Big Data Management
Assignment 1

Description

In this assignment your task is to prepare the batch layer (off-line processing pipeline) of the lambda architecture that will enable us to perform some analytics on a dataset. You are required to use Apache Spark to compute some simple analytics. You will be using the CRAN package download logs (`http://cran-logs.rstudio.com`). These log files contain all hits to `http://cran.rstudio.com` mirror related to downloads of the R packages. The raw log files have been parsed into CSV and anonymised. Since these logs contain massive amount of data (from 2012 to date), we will only be using a recent one which represent the logs for 31st of October, 2021. This log file are available at:

`http://cran-logs.rstudio.com/2021/2021-10-31`

The package download logs contain data about the following variables:

```
date: Download date
time: Download time (in UTC)
size: Package size (in bytes)
r_version: Version of R used to download package
r_arch: Processor architecture (i386 = 32 bit, x86_64 = 64 bit)
r_os: Operating System (darwin9.8.0 = mac, mingw32 = windows)
package: Name of the package downloaded
country: Two letter ISO country code
ip_id: A daily unique id assigned to each IP address
```

Setup

In order to complete this work and the future assignments you need to have a working VM. The VM must have the Hadoop installed in Pseudo-Distributed Operation. It must have the Apache Spark setup and Jupyter Notebook linked with PySpark. If your VM has everything ready - go to the next section. Otherwise follow the guide provided on Moodle on how to setup Ubuntu VM, Hadoop, and Apache Spark.

Load Data

The method we discussed to load data is `sc.textFile`. This method takes a URI of a file to load it. It then reads the data as a collections of lines. Download the data mentioned above (the CRAN log files) on your VM and create an RDD of the data by executing, for example (remember to change the path according to your VM):

```
downloads_RDD = sc.textFile("file:///home/bdm/assignment1_data/*.gz")

# You can also store the log files on your HDFS cluster and load them using:
# downloads_RDD = sc.textFile("hdfs://localhost:9000/input/*.gz")
# If using this option make sure you have HDFS and YARN services running.
```

- You can check the type of `downloads_RDD`:

  ```
  type(downloads_RDD)
  # output: pyspark.rdd.RDD
  ```

- Count the number of lines/records in `downloads_RDD`:

  ```
  downloads_RDD.count()
  # output: 4297202
  ```

- Take a look at the first 5 lines in `downloads_RDD`:

  ```
  downloads_RDD.take(5)
  # output: 5 lines from the downloads_RDD
  ```

- As each line has comma-separated values, you can split it by comma:

  ```
  downloads_RDD = downloads_RDD.map(lambda x: x.split(','))
  ```

- Take a look at the first 5 lines again in `downloads_RDD`:

  ```
  downloads_RDD.take(5)
  # output: 5 lines from the downloads_RDD
  ```

- You may have noticed the double quotation marks in the transformed `downloads_RDD` above, you can remove them:

  ```
  def remove_quotation(x):
          return([xx.replace('"', '') for xx in x])

  downloads_RDD = downloads_RDD.map(remove_quotation)
  ```

- Take a look at the first 5 lines again in downloads_RDD:

```
downloads_RDD.take(5)
# output: 5 lines from the downloads_RDD
```

- Now the downloads_RDD is all set to perform some basic analytics. For example, let's say you want to know the total number of downloads for each package.

```
package_download_count = downloads_RDD.map(lambda x: (x[6], 1))
# Note: x[6] is the 7th element of each row, which is the package name.

package_download_count = package_download_count.reduceByKey(lambda a,b: a+b)
# Add number of downloads for each package name.
```

- Take a look at first 20 packages and their total downloads:

```
package_download_count.take(10)
# output:
[('CNull', 13),
 ('covafillr', 14),
 ('perccalc', 8),
 ('liso', 3),
 ('logbin', 11),
 ('linkR', 13),
 ('nlmeODE', 18),
 ('HDcpDetect', 7),
 ('MPDiR', 18),
 ('mudata2', 15)]
```

- To see a list of top 10 downloaded packages:

```
package_download_count.sortBy(lambda a: a[1], ascending=False).take(10)
# output:
[('rlang', 55592),
 ('Rcpp', 50448),
 ('tibble', 45020),
 ('pillar', 40948),
 ('dplyr', 39443),
 ('stringr', 39439),
 ('R6', 39063),
 ('ggplot2', 38729),
 ('yaml', 38422),
 ('fansi', 37598)]
```

Questions

Your task is to prepare code that could be run in the batch layer of a system, and will help performing the following analyses:

1. Show number of downloads for package `ggplot2`.
2. List the highest number of downloads by a country.
3. Show top 10 largest sized packages.
4. What were the top 10 most popular packages?
5. What OS is used for downloading the most popular package?
6. What is the most popular package in Ireland?
7. What is the highest number of downloads by a single machine? What OS it has?
8. What OS is most popular among the R programmers?
9. How many R users still use 32 bit machines?
10. List total number of incomplete records - lines which have missing values.

Submission

- Create a Python notebook - name it `assignment1.ipynb`. The notebook should be exported as iPython Notebook with *.ipynb extension. Make sure you have run the entire code and then exported the notebook. If the code in your notebook does not run, it will result in 20% penalty.
- Take one screenshot of your solution to each question (show code + output) and put it in document, generate a pdf of this document.
- Zip both files together and submit your solution on Moodle by the deadline.
- Do not submit work thats not your own and do not let others copy work that is your own. Both Copyier and Copyee will get ZERO marks.