**Introduction**

The techniques which are used in our project are a variety of skills and knowledge that we have learnt in data wrangling class. The programming languages which are used include R and Julia. They are not only open sources which are free from Internet, but also have various packages and function which can collect data, clean data, store data, visualize data and analyze data in memory. These are the basic steps in which data scientists and statisticians usually manipulate data. The data processes are showed below:

Step **Question** is the first step to start a new data analysis project. In this case, we want to investigate the endangered animals’ situations and influence factors in each continent all over the world. Step **Data Collection** is the step where we use some techniques to get data from Internet. These techniques include scraping websites and downloading data by using R and Julia languages. Step **Data Processing** includes cleaning and storing data. Next, step **Analyze** is the relationship between endangered animal count and influence factors. The final step is **Presentation**, which needs to visualize our research results to present the audience. This document is involved in **Data Collection** and **Data Processing** phases in which we use to realize our purposes.

**Data Collection**

In short words, the techniques which are used in this step contain R *rvest, purrr, tidyverse,* and *glue* packages. The summary is showed in the table below.

Table 1- Collection Data Info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Order | Website url | Purpose | techniques | Results |
| 1 | https://a-z-animals.com/animals/ | Scrape the animal data | *rvest, purrr, tidyverse, glue* | 593 rows, 61 columns |
| 2 | https://a-z-animals.com/animals/ | Scrape the animal image | *rvest, purrr, tidyverse, glue* | 592 images |
| 3 | <http://earthsendangered.com/continent.asp?gr=&view=&ID=1> | Scrape the mammal animal data | *rvest, purrr, tidyverse, glue*, regular expression | 1329 rows, 10 columns, 11 files |
| 4 | <https://www.ncdc.noaa.gov/cag/global/time-series/hawaiianRegion/land_ocean/ann/12/1880-2019> | Download temperature change data | *NA* | 14 files |
| 5 | <https://datahub.io/core/population#r> | API | *jsonlite* | NA |

The first three data are collected by programming and the last one is collected by downloading csv. The processes of scraping are almost the same. Next, we will show the outline of the processes.

**a-z-animals Data**

We scrape a-z-animals website using R. The main functions we use are ***read\_html, html\_nodes, html\_attr, html\_node, add\_row, html\_text, add\_column*** and ***full\_join***. Firstly, we scrape the html script from <https://a-z-animals.com/animals/> as the main entrance of scraping. There are 593 animals in this page. That is to say there are 593 links. We use ***html\_nodes*** to find all these links html tags, which can be located by searching for html tags with the css class name beginning with **az-phobia-link**. Next, we use 593 times loop to scrape every animal data. In every loop, we combine the basic url and animal name as the new url link. After scraping the new page, we extract the table element with the css class name **az-facts**. We read every row of this table, by finding the **tr** tag in another loop. In each row, there are almost two columns which are controlled by **td** tag. The first column is the attribute and the second is the value of that attribute. We put the attribute as the column name and the value as the cell. We add a new column to a data frame. After this loop, we get a new row, which is added to our final data frame. This step is done by using ***full\_join*** function. Finally, we write this data frame into a file by using ***write\_csv*** function. It is noteworthy that in the outer loop, we pause the program for one second every loop which can relieve the burden of the website and prevent our program to scrape so fast as to be considered as malevolence.

**a-z-animal images**

We scrape animal images from the same website. The main functions we use are ***read\_html, html\_nodes, html\_node, html\_attr, html\_text, add\_row*** and ***download.file***. The packages we use are the same as the previous part and the processes are also almost the same. Firstly, we scrape the outer page which contains 593 animals. Then in every loop, we scrape the inner page and locate the image by using css name **content** and **img** tag. We extract the **src** attribute to get the link of the image. From this link, we can download the image from the Internet to the local, using **download.file** function. We also extract the animal names and store them in a csv file.

**Endangered mammal animals Data**

We scrape endangered mammal animals from <http://earthsendangered.com/continent.asp?gr=&view=&ID=1> website. The main functions used are ***read\_html, html\_nodes, html\_attr, html\_text, add\_row, add\_column*** and ***full\_join***. The processes are almost the same with the previous ones, except the different elements which are used to locate the tags. For example, we get titles by locating a tag with an attribute **data-title** equals to **Common Name**.

