balicku
library(scatterplot3d)
library(arcgisbinding)
```

Obr. 27: Použité balíčky v příkladu 3D Grafy

Proměnné

Úloha je nastavena na celkem dvanáct parametrů, deset parametrů je vstupních a dva výstupní. Mezi vstupní parametry, které je nutné zadat do R-Skriptu patří parametr in_shp, do této proměnné je uložen vstupní shapefile. Dalšími vstupními parametry, které vstupují do R-Skriptu jsou in_x, in_y a in_z. Tyto proměnné definují, který sloupec ze shapefile uživatel zadá jako vstupní data pro osu x, y a z. Dalšími parametry, které obsahuje tento R-Skript jsou dva výstupní parametry. Do proměnné

out_slozka uživatel zadá složku, kam má být soubor uložen. Do druhého parametru, out_soubor, uživatel nastaví název výstupního souboru.

Mezi nepovinné parametry, které vstupují do tohoto R-Skriptu patří parametr in_type, tento parametr obsahuje v uživatelském rozhraní toolboxu možnost zaškrtnout, zda uživatel chce mít body spojeny se základnou 3D grafu. Druhým parametrem je parametr in_titul, který nastaví titul grafu. Parametry in_pop_os_x, in_pop_os_y, in_pop_os_z uživatel zadá popisky k osám x, y a z. Posledním parametrem, který tento skript obsahuje, je parametr in_barva. Tento parametr nastavuje barvu bodu a v případě, že je aktivováno spojení bodů se základnou, tak také barvu spojnic.

```
# parametry
print(in_params)
in_shp = in_params[[1]]
in_x = in_params[[2]]
in_y = in_params[[3]]
in_z = in_params[[4]]
in_type = in_params[[5]]
in_titul = in_params[[6]]
in_pop_os_x = in_params[[7]]
in_pop_os_y = in_params[[8]]
in_pop_os_z = in_params[[9]]
in_barva = in_params[[10]]
out_slozka = in_params[[11]]
out_soubor = in_params[[12]]
```

Obr. 28: Použité proměnné v příkladu 3D Grafy

Načtení dat

V této části R-Skriptu je využíváno balíčku "arcgisbinding", tedy balíčku, který obsahuje funkce pro import shapefilu do R Project. Jako první je zde užito funkce arc.open. Tato funkce načte shapefile do R Project. Po načtení, je shapefile načten do proměnné shp. Ve druhém kroku této části R-Skriptu je vytvořena tabulka, obsahující pole in_x, in_y a in_z, tedy x, y a z souřadnici 3D grafu, kterou vybere uživatel z polí shapefilu. Tato tabulka je následně uložena do proměnné dataset.

```
# otevreni shp
shp <- arc.open(in_shp)

# vytvoreni tabulky se sloupci x, y, z
dataset<- arc.select(object = shp, fields = c(in_x,in_y,in_z))
```

Obr. 29: Načtení dat v příkladu 3D Grafy

Úprava dat

Kroky v této části R-Skriptu je docíleno, aby vstupní data, která byla ve fázi načtení dat, byla schopna vstoupit do řešení příkladu. V této úloze obsahuje tato část R-Skriptu funkce na převedení hodnot proměnných in_x, in_y a in_z na numerické. Je užito funkce as.numeric. Proměnná in_x se po převedení uloží do proměnné xcol, proměnná in_y se po převedení uloží do proměnné ycol a poslední proměnná in_z se po převedení uloží do zcol.

```
# prevod in_x na numerickou hodnotu
xcol= as.numeric(sub(",", ".", dataset[[in_x]], fixed = TRUE));

# prevod in_y na numerickou hodnotu
ycol= as.numeric(sub(",", ".", dataset[[in_y]], fixed = TRUE));

# prevod in_z na numerickou hodnotu
zcol= as.numeric(sub(",", ".", dataset[[in_z]], fixed = TRUE));
```

Obr. 30: Úprava dat v příkladu 3D Grafy

Samostatné řešení příkladu

Pro řešení úlohy 3D grafů byla vybrána funkce `scatterplot3d`. Uvnitř této funkce jsou zadány tři pole, ze kterých bere 3D graf data pro osu x, y a z. Pro tento účel jsou připraveny pro osu x proměnná `xcol`, pro osu y proměnná `ycol` a pro osu z proměnná `zcol`. Tyto proměnné byly vytvořeny při vytváření numerických hodnot ze vstupních dat. Další, co je nutné, aby obsahovala funkce, jsou limity grafu, aby bylo pole grafu zobrazeno tam, kde jsou data. Tyto limity získáme pomocí funkcí `min` a `max` každé osy a jejichž dosazení do `xlim`, `ylim` respektive do `zlim`. Další, co se nachází uvnitř funkce `scatterplot3d` zapsáno je barva grafu. Barva je nastavena v proměnné `in_barva` a je zadána uživatelem. Po barvě následují ve funkci popisky grafu. Do funkce `main`, tedy název grafu, je přiřazena proměnná `in_titul`. Popisky os x, y a z jsou uloženy v proměnných `in_pop_os_x`, `in_pop_os_y` a `in_pop_os_z`. Poslední funkcí, kterou obsahuje `scatterplot3d` je, zda uživatel požaduje pouze body v prostoru, nebo by chtěl spojit tyto body s podstavou, a to kolmou linií na podstavu. Toto vyřeší jednoduchý cyklus, který pokud je zaškrtnuto od uživatele, aby byly body grafu spojeny s podstavou, tak zadá typ grafu "h". V ostatních případech, tedy pokud uživatel nezaškrtně pole, je vytvořen graf typu "p", který zobrazí pouze body v prostoru.

K samostatnému řešení příkladu je ještě přidána funkce, která uloží vytvořený obrázek 3D grafu v typu `.png` do vybrané složky, která byla ve vstupních parametrech vybrána. První částí je vytvoření adresáře, kam se má obrázek uložit. Toho je docíleno pomocí funkce `setwd` (set workspace directory). Následně se funkcí `png` vytvoří ve složce, kterou zadáme do vstupního parametru `out_slozka` s názvem souboru, který zadáme do proměnné `out_soubor`. V kulatých závorkách funkce `png` je nutné spojit řetězec názvu výstupního souboru a přípony `.png`. Po vytvoření 3D grafu, je nutné ukončit zapisování do souboru `out_soubor` pomocí funkce `dev.off()`.

```
# Nastaveni vystupni slozky
setwd(out_slozka)

# vytvoreni png
png(filename = paste(out_soubor, ".png", sep=""))

# typ grafu
if(in_type){
  scattertype="h";
}else{
  scattertype="p";
}

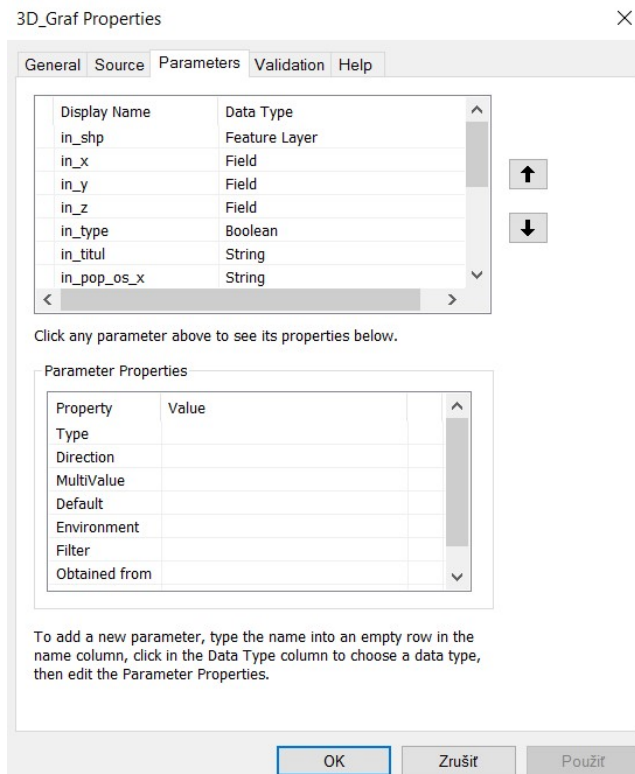
with(dataset, {
  scatterplot3d(xcol, ycol, zcol, # x y z
               xlim=c(min(xcol),max(xcol)), # limita osy x
               ylim=c(min(ycol),max(ycol)), # limita osy y
               zlim=c(min(zcol),max(zcol)), # limita osy z
               color=in_barva, # Color = barva
               pch=19, # pch = Typ znacky
               type=scattertype, # h - Typ grafu kde linie spojuje body s podlozkou
               main=in_titul, # Nazev grafu
               xlab=in_pop_os_x, # Popisek osy x
               ylab=in_pop_os_y, # Popisek osy y
               zlab=in_pop_os_z) # Popisek osy z
})

