Part 1: Apache Hadoop. PIG. HIVE and SQL

Importing Data and Uploading to HDFS

- Import RStudio CRAN Log Files of three weekdays (October 2020) into HDFS.

Importing 3 files from http://cran-logs.rstudio.com/. Decided for 5th, 6th and 7th days of October and uploaded the files to HDFs location /user/htw/LogAnalysis.

Unpacking .gz files to local folder

```
htw@master:~/Downloads$ gzip -d 2020-10-05.csv.gz
htw@master:~/Downloads$ gzip -d 2020-10-06.csv.gz
htw@master:~/Downloads$ gzip -d 2020-10-07.csv.gz
```

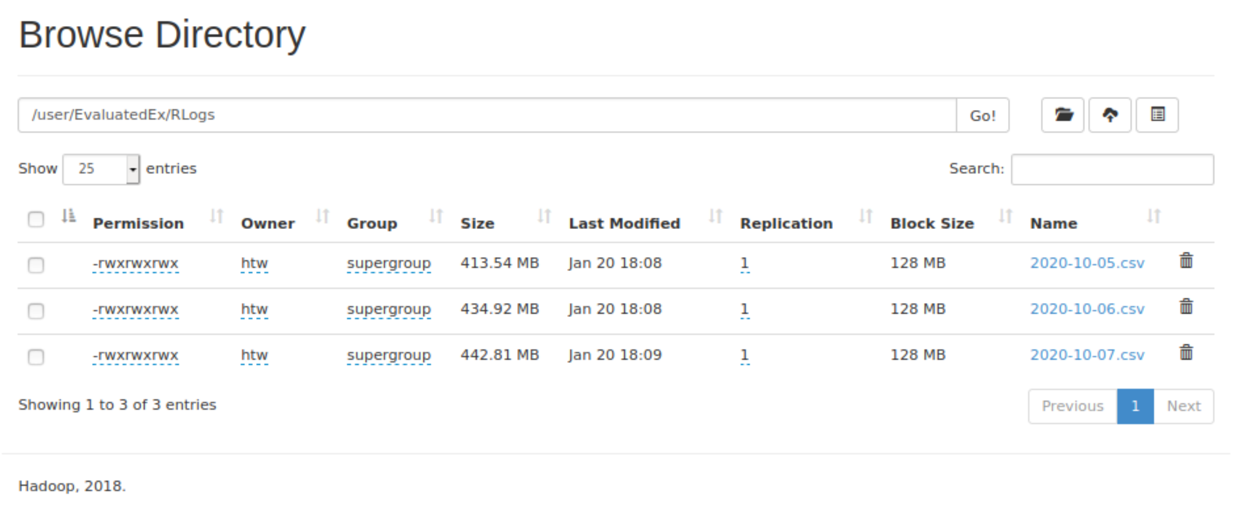
Uploading files to HDFs folder

```
htw@master:~/Downloads$ hdfs dfs -put 2020-10-07.csv /user/EvaluatedEX/RLogs/
htw@master:~/Downloads$ hdfs dfs -put 2020-10-06.csv /user/EvaluatedEX/RLogs/
htw@master:~/Downloads$ hdfs dfs -put 2020-10-05.csv /user/EvaluatedEX/RLogs/
```

Changing writing permissions

```
hdfs dfs -chmod -R 777 /user/EvaluatedEX/RLogs
```

All set in HDFS:



Browse Directory

/user/EvaluatedEx/RLogs

Show 25 entries

Search:

☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	-rwxrwxrwx	htw	supergroup	413.54 MB	Jan 20 18:08	1	128 MB	2020-10-05.csv
☐	-rwxrwxrwx	htw	supergroup	434.92 MB	Jan 20 18:08	1	128 MB	2020-10-06.csv
☐	-rwxrwxrwx	htw	supergroup	442.81 MB	Jan 20 18:09	1	128 MB	2020-10-07.csv

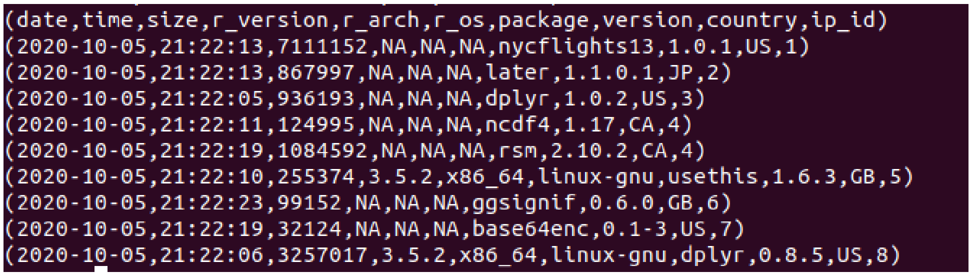
Showing 1 to 3 of 3 entries

Hadoop, 2018.

To get a first overview, which packages are downloaded at all and which operating systems are currently used, run a first analysis using Apache Pig and/or Hadoop wordcount.

