CODATA-RDA School of Research Data:
Hands-on exercises with the EGI Jupyter Notebooks

INTRODUCTION

Requirements: For this lab session, we will use the following R libraries: gdata() an open-source, library to import data frame from external data files and plot() to plot datasets.

Data Import in R

It is often necessary to import sample textbook data into R before you start working on your homework. We will import data from Excel but there are a number of options for this:

1.) From table text file

A data table can resides in a text file. The cells inside the table are separated by blank characters. Here is an example of a table with 4 rows and 3 columns.

```
100   a1   b1
200   a2   b2
300   a3   b3
400   a4   b4
```

Now copy and paste the table above in a file named "mydata.txt" with a text editor. Then load the data into the workspace with the function read.table.

```
> mydata = read.table("mydata.txt")  # read text file
> mydata                             # print data frame
   V1 V2 V3
1 100 a1 b1
2 200 a2 b2
```

```
3 300 a3 b3
4 400 a4 b4
```

2.) From Excel file

Quite frequently, the sample data is in **Excel** format, and needs to be imported into R prior to use. For this, we can use the function read.xls from the gdata package. It reads from an Excel spreadsheet and returns a data frame. The following shows how to load an Excel spreadsheet named "mydata.xls". This method requires Perl runtime to be present in the system.

```
> install.package(gdata)            # Install the gdata package
> library(gdata)                    # load gdata package
> mydata = read.xls("mydata.xls")   # read from first sheet
```

3.) From CSV File (Comma Separated Values)

The sample data can also be in comma separated values (CSV) format. Each cell inside such data file is separated by a special character, which usually is a comma, although other characters can be used as well. The first row of the data file should contain the column names instead of the actual data. Here is a sample of the expected format.

```
Col1,Col2,Col3
100,a1,b1
200,a2,b2
300,a3,b3
```

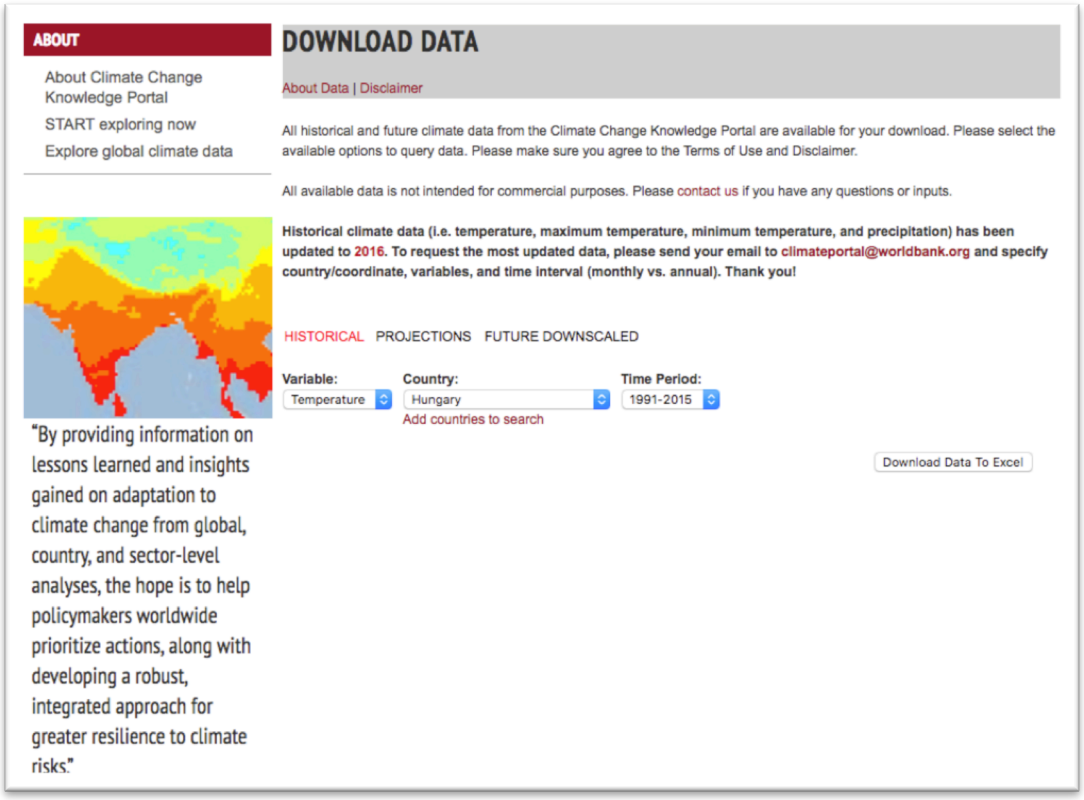
After we copy and paste the data above in a file named "mydata.csv" with a text editor, we can read the data with the function read.csv.

```
> mydata = read.csv("mydata.csv")  # read csv file
> mydata
  Col1 Col2 Col3
1  100   a1   b1
2  200   a2   b2
3  300   a3   b3
```