# Uzavreni zapisovani souboru png
dev.off()
```

Obr. 31: Samotné řešení v příkladu 3D Grafy

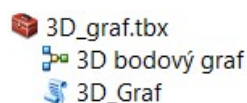
2) Toolbox v aplikaci ArcGIS for Desktop

V aplikaci ArcGIS for Desktop je vytvořen toolbox podle návodu a v něm je vytvořen skript, který je uživatelem pojmenován. Dále je do skriptu nahrán R-Skript. Při vytváření skriptu je nutné zadat stejné parametry jako jsou v R-Skriptu, aby se R-Skript se skriptem spojili a fungovali tak jak mají. V R-Skriptu musejí být očíslovány vstupní parametry podle toho, jaké pořadí parametrům náleží ve skriptu.



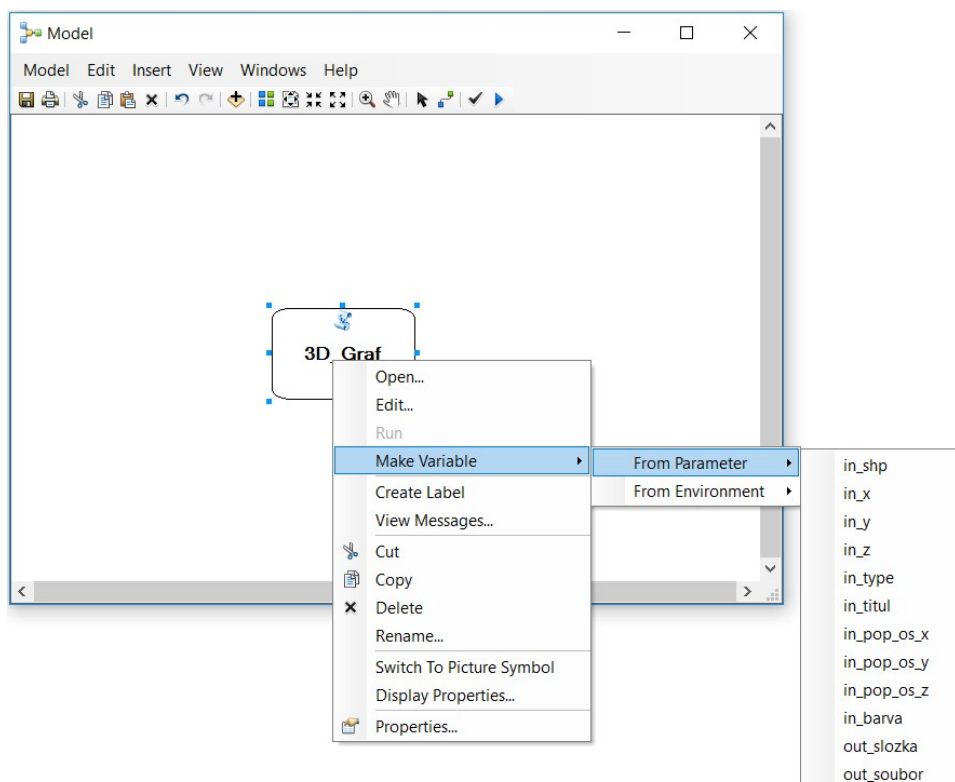
Obr. 32: Parametry skriptu v příkladu 3D Grafy

Parametry skriptu jde i později upravovat dle vlastního uvážení. Dalším krokem je vytvoření modelu v prostoru toolboxu, který však v tomto případě nemusí být vytvořen. Úlohu totiž lze spustit i přímo ze skriptu v aplikaci ArcGIS for Desktop.



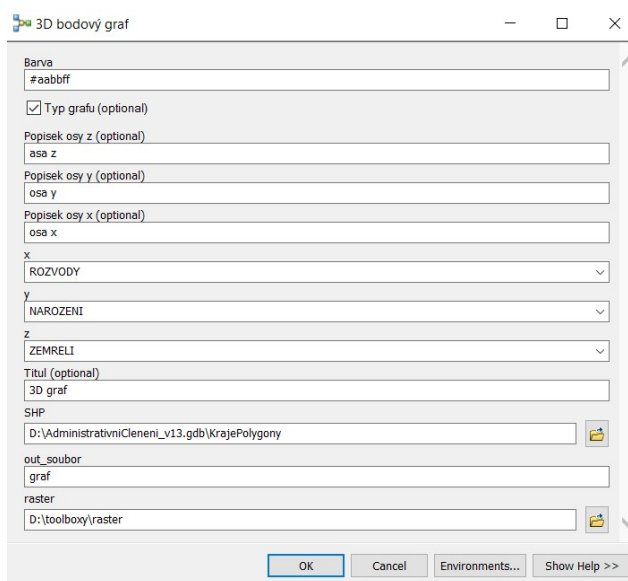
Obr. 33: Toolbox příkladu 3D Grafy

V dalším kroku se otevře editační okno modelu a skript se přetáhne z okna ArcCatalogu do editačního okna. Dále je nutné přidat parametry do editačního okna modelu, a to se provede tak, že uživatel pravým kliknutím myši zadá funkci Make Variable - From Parameter a poté vybere parametry, které chce do modelu přidat. Podle názvů v editačním okně modelu, jsou pak parametry nazvány i v rozhraní, kde se zadávají vstupní parametry pro spuštění modelu. V editačním okně lze také u každé proměnné nastavit předefinovanou hodnotu, která se uživateli objeví při zadávání parametrů do modelu či skriptu.



Obr. 34: Model příkladu 3D Grafy

Dále je nutné nastavit parametry, u kterých má být možnost zadat je uživatelem. Tohoto kroku je docíleno tím způsobem, že na daný parametr klikneme pravým tlačítkem myši a zadáme Model Parameter. Tento model má všechny vybrané parametry nastavené tak, že se dají uživatelem vyplnit. Bez vyplněného povinného parametru model není spuštěn. Pokud má být parametr předdefinovaný, musí být daný parametr vyplněný příslušnou hodnotou. Toto je použito hlavně u parametrů, které nejsou pro uživatele povinné na zadávání. Patrné parametry, které musí uživatel vyplnit (viz obr.35), aby mohl skript spustit.



Obr. 35: Rozhraní příkladu 3D Grafy

5.3 Kriging

1) R-Skript

Balíčky

Tato část R Skriptu zajišťuje fungování všech funkcí, které jsou v R Skriptu obsaženy. Uživateli umožňuje využívat funkce, které základní verze R Project nabízí. Pokud balíček, který je nutný v tomto skriptu, není přítomen, funkce v R-Skriptu jej stáhne a nainstaluje. Po instalaci jsou všechny balíčky aktivovány.

Pro tuto úlohu je nutné mít aktivované celkem čtyři balíčky. Jsou jimi balíček "geoR", balíček "arctgisbinding", balíček "maptools" a balíček "sp". Balíček "arctgisbinding" je nutný pro načtení dat z prostředí ArcGIS for Desktop. Balíčky "geoR", "maptools" a "sp" obsahuje R-Skript, aby bylo možné vytvořit kriging.

```
# instalace balicku
if (!requireNamespace("geoR", quietly = TRUE))
  install.packages("geoR")
require(geoR)

if (!requireNamespace("arctgisbinding", quietly = TRUE))
  install.packages("arctgisbinding")
require(arctgisbinding)

if (!requireNamespace("maptools", quietly = TRUE))
  install.packages("maptools")
require(maptools)

if (!requireNamespace("sp", quietly = TRUE))
  install.packages("sp")
require(sp)

library(maptools)
library(geoR)
library(arctgisbinding)
library(sp)
```

Obr. 36: Použité balíčky v příkladu Kriging

Nastavení parametrů

R-Skript má celkem tři vstupní parametry, které může uživatel zadat. Dva parametry, parametr `in_shp` a parametr `in_rozliseni` jsou parametry, které jsou od uživatele nutné k zadání. Třetím parametrem je parametr `z`, tento parametr je zadán, pokud bodový shapefile má definované nadmořské výšky v atributech. Pokud tomu tak není, jsou zjištěny ze shapefile pomocí funkce `shape$coords.x1` pro osu `x` a `shape$coords.x2` pro osu `y`. Do parametru `in_shp` se načítá bodový shapefile. Parametrem `in_rozliseni` uživatel určí rozlišení pixelu výsledného krigingu.

Posledním parametrem je proměnná `zcolum`. Tato proměnná je vázána na vstupní shapefile a pokud se v shapefile nachází pole s názvem „GRID_DODE“, tak toto pole vloží do proměnné `zcolum`.

```
# vstupní parametry
in_shp = in_params[[1]]
z = in_params[[2]]
in_rozliseni = in_params[[3]]
zcolum = "GRID_CODE"
```

Obr. 37: Vstupní parametry v příkladu Kriging

Načtení dat

V této fázi R-Skriptu je řešeno načtení dat do R-Skriptu. Prvním krokem je otevření shapefilu `in_shp` a to funkcí `readShapePoints`. Po otevření jsou data nahrána do proměnné `shape`.

Druhým krokem je získání souřadnic bodů. K tomuto účelu slouží funkce `shape$coords.x1`, která získá souřadnici `x` z bodového shapefilu a `shape$coords.x2`, která získá souřadnici `y` z bodového shapefilu. Souřadnice `z` je nahrána z proměnné `zcolumn`. Pokud je však proměnná `zcolumn` prázdná, je použito funkce `shape$coords.x3`, která zjistí souřadnici `z` z bodového shapefilu. Souřadnice `z` představuje nadmořskou výšku v bodech.

```
# otevření shapefilu a uložení do promene shape
shape <- readShapePoints(in_shp)

# zjistiění souradnic ze shapefilu
x <- shape$coords.x1
y <- shape$coords.x2
if(zcolumn==""){
  z <- shape$coords.x3
}else{
  z <- shape$GRID_CODE
}
```

Obr. 38: Načtení dat v příkladu Kriging

Úprava dat

V této části R Skriptu se upravují data, která byla načtena, aby mohla být použita v samostatném řešení příkladu. Prvním krokem je převedení souřadnice `z`, tedy výšky na numerickou hodnotu. Tento krok je proveden funkcí `as.numeric`, která ze vstupního řetězce (string) vytvoří pole numerických hodnot, a v desetinných číslech zamění čárku za tečku. Druhým krokem je převedení souřadnic bodů na numerické hodnoty. Tento krok je opět proveden pomocí funkce `as.numeric`, kde tato funkce vytvoří ze souřadnic numerické hodnoty, se kterými již může R Project pracovat.

Třetím krokem v úpravě dat je převedení řetězce, který zadal uživatel do proměnné `in_rozliseni`, na numerickou hodnotu. Numerické `in_rozliseni` je následně uloženo do proměnné `rozliseni`. Posledním krokem je vytvoření tabulky ze souřadnic `x`, `y` a `z` pomocí funkce `data.frame` a následně uložení do proměnné `a`.

```
# prevedení GRID_CODE na numerickou hodnotu
z <- as.numeric(sub(",", ".", shape$GRID_CODE, fixed = TRUE));

#prevod coords.x1 na numerickou hodnotu
x <- as.numeric(sub("-", "", shape$coords.x1, fixed = TRUE));

# prevod coords.x2 na numerickou hodnotu
y <- as.numeric(sub("-", "", shape$coords.x2, fixed = TRUE));

# prevod retezce v promenne rozliseni na numerickou hodnotu
rozliseni = as.numeric(in_rozliseni)

# vytvoreni tabulky s poli x, y a z
a <- data.frame(x,y,z)
```

Obr. 39: Úprava dat v příkladu Kriging

Samotné řešení příkladu

V této části R-skriptu jsou již připravena všechna data a je na řadě vytvoření krigingu. Prvním krokem, který se provede je funkce `shapiro.test`. Tato funkce zjistí hodnotu `p-value` a rovnou je použita funkce `as.numeric`, která slouží pro převedení dat na numerické hodnoty a uložena do proměnné `p-value`. Pokud je `p-value` větší jak 0,05,

je potřeba logaritmovat data pro jednodušší výpočet. Následující cyklus zjistí, hodnotu p-value. Pokud je tato hodnota vyšší, tak zlogaritmuje třetí sloupec dat (nadmořské výšky) a uloží je do nově vytvořeného čtvrtého sloupce. Pokud jsou data menší, jak p-value, tak se hodnoty ze třetího sloupce pouze překopírují do čtvrtého sloupce.

Následuje vytvoření geodat. Provede se pomocí funkce `as.geodata` a uvnitř funkce se zadá nejprve proměnná ze které se vezmou data, poté jaké sloupce patří ke které ose a jako poslední se zadává nový sloupec vytvořený v minulém kroku. V tomto případě jde o sloupec čtvrtý.

Dalším krokem, který je nutné v postupu vytvoření krigingu udělat je vytvoření variogramu z geodat. Toto provedeme pomocí funkce `variog` a uložíme do proměnné `var`. Dalším krokem, který není nutný je vyobrazení variogramu funkcí `plot`. Po vykreslení následuje proložení variogramu křivkou, která je následně uložena do proměnné `varfit`.

```
#Shapiro test pro 3. sloupec dat (souradnice z), pokud je p-value vetsi jak 0,05 nutnost logaritmovat
pvalue = as.numeric(sub(",", ".", shapiro.test(z[1:3000])$p.value, fixed = TRUE));

#logaritmovani dat pokud je p-value vetsi jak 0.05
if(pvalue>0.05){
  a[,4] <- log(a[,3])
}
else{
  a[,4] <- a[,3]
}

#kde coords.col = 1:2 urcuje ze se za x a y souradnici dosazuji hodnoty z prvnio
#a druhoho sloupce, col=4 znamena, ze se za zetovou souradnici (vyska) dosazuje 4. sloupec.
geodata <- as.geodata(a, coords.col=1:2, a.col=4)

#vytvoreni variogramu z geodat a ulozeni do promenne var
var<-variog(geodata)

# zobrazeni variogramu
plot(var)

#prolozeni variogramu krivkou
varfit <- variofit(var)
```

Obr. 40: Samotné řešení příkladu Kriging

V dalším kroku je nutné zjistit, minima a maxima souřadnic, aby se mohl vytvořit rastr, který se následně naplní hodnotami o nadmořské výšky. Pro zjištění minima a maxima je určená funkce `summary`. Tato funkce vypíše základní statistiku obou souřadnic. Následně je pomocí funkce `print` tato základní statistika také zobrazena uživateli. V dalším kroku je vytvářen rastr. Je použita funkce `expand.grid`. Do funkce jsou zadána minima a maxima `x` a `y` a také rozlišení, které určuje velikost, jakou bude mít pixel výsledného rastru. Dalším krokem je vytvoření krigingu. Je využita funkce `krige.conv` a v těle funkce jsou zadány geodata, vytvořený raster, který byl uložen do proměnné `loci` a křivka, proložená variogramem, která byla uložena do proměnné `varfit`. Celá funkce `expand.grid` je uložena do proměnné `kc`. Po tomto kroku následuje odlogaritmování dat, pokud byla logaritmována a je zobrazen kriging pomocí funkce `image`. Posledním krokem je vložení vrstevnic, a to je učiněno pomocí funkce `contour`.