Loading the three files by LOAD (specifying the folder in HDFs will read the files)

```
grunt>
A = LOAD '/user/EvaluatedEX/RLogs USING org.apache.pig.piggybank.storage.CSVExcelStorage(',')
AS (date:chararray, time:chararray, size:chararray, r_version:chararray, r_arch:chararray,
r_os:chararray, package:chararray, version:chararray, country:chararray, ip_id:chararray);

head = LIMIT A 10;

DUMP head;
```



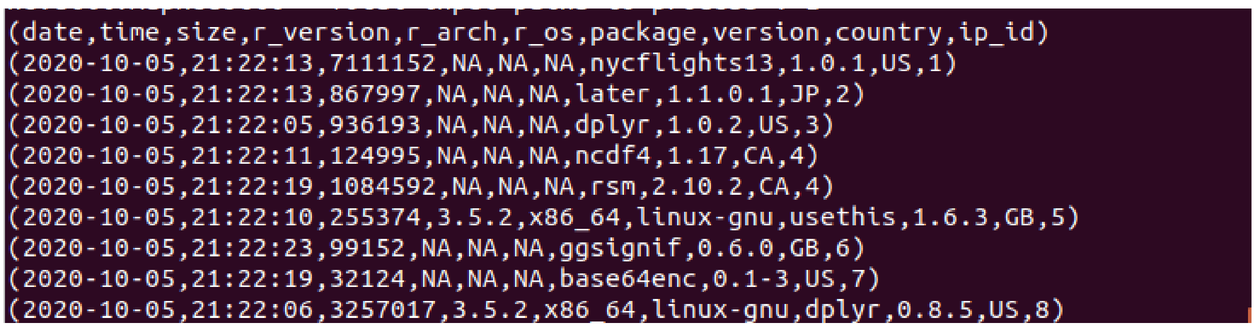
```
(date,time,size,r_version,r_arch,r_os,package,version,country,ip_id)
(2020-10-05,21:22:13,7111152,NA,NA,NA,nycflights13,1.0.1,US,1)
(2020-10-05,21:22:13,867997,NA,NA,NA,later,1.1.0.1,JP,2)
(2020-10-05,21:22:05,936193,NA,NA,NA,dplyr,1.0.2,US,3)
(2020-10-05,21:22:11,124995,NA,NA,NA,ncdf4,1.17,CA,4)
(2020-10-05,21:22:19,1084592,NA,NA,NA,rsm,2.10.2,CA,4)
(2020-10-05,21:22:10,255374,3.5.2,x86_64,linux-gnu,usethis,1.6.3,GB,5)
(2020-10-05,21:22:23,99152,NA,NA,NA,ggsignif,0.6.0,GB,6)
(2020-10-05,21:22:19,32124,NA,NA,NA,base64enc,0.1-3,US,7)
(2020-10-05,21:22:06,3257017,3.5.2,x86_64,linux-gnu,dplyr,0.8.5,US,8)
```

Top 25 packages by operating system

- Load log-files into Apache Pig, set variable names (please check documentation);

```
grunt>
----------
A = LOAD '/user/EvaluatedEX/RLogs USING org.apache.pig.piggybank.storage.CSVExcelStorage(',')
AS (date:chararray, time:chararray, size:chararray, r_version:chararray, r_arch:chararray,
r_os:chararray, package:chararray, version:chararray, country:chararray, ip_id:chararray);

HEAD = LIMIT A 10;

DUMP HEAD;
```

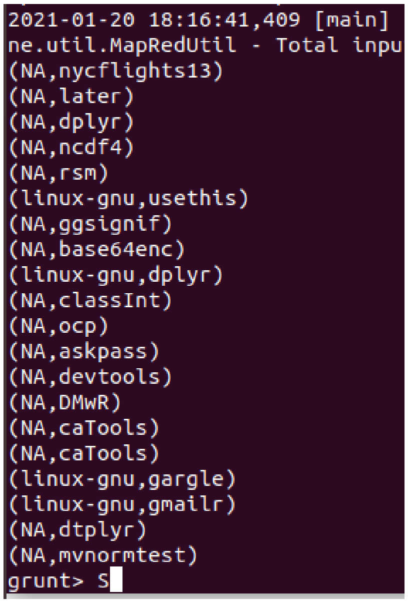
- Dump the first 10 entries on screen (*attach a screen shot into your report*) to check if it works or not;



```
(date,time,size,r_version,r_arch,r_os,package,version,country,ip_id)
(2020-10-05,21:22:13,7111152,NA,NA,NA,nycflights13,1.0.1,US,1)
(2020-10-05,21:22:13,867997,NA,NA,NA,later,1.1.0.1,JP,2)
(2020-10-05,21:22:05,936193,NA,NA,NA,dplyr,1.0.2,US,3)
(2020-10-05,21:22:11,124995,NA,NA,NA,ncdf4,1.17,CA,4)
(2020-10-05,21:22:19,1084592,NA,NA,NA,rsm,2.10.2,CA,4)
(2020-10-05,21:22:10,255374,3.5.2,x86_64,linux-gnu,usethis,1.6.3,GB,5)
(2020-10-05,21:22:23,99152,NA,NA,NA,ggsignif,0.6.0,GB,6)
(2020-10-05,21:22:19,32124,NA,NA,NA,base64enc,0.1-3,US,7)
(2020-10-05,21:22:06,3257017,3.5.2,x86_64,linux-gnu,dplyr,0.8.5,US,8)
```

