

GRIFFITH COLLEGE DUBLIN

BIG DATA MANAGEMENT AND ANALYTICS

BIG DATA MANAGEMENT ASSIGNMENT 1

ARAVINDAN SRINIVASAN(2981707)
aravindan.srinivasan@student.griffith.ie

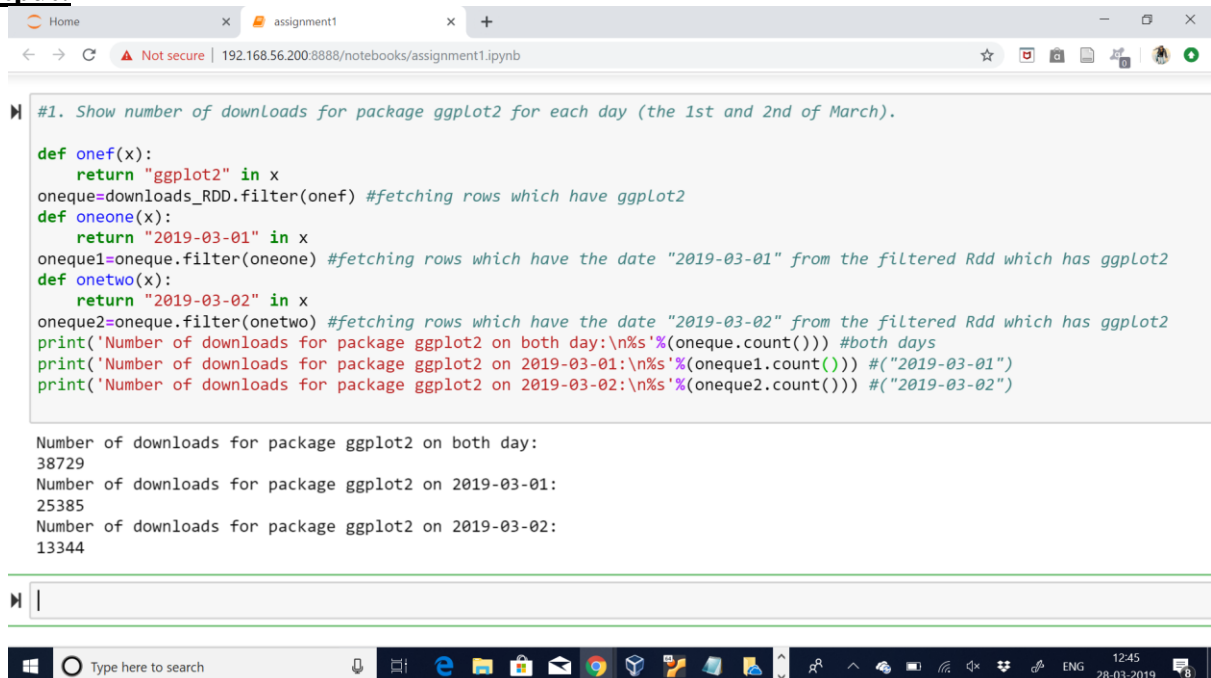
28/03/2019

1) Show number of downloads for package ggplot2 for each day (the 1st and 2nd of March).

Coding:

```
def onef(x):
    return "ggplot2" in x
oneque=downloads_RDD.filter(onef) #fetching rows which have ggplot2
def oneone(x):
    return "2019-03-01" in x
oneque1=oneque.filter(oneone) #fetching rows which have the date "2019-03-01" from the
filtered Rdd which has ggplot2
def onetwo(x):
    return "2019-03-02" in x
oneque2=oneque.filter(onetwo) #fetching rows which have the date "2019-03-02" from the
filtered Rdd which has ggplot2
print('Number of downloads for package ggplot2 on both day:\n%s'%(oneque.count())) #both
days
print('Number of downloads for package ggplot2 on 2019-03-01:\n%s'%(oneque1.count()))
#("2019-03-01")
print('Number of downloads for package ggplot2 on 2019-03-02:\n%s'%(oneque2.count()))
#("2019-03-02")
```

Output:



The screenshot shows a Jupyter Notebook interface with a single cell containing the code from the 'Coding' section. The output of the code is displayed below the cell, showing the number of downloads for the 'ggplot2' package on both days of March 2019.

```
#1. Show number of downloads for package ggplot2 for each day (the 1st and 2nd of March).

def onef(x):
    return "ggplot2" in x
oneque=downloads_RDD.filter(onef) #fetching rows which have ggplot2
def oneone(x):
    return "2019-03-01" in x
oneque1=oneque.filter(oneone) #fetching rows which have the date "2019-03-01" from the filtered Rdd which has ggplot2
def onetwo(x):
    return "2019-03-02" in x
oneque2=oneque.filter(onetwo) #fetching rows which have the date "2019-03-02" from the filtered Rdd which has ggplot2
print('Number of downloads for package ggplot2 on both day:\n%s'%(oneque.count())) #both days
print('Number of downloads for package ggplot2 on 2019-03-01:\n%s'%(oneque1.count())) #("2019-03-01")
print('Number of downloads for package ggplot2 on 2019-03-02:\n%s'%(oneque2.count())) #("2019-03-02")
```

