Exercise 1: Command line and Git

Matthias Gassilloud, Matr.-Nr.: 3543420

GitHub name: MatthiasGass

Exercise 1

1. GitHub Account created.

2. GitHub repository https://github.com/redfrexx/OpenSourceGIS_exercise01.git

   a) What does “forking” mean in GitHub and why is it useful?
   ”Forking” means creating a copy of an existing repository on your own GitHub account by copying the repository from someone else. The advantage is that you can use the code from someone else and adapt it to your needs without interfering with the original code.

   b) What does “cloning” mean in GitHub?
   Cloning gets a local copy on your device (eg. on your computer) of a remote repository (eg. from GitHub).

Exercise 2

1. Open the command line (“Eingabebaufforderung”). Check in which directory you are located and list all files. Provide the command.

   ```
   # Windows Search
   cmd             # open command line
   cd              # print working directory
   dir             # check in which directory you are located and list all files
   ```

2. Create a folder called “exercise01” and navigate into this folder. Provide the commands.

   ```
   mkdir exercise01        # creates a folder named “exercise01”
   cd exercise01           # navigate into this folder
   ```

3. Clone your repository using the url of your forked repository.

   ```
   git clone "https://github.com/MatthiasGass/OpenSourceGIS_exercise01.git"
   # repository cloned
   ```

4. Now we want to check whether all files were downloaded. So navigate into the top folder of your repository and list all files contained in it.

```
dir                    # check in which directory you are located and list all files
cd OpenSourceGIS_exercise01  # change directory to “OpenSourceGIS_exercise01”
dir                    # check in which directory you are located and list all files
```

   a) How many raster files are displayed?
      1 Raster file is displayed (.tiff)

   b) How many vector files are displayed?
      2 main Vector files are displayed, one is the “india-disputed-lsib.geojson” file and the other is the “san_andres_y_providencia_highway.shp” file with its specific file extensions like .dbf for feature attribute information, .prj for projection information, .shx for indexing the feature geometry.

Exercise 3

Find out what the command *gdalinfo* does using the GDAL documentation (https://www.gdal.org) or further online research. Describe it in the documentation.

1. Find out what the command *gdalinfo* does using the GDAL documentation (https://www.gdal.org) or further online research.

   # The command “gdalinfo” is part of the raster utilities program. The “gdalinfo” program lists various information about a GDAL supported raster dataset, for eg. (if known):
   - The format driver used to access the file.
   - Raster size (in pixels and lines).
   - The coordinate system for the file (in OGC WKT).
   - The geotransform associated with the file (rotational coefficients are currently not reported).
   - Corner coordinates in georeferenced, and if possible lat/long based on the full geotransform (but not GCPs).

2. Open the OSGeo4W shell and print your current working directory. By default you are located in C:\Windows\System32. Navigate to your user directory.

   ```
   m:
   ```
   # changes directory to M:\, **command “cd M:\” did not work!**

3. Therefore, execute the following command in the OSGeo4W shell:

   ```
   gdalinfo relief_san_andres.tif
   ```

   a) What is the coordinate reference system of the raster file?
      The coordinate reference system is “WGS 84”.

b) What is the pixel size? Don’t forget to provide the units!
The Pixel Size is 0.016666666666667 Degree (length of a Pixel border)

4. What happens if you add the flag “-json” to the command and execute it? What is the difference to the output that is generated without it (see previous command)?

   The output now is displayed in .json format. Java Script Object Notation (JSON) is a text-based, lightweight, data interchange format to share GIS data between ArcGIS and other systems.

5. Redirect the output of the last command to a new file called *“relief_gdalinfo.json”.* List all files of the directory to check if the new file was created. Then display the contents of the file in the command line to check whether it is not empty. Provide all commands and along with brief comments in the documentation.

```
gdalinfo relief_san_andres.tif -json > relief_gdalinfo.json
# creates an file called “relief_gdalinfo.json” and stores the output from the command “gdalinfo relief_san_andres.tif” in it.

dir
# check that relief_gdalinfo.json was created

ogrinfo relief_gdalinfo.json
# Lists information about an OGR supported data source, as it is no raster file anymore.
```

Output:

```
INFO: Open of `relief_gdalinfo.json'
     using driver `GeoJSON' successful.
1: wgs84Extent (Polygon)
```

Exercise 4

The git commands are not available in the OSGeo4W shell. Therefore, we need to switch back to the normal command line. If the normal command line is not open anymore, open it again and navigate to the top folder of your repository.