****** For easiness and computation speed, I will be reducing the full data into only **R_OS** and **PACKAGE** columns and make the operations over this data set:

```
grunt>
----------
A = LOAD '/user/EvaluatedEX/RLogs USING org.apache.pig.piggybank.storage.CSVExcelStorage(',')
AS (date:chararray, time:chararray, size:chararray, r_version:chararray, r_arch:chararray,
r_os:chararray, package:chararray, version:chararray, country:chararray, ip_id:chararray);

REDUCED = FOR EACH A GENERATE $5 AS r_os, $6 AS pckg;

REDHEAD = LIMIT REDHEAD 10;

DUMP REDHEAD;
```



```
2021-01-20 18:16:41,409 [main]
ne.util.MapRedUtil - Total inpu
(NA,nycflights13)
(NA,later)
(NA,dplyr)
(NA,ncdf4)
(NA,rsm)
(linux-gnu,usethis)
(NA,ggsignif)
(NA,base64enc)
(linux-gnu,dplyr)
(NA,classInt)
(NA,ocp)
(NA,askpass)
(NA,devtools)
(NA,DMwR)
(NA,caTools)
(NA,caTools)
(linux-gnu,gargle)
(linux-gnu,gmailr)
(NA,dtplyr)
(NA,mvnormtest)
grunt> S
```

- Use Pig to get rid of the quotation marks!

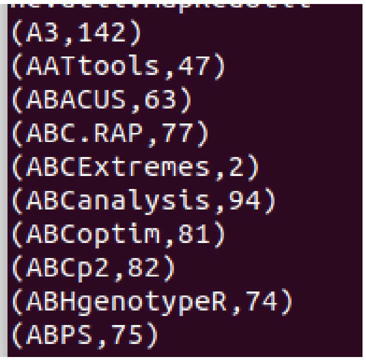
The loading phase gets rid automatically of the double quotes with:

```
USING org.apache.pig.piggybank.storage.CSVExcelStorage(',')
```

- Count the number of occurrences of different packages; Use either a Apache Pig script or store the modified data into HDFS and use Hadoop wordcount.

Each block of the following code represents a standalone instruction. Information can be loaded from hdfs and reduced only once:

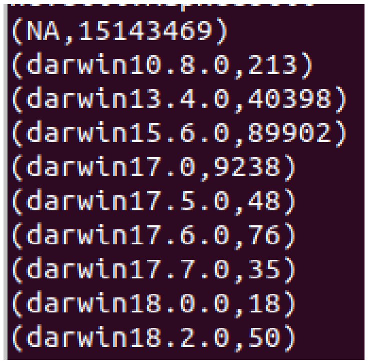
```
-       grunt>
----------
A = LOAD '/user/EvaluatedEx/Rlogs' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',')
AS (date:chararray, time:chararray, size:chararray, r_version:chararray, r_arch:chararray,
r_os:chararray, package:chararray, version:chararray, country:chararray, ip_id:chararray);

REDUCED = FOR EACH A GENERATE $5 AS r_os, $6 AS pckg;

GROUPED_PCK = GROUP REDUCED BY $1;

GROUPED_PCK_COUNT = FOREACH GROUPED_PCK GENERATE group, COUNT($1) AS cnt;
GROUPED_PCK_COUNT = ORDER GROUPED_PCK_COUNT BY $0 ASC;

STORE GROUPED_PCK_COUNT INTO ‘/user/EvaluatedEx/PCK_Count/’ using PigStorage(‘,’);
```



```
(A3,142)
(AATtools,47)
(ABACUS,63)
(ABC.RAP,77)
(ABCExtremes,2)
(ABCanalysis,94)
(ABCoptim,81)
(ABCp2,82)
(ABHgenotypeR,74)
(ABPS,75)
```