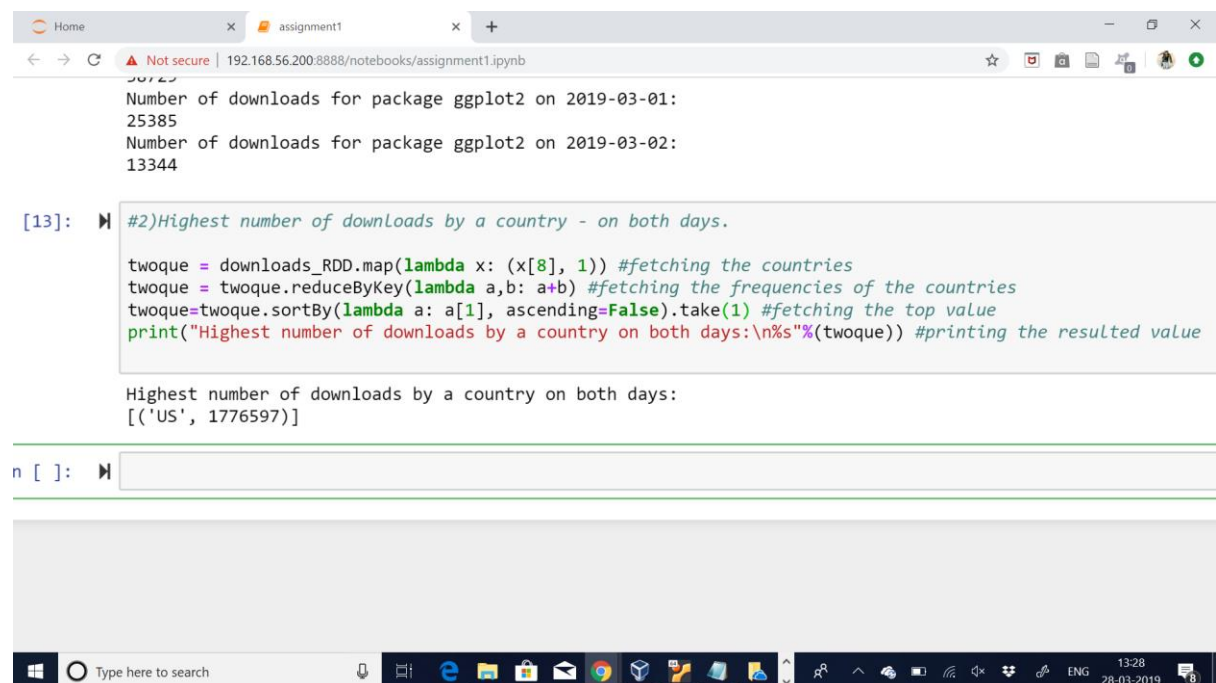
Number of downloads for package ggplot2 on both day:
38729
Number of downloads for package ggplot2 on 2019-03-01:
25385
Number of downloads for package ggplot2 on 2019-03-02:
13344

2) Highest number of downloads by a country - on both days.

Coding:

```
twoque = downloads_RDD.map(lambda x: (x[8], 1)) #fetching the countries
twoque = twoque.reduceByKey(lambda a,b: a+b) #fetching the frequencies of the
countries
twoque=twoque.sortBy(lambda a: a[1], ascending=False).take(1) #fetching the top value
print("Highest number of downloads by a country on both days:\n%s"%(twoque))
#printing the resulted value
```