```

#zjistí minimum a maximum souřadnice x a y a které se dosadí do následujícího nastavení velikosti rastru
summary(geodata$coords)

# vypíše základní statistiku souřadnic x a y
print(summary(geodata$coords))

#vytvorí raster do kterého se vykreslí digitální model,
#v seq se zapisuje nejdrive x souřadnice a to v poradí minimum, maximum a krok (velikost pixelu).
loci <- expand.grid(seq(min(x),max(x),b=rozliseni),seq(min(y),max(y),b=rozliseni))

kc <- krige.conv(geodata,loc=loci,krige=krige.control(obj.model=varfit))
# odlogaritmování dat
if(pvalue>0.05){
  print("Logaritmovano")
  krg_predict=kc$predict
}
else{
  print("Ne-logaritmovano")
  krg_predict=kc$predict
}

#vytvoríme kriging
image(kc)

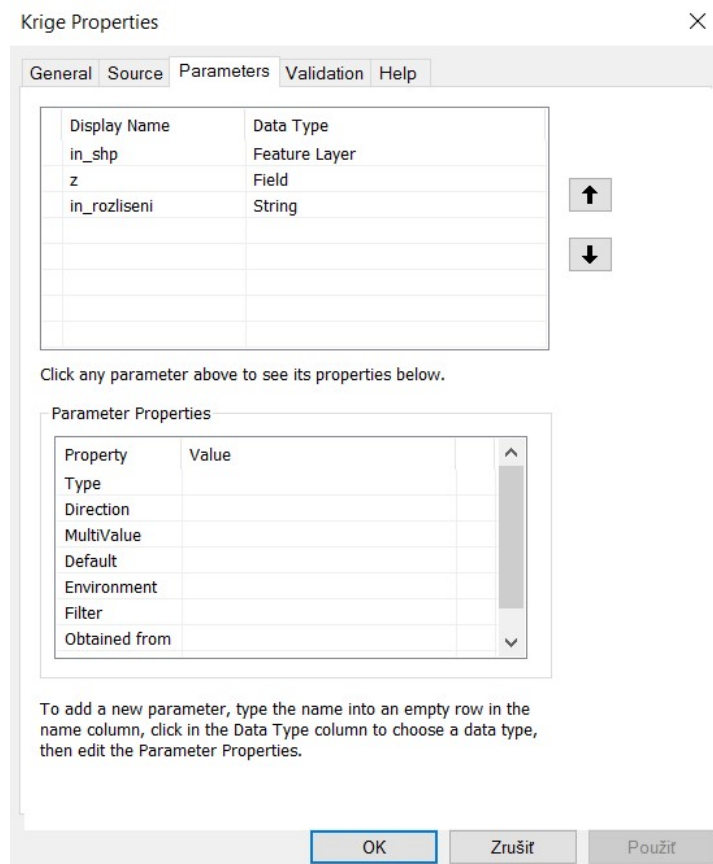
#vlození vrstevnic
contour(kc,value = krg_predict, nlev = 5, add=T)

```

Obr. 41: Samotné řešení příkladu Kriging

2) Toolbox v ArcGIS for Desktop

V ArcGIS for Desktop se vytvoří toolbox podle návodu a v něm se vytvoří skript, který si uživatel pojmenuje. Dále je do skriptu nahrán R-Skript. Při vytváření skriptu je nutné zadat stejné parametry jako jsou v R-Skriptu, aby se R-Skript se skriptem spojili a fungovali tak jak mají. V R-Skriptu musí být očíslovány vstupní parametry stejně tak, jako je pořadí parametrům v náležitém skriptu.



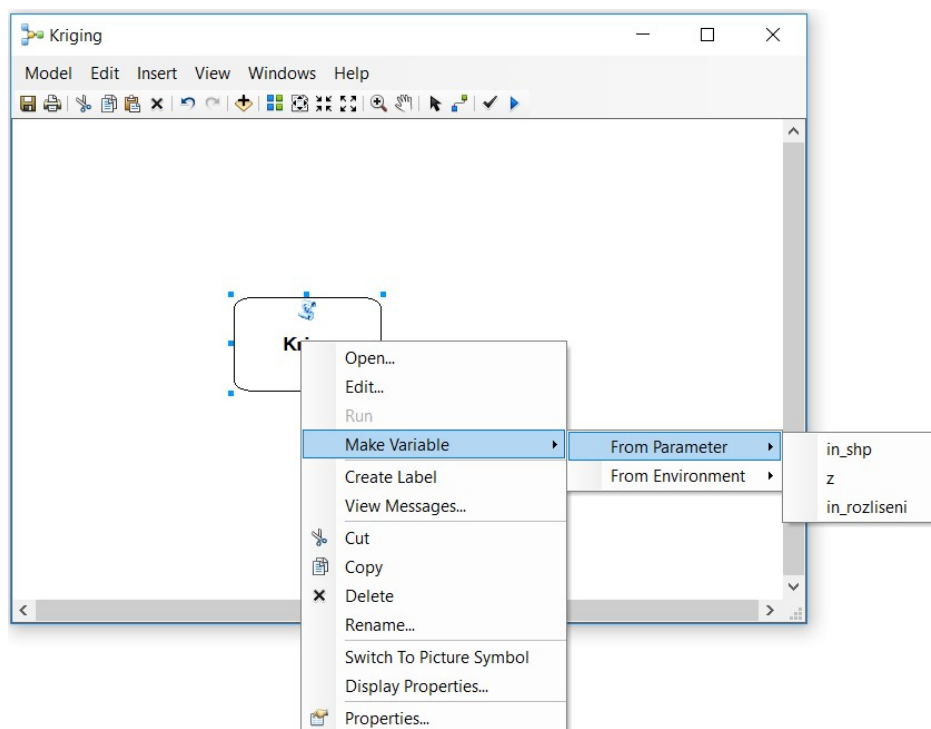
Obr. 42: Parametry skriptu v příkladu Kriging

Parametry skriptu lze i později upravovat dle vlastního uvážení. Dalším krokem je vytvoření modelu v prostoru toolboxu, který však v tomto případě nemusí být vytvořen. Úlohu totiž lze spustit i přímo ze skriptu v aplikaci ArcGIS for Desktop.



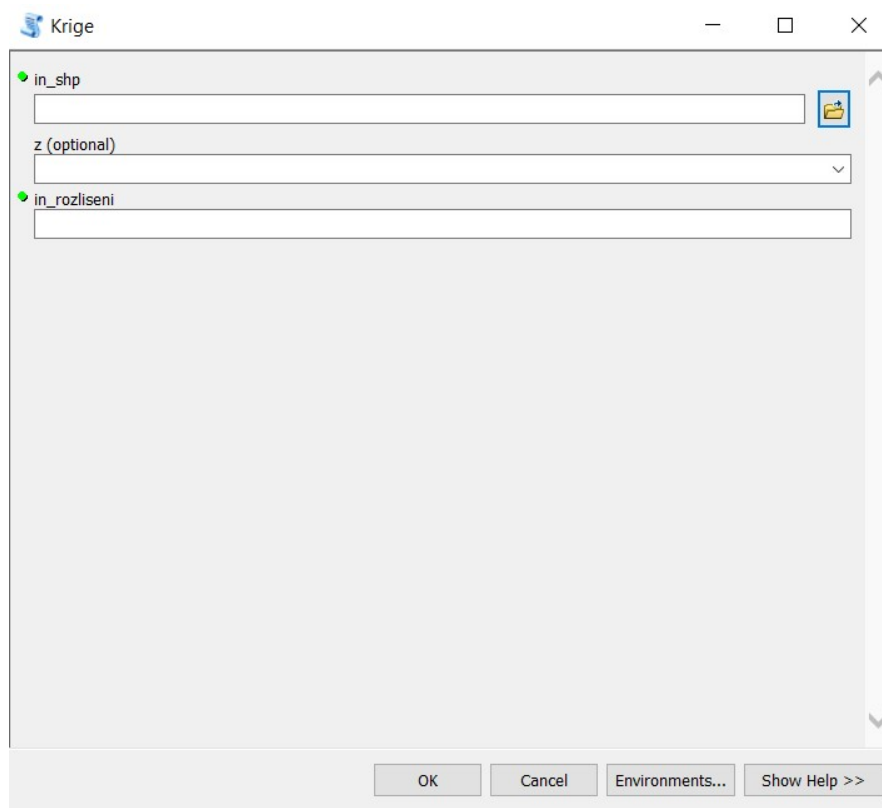
Obr. 43: Toolbox příkladu Kriging

V dalším kroku je otevřeno editační okno modelu a skript je přetažen z okna ArcCatalogu do editačního okna. Dále je nutné, aby byly přidány parametry do editačního okna modelu, a to se provede tak, že uživatel pravým kliknutím myši zadá funkci Make Variable - From Parameter a poté vybere parametry, které chce do modelu přidat. Podle názvů v editačním okně modelu, jsou pak parametry nazvány i v rozhraní, kde se zadávají vstupní parametry pro spuštění modelu. V editačním okně lze také u každé proměnné nastavit předdefinovanou hodnotu, která se uživateli objeví při zadávání parametrů do modelu či skriptu.



Obr. 44: Model příkladu Kriging

V dalším kroku, se nastavují parametry, u kterých má být možnost zadat je uživatelem. Tohoto kroku je docíleno tím způsobem, že na daný parametr klikneme pravým tlačítkem myši a zadáme Model Parameter. Tento model má všechny vybrané parametry nastavené tak, že se dají uživatelem nastavit. Pokud má být parametr předdefinovaný, musí být daný parametr vyplněný příslušnou hodnotou. Toto je použito hlavně u parametrů, které nejsou pro uživatele povinné na zadávání. Z obrázku jsou patrné parametry, které musí uživatel vyplnit, aby mohl skript spustit. U parametrů, které musí uživatel vyplnit je zelená tečka.



Obr. 45: Rozhraní příkladu Kriging

5.4 Wordcloud

Tento příklad využívá propojení z R Project do ArcGIS Desktop. Jde o příklad, na kterém je zobrazena vizualizace z R Project. Vytvořený wordcloud je možné použít jako doplňkový kompoziční prvek k mapě či jakékoliv jiné práci.

1) R-Skript

Balíčky

Pro tuto úlohu je nutné si aktivovat celkem tři balíčky. Jsou jimi knihovny "wordcloud", "RColorBrewer" a "arcgisbinding". Knihovna "wordcloud" je důležitá pro finální vytvoření wordcloudu a obsahuje pro to nezbytné funkce. Druhou knihovnou, která je využívána pro tento R-Skript je knihovna "arcgisbinding". Takto knihovna zajišťuje, aby se shapefile mohl načíst do R Project a dále s ním pracovat. Třetí knihovnou je knihovna "RColorBrewer", která je zde pouze jako doplňková, jelikož slouží pouze k obarvení výsledného wordcloudu.

R-Skript nejprve projde všechny knihovny R Project a zkontroluje, zda jsou tyto knihovny nainstalované. Pokud ne, stáhne tyto knihovny a nainstaluje je.

Dalším krokem po instalaci balíčků je jejich aktivování. To se provede pomocí funkce library a v kulatých závorkách se následně zapíše v uvozovkách název balíčku, který chceme aktivovat. Po tomto kroku již jsou balíčky plně připraveny k použití.