Exercise 1: Calculate and plot average temperature

Download dataset from the Climate Change Knowledge portal

1. Visit http://sdwebx.worldbank.org/climateportal/index.cfm
2. Click on the area, then country of your interest on the interactive map
3. Click the 'Click to download historical data' link
4. Select 'Temperature', the country and the time period you are interested in (See screenshot below)
5. Click 'Download Data To Excel', and save the file on your computer as *temperatures.xls*



Excel files downloaded from the [Climate Change Knowledge portal] are structured like this:

- tas: average temperature of a given country in a specific year and month

	A	B	C	D	E	F
1	tas	Year	Month	Country	ISO3	ISO2
2	-0,4326	1991	1	HUN		
3	-2,6597	1991	2	HUN		
4	7,37497	1991	3	HUN		
5	8,91267	1991	4	HUN		
6	12,3143	1991	5	HUN		
7	18,5289	1991	6	HUN		
8	21,6088	1991	7	HUN		
9	19,8747	1991	8	HUN		
10	16,9864	1991	9	HUN		
11	9,65744	1991	10	HUN		
12	5,57551	1991	11	HUN		
13	-1,9335	1991	12	HUN		
14	-0,2991	1992	1	HUN		
15	2,40148	1992	2	HUN		
16	5,90079	1992	3	HUN		
17	11,3545	1992	4	HUN		
18	15,7719	1992	5	HUN		
19	19,4253	1992	6	HUN		
20	21,3481	1992	7	HUN		
21	24,0015	1992	8	HUN		
22	16,3708	1992	9	HUN		
23	10,059	1992	10	HUN		
24	5,77228	1992	11	HUN		
25	-0,4692	1992	12	HUN		
26	-0,4635	1993	1	HUN		
27	-1,8311	1993	2	HUN		
28	3,94041	1993	3	HUN		
29	10,6324	1993	4	HUN		

Calculate the Average Monthly Temperature with the R kernel of Jupyter

We will use R code in the EGI Jupyter Notebook service to process and plot the data:

1. Go to the training instance of the EGI Jupyter Notebook service: https://training.fedcloud-tf.fedcloud.eu (Note: the production instance of the service is at a different URL. You'll hear about that later.)
2. Click on 'Login with Check-in' and use your institutional account, or a Social Account for login (e.g ORCID, Google, Facebook, LinkedIn)
   - For further info, please check the **"Instructions for accessing the EGI Jupyter Notebook" document**.
3. Wait for your Jupyter server to boot up
4. Open a new **R** Notebook and save it under a new name (File/Save Notebook as)
5. Upload the XLS file into your Jupyter online folder residing on the left panel (use the same online folder where your notebook file is saved)
6. Use the gdata() library and the read.xls() method to read an Excel file from remote:

```
library(gdata)
raw <- read.xls("temperatures.xls")
```

Note: Click on the play button to run the code segment where your cursor stands.