Output:



```
Home x assignment1 x +
← → ↻ Not secure | 192.168.56.200:8888/notebooks/assignment1.ipynb ☆ 📄 📁 📧 📞 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿
2019-03-01
Number of downloads for package ggplot2 on 2019-03-01:
25385
Number of downloads for package ggplot2 on 2019-03-02:
13344

[13]: #2)Highest number of downloads by a country - on both days.

twoque = downloads_RDD.map(lambda x: (x[8], 1)) #fetching the countries
twoque = twoque.reduceByKey(lambda a,b: a+b) #fetching the frequencies of the countries
twoque=twoque.sortBy(lambda a: a[1], ascending=False).take(1) #fetching the top value
print("Highest number of downloads by a country on both days:\n%s"%(twoque)) #printing the resulted value

Highest number of downloads by a country on both days:
[('US', 1776597)]

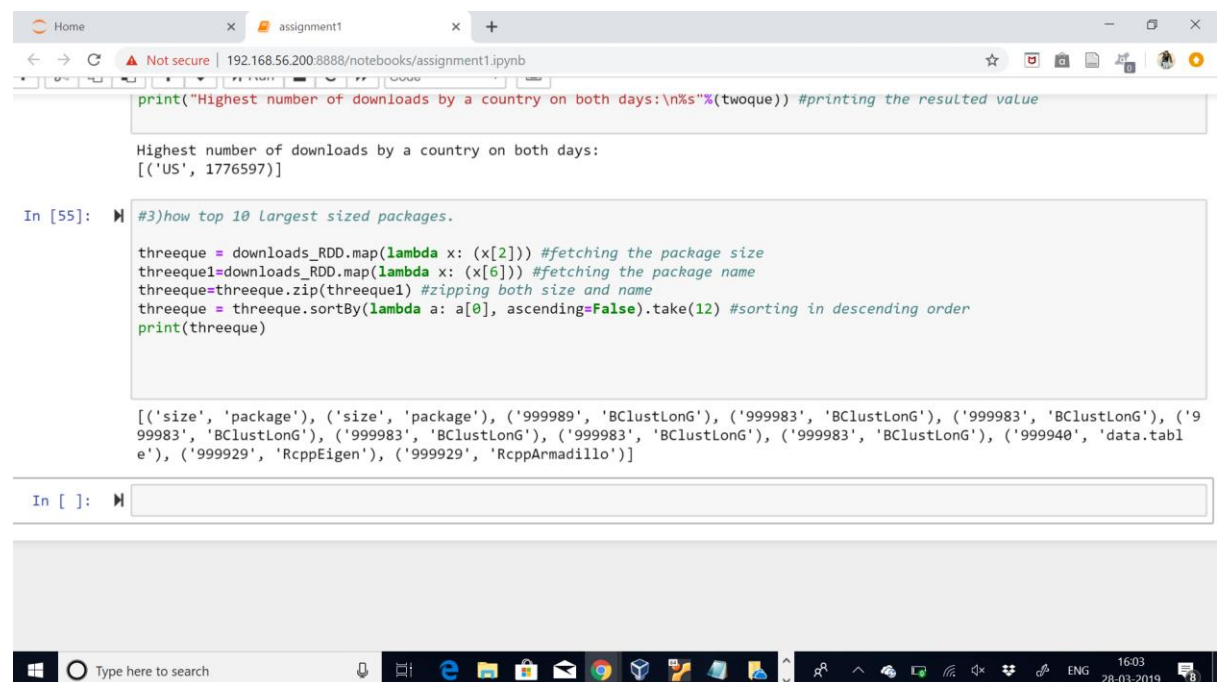
In [ ]: 
```

3)how top 10 largest sized packages.

Coding:

```
threeque = downloads_RDD.map(lambda x: (x[2])) #fetching the package size
threeque1=downloads_RDD.map(lambda x: (x[6])) #fetching the package name
threeque=threeque.zip(threeque1) #zipping both size and name
threeque = threeque.sortBy(lambda a: a[0], ascending=False).take(12) #sorting in
descending order
print(threeque)
```

Output:



The screenshot shows a Jupyter Notebook interface with a browser window at the top displaying the URL `192.168.56.200:8888/notebooks/assignment1.ipynb`. The notebook contains two code cells. The first cell executes a `print` statement that outputs the highest number of downloads by a country on both days: `[('US', 1776597)]`. The second cell, labeled `In [55]:`, contains the code for finding the top 10 largest sized packages. The output of this cell is a list of 12 tuples, each containing package size, package name, and package ID. The packages are sorted in descending order of size.

```
print("Highest number of downloads by a country on both days:\n%s"%(twoque)) #printing the resulted value

Highest number of downloads by a country on both days:
[('US', 1776597)]

In [55]: #3)how top 10 largest sized packages.

threeque = downloads_RDD.map(lambda x: (x[2])) #fetching the package size
threeque1=downloads_RDD.map(lambda x: (x[6])) #fetching the package name
threeque=threeque.zip(threeque1) #zipping both size and name
threeque = threeque.sortBy(lambda a: a[0], ascending=False).take(12) #sorting in descending order
print(threeque)

[('size', 'package'), ('size', 'package'), ('999989', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999940', 'data.tabl e'), ('999929', 'RcppEigen'), ('999929', 'RcppArmadillo')]
```

4)What were the top 10 most popular packages on 2nd of March?