```

# instalace balicku
if (!requireNamespace("wordcloud", quietly = TRUE))
  install.packages("wordcloud")
require(wordcloud)

if (!requireNamespace("RColorBrewer", quietly = TRUE))
  install.packages("RColorBrewer")
require(RColorBrewer)

if (!requireNamespace("arcgisbinding", quietly = TRUE))
  install.packages("arcgisbinding")
require(arcgisbinding)

# nacteni balicku
library("wordcloud")
library("RColorBrewer")
library("arcgisbinding")

```

Obr. 46: Použité balíčky v příkladu Wordcloud

Proměnné

Proměnné jsou vstupními parametry skriptu. Dělí se na parametry povinné, kde musí být zadána hodnota a na parametry nepovinné. Povinnými parametry v tomto R-skriptu jsou parametry `in_shp`, `in_x` a `in_y`. Parametr `in_shp` značí vstupní shapefile, ze kterého jsou vybírána slova a k nim jejich hodnoty pro vytvoření wordcloudu. Parametrem `in_x` vybíráme slova, která budou vyobrazena na výsledném wordcloudu a parametrem `in_y` k nim přiřazené hodnoty. Dalšími dvěma povinnými parametry tohoto R-Skriptu je umístění souboru, kam se uloží obrázek wordcloudu, v tomto R-Skriptu je tento parametr nazván `out_slozka`, a název výstupního souboru, který je v tomto R-Skriptu nazván `out_soubor`.

Nepovinnými parametry v této úloze jsou parametry `in_stupnice`, `in_poc_barev`, `in_poc_slov` a `in_min_poc_slov`. Parametr `in_stupnice` určuje, kterou barevnou stupnici ze sady `RColorBrewer` použijeme a parametr `in_pocet_barev` nastaví počet intervalů barev. Dalším nepovinným parametrem, který vstupuje do této úlohy je parametr `in_poc_slov`, tento parametr nastavuje maximální počet slov, která jsou ve výsledném wordcloudu vykreslena. Posledním nepovinným parametrem je parametr `in_min_poc_slov`, který nastaví hodnotu minimální hodnoty slova, které je vykresleno.

```

# vstupni parametry
in_shp = in_params[[1]]
in_x = in_params[[2]]
in_y = in_params[[3]]
in_stupnice = in_params[[4]]
in_poc_barev = in_params[[5]]
in_poc_slov = in_params[[6]]
in_min_poc_slov = in_params[[7]]
out_slozka = in_params[[8]]
out_soubor = in_params[[9]]

```

Obr. 47: Proměnné v příkladu Wordcloud

Načtení dat

Dalším krokem v R-Skriptu je načtení dat. Vstupním souborem do této tabulky může být tabulka či shapefile. Pomocí funkce `arc.open` je soubor z ArcGISu otevřen a následně uložen do proměnné `shp`. Dalším krokem je nutnost vybrat ze souboru sloupce se slovy (`in_x`) a sloupce s hodnotami (`in_y`). Tento krok provedeme pomocí funkce `arc.select`. Tato funkce z objektu `shp` vybere sloupec, který byl vybrán jako slova, tedy `in_x` a sloupec, který byl vybrán jako hodnoty, tedy `in_y`. A tyto dva sloupce uloží do proměnné `dataset`.


```
# otevreni shp
shp <- arc.open(in_shp)

# vytvoreni tabulky se sloupci in_x, in_y
dataset<- arc.select(object = shp, fields = c(in_x,in_y))
```

Obr. 48: Načtení dat v příkladu Wordcloud

Úprava dat

První fáze úpravy dat je převedení stringů, tedy řetězců, na numerické hodnoty. Tato změna se týká vstupního parametru `in_poc_barev`, tedy počet barev barevné stupnice. Také se týká proměnné `in_poc_slov`, tedy maximální počtu slov ve wordcloudu a v poslední řadě také proměnné `in_min_poc_slov`, tedy minimální hodnota slova, která je ještě na výsledném wordcloudu zobrazena.

Druhou úpravou dat v této úloze je převedení hodnot, tedy sloupce `in_y`, na numerické hodnoty. Tento krok je proveden pomocí funkce `as.numeric`, který v desetinných číslech zamění čárku za tečku. Výsledek tohoto kroku je následně uložen do proměnné `ycol`, která už může vstupovat do funkce `wordcloud`.

```
# převedení stringu na numericke hodnoty
in_poc_barev = as.numeric(in_poc_barev)
in_poc_slov = as.numeric(in_poc_slov)
in_min_poc_slov = as.numeric(in_min_poc_slov)

#in_stupnice = as.factor(in_stupnice)

# prevod in_y na numericou hodnotu
ycol= as.numeric(sub(",", ".", dataset[[in_y]], fixed = TRUE));
```

Obr. 49: Úprava dat v příkladu Wordcloud

Samostatné řešení příkladu

V této úloze je hlavním úkolem vytvořit wordcloud, tedy mrak ze slov. Funkce, pro vytvoření wordcloudu se nazývá `wordcloud`. V kulatých závorkách se nejprve zadává sloupec z tabulky `dataset`, tedy sloupec se slovy (`in_x`), které má wordcloud vypsat. Druhým parametrem, který se zadává je frekvence slov, v tomto případě proměnná `ycol` do které byly uloženy numerické hodnoty k daným slovům. Dalším parametrem, který ve funkci `wordcloud` zadáváme je minimální frekvence slov, tedy jaká je minimální hodnota slova, která je ještě vyobrazena. Další, tedy již čtvrtým parametrem je maximální počet slov, kterým se určí, kolik slov se maximálně na výsledném wordcloudu zobrazí. Další je pak funkce `random.order`, která vytvoří vždy jiné vyobrazení wordcloudu. Posledním parametrem, který se zadává, je barevná stupnice z `ColorBrewer` a počet barev této stupnice.

K samotnému řešení příkladu jsou ještě přidány funkce, které uloží vytvořený obrázek wordcloudu v typu `.png` do vybrané složky, která byla ve vstupních parametrech vybrána. První částí je vytvoření adresáře, kam se má obrázek wordcloudu uložit. Toho je docíleno pomocí funkce `setwd` (`set workspace directory`). Následně se funkcí `png` vytvoří ve složce, kterou zadáme do vstupního parametru `out_slozka` s názvem souboru, který zadáme do proměnné `out_soubor`. V kulatých závorkách funkce `png` je nutné spojit řetězec názvu výstupního souboru a přípony `.png`. Po vytvoření wordcloudu je nutné ukončit zapisování do souboru `out_soubor` pomocí funkce `dev.off()`.

```

# Nastavení výstupní složky
setwd(out_složka)

# Vytvoření png
png(filename = paste(out_soubor, ".png", sep=""))

wordcloud(words = dataset[[in_x]], # slova
          freq = ycol, # hodnoty
          min.freq = in_min_poc_slov, # minimální hodnota slova
          max.words=in_poc_slov, # maximální počet slov
          random.order=FALSE,
          rot.per=0.35,
          colors=brewer.pal(in_poc_barev, in_stupnice)) # obarvení slov, první počet barev a druhý druh stupnice