**Data Processing**

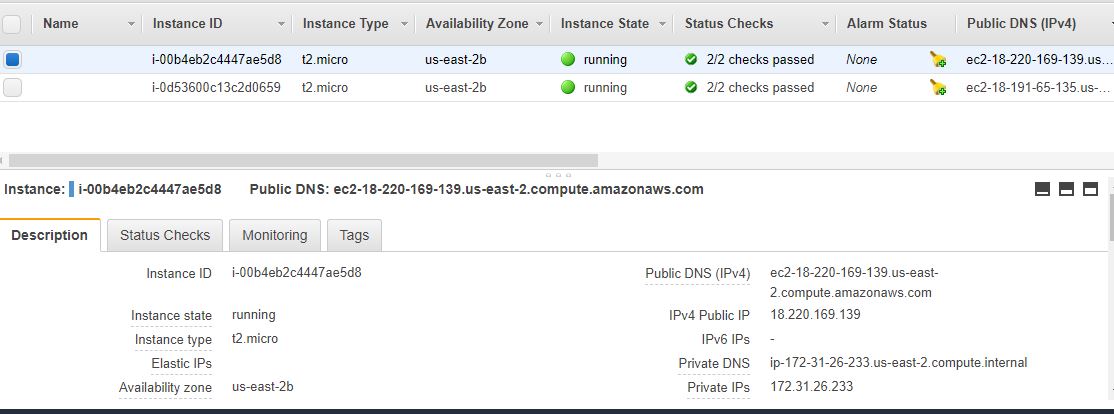
Step **Data Processing** includes cleaning and storing data. Data cleaning is involved in combining 11 endangered mammal animals to one single file, filtering NA data, adding new columns and group rows together. For example, in the (script new-animal-clean.R) file, we create a function (create.Area) which can add a new column named Area. Its value is based on another column “Endangered Animals of Area”. We extract the following part of the sub sentence “Endangered Mammals of ” and then use regular expression [^a-zA-Z0-9 -] to remove all non-numeric and non-alphabet characters.

**Amazon EC2**

Amazon EC2 is the short name of Amazon Elastic Compute Cloud. It is a web service that provide secure, resizable computer capacity in the cloud. It is designed as a cloud-computing platform by allowing users to rent virtual computers on which to run their own computer applications. It is developed by Amazon.com. It supports many operating system, like Linux, Microsoft Windows and FreeBSD. We use Amazon Linux system and run our R programs on Amazon EC2.

* Launch a new Amazon new instance

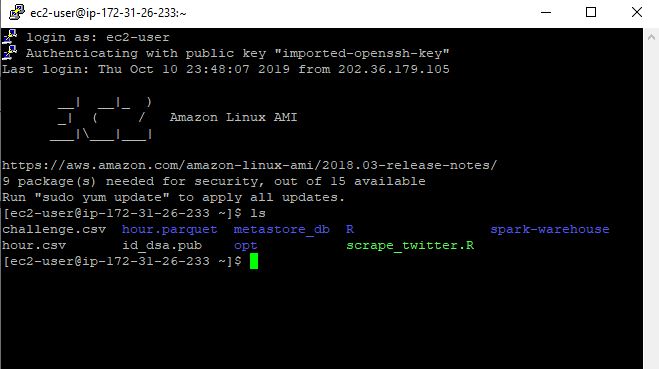
How to launch a new Amazon instance, please refer to <https://docs.aws.amazon.com/quickstarts/latest/vmlaunch/step-1-launch-instance.html>. After launching a new instance, the result is showed below:

We use **Public DNS (IPv4**) = **ec2-18-220-169-139.us-east-2.compute.amazonaws.com**

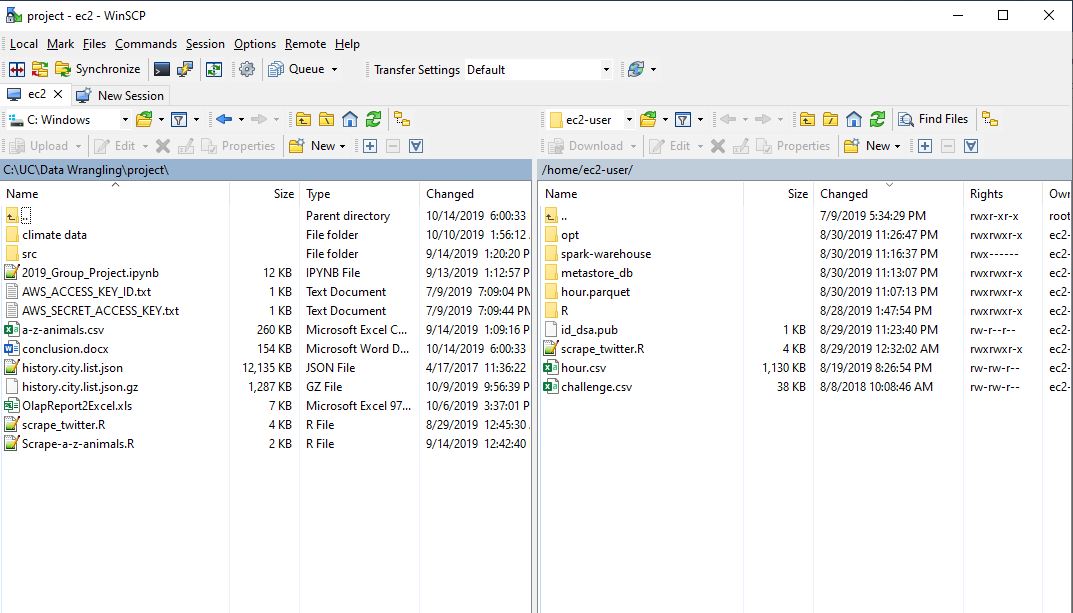
The instance State is running.