Coding:

```
def fourfunc(x): #function for fetching 2nd of march data's
    return "2019-03-02" in x
fourque=downloads_RDD.filter(fourfunc) #accessing function
fourque = fourque.map(lambda x: (x[6], 1)) #fetching package column
fourque = fourque.reduceByKey(lambda a,b: a+b) #adding the frequencies
fourque=fourque.sortBy(lambda a: a[1], ascending=False).take(10) #fetching top 10
packages of 2nd march
print("top 10 most popular packages on 2nd of March:")
print(fourque)
```

Output:

The screenshot shows a web browser window displaying a Jupyter Notebook. The address bar indicates a local file path. The notebook's menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for saving, undo, redo, running code, and other functions. The main area contains two input prompts. The first prompt, labeled 'In [58]:', contains a multi-line Python function definition named 'fourfunc'. This function takes an argument 'x' and returns a date string '2019-03-02'. It then uses several RDD operations: filtering by date, selecting specific columns, adding frequency values, and sorting by frequency to return the top 10 most popular packages. The second prompt, labeled 'In []:', is currently empty. The output of the first prompt is displayed below it, showing a list of tuples where each tuple consists of a package name and its frequency value. The packages listed are 'rlang', 'Rcpp', 'tibble', 'pillar', 'yaml', 'openssl', 'stringr', 'R6', 'fansI', and 'cli'.

Home assignment1

Not secure | 192.168.56.200:8888/notebooks/assignment1.ipynb

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

```
[('size', 'package'), ('size', 'package'), ('999989', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('9  
99983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999983', 'BClustLonG'), ('999940', 'data.tabl  
e'), ('999929', 'RcppEigen'), ('999929', 'RcppArmadillo')]
```

In [58]: #4)What were the top 10 most popular packages on 2nd of March?

```
def fourfunc(x): #function for fetching 2nd of march data's  
    return "2019-03-02" in x  
fourque=downloads_RDD.filter(fourfunc) #accessing function  
fourque = fourque.map(lambda x: (x[6], 1)) #fetching package column  
fourque = fourque.reduceByKey(lambda a,b: a+b) #adding the frequencies  
fourque=fourque.sortBy(lambda a: a[1], ascending=False).take(10) #fetching top 10 packages of 2nd march  
print("top 10 most popular packages on 2nd of March:")  
print(fourque)
```

top 10 most popular packages on 2nd of March:

```
[('rlang', 19600), ('Rcpp', 18384), ('tibble', 16290), ('pillar', 14957), ('yaml', 14630), ('openssl', 14407), ('stringr', 1  
4112), ('R6', 13965), ('fansI', 13796), ('cli', 13678)]
```

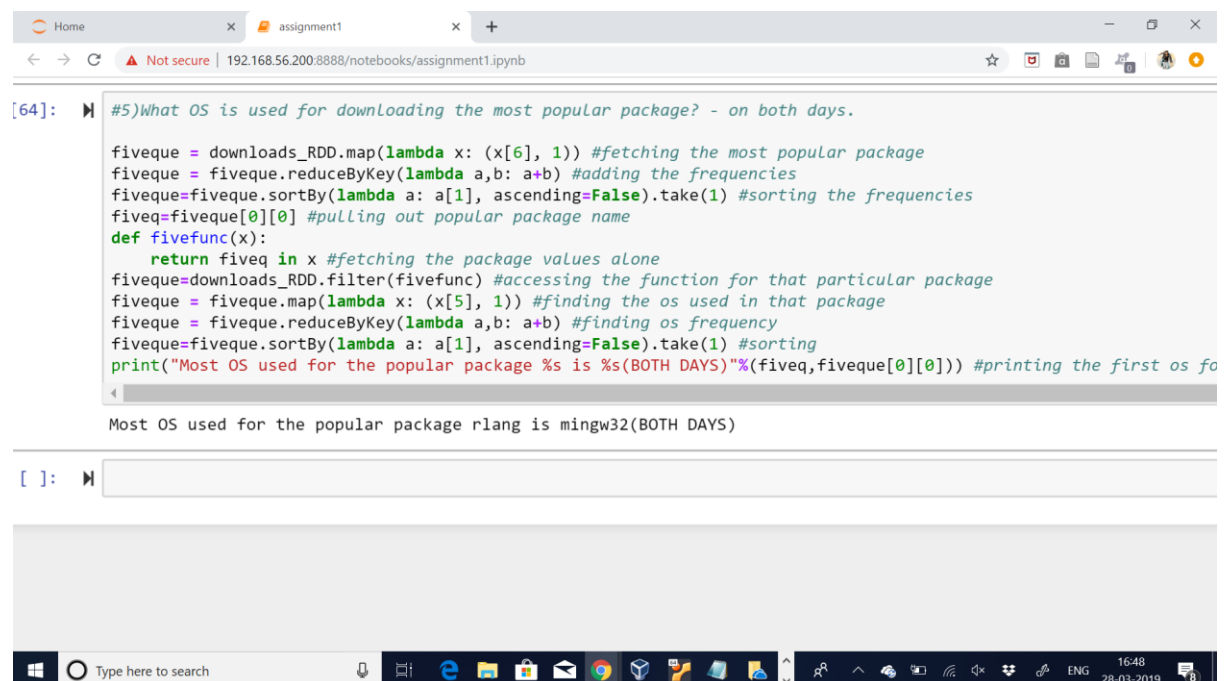
In []:

5)What OS is used for downloading the most popular package? - on both days.