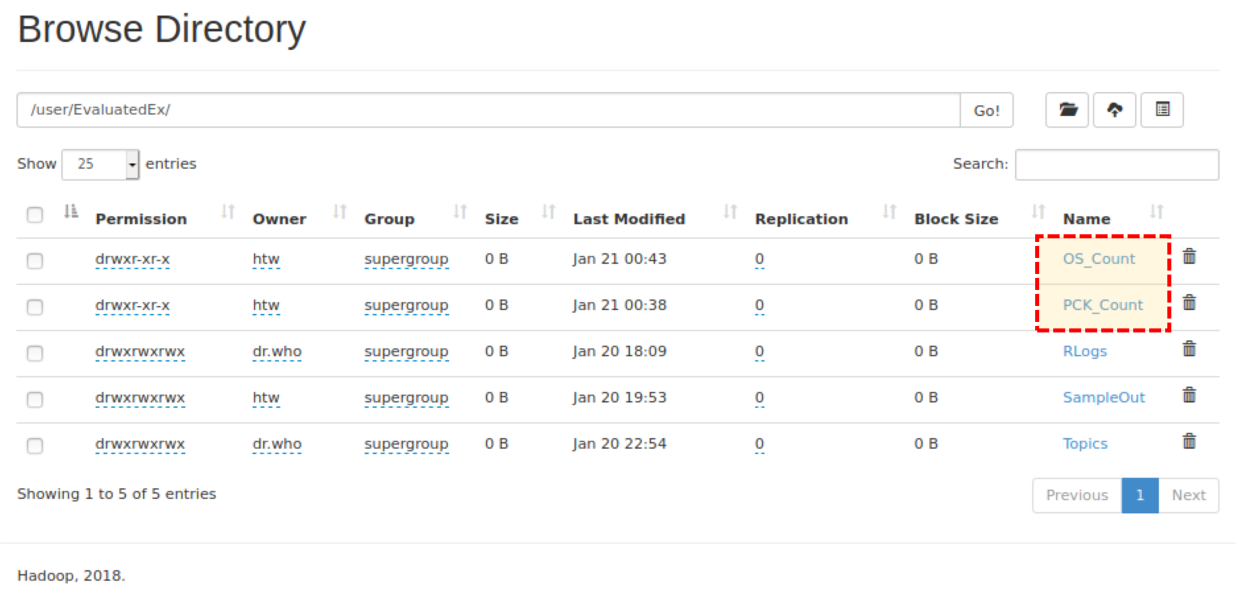
- Count the number of occurrences of different packages by operating system;

```
-       grunt>
----------
A = LOAD '/user/EvaluatedEx/Rlogs' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',')
AS (date:chararray, time:chararray, size:chararray, r_version:chararray, r_arch:chararray,
r_os:chararray, package:chararray, version:chararray, country:chararray, ip_id:chararray);

REDUCED = FOR EACH A GENERATE $5 AS r_os, $6 AS pckg;

GOUPED_OS = GROUP REDUCED BY $0;

GROUPED_OS_COUNT = FOREACH GROUPED_OS GENERATE group, COUNT($1) AS cnt;
GROUPED_OS_COUNT = ORDER GROUPED_OS_COUNT BY $0 ASC;

STORE GROUPED_OS_COUNT INTO ‘/user/EvaluatedEx/OS_Count/’ using PigStorage(‘,’);
```



```
(NA,15143469)
(darwin10.8.0,213)
(darwin13.4.0,40398)
(darwin15.6.0,89902)
(darwin17.0,9238)
(darwin17.5.0,48)
(darwin17.6.0,76)
(darwin17.7.0,35)
(darwin18.0.0,18)
(darwin18.2.0,50)
```

- Store the results of both operations in HDFS;



Browse Directory

☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
☐	drwxr-xr-x	htw	supergroup	0 B	Jan 21 00:43	0	0 B	OS_Count	🗑
☐	drwxr-xr-x	htw	supergroup	0 B	Jan 21 00:38	0	0 B	PCK_Count	🗑
☐	drwxrwxrwx	dr.who	supergroup	0 B	Jan 20 18:09	0	0 B	RLogs	🗑
☐	drwxrwxrwx	htw	supergroup	0 B	Jan 20 19:53	0	0 B	SampleOut	🗑
☐	drwxrwxrwx	dr.who	supergroup	0 B	Jan 20 22:54	0	0 B	Topics	🗑

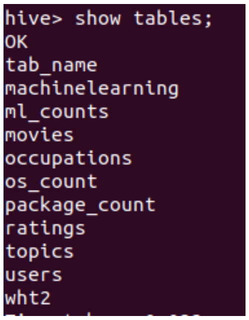
Showing 1 to 5 of 5 entries

Hadoop, 2018.

HIVE

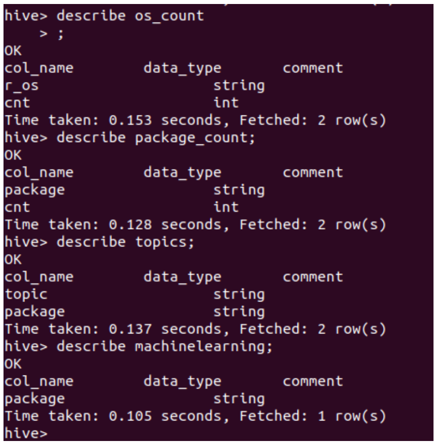
- Import all relevant tables into HIVE (register the tables and import the data)

```
-       hive>
----------
CREATE TABLE os_count (r_os STRING, cnt INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

CREATE TABLE package_count (package STRING, cnt INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY
',' STORED AS TEXTFILE;

CREATE TABLE topics (topic STRING, package STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY
',' STORED AS TEXTFILE;

CREATE TABLE machinelearning (package STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```



```
hive> show tables;
OK
tab_name
machinelearning
ml_counts
movies
occupations
os_count
package_count
ratings
topics
users
wht2
```

```
-       hive>
----------
LOAD DATA INPATH ‘/user/EvaluatedEx/OS_count’ OVERWRITE INTO TABLE os_count;

LOAD DATA INPATH ‘/user/EvaluatedEx/PCK_count’ OVERWRITE INTO TABLE package_count;