1. Check the status of your repository. Provide the command along with the output in the documentation.

```
git status
# compares the local repository with the github repository and detects changes and new files
```

Output:

```
On branch master
Your branch is up to date with 'origin/master'.
```

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        relief_gdalinfo.json

nothing added to commit but untracked files present (use "git add" to track)
```

2. In the section “Untracked files” of the output you should see the newly created “relief_gdalinfo.json” marked in red. (If not, make sure you are located in the right folder and that it exists.) What does this mean?

   It means that the untracked file is not yet tracked and not in the github repository. You need to add the untracked file to the commit.

3. As a final step, add “relief_gdalinfo.json” to your GitHub repository by staging the file, creating a commit and pushing your changes to the remote repository. Provide all commands along with brief comments in the documentation. Verify that “relief_gdalinfo.json” has been successfully added to the remote repository by opening your repository on github.com using an internet browser.

   ```
   git add relief_gdalinfo.json
   # added the file to be included in the next commit

   git commit -m "created_new_.json_file"
   # created a commit with a description of the changes

   git push
   # uploaded the commit to the remote repository on github

   # Verified that it was successfully added.
   ```

Exercise 5

1. You are given a data set containing the locations of trees in Heidelberg along with information about their species, height and crown diameter. Come up with a geographic question or geospatial service that could be answered/provided using this data set. Explain your topic in one paragraph in your documentation file. If you need additional datasets for your topic, mention them in your description.

   Topic: Spatial analysis of tree species and their development.

   It would be possible to do a spatial analysis of the tree species to identify possible regional patterns. Additional it is possible to see with the parameters “height” and “crown” diameter how well the species developed, maybe also regional patterns will appear.

   Therefore you could use other databases of trees with the parameter “species”, “height” and “crown” to compare and conclude how well the trees are growing and if there are differences from the “average”.

Furthermore you could use soil data to calculate a correlation between soil attributes and the existence of different species and their development.

Of course other parameters could be taken into account like sunlight exposure, which could be calculated by using a digital elevation model of the landscape an buildings. Also you would need the sunlight angle and position during a one year period.

2. You are the proud owner of a new food truck in Heidelberg. For the next "Schlossbeleuchtung" (fireworks at the castle), you want to place you're food truck at a location where you will have a large number of customers. Therefore, the location should fulfill the following requirements:

   a) It needs to have a good view of the castle:
   It would make sense to have a digital elevation model of the buildings and the surface to find a good view of the castle.

   b) It should be located in an area where many potential customers pass by:
   For this task you could use a street layer with an attribute table in which it is mentioned how high the traffic of pedestrians is. If there is no such information you could go by the classification of streets: Big streets for car traffic shouldn't be taken into account, better are streets and places with only limited or no accessed by car. Also tourist attractions could play an important role in pedestrian and tourist traffic. Here, personal knowledge also plays an important role, I would suggest that grassland nearby the river is a better spot to stay and relax for a longer period of time than staying on a street, so many different attributes could be taken into account.

   c) It should be located in an area where there are no or only a few other options to eat:
   You can use a layer of buildings with an attribute table in which it is mentioned if you somehow can buy food (Restaurant, Imbiss, …). You can do a spatial analysis to evaluate zones in within a specific distance there are no or only a few options to eat. Additionally if you attended the firework already you can add your personal experience and knowledge of the positions of other possible foodtrucks. Maybe you even will find further information on "food locations/ trucks" on the event-website.

Bonus Exercise:

Data could be accessed on OpenStreetMap. There you have different data like a building layer with restaurant information, a street/ places layer, some sort of land classification (grassland/ street/ river) and a digital elevation model.