Coding:

```
fiveque = downloads_RDD.map(lambda x: (x[6], 1)) #fetching the most popular package
fiveque = fiveque.reduceByKey(lambda a,b: a+b) #adding the frequencies
fiveque=fiveque.sortBy(lambda a: a[1], ascending=False).take(1) #sorting the frequencies
fiveq=fiveque[0][0] #pulling out popular package name
def fivefunc(x):
    return fiveq in x #fetching the package values alone
fiveque=downloads_RDD.filter(fivefunc) #accessing the function for that particular
package
fiveque = fiveque.map(lambda x: (x[5], 1)) #finding the os used in that package
fiveque = fiveque.reduceByKey(lambda a,b: a+b) #finding os frequency
fiveque=fiveque.sortBy(lambda a: a[1], ascending=False).take(1) #sorting
print("Most OS used for the popular package %s is %s(BOTH
DAYS)"%(fiveq,fiveque[0][0])) #printing the first os for popular package
```

Output:



The screenshot shows a Jupyter Notebook window with a single code cell. The code cell contains the same Python code as shown in the 'Coding' section. Below the code cell, the output is displayed: 'Most OS used for the popular package rlang is mingw32(BOTH DAYS)'. The notebook interface includes a browser window at the top with the URL '192.168.56.200:8888/notebooks/assignment1.ipynb'. The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 16:48 on 28-03-2019.

```
[64]: #5)What OS is used for downloading the most popular package? - on both days.

fiveque = downloads_RDD.map(lambda x: (x[6], 1)) #fetching the most popular package
fiveque = fiveque.reduceByKey(lambda a,b: a+b) #adding the frequencies
fiveque=fiveque.sortBy(lambda a: a[1], ascending=False).take(1) #sorting the frequencies
fiveq=fiveque[0][0] #pulling out popular package name
def fivefunc(x):
    return fiveq in x #fetching the package values alone
fiveque=downloads_RDD.filter(fivefunc) #accessing the function for that particular
package
fiveque = fiveque.map(lambda x: (x[5], 1)) #finding the os used in that package
fiveque = fiveque.reduceByKey(lambda a,b: a+b) #finding os frequency
fiveque=fiveque.sortBy(lambda a: a[1], ascending=False).take(1) #sorting
print("Most OS used for the popular package %s is %s(BOTH DAYS)"%(fiveq,fiveque[0][0])) #printing the first os fo

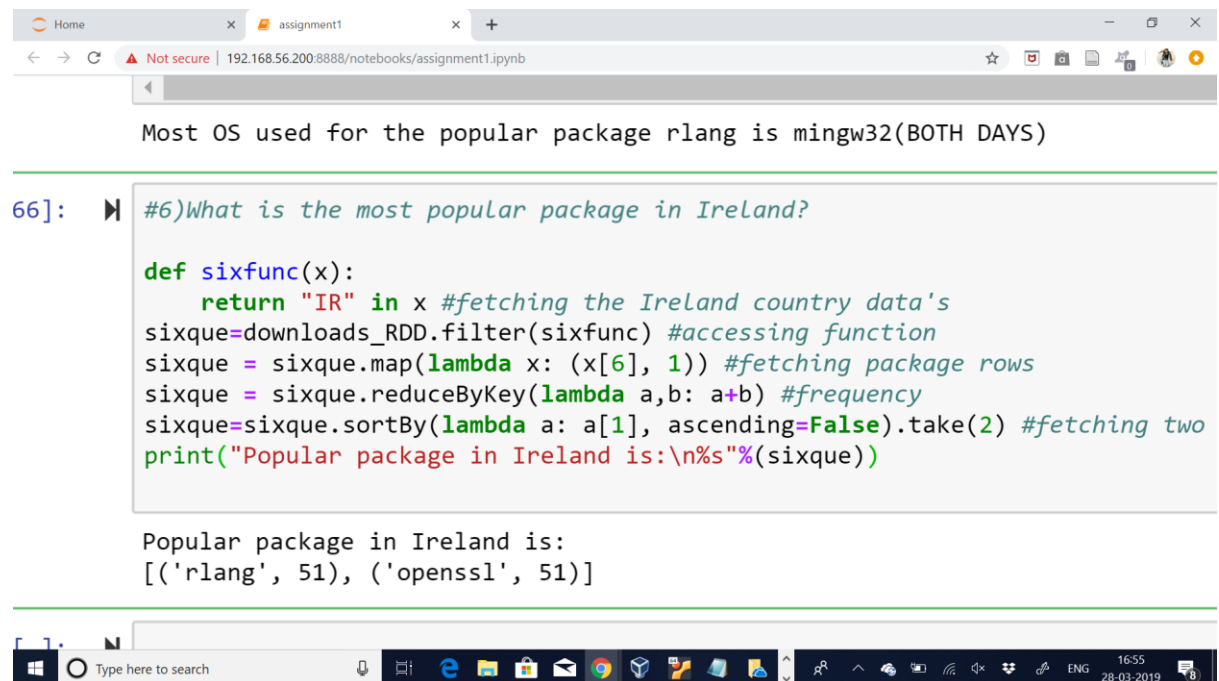
Most OS used for the popular package rlang is mingw32(BOTH DAYS)
```

