Universität Hamburg	Exercise Sheet 6
Department Informatik	Big Data Analytics
Wissenschaftliches Rechnen	WiSe 2016/2017
Dr. J. Kunkel, H. Hatzel	**Due Date: 2016-12-02**, 420 P

Please document your code, without sufficent documentation you won't recive any points.

1 Exploration of NetCDF Data (R) (180 P)

In this exercise, we will analyze weather data from the NetCDF4 file `/home/bigdata/6/weather-full.nc`. NetCDF is a matrix based (structured) data model and file format that is primarily used for earth-system data. It allows to store multiple N-dimensional variables together with descriptive metadata. Use the methods of descriptive statistics to analyze the overall data.
Document your results. Create a lab notebook PDF containing your results and their analysis.

1.1 Guiding questions

- How does the mean temperature (for each year) in $^{\circ}C$ for Hamburg behave?
- Is there a difference to Tokio?
- Is there a correlation between temperature, precipitation and sun duration?
- How much difference is between regional values (pick any of the variables)? Why could the value be higher in some cases than in others?

1.2 Hints

The dataset contains data for three years.
To access the NetCDF data, we will use the ncdf4 package. Please see `http://geog.uoregon.edu/GeogR/topics/netCDF-read-ncdf4.html` for a description how to read the data.
The most time consuming part probably is the investigation of regional differences. To investigate the difference between regional values, you have do compute deltas between neighboring data points. Manually code an approach similar to the semivariogram from the statistics lecture (Slide 32), i.e., distance in (x,y) and/or time. That means for a given value d(x,y) (of one time step), compute the four deltas: $d(x,y) - d(x,y-1)$, $d(x,y) - d(x,y+1)$, $d(x,y) - d(x-1,y)$, ... Similarly, you can compute the difference across timesteps. If you are interested, you can increase the offset, taking more neighbors into account.

Submission:

`1-analysis.pdf`	A lab notebook with your code, analysis and results.

2 Spatial Data in PostGIS (60 P)

In this task, we want to locate the nearest point of interest, given any location on the planet as a coordinate. Luckily we already have data extracted from the Wikipedia, we will consider each article with coordinates as a place of interest.
Write a python script that, given any coordinate will find the "closest" Wikipedia article in our database. Keep in mind that euclidian distances are not an appropriate approximation for distances between coordinates. Explore options to speed up your queries.

2.1 Hints

The data has already been imported to the database **postgis** on Abu1. To connect use the command
`$ psql postgis`
To inspect the data, use the SQL standard way to inspect the available tables:

```
# Identify available tables:
select table_name from INFORMATION_SCHEMA.TABLES ;

# Inspect a table
select column_name, data_type from INFORMATION_SCHEMA.COLUMNS where table_name = 'TABLE';
```

Submission:

`2-nearest.py`	The script to get the closest known place to any given coordinates.

3 Character Recognition using Machine Learning (R) (180 P)

In this task you will use different machine learning approaches to recognize handwritten characters. We will use the MNIST dataset containing a set of handwritten images that have been preprocessed.
You are provided with four files. Files `/home/bigdata/6/mnist-train-images` and `/home/bigdata/6/mnist-train-labels` hold the training images and their correct labels respectively.
The files `/home/bigdata/6/mnist-t10k-images-idx3-ubyte` and `/home/bigdata/6/mnist-t10k-labels-idx1-ubyte` hold validation data and should not to be used for training.

Perform the following tasks:

- Train a linear model.
- Train a classification tree.
- Consider the problem as regression problem and create a regression tree.
- Evaluate the performance of your models on the training sets. Are there some numbers that are harder to be clasified correctly?
- Evaluate the performance of your models on the validation sets. Is there a way to improve the accuracy?

Discuss how this approach can be scaled in a big data environment.

3.1 Hints

Use the code below to import the images:

```
library(softmaxreg)
x = load_image_file("train-images-idx3-ubyte")
y = load_label_file("train-labels-idx1-ubyte")
image(matrix(x[98,], nrow=28), col=grey(seq(0, 1, length = 256)))
```

Submission:

`3-recognition.pdf`	Your R notebook for creating and evaluating the models. Include the discussion as well.