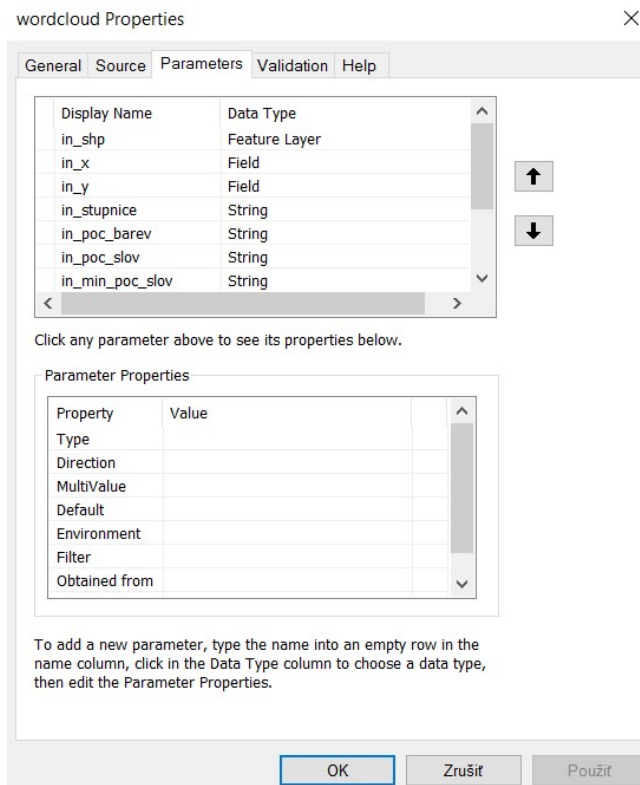
# Uzavření zapisování souboru png
dev.off()

```

Obr. 50: Samotné řešení příkladu Wordcloud

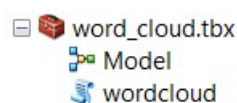
2) Toolbox v aplikaci ArcGIS for Desktop

V aplikaci ArcGIS for Desktop se vytvoří toolbox podle návodu a v něm se vytvoří skript. Dále se do skriptu nahraje R-Skript. Při vytváření skriptu je nutné zadat stejné parametry jako jsou v R-Skriptu, aby se R-Skript se skriptem spojili a fungovali tak jak mají.



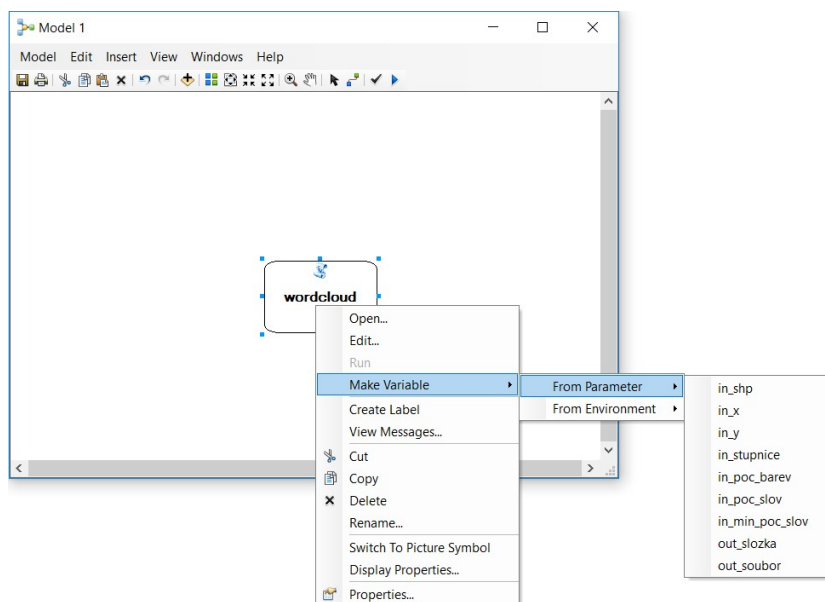
Obr. 51: Parametry skriptu příkladu Wordcloud

Parametry skriptu jde i později upravovat dle vlastního uvážení. Dalším krokem je vytvoření modelu v prostoru toolboxu, který však v tomto případě nemusí být vytvořen. Úlohu totiž lze spustit i přímo ze skriptu v aplikaci ArcGIS for Desktop.



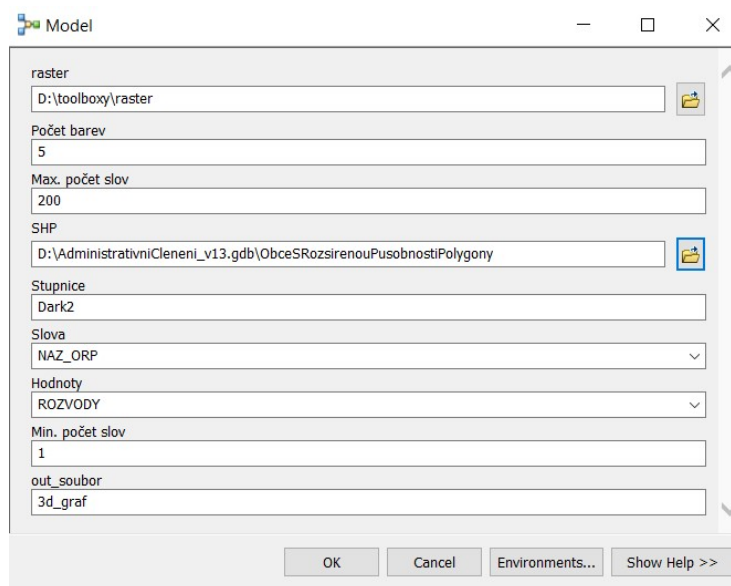
Obr. 52: Toolbox příkladu Wordcloud

V dalším kroku se otevře editační okno modelu a skript se přetáhne z ArcCatalogu do editačního okna. Dále je nutné přidat parametry do editačního okna modelu, a to se provede, pokud uživatel pravým kliknutím myši zadá funkci Make Variable - From Parameter a poté jsou vybrány parametry, které chce uživatel do modelu přidat. Podle názvů v editačním okně modelu, jsou pak parametry nazvány i v rozhraní, kde se zadávají vstupní parametry pro spuštění modelu.



Obr. 53: Model příkladu Wordcloud

Dále je nutné nastavit parametry, u kterých má být možnost zadat je uživatelem. Tohoto kroku je docíleno tím způsobem, že na daný parametr klikneme pravým tlačítkem myši a zadáme Model Parameter. Tento model má všechny vybrané parametry nastavené tak, že se dají uživatelem nastavit. Pokud má být parametr předdefinovaný, musí být daný parametr vyplněný příslušnou hodnotou. Toto je použito hlavně u parametrů, které nejsou pro uživatele povinné na zadávání. Z obrázku jsou patrné parametry, které musí uživatel vyplnit, aby mohl skript spustit.



Obr. 54: Rozhraní příkladu Wordcloud

6 VÝSLEDKY

6.1 Obecné vytvoření příkladu

Prvním krokem, pokud si chce uživatel vytvořit příklad založený na propojení R-Bridge je nastudování si problematiky příkladu v aplikaci R Project. Uživatel by měl nejdříve zjistit, jaké funkce bude potřebovat pro řešení úlohy a také si promyslet co od takové úlohy očekává. S náročností příkladu roste doba, kterou uživatel stráví při vytváření R-Skriptu. Také je nutné, aby měl uživatel připravena testovací data, na kterých bude moct po částech daný příklad testovat. V některých případech je lepší mít více variant testovacích dat, protože když jednou R-Skript proběhne v pořádku, tak s jinými daty už se to povést nemusí.

6.2.1 R-Skript

Balíčky

Při vytváření R-Skriptu je prvním krokem aktivování balíčků neboli knihoven, které obsahují funkce, jaké základní verze R Project neobsahuje. Těchto balíčků jsou v této době již tisíce. Tyto balíčky se úlohou od úlohy liší, některé pomáhají funkcemi při načítání dat, jiné jsou nutné při úpravě dat a některé obsahují funkce pro řešení vlastního příkladu.

Při zadávání knihovny, která má být aktivována je nejprve nutné vytvořit cyklus, který projde všechny nainstalované knihovny v počítači uživatele a pokud knihovnu nenajde, tak ji stáhne a nainstaluje. Pro tento krok je nutné, aby uživatel byl připojen k internetu. Dalším krokem po nainstalování balíčku je aktivace těchto balíčků.

Nastavení parametrů

Druhou částí R-Skriptu, která je obsažena je část, ve které se zadávají vstupní a výstupní parametry a ukládají se do proměnných. Dělí se na parametry vstupní, které většinou zadává uživatel a parametry výstupní. Vstupní parametry se dále dělí na parametry povinné a nepovinné. Povinnými parametry jsou myšleny ty, které jsou nutné pro průběh dané úlohy. Příkladem povinného parametru je například vstupní tabulka či shapefile Na druhou stranu nepovinné parametry jsou parametry, které uživatel nemusí nutně zadat. Příkladem nepovinného parametru je třeba popisek osy či název grafu.

Načtení dat

Třetí částí, která se v R-Skriptu řeší je načtení dat. Většinou jsou data načítána z aplikace ArcGIS for Desktop, tudíž jsou tato data nejčastěji ve formě tabulek, shapefilů či rastrů. V této části R Skriptu je často využíván balíček `arcgisbinding`, který svou funkcí `arc.open` pomáhá otevírat tabulky a shapefilly. Tato funkce, která otevře například `shapefile` je následně uložena do proměnné, aby se s daty mohlo dále pracovat. Dalším krokem je vybrání polí z načteného shapefilu a vytvoření tabulky. Tabulka je opět uložena, aby se s ní mohlo pracovat při úpravě dat.

Druhou možností, jak načíst data, je funkce `readShapePoints`. Tato funkce načte bodová data a dokáže z nich zjistit jejich souřadnice, i když se souřadnice těchto bodů nevyskytují v atributové tabulce shapefilu a to pomocí funkce `shape$coords.x1`, kde

shape určuje název proměnné, do které byly vstupní data uložena a x1 první souřadnici. Druhá souřadnice by vypadala takto: shape\$coords.x2.

Úprava dat

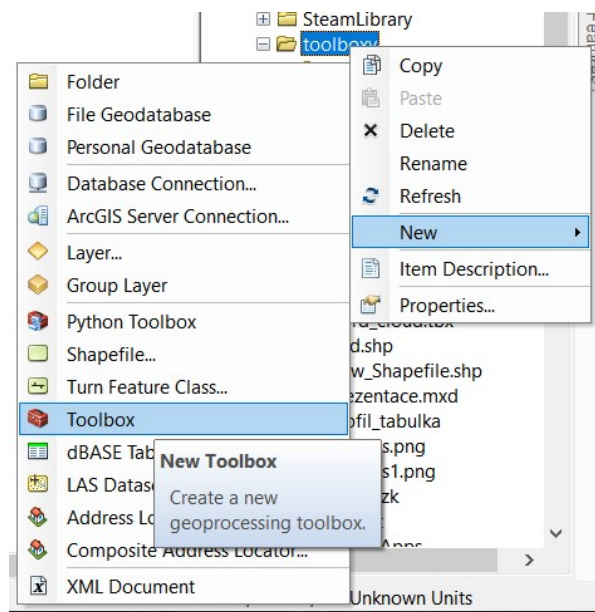
Ve čtvrté fázi R-Skriptu je nutné data upravit, aby je bylo možno použít pro fázi vlastního řešení příkladu. Většinou se v této fázi vytváření R-Skriptu pracuje s převodem řetězců (string) na numerické hodnoty. Tento krok je zajištěn funkcí `as.numeric`. Převedené hodnoty se následně uloží do proměnné, aby bylo možno s nimi dále počítat. Dalším možným krokem, který je prováděn v této fázi je vytvoření tabulek s převedenými hodnotami pro pozdější pracování. Ve výsledku má tato fáze R-Skriptu připravit data, aby je bylo možno zadat do vlastního řešení příkladu.

Vlastní řešení příkladu

V této fázi přichází na řadu použití funkce či funkcí, které vyřeší problematiku, kterou chtěl uživatel vyřešit. Na řadu většinou přichází balíček, který zajišťuje, aby funkce fungovaly. V těle funkce se využijí nejen vstupní data, ale také povinné a nepovinné parametry, které jsou použity v proměnných, do kterých byly uloženy, a které uživatel zadával na počátku R-Skriptu. Tyto parametry nahrazují přímo zadané hodnoty ve funkci, a proto může uživatel, který následně využívá vytvořený skript z toolboxu zadat hodnotu dle vlastního uvážení.

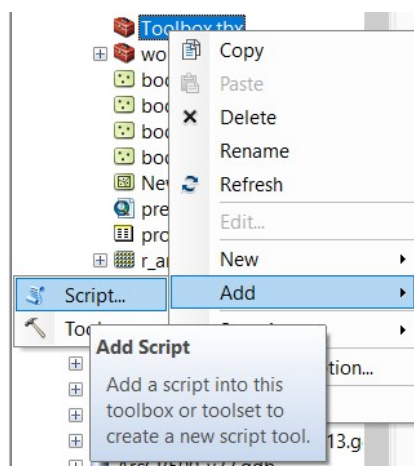
6.2.2 Toolbox

Pro použití vytvořeného R-Skriptu při implementaci R Project do ArcGIS for Desktop je nutné R-Skript nahrát do skriptu, který pak můžeme v ArcGIS for Desktop spustit. Prvním krokem, než je však R-Skript nahrát do skriptu je vytvoření toolboxu. V Arc Catalogu se vybere složka, ve které má být toolbox vytvořen a přes pravé tlačítko myši se vybere New -> toolbox. Následuje pojmenování toolboxu.



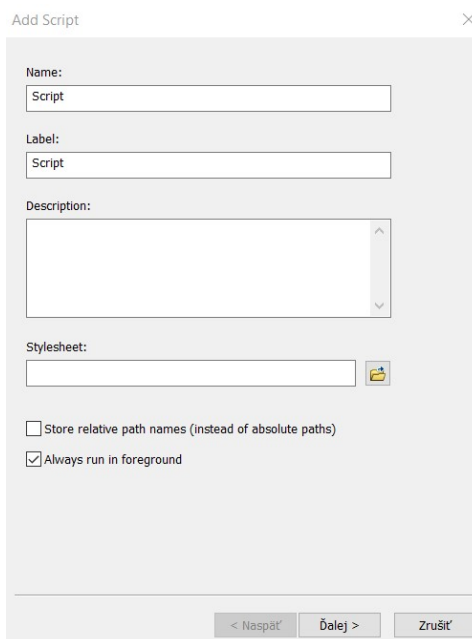
Obr. 55: Nový Toolbox

Dalším krokem je vytvoření skriptu. Tento úkon je opět prováděn v Arc Catalogu a přes pravé tlačítko myši na právě vytvořený toolbox a záložku Add -> script se v toolboxu vytvoří nový skript.



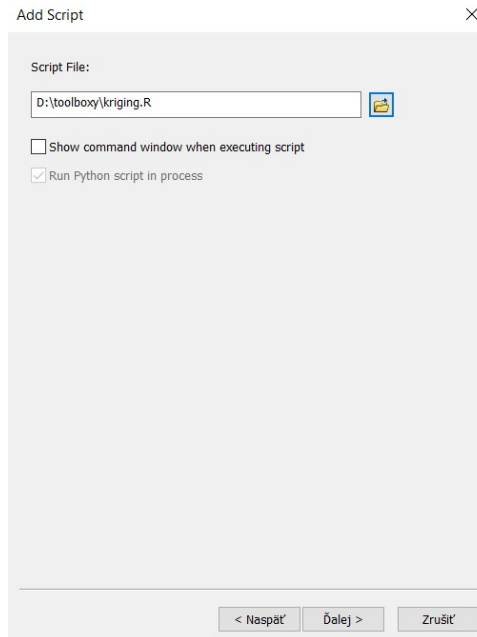
Obr. 56: Přidání skriptu

Po vytvoření skriptu uživatel zadá jméno skriptu a popisek skriptu. Uživatel může také dopsat popisné informace do kolonky Description.



Obr. 57: Název a popisek skriptu

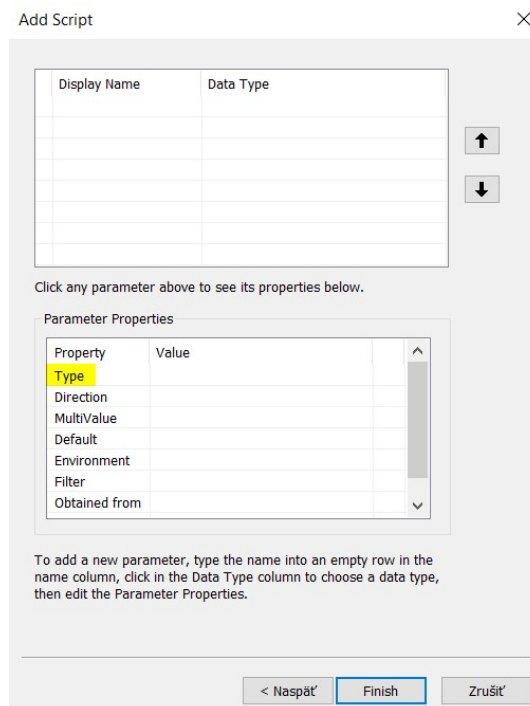
Dalším krokem je zadání R-Skriptu, který do skriptu má být vložen.



Obr. 58: Zadání R-Skriptu

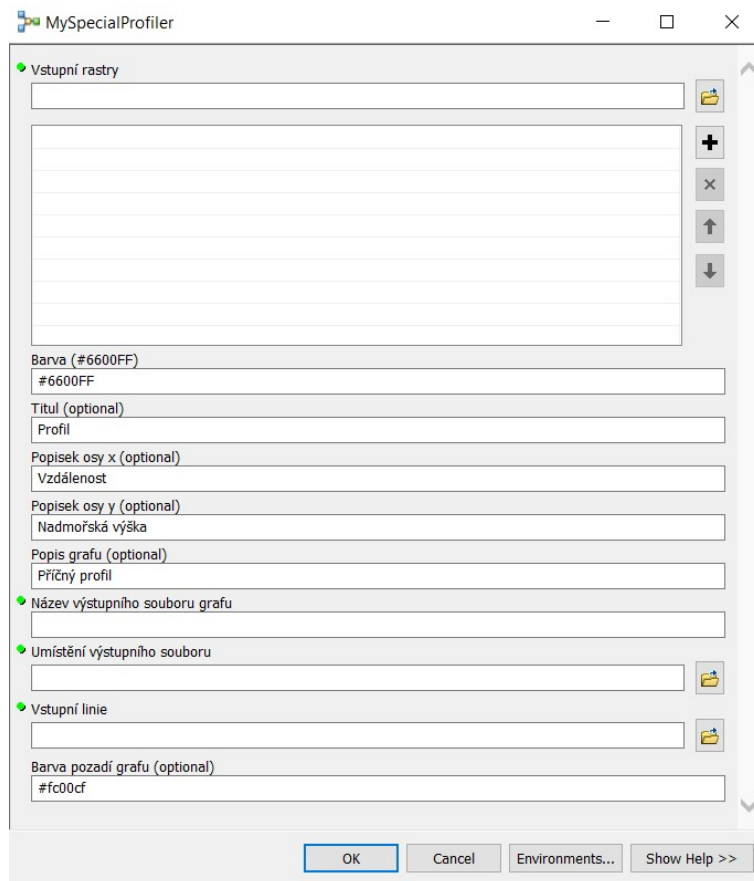