LOAD DATA INPATH ‘/user/EvaluatedEx/Topics/threetopics’ OVERWRITE INTO TABLE topics;

LOAD DATA INPATH ‘/user/EvaluatedEx/Topics/machinelearning’ OVERWRITE INTO TABLE
machinelearning;
```

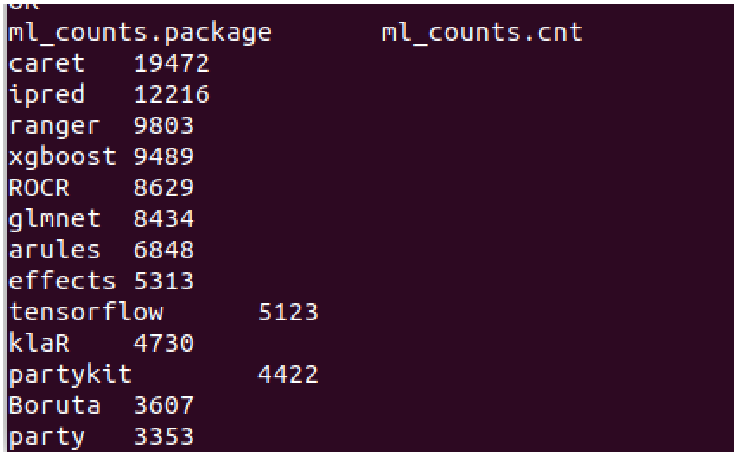


```
hive> describe os_count
    > ;
OK
col_name        data_type       comment
r_os                    string
cnt                     int
Time taken: 0.153 seconds, Fetched: 2 row(s)
hive> describe package_count;
OK
col_name        data_type       comment
package                 string
cnt                     int
Time taken: 0.128 seconds, Fetched: 2 row(s)
hive> describe topics;
OK
col_name        data_type       comment
topic                   string
package                 string
Time taken: 0.137 seconds, Fetched: 2 row(s)
hive> describe machinelearning;
OK
col_name        data_type       comment
package                 string
Time taken: 0.105 seconds, Fetched: 1 row(s)
hive>
```

- Count download figures of packages which belong to Task View Machine Learning

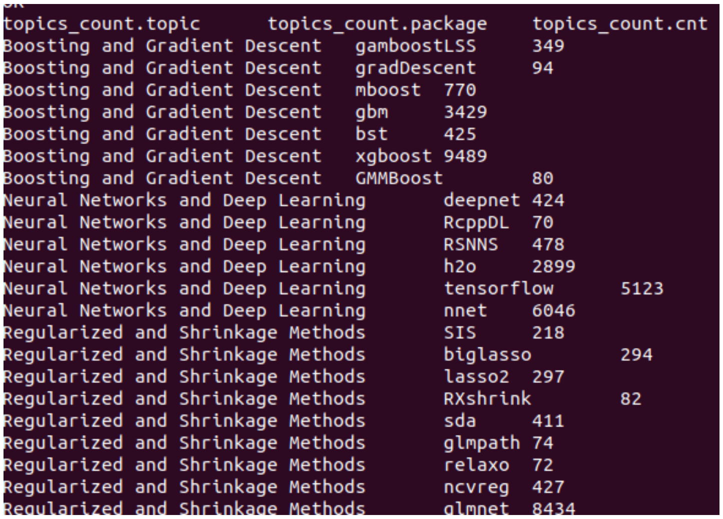
For this HIVE statements, tables have not been short labeled and have been written completely for reading easiness.

```
hive>
----------
CREATE TABLE ml_counts AS
SELECT machinelearning.package, package_count.cnt
FROM machinelearning LEFT JOIN package_count
ON machinelearning.package = package_count.package
ORDER BY package_count.cnt DESC;
```



```
ml_counts.package       ml_counts.cnt
caret   19472
ipred   12216
ranger  9803
xgboost 9489
ROCR    8629
glmnet  8434
arules  6848
effects 5313
tensorflow      5123
klaR    4730
partykit        4422
Boruta  3607
party   3353
```

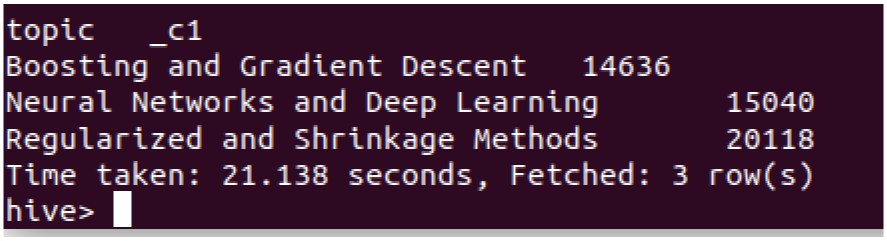
```
hive>
----------
CREATE TABLE topics_count AS SELECT t.topic, t.package, c.cnt FROM topics t LEFT JOIN
package_count c ON t.package = c.package;
```