* Connect this instance by putty and WinSCP from local machine

Please refer to <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>. The putty login UI is showed below:



The WinSCP UI is showed below:



* Install R on Linux

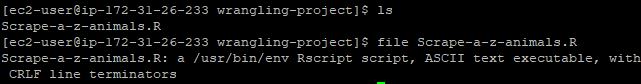
Refer to <https://linuxize.com/post/how-to-install-r-on-centos-7/> to install R on our EC2 virtual machine.

* Upload R script to the cloud and modify the format

Make a new directory wrangling-project and enter this directory.

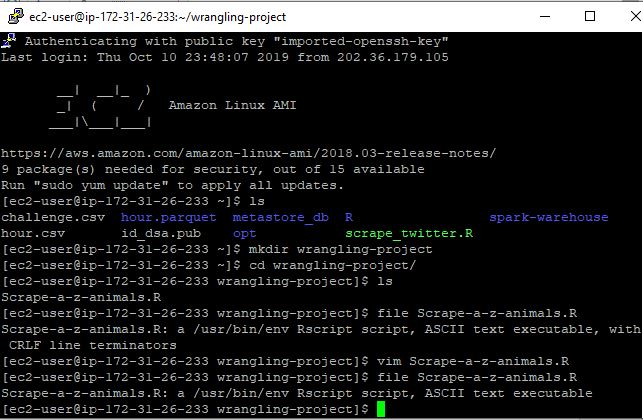


Use WinSCP to upload our Scrape-a-z-animals.R



Because this file is created by Windows, each line contains “with CRLF line terminators” which means it cannot run on the Linux system directly. Hence, convert the file format to unix in the vim editor.

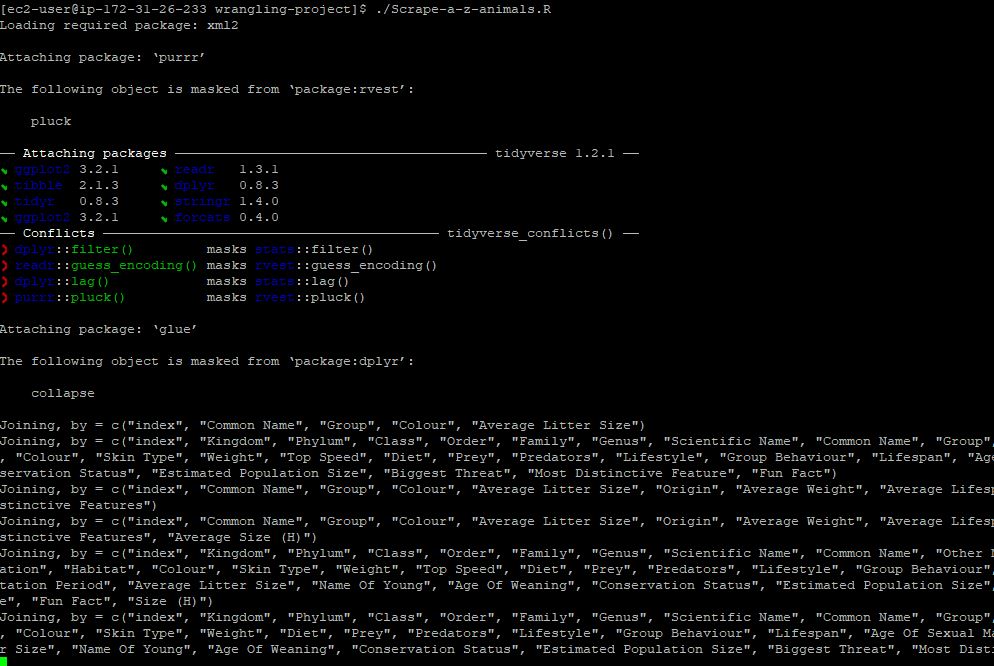




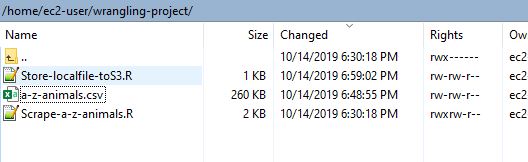
Change the file to an executable file



* Run R program on the cloud



The result is the same as we run on the local. The generated csv file is stored on the EC2 virtual machine in the current directory.