Posledním krokem vytváření skriptu je zadání parametrů. Jelikož lze skript vytvářet dříve, než je R-Skript úplně dokončen, tak se parametry doplňují průběžně. Zadané parametry musí mít stejný název, a hlavně pořadí, jak jsou nadefinovány v sekci nastavení parametrů R-Skriptu.

Pomocí funkce type nastaví autor, zda chce, aby byl parametr povinný při zadávání parametrů.



Obr. 59: Parametry skriptu

Po spuštění skriptu může vypadat rozhraní v toolboxu takto. Příkladem je skript na vytváření profilu z rastru a liniové vrstvy. Parametry, které jsou označené zelenou tečkou, jsou parametry povinné, uživatel je musí zadat. Ostatní parametry jsou nepovinné a lze je předvyplnit, aby je nemusel uživatel vyplňovat.



Obr. 60: Uživatelské rozhraní

7 DISKUZE

Práce byla vytvořena, aby byla popsána možnost propojení mezi aplikacemi ESRI ArcGIS for Desktop a aplikace The R Project for Statistical computing. Výhodou takového propojení je, že u vytvořených úloh již nemusí uživatel otevírat druhý software. Například, pokud má vytvořený toolbox v aplikaci ArcGIS for Desktop, spustí jej přímo z ArcGIS for Desktop a nemusí aplikaci R Project vůbec otevírat. Nutnost instalace obou aplikací však zůstává, navíc je nutné mít nainstalované propojení, které však je velmi jednoduché. Jedná se pouze o spuštění dvou instalačních skriptů.

Úlohy, které byly vytvořeny pomocí daného propojení by měli sloužit k účelům pomoci uživatelům jednodušeji vytvořit složitější a komplexnější úlohy. Hlavním přínosem je, že lze využívat funkce z obou softwarů zároveň. Tudíž lze například využívat složité statistické funkce, které se nevyskytují v ArcGIS for Desktop a nemuset R Project ani otevřít. Dalším krokem, kam by podle mého názoru měla práce ubírat svou cestu, je implementování úlohy vyšších rozměrů. Tedy komplexní úlohy, která by byla přínosná a ideálně by alespoň polovinu funkcí tvořily funkce z jedné z aplikací.

Při srovnání prostředí, ve kterých se úlohy dokončují, je každopádně uživatelsky přijatelnější prostředí toolboxů v ArcGIS for Desktop. Tím že je R-Skript nahrán do skriptu v toolboxu aplikace ArcGIS for Desktop, je zde možnost modelování skriptu do podoby, kterou požaduje uživatel, a která je snadno pochopitelná pro každého uživatele, který by s daným skriptem pracoval. Pokud implementujeme R Project do ArcGIS for Desktop, je uživatelským rozhraním, kde uživatel zadává povinné a volitelné parametry, toolbox. V R Project však tato varianta chybí, tudíž je vytvořený R-Skript přímo spouštěn z okna R Project.

Při instalaci propojení R-Bridge byl hned prvním problémem špatný název počítače. Název počítače nesmí obsahovat písmena s háčky a čárky, ani další nadstavbové znaky, jinak instalace modulu R-Bridge neproběhne správně. Nepomáhá ani počítač přejmenovat, modul si pamatuje celou cestu, jak byla s původním názvem počítače. Řešením tohoto problému bylo nakonec velmi jednoduché a stačilo vytvořit v počítači nový uživatelský úče již se správnými znaky názvu počítače.

Dalším krokem, ve kterém byly problémy při dokončování funkčnosti propojení bylo převedení 64 bitové verze R Project na 32 bitovou verzi, ve které propojení pomocí modulu R-Bridge funguje. Tento problém byl vyřešen ručním přepnutím verzí v R Project.

V realizaci R-Skriptu nastaly asi největší problémy s převodem vstupních dat do formy, kterou již R Project hravě zvládne přečíst. Tato forma byla u každé úlohy trochu jiná, ale vždy zdárně testováním nalezena. Druhý, již méně velký problém pak nastal při definování proměnných a jejich následných datových typů, když byl R-Skript v toolboxu aplikace ArcGIS for Desktop nahráván do skriptu.

V úloze kriging se vyskytl problém s počtem vstupních bodů. Menší zobrazené území s menším počtem bodů v pravidelné mřížce prošlo, ale větší území již ne. Tento fakt je důsledkem toho, že v aplikaci ArcGIS for Desktop je území interpolováno po částech, a tak paměť RAM není naplněna.

Realizace jedné úlohy, od sepsání R-Skriptu, průběžném testování na zkušebních datech a následném vytvoření toolboxu v ArcGIS for Desktop zabrali kolem pár dnů práce. Rychlost vytváření příkladů se také odvíjela od počtu vytvořených úloh. Čím více jich bylo vytvořených, tím jednodušší poté bylo další úlohu vyřešit. Doba, za kterou je úloha vytvořena také závisí na složitosti příkladu, na autorovi příkladu, který vytváří

danou úlohu a počtu proměnných, které musí být nadefinovány. Tento výčet reprezentuje hlavní části při realizování dané úlohy, dále jsou zde také další problémy, které mohou nastat při realizaci příkladu, ale všechny byly postupným testováním na zkušebních datech vyřešeny a odstraněny.

8 ZÁVĚR

Cílem této práce bylo vytvořit step by step návod, jak realizovat propojení mezi ArcGIS for Desktop a The R Project for Statistical Computing. Druhým krokem bylo vytvoření cvičných úloh a následné otestování propojení těmito úlohami. Poslední fází práce bylo z nastudované referenční úlohy a úloh samostatně vytvořených vytvořit obecný návod, jak vytvořit úlohu.

Bylo popsáno propojení pomocí modulu R-Bridge a vytvořen step by step návod, jak dané propojení realizovat. Také bylo vytvořeno celkem sedm úloh, z toho jedna referenční úloha (vytvořená). Čtyři úlohy demonstrovali propojení z R Project do ArcGIS for Desktop a dvě úlohy demonstrovali propojení z ArcGIS for Desktop do R Project.

Při instalaci propojení R-Bridge nastal první problém, který však byl vyřešen přejmenováním, aby na cestě k místu uložení ArcGIS a R Project nebyla použita žádná diakritika. Dalším problémem, který se objevil se týkal verze R Project. Tento problém byl vyřešen ručním přepnutím z 64-bitové verze na 32-bitovou verzi aplikace R Project. Při tvorbě úloh bylo testováno po každé menší úpravě, zda skript funguje, aby se předešlo případným komplikacím.

Možnost propojení pomocí mostu R-Bridge dává možnost vytvářet úlohy v ArcGIS for Desktop pomocí R Project. Zatímco ArcGIS for Desktop je založen na pythonu, který je vcelku náročný, R Project je uživatelsky přijatelnější. Pokud R Project implementujeme do ArcGIS for Desktop dostaneme tak uživatelsky přijatelnou aplikaci s velkou možností statistických analýz a vizualizací do známého prostředí.

POUŽITÁ LITERATURA

Using the R-ArcGIS Bridge: the arcgisbinding Package [online]. 2016 [cit. 2017-08-03]. Dostupné z: <https://r-arcgis.github.io/assets/arcgisbinding-vignette.html>

Package 'arcgisbinding' [online]. 2016 [cit. 2017-08-05]. Dostupné z: <https://r-arcgis.github.io/assets/arcgisbinding.pdf>

Combining ArcGIS and R - Clustering Toolbox [online]. 2016 [cit. 2017-08-10]. Dostupné z: <http://r-video-tutorial.blogspot.cz/2016/07/combine-arcgis-and-r-clustering-toolbox.html>

Install the R-ArcGIS Bridge [online]. 2016 [cit. 2017-03-21]. Dostupné z: <https://github.com/R-ArcGIS/r-bridge-install>

Stack Profile [online]. 2017 [cit. 2017-04-24]. Dostupné z: <http://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/stack-profile.htm>

INFORMAČNÍ ZDROJE

<http://www.statmethods.net/graphs/line.html>

<http://www.statmethods.net/graphs/scatterplot.html>

<http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know>

PŘÍLOHY

SEZNAM PŘÍLOH

Vázané přílohy:

Příloha 1	Načtení rasteru
Příloha 2	Načtení SHP
Příloha 3	Profil
Příloha 4	3D Grafy
Příloha 5	Kriging
Příloha 6	Worldcloud

Volné přílohy:

Příloha 7	Poster
Příloha 8	DVD

Popis struktury DVD

Adresáře:

- Text práce
- Webové stránky
- Toolboxy
- R-Skripty

PŘÍLOHA 1 (Načtení rasteru)