6. What is the most popular package in Ireland?

Coding:

```
def sixfunc(x):
    return "IR" in x #fetching the Ireland country data's
sixque=downloads_RDD.filter(sixfunc) #accessing function
sixque = sixque.map(lambda x: (x[6], 1)) #fetching package rows
sixque = sixque.reduceByKey(lambda a,b: a+b) #frequency
sixque=sixque.sortBy(lambda a: a[1], ascending=False).take(2) #fetching two values
since both have same frequencies
print("Popular package in Ireland is:\n%s"%(sixque))
```

Output:



The screenshot shows a web browser window with a Jupyter Notebook interface. The address bar shows a local IP address. The notebook has a single code cell with the following Python code:

```
#6)What is the most popular package in Ireland?

def sixfunc(x):
    return "IR" in x #fetching the Ireland country data's
sixque=downloads_RDD.filter(sixfunc) #accessing function
sixque = sixque.map(lambda x: (x[6], 1)) #fetching package rows
sixque = sixque.reduceByKey(lambda a,b: a+b) #frequency
sixque=sixque.sortBy(lambda a: a[1], ascending=False).take(2) #fetching two
print("Popular package in Ireland is:\n%s"%(sixque))
```

The output of the code cell is:

```
Popular package in Ireland is:
[('rlang', 51), ('openssl', 51)]
```

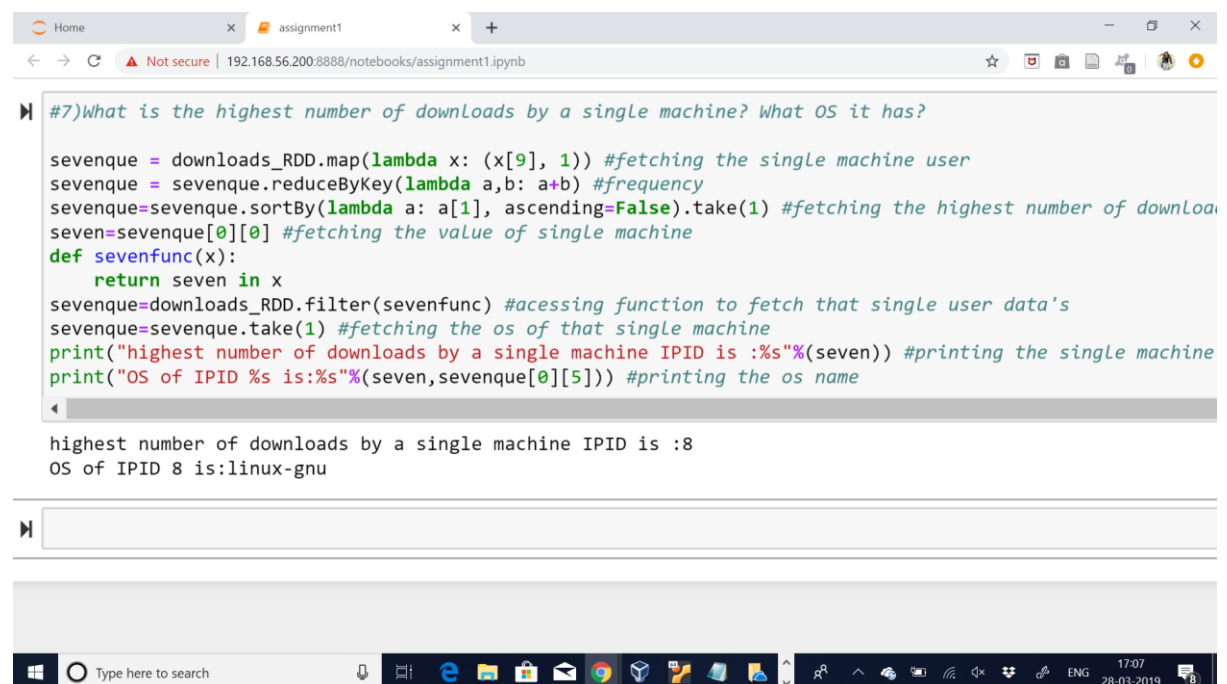
The bottom of the screenshot shows a Windows taskbar with various icons and a system clock indicating 16:55 on 28-03-2019.

7)What is the highest number of downloads by a single machine? What OS it has?