Furthermore, we can change the store location to S3.

**Amazon S3**

Amazon S3 is a service offered by Amazon Web Services that provides object storage through a web service interface. Its type is cloud storage developed by Amazon.com.

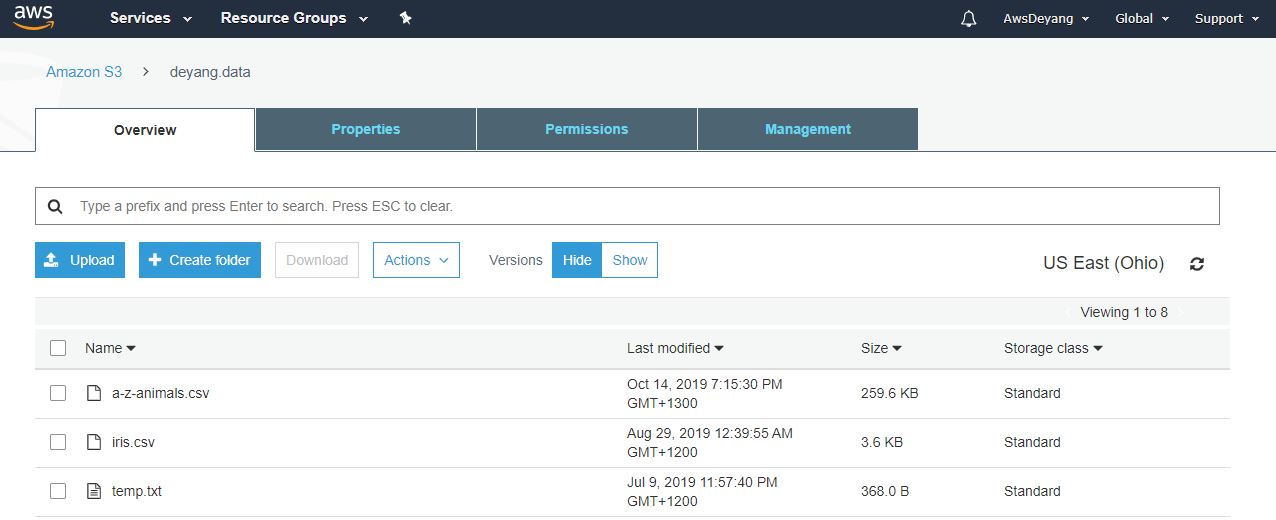
* Start Amazon S3

How to start a new Amazon S3, please refer to <https://aws.amazon.com/s3/getting-started/>

* Write file to S3



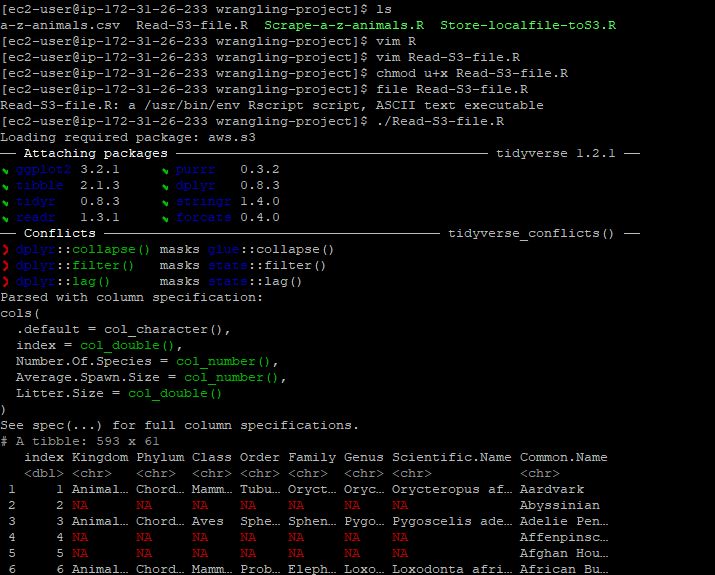




It is clear to see that the a-z-animals.csv file is stored in Amazon S3.

* Read file from S3

Do the same to the “Read-S3-file.R” file, making it an executable file in the Linux. The running result is shown below.



**Conclusion:**

The techniques which we use are R language, Julia language, Github, Amazon EC2 and S3. We store our output on S3, so that we can read the file on our local machines or on EC2 applications. The pity is that we don’t use Amazon RDS like mysql to store our output data. However, we use the knowledge and skills that we have learnt from the wrangling class plus in our spare time by ourseleves.