```
topics_count.topic      topics_count.package    topics_count.cnt
Boosting and Gradient Descent   gamboostLSS     349
Boosting and Gradient Descent   gradDescent     94
Boosting and Gradient Descent   mboost  770
Boosting and Gradient Descent   gbm     3429
Boosting and Gradient Descent   bst     425
Boosting and Gradient Descent   xgboost 9489
Boosting and Gradient Descent   GMMBoost        80
Neural Networks and Deep Learning       deepnet 424
Neural Networks and Deep Learning       RcppDL  70
Neural Networks and Deep Learning       RSNNS   478
Neural Networks and Deep Learning       h2o     2899
Neural Networks and Deep Learning       tensorflow      5123
Neural Networks and Deep Learning       nnet    6046
Regularized and Shrinkage Methods       SIS     218
Regularized and Shrinkage Methods       biglasso        294
Regularized and Shrinkage Methods       lasso2  297
Regularized and Shrinkage Methods       RXshrink        82
Regularized and Shrinkage Methods       sda     411
Regularized and Shrinkage Methods       glmpath 74
Regularized and Shrinkage Methods       relaxo  72
Regularized and Shrinkage Methods       ncvreg  427
Regularized and Shrinkage Methods       glmnet  8434
```

Whith this table we can get a summary by topic (Boosting and Gradient Descent, Neural Networks and Deep Learning and Regularized and Shrinkage Methods) each one with its total sum:

```
hive>
----------
SELECT topic, SUM(cnt) as total_count FROM topics_count GROUP BY topc;
```



```
topic   _c1
Boosting and Gradient Descent   14636
Neural Networks and Deep Learning       15040
Regularized and Shrinkage Methods       20118
Time taken: 21.138 seconds, Fetched: 3 row(s)
hive>
```

```
hive>
----------

INSERT OVERWRITE ELOCAL DIRECTORY ‘Desktop/ml_counts’ ROW FORMAT DELIMITED FIELDS TERMINATED
BY ‘,’ SELECT * FROM ml_counts;