Coding:

```
sevenque = downloads_RDD.map(lambda x: (x[9], 1)) #fetching the single machine user
sevenque = sevenque.reduceByKey(lambda a,b: a+b) #frequency
sevenque=sevenque.sortBy(lambda a: a[1], ascending=False).take(1) #fetching the
highest number of downloaded by a single machine
seven=sevenque[0][0] #fetching the value of single machine
def sevenfunc(x):
    return seven in x
sevenque=downloads_RDD.filter(sevenfunc) #accessing function to fetch that single user
data's
sevenque=sevenque.take(1) #fetching the os of that single machine
print("highest number of downloads by a single machine IPID is :%s"%(seven)) #printing
the single machine
print("OS of IPID %s is:%s"%(seven,sevenque[0][5])) #printing the os name
```

Output:



```
#7)What is the highest number of downloads by a single machine? What OS it has?

sevenque = downloads_RDD.map(lambda x: (x[9], 1)) #fetching the single machine user
sevenque = sevenque.reduceByKey(lambda a,b: a+b) #frequency
sevenque=sevenque.sortBy(lambda a: a[1], ascending=False).take(1) #fetching the highest number of download
seven=sevenque[0][0] #fetching the value of single machine
def sevenfunc(x):
    return seven in x
sevenque=downloads_RDD.filter(sevenfunc) #accessing function to fetch that single user data's
sevenque=sevenque.take(1) #fetching the os of that single machine
print("highest number of downloads by a single machine IPID is :%s"%(seven)) #printing the single machine
print("OS of IPID %s is:%s"%(seven,sevenque[0][5])) #printing the os name
```

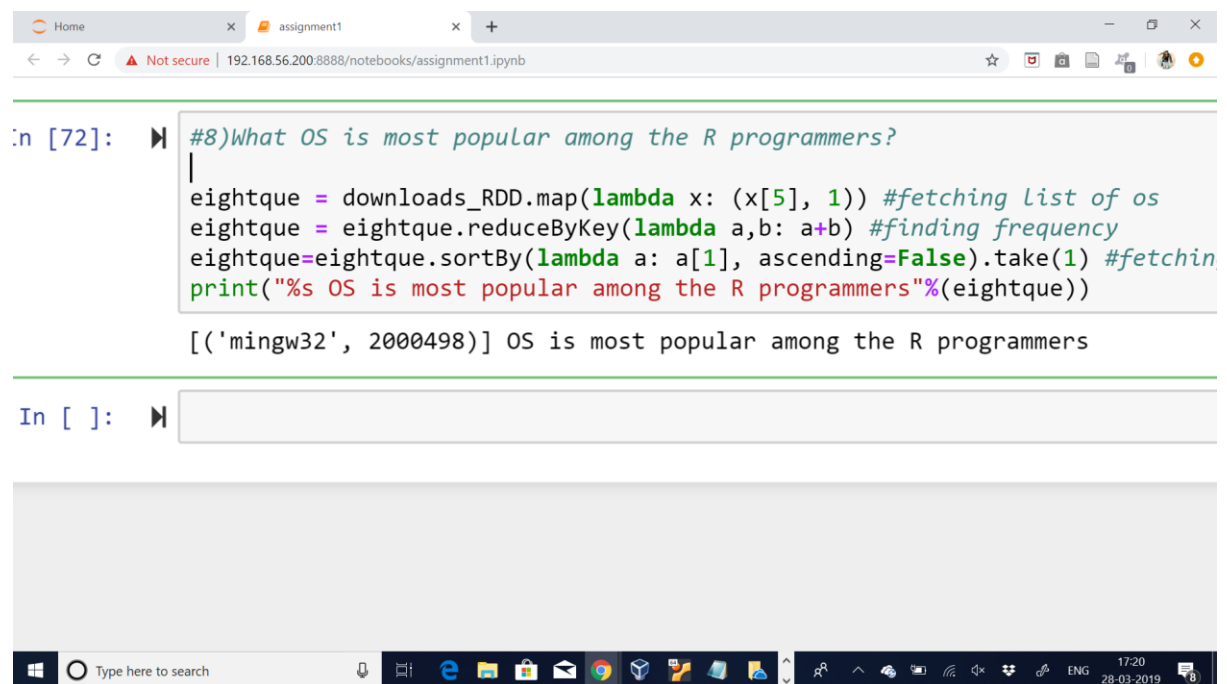
highest number of downloads by a single machine IPID is :8
OS of IPID 8 is:linux-gnu

8) What OS is most popular among the R programmers?