```
tool_exec <- function(in_params, out_params)
{
  if (!requireNamespace("raster", quietly = TRUE))
    install.packages("raster")
  require(raster)

  if (!requireNamespace("sp", quietly = TRUE))
    install.packages("sp")
  require(sp)

  if (!requireNamespace("rgdal", quietly = TRUE))
    install.packages("rgdal")
  require(rgdal)

  if (!requireNamespace("gdalutils", quietly = TRUE))
    install.packages("gdalutils")
  require(gdalutils)

  library(raster)
  library(sp)
  library(rgdal)

  setwd("D:/toolbox")

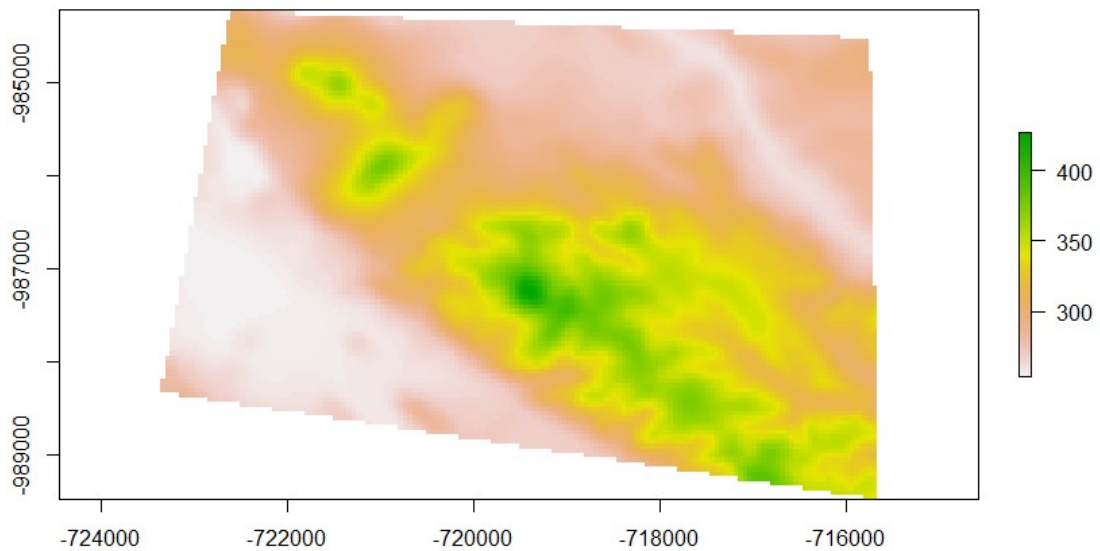
  DEM <- raster("bo/raster_zk1.tif")

  DEM

  plot(DEM)

  return(out_params)
}
```

Obr. 1: R-Skript úlohy Načtení rasteru



Obr. 2: Výsledek úlohy Načtení rasteru

PŘÍLOHA 2 (Načtení SHP)

```
tool_exec <- function(in_params, out_params)
{
  #načtení balíčků
  if (!requireNamespace("arctgisbinding", quietly = TRUE))
    install.packages("arctgisbinding")
  require(arctgisbinding)

  #aktivace balíčků
  library(arctgisbinding)

  #vstupní parametry
  in_table = in_params

  #otevření shp
  tab <- arc.open(in_table)

  #vytvoření datasetu a vybrání příslušných sloupců
  dataset<- arc.select(object = shp, fields = c(in_x,in_y))

  return(out_params)
}
```

Obr. 3: R-Skript úlohy Načtení SHP

PŘÍLOHA 3 (Profil)

```
tool_exec <- function(in_params, out_params)
{
  if (!requireNamespace("readr", quietly = TRUE))
    install.packages("readr")
  require(readr)

  if (!requireNamespace("arctgisbinding", quietly = TRUE))
    install.packages("arctgisbinding")
  require(arctgisbinding)

  library(arctgisbinding)
  library(readr)

  print(in_params)
  arc.progress_label("-")

  in_table = in_params[[1]]

  visual_variables = 3
  graph_color = in_params[[visual_variables+1]]
  titul = in_params[[visual_variables+2]]
  popisek_osa_x = in_params[[visual_variables+3]]
  popisek_osa_y = in_params[[visual_variables+4]]
  popisek_grafu = in_params[[visual_variables+5]]

  color_variables = visual_variables+5
  bg_color = in_params [[color_variables+1]]

  tab <- arc.open(in_table)
  #print(tab@fields) #seznam poli v tabulce
  dataset<- arc.select(object = tab, fields = c("FIRST_DIST", "FIRST_Z"))

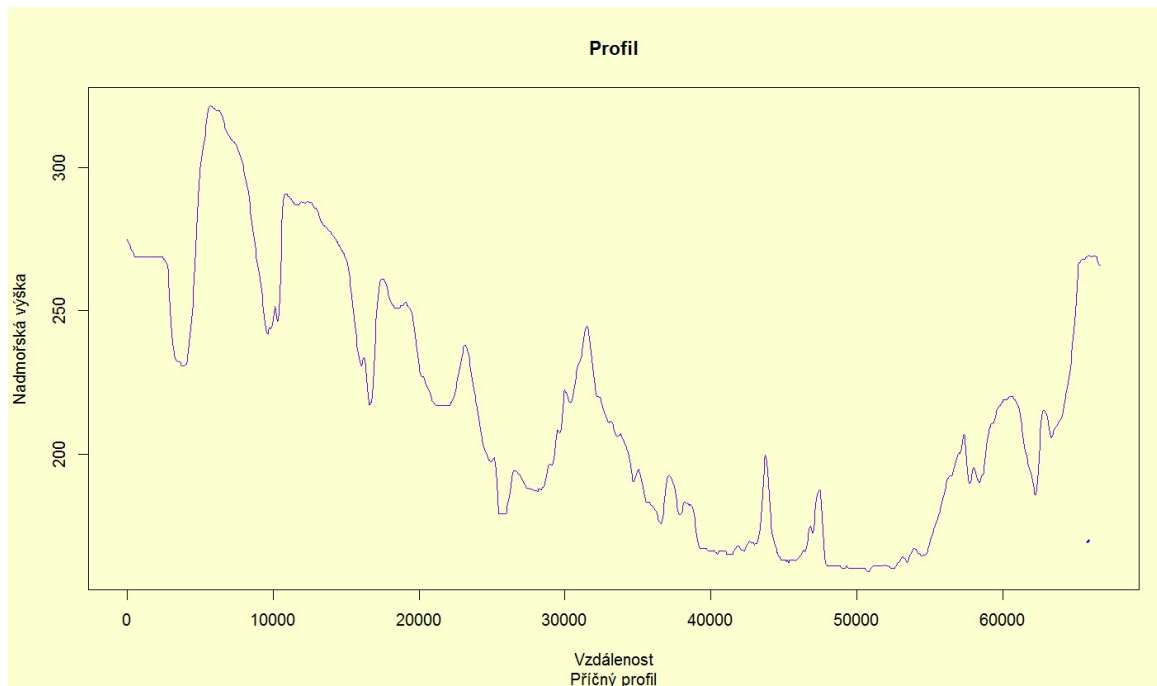
  #text <- read_delim(in_table,
  #                  ";", escape_double = FALSE, col_types = cols(FIRST_Z = col_character()),
  #                  trim_ws = TRUE) #pretypovani sloupce
  #attach(text)

  xcol= as.numeric(sub(",", ".", dataset$FIRST_DIST, fixed = TRUE));
  ycol= as.numeric(sub(",", ".", dataset$FIRST_Z, fixed = TRUE));

  par(bg = bg_color)
  plot(xcol,ycol ,xlim=c(min(xcol),max(xcol)),ylim=c(min(ycol),max(ycol)), main=titul, sub = popisek_grafu,
       xlab= popisek_osa_x , ylab= popisek_osa_y , type="l", asp=0, col=graph_color)

  return(out_params)
}
```

Obr. 4: R-Skript úlohy Profil



Obr. 5: Výsledek úlohy Profil

PŘÍLOHA 4 (3D Grafy)

```

tool_exec <- function(in_params, out_params)
{
  # nainstalování potřebných balíčků
  if (!requireNamespace("readr", quietly = TRUE))
    install.packages("readr")
  require(readr)

  if (!requireNamespace("scatterplot3d", quietly = TRUE))
    install.packages("scatterplot3d")
  require(scatterplot3d)

  if (!requireNamespace("arcgisbinding", quietly = TRUE))
    install.packages("arcgisbinding")
  require(arcgisbinding)

  # aktivování balíčků
  library(scatterplot3d)
  library(readr)
  library(arcgisbinding)

  # parametry
  print(in_params)
  in_shp = in_params[[1]]
  in_x = in_params[[2]]
  in_y = in_params[[3]]
  in_z = in_params[[4]]
  in_type = in_params[[5]]
  in_titul = in_params[[6]]
  in_pop_os_x = in_params[[7]]
  in_pop_os_y = in_params[[8]]
  in_pop_os_z = in_params[[9]]
  in_barva = in_params[[10]]

  print("a")
  shp <- arc.open(in_shp) # otevření shp

  print("b")

  # vytvoření tabulky se sloupci x,y,z
  dataset <- arc.select(object = shp, fields = c(in_x, in_y, in_z))

  print("c")

  xcol = as.numeric(sub(",", ".", dataset[[in_x]], fixed = TRUE));
  ycol = as.numeric(sub(",", ".", dataset[[in_y]], fixed = TRUE));
  zcol = as.numeric(sub(",", ".", dataset[[in_z]], fixed = TRUE));
}

```

Obr. 6: První část R-Skriptu úlohy 3D Grafy

```

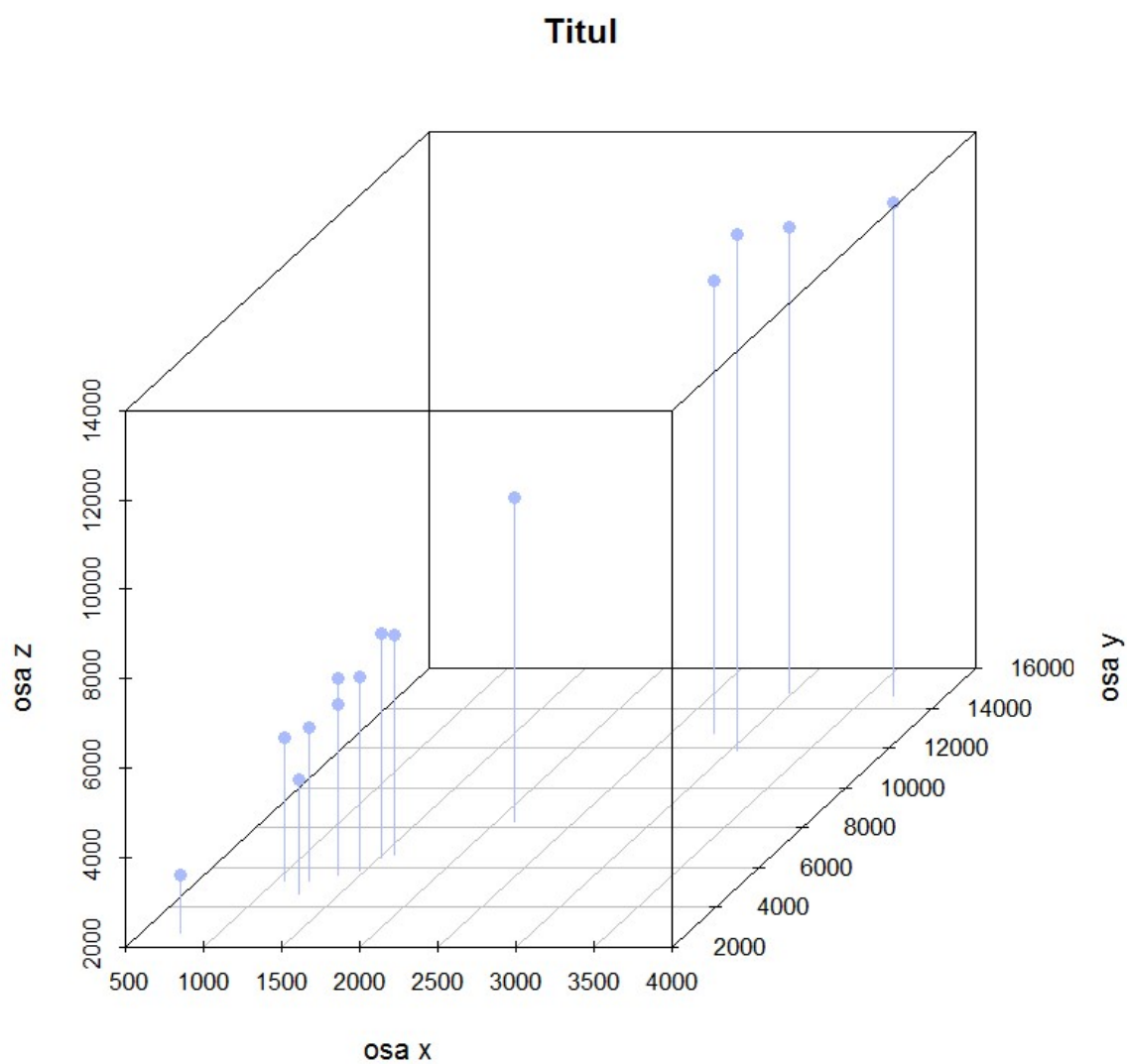
print("")
if(in_type){
  scattertype="h";
}else{
  scattertype="p";
}

with(dataset, {
  scatterplot3d(xcol, ycol, zcol, # x y z
    xlim=c(min(xcol),max(xcol)),
    ylim=c(min(ycol),max(ycol)),
    zlim=c(min(zcol),max(zcol)),
    color=in_barva, # color = barva
    pch=19, # pch = typ značky
    type=scattertype, # h - typ grafu kde linie spojuje body s podložkou
    main=in_titul, # Název grafu
    xlab=in_pop_os_x, # Popisek osy x
    ylab=in_pop_os_y, # Popisek osy y
    zlab=in_pop_os_z) # Popisek osy z
})

return(out_params)
}

```

Obr. 7: Druhá část R-Skriptu úlohy 3D Grafy



Obr. 8: Výsledek úlohy 3D Grafy

PŘÍLOHA 5 (Kriging)