INSERT OVERWRITE ELOCAL DIRECTORY ‘Desktop/topic_counts’ ROW FORMAT DELIMITED FIELDS
TERMINATED BY ‘,’ SELECT * FROM topic_counts;
```

Management Report

- Top package Downloads from CRAN.

After the data wrangling, R studio is used for deeper analysis.

Packages:

There were a total of 18,962,446 of package downloads. These downloads come from a total of 19,526 different packages available to the public.

The top downloaded package was magrittr with 352,838 and jsonlite with 258,284 downloads respectively. 3rd place belongs to aws.s3 for its popularity rise in the latest years with 254,284.

The following graph explores the top 25 packages by download amounts.

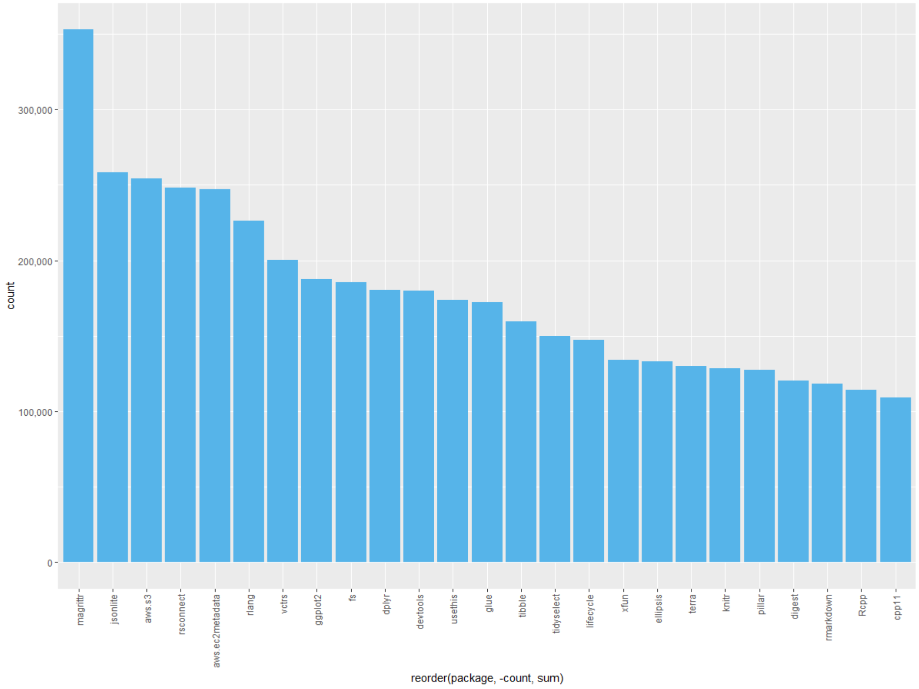


Figure 1 Package Download Count

Operating System:

In contrast, the operating system race is not as similar as the packages uniformity. We encounter an outlier of non-defined operating system which takes over the count with 15,143,469, and skews the analysis.

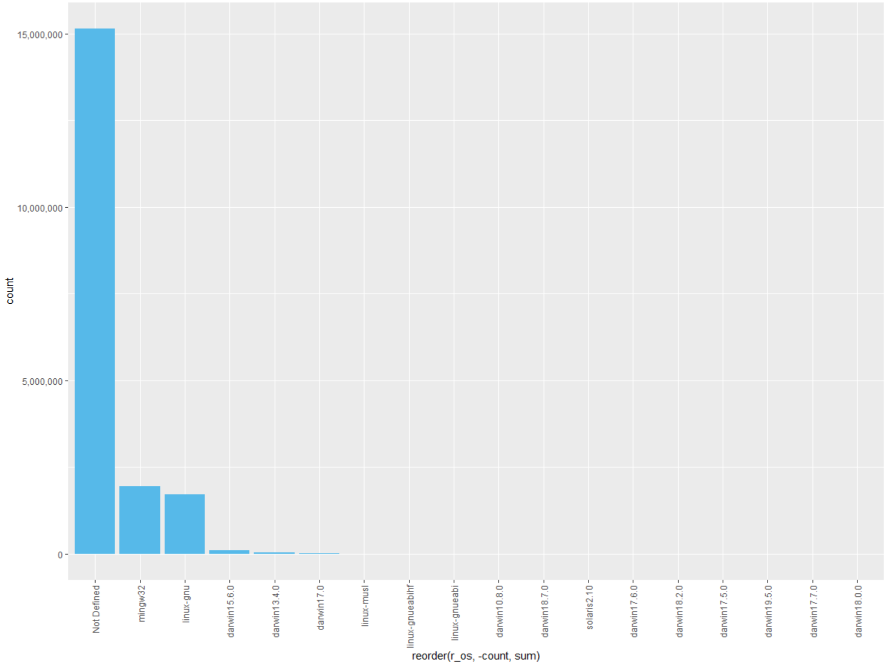


Figure 2 Downloads per Operating System