7. Use the head() method to display the first few rows of the imported dataset:

```
head(raw)
```

then click on the play button to run your code. A similar table should be displayed:

tas	X.Year	X.Month	X.Country	X.ISO3	X.ISO2
-0.43260	1991	1	HUN	NA	NA
-2.65970	1991	2	HUN	NA	NA
7.37497	1991	3	HUN	NA	NA
8.91267	1991	4	HUN	NA	NA
12.31430	1991	5	HUN	NA	NA
18.52890	1991	6	HUN	NA	NA

8. use the aggregate() method to group temperatures per Year, and to calculate the mean average for each year:

```
datasets = aggregate(raw[, 1:2], list(raw$X.Year), mean)
```

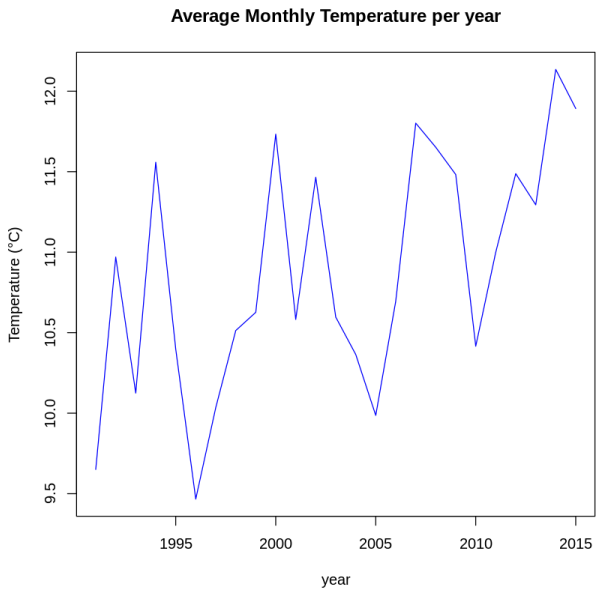
9. Print the average mean temperature per year:

```
print(datasets)
```

```
   Group.1       tas X.Year
1     1991  9.650657   1991
2     1992 10.969779   1992
3     1993 10.124949   1993
4     1994 11.558376   1994
5     1995 10.397354   1995
6     1996  9.466537   1996
7     1997 10.034405   1997
8     1998 10.513293   1998
9     1999 10.626378   1999
10    2000 11.732897   2000
11    2001 10.582294   2001
12    2002 11.465237   2002
13    2003 10.596760   2003
14    2004 10.364570   2004
15    2005  9.986383   2005
16    2006 10.695692   2006
17    2007 11.801325   2007
18    2008 11.651632   2008
19    2009 11.482057   2009
20    2010 10.415981   2010
21    2011 10.999680   2011
22    2012 11.487604   2012
23    2013 11.294179   2013
24    2014 12.134994   2014
25    2015 11.892313   2015
```

10. Drop the duplicate column (X.Year) and plot results:

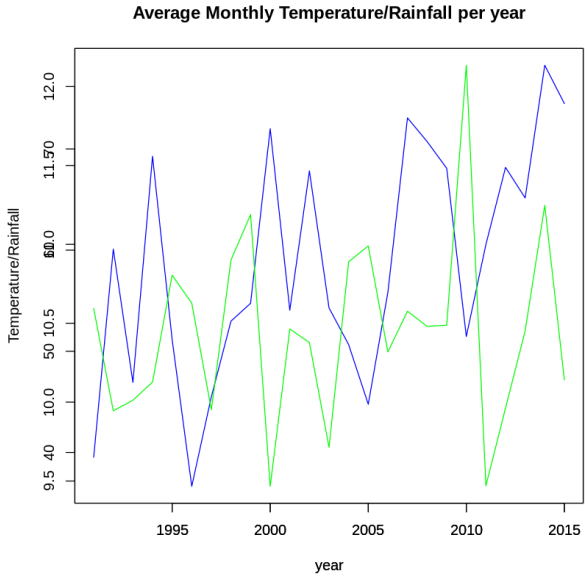
```
plot (datasets[-3], type="l", col="blue", main="Average Monthly Temperature per year", xlab="year", ylab="Temperature (°C)")
```



See the CCKP_AverageMonthlyTemperature.ipynb notebook file as reference in the Samples folder.

Exercise 2: Calculate the average monthly rainfall

1.) Modify the CCKP_AverageMonthlyTemperature.ipynb file to calculate the average monthly rainfall as datasets.
2.) Plot the average monthly temperature and rainfall in the same plot.
Tip: Download rainfall data into another file, then use `par(new=TRUE)` with plot().



See the CCKP_AverageMonthlyTemperatureRainfall.ipynb notebook file as reference in the Samples folder.