```
tool_exec <- function(in_params, out_params)
{
  if (!requireNamespace("geoR", quietly = TRUE))
    install.packages("geoR")
  require(geoR)

  if (!requireNamespace("arctgisbinding", quietly = TRUE))
    install.packages("arctgisbinding")
  require(arctgisbinding)

  if (!requireNamespace("maptools", quietly = TRUE))
    install.packages("maptools")
  require(maptools)

  if (!requireNamespace("sp", quietly = TRUE))
    install.packages("sp")
  require(sp)

  library(maptools)
  library(geoR)
  library(arctgisbinding)
  library(sp)

  # vstupní parametry
  in_shp = in_params[[1]]
  #x = in_params[[2]]
  #y = in_params[[3]]
  #z = in_params[[4]]
  #rozliseni = in_params[[5]]
  zcolumn = "GRID_CODE"

  tab <- arc.open(in_shp)
  print(tab@fields)
  shape <- readShapePoints("D:/toolbox/bod.shp")

  x <- shape$coords.x1
  y <- shape$coords.x2
  if(zcolumn==""){
  z <- shape$coords.x3
  }else{
  z <- shape$GRID_CODE
  }
  for(n in c(2,5,10,20,50)){
  print(paste("X",x[n]," Y", y[n], " Z",z[n]))
  }
  #z <- as.numeric(sub(",", ".", shape$GRID_CODE, fixed = TRUE));
  mx <- x/1000
  my <- y/1000
  mz <- z/1000
}
```

Obr. 9: První část R-Skriptu úlohy Kriging

```

a <- data.frame(x,y,z)
#a <- as.matrix(a)
print(a)
#Shapiro test pro 3. sloupec dat (souřadnice z), pokud je p-value větší jak 0,05 musíme logaritmovat
#print(shapiro.test(z[1:3000]))$p.value)

pvalue= as.numeric(sub(" ", ".", shapiro.test(z[1:3000]))$p.value, fixed = TRUE));

print(pvalue)
#logaritmování dat pokud je p-value větší jak 0.05
if(pvalue>0.05){
  a[,4] <- log(a[,3])
}
else{ a[,4] <- a[,3]}
#logaritmování dat - přidání sloupce zlogaritmovaných dat
#data[,4] <- log(z)
print("a")
#kde coords.col = 1:2 určuje že se za x a y souřadnici dosazují hodnoty z prvního
#a druhého sloupce, col=4 znamená, že se za zetovou souřadnici (výška) dosazuje 4. sloupec.
geodata <- as.geodata(a, coords.col=1:2, a.col=4)

#vytvoříme variogram z geodet a uložíme ho do proměnné var
var<-vario(geodata)

#proložíme variogram křivkou
varfit <- variofit(var)

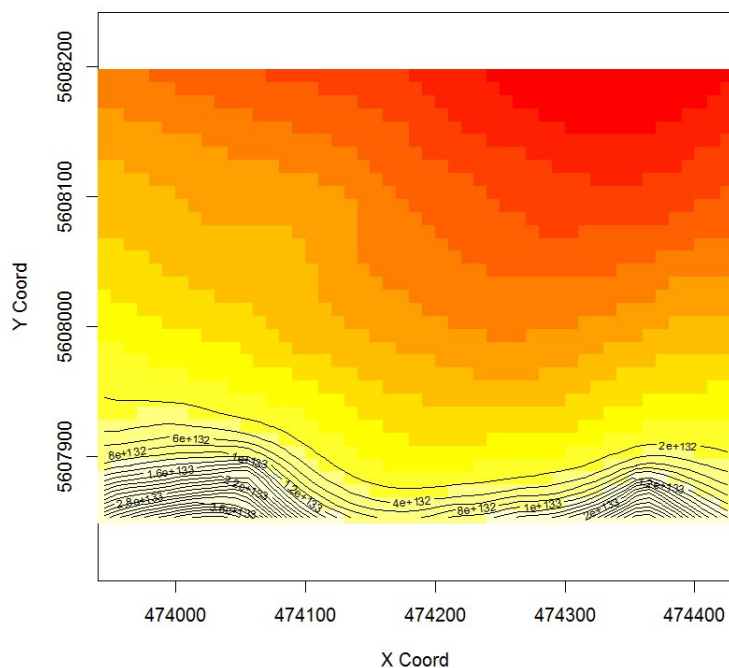
#Zjistíme minimum a maximum souřadnice x a y a dosadíme ho do následujícího nastavení velikosti rastru
summary(geodata$coords)
print(geodata$coords)
#vytvoříme si raster do kterého vykreslíme digitální model,
#v seq zapíšeme nejdříve x souřadnici a to v pořadí minimum, maximum a krok (velikost pixelu).
#Hodnoty zaokrouhlujeme nahoru.
#Rozdíl mezi minimem a maximem u x souřadnice je 290, z čehož vyplývá
#že na x souřadnicích bude 29 pixelů vykresleno jelikož je jeden pixel 10.
loci <- expand.grid(seq(-715686.1,-723336.1,b=50),seq(-984252.4,-989452.4,b=50))
print("c")
# odlogaritmování dat
if(pvalue>0.05){
  kc <- krige.conv(geodata,loc=loci,krige=krige.control(obj.model=varfit))
}
else{ kc <- (loci)}

print(kc)
#poté odlogaritmuje data
#kc <- krige.conv(geodata,loc=loci,krige=krige.control(obj.model=varfit))
print("d")
#vytvoříme kriging
image(kc)
print("e")
#vložíme vrstevnice
contour(kc,value = exp(kc$predict), nlev = 20, add=T)

return(out_params)
}

```

Obr. 10: Druhá část R-Skriptu úlohy Kriging



Obr. 11: Výsledek úlohy Kriging

PŘÍLOHA 6 (Wordcloud)

```
tool_exec <- function(in_params, out_params)
{
  # instalace balíčků
  if (!requireNamespace("tm", quietly = TRUE))
    install.packages("tm")
  require(tm)

  if (!requireNamespace("SnowballC", quietly = TRUE))
    install.packages("SnowballC")
  require(SnowballC)

  if (!requireNamespace("wordcloud", quietly = TRUE))
    install.packages("wordcloud")
  require(wordcloud)

  if (!requireNamespace("RColorBrewer", quietly = TRUE))
    install.packages("RColorBrewer")
  require(RColorBrewer)

  if (!requireNamespace("arccgisbinding", quietly = TRUE))
    install.packages("arccgisbinding")
  require(arccgisbinding)

  # Načtení balíčků
  library("tm")
  library("SnowballC")
  library("wordcloud")
  library("RColorBrewer")
  library("arccgisbinding")

  print(in_params)
  # vstupní parametry
  in_shp = in_params[[1]]
  in_x = in_params[[2]]
  in_y = in_params[[3]]
  in_stupnice = in_params[[4]]
  in_poc_barev = in_params[[5]]
  in_poc_slov = in_params[[6]]
  in_min_poc_slov = in_params[[7]]

  # převedení stringů na numerické hodnoty
  in_poc_barev = as.numeric(in_poc_barev)
  in_poc_slov = as.numeric(in_poc_slov)
  in_min_poc_slov = as.numeric(in_min_poc_slov)

  shp <- arc.open(in_shp) # otevření shp
  dataset<- arc.select(object = shp, fields = c(in_x,in_y))

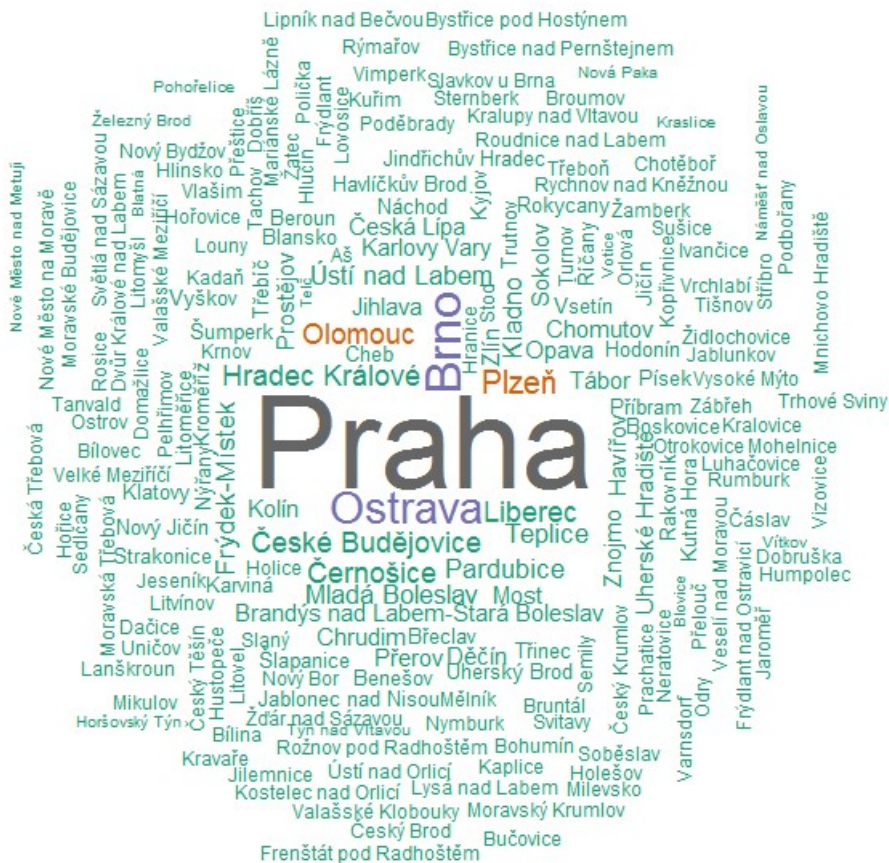
  print("c")
  ycol= as.numeric(sub(",", "", ".", dataset[[in_y]], fixed = TRUE));
  . . . " "

```

Obr. 12: První část R-Skriptu úlohy Wordcloud

```
wordcloud(words = dataset[[in_x]],
          freq = ycol,
          min.freq = in_min_poc_slov,
          max.words=in_poc_slov,
          random.order=FALSE,
          rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
  return(out_params)
}
```

Obr. 13: První část R-Skriptu úlohy Wordcloud



Obr. 14: Výsledek úlohy Wordcloud