Even after the removal of the non-defined, we can see a high lead by mingw32 with 1,956,919 and linux-gnu with 1,714,385:

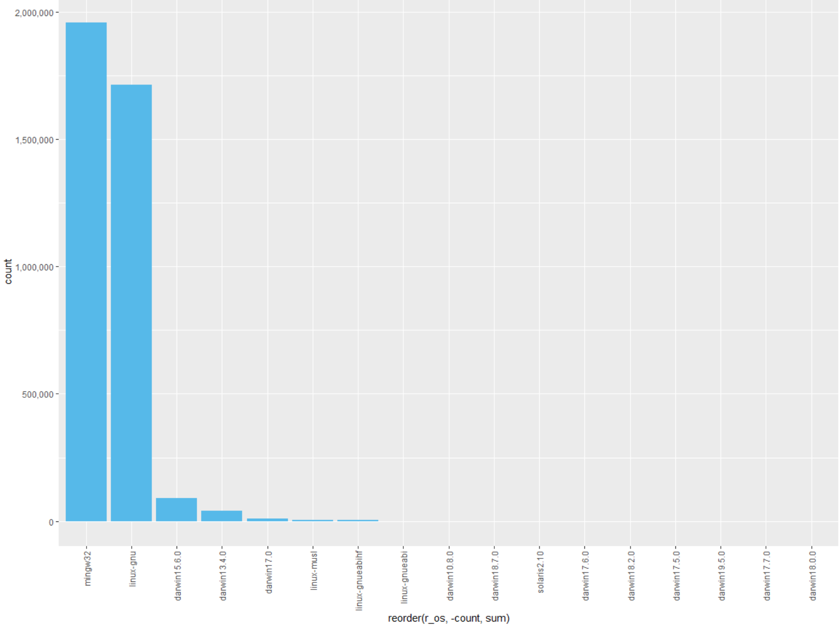
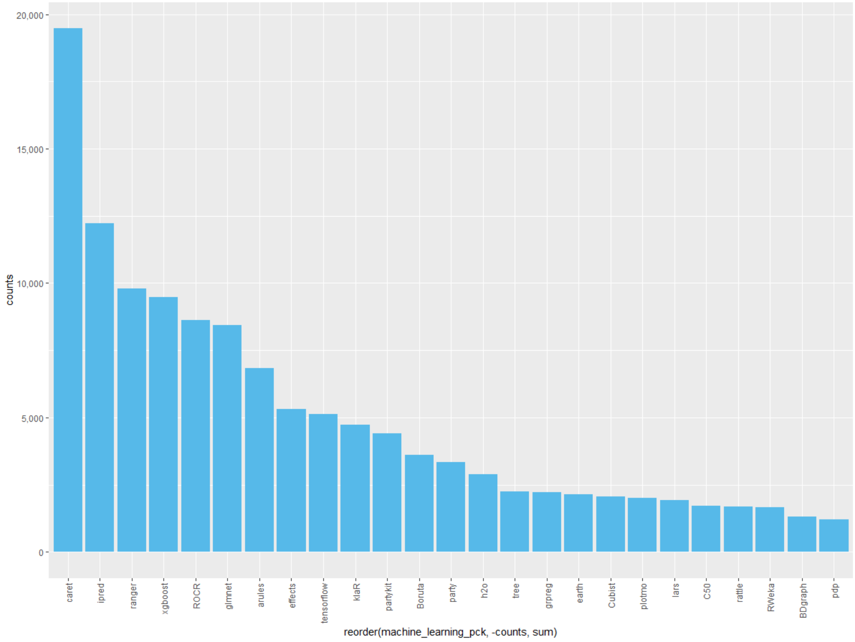


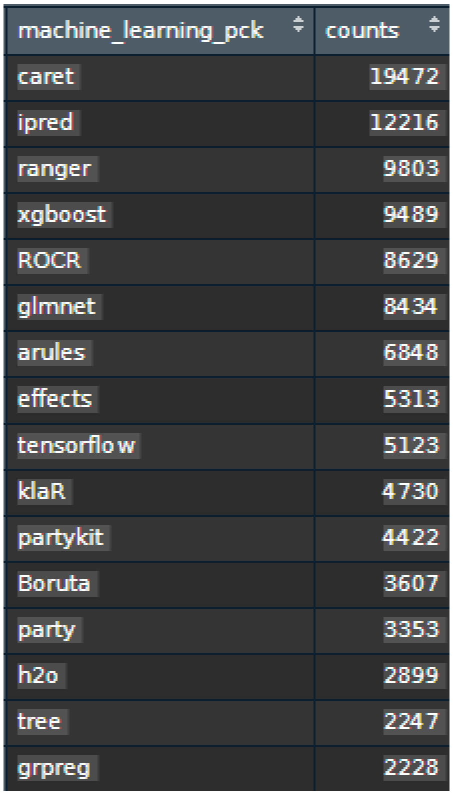
Figure 3 Downloads per OS excluding Non-Defined

Machine Learning Packages:

The leading machine learning packages are caret, ipred and ranger, with 19,472; 12,216 and 9,803 respectively



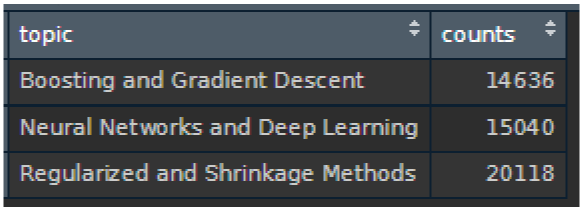
.



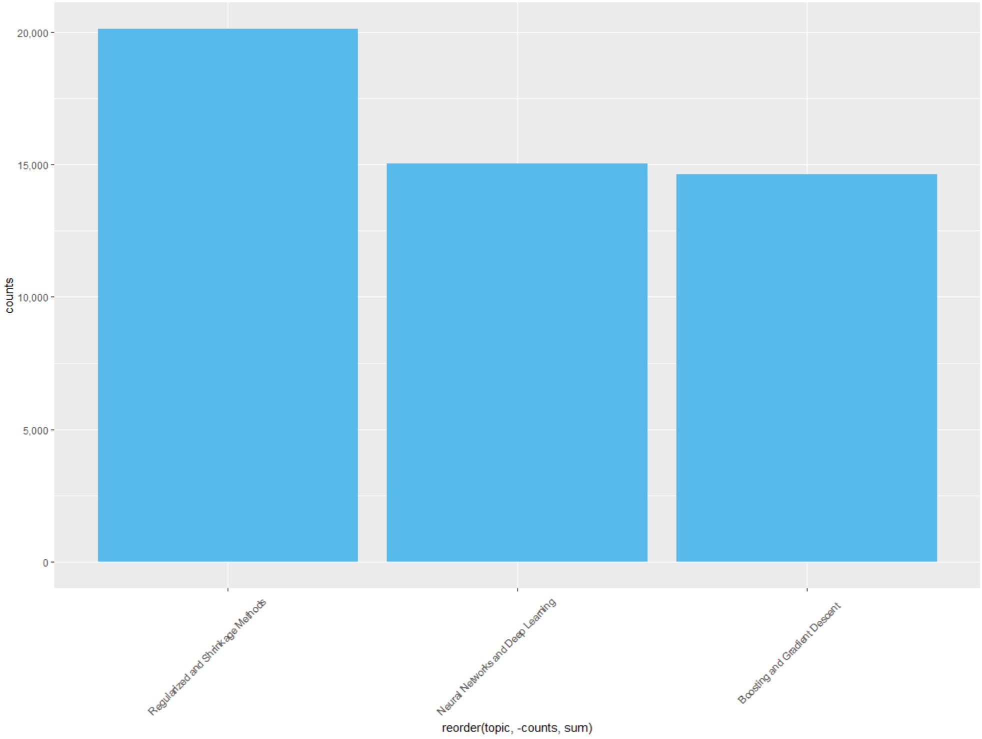
machine_learning_pck	counts
caret	19472
ipred	12216
ranger	9803
xgboost	9489
ROCR	8629
glmnet	8434
arules	6848
effects	5313
tensorflow	5123
klaR	4730
partykit	4422
Boruta	3607
party	3353
h2o	2899
tree	2247
grpreg	2228

Three Major Topics:

The leading machine learning packages are caret, ipred and ranger, with 19,472; 12,216 and 9,803 respectively



topic	counts
Boosting and Gradient Descent	14636
Neural Networks and Deep Learning	15040
Regularized and Shrinkage Methods	20118


counts
Regularized and Shrinkage Methods
Neural Networks and Deep Learning
Boosting and Gradient Descent
reorder(topic, -counts, sum)