Coding:

```
eightque = downloads_RDD.map(lambda x: (x[5], 1)) #fetching list of os
eightque = eightque.reduceByKey(lambda a,b: a+b) #finding frequency
eightque=eightque.sortBy(lambda a: a[1], ascending=False).take(1) #fetching first values
print("%s OS is most popular among the R programmers"%(eightque))
```

Output:



```
In [72]: #8)What OS is most popular among the R programmers?
|
| eightque = downloads_RDD.map(lambda x: (x[5], 1)) #fetching list of os
| eightque = eightque.reduceByKey(lambda a,b: a+b) #finding frequency
| eightque=eightque.sortBy(lambda a: a[1], ascending=False).take(1) #fetchin
| print("%s OS is most popular among the R programmers"%(eightque))
|
| '['mingw32', 2000498]' OS is most popular among the R programmers
```

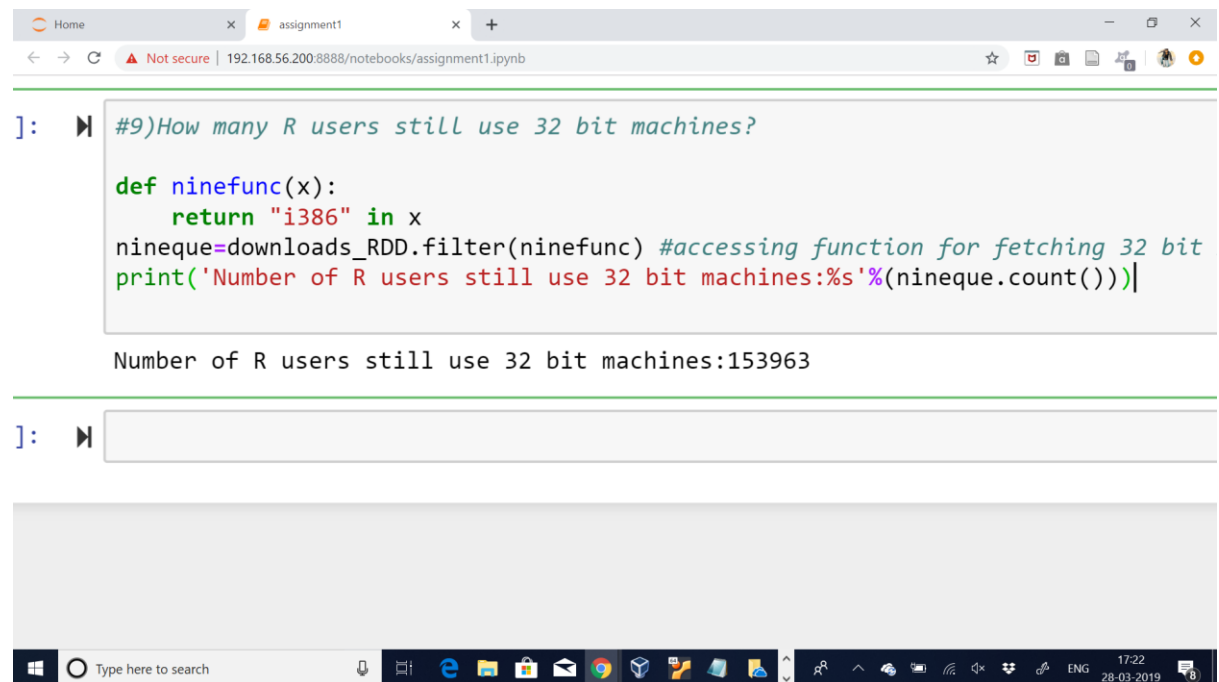
```
In [ ]:
```


9) How many R users still use 32 bit machines?

Coding:

```
def ninefunc(x):  
    return "i386" in x  
nineque=downloads_RDD.filter(ninefunc) #accessing function for fetching 32 bit  
machines  
print('Number of R users still use 32 bit machines:%s'%(nineque.count()))
```

Output:



The screenshot shows a web browser window with a Jupyter Notebook interface. The browser's address bar shows a local IP address. The notebook has two cells. The first cell contains the following Python code:

```
] : ▶ #9)How many R users still use 32 bit machines?  
  
def ninefunc(x):  
    return "i386" in x  
nineque=downloads_RDD.filter(ninefunc) #accessing function for fetching 32 bit  
machines  
print('Number of R users still use 32 bit machines:%s'%(nineque.count()))
```

The output of the first cell is displayed below the code:

```
Number of R users still use 32 bit machines:153963
```

The second cell is currently empty and shows a prompt character.

10) List total number of incomplete records - lines which have missing values.

Coding:

```
def tenfunc(x):  
    return "NA" in x  
tenque=downloads_RDD.filter(tenfunc) #accessing function for NA values  
print("Total number of incomplete records:%s"%(tenque.count()))
```

Output:

The screenshot displays a web browser window with a Jupyter Notebook interface. The browser's address bar shows a local IP address: 192.168.56.200:8888/notebooks/assignment1.ipynb. The notebook contains a code cell with the following Python code:

```
#10)List total number of incomplete records - Lines which have m  
  
def tenfunc(x):  
    return "NA" in x  
tenque=downloads_RDD.filter(tenfunc) #accessing function for NA  
print('Total number of incomplete records:%s'%(tenque.count()))
```

Below the code cell, the output is displayed as a text string: "Total number of incomplete records:468436". The interface includes a toolbar with various icons for file operations and a Windows taskbar at the bottom showing the search bar, taskbar icons, and system clock (17:34, 28-03-2019).