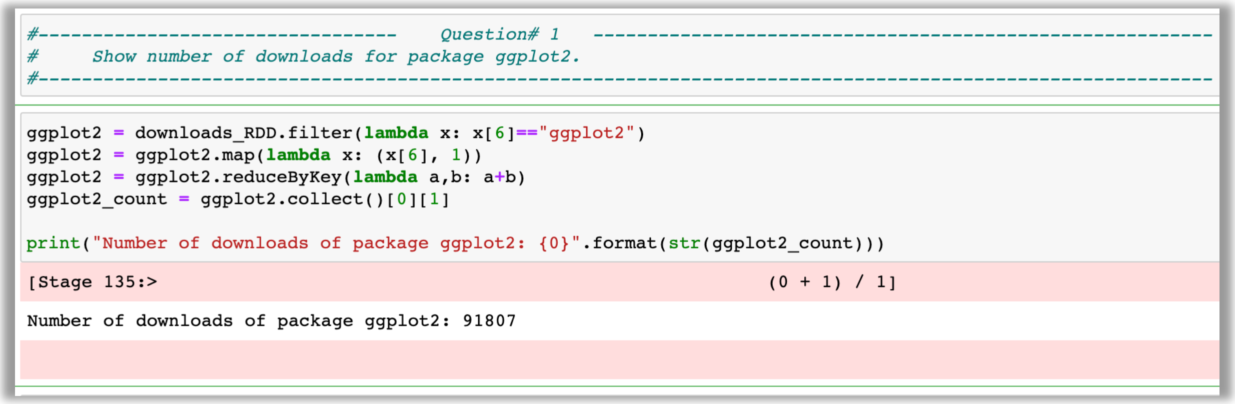
Assignment 01 – Big Data Management

Student Number: 3049429 **Student Name: Jawad Adil**

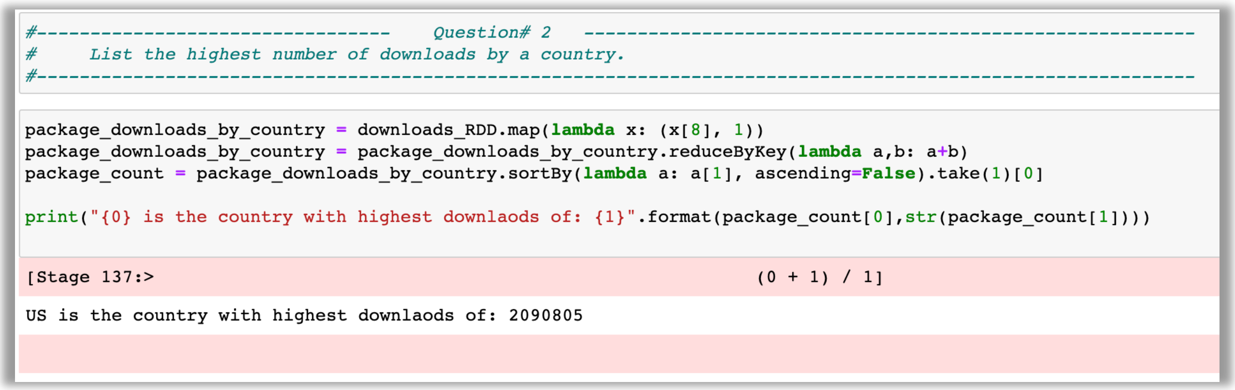
Question#1:



```
#--------------------------------     Question# 1   ----------------------------------------------------------
#     Show number of downloads for package ggplot2.
#-------------------------------------------------------------------------------------------------------------

ggplot2 = downloads_RDD.filter(lambda x: x[6]=="ggplot2")
ggplot2 = ggplot2.map(lambda x: (x[6], 1))
ggplot2 = ggplot2.reduceByKey(lambda a,b: a+b)
ggplot2_count = ggplot2.collect()[0][1]

print("Number of downloads of package ggplot2: {0}".format(str(ggplot2_count)))
```

```
[Stage 135:>                                                          (0 + 1) / 1]

Number of downloads of package ggplot2: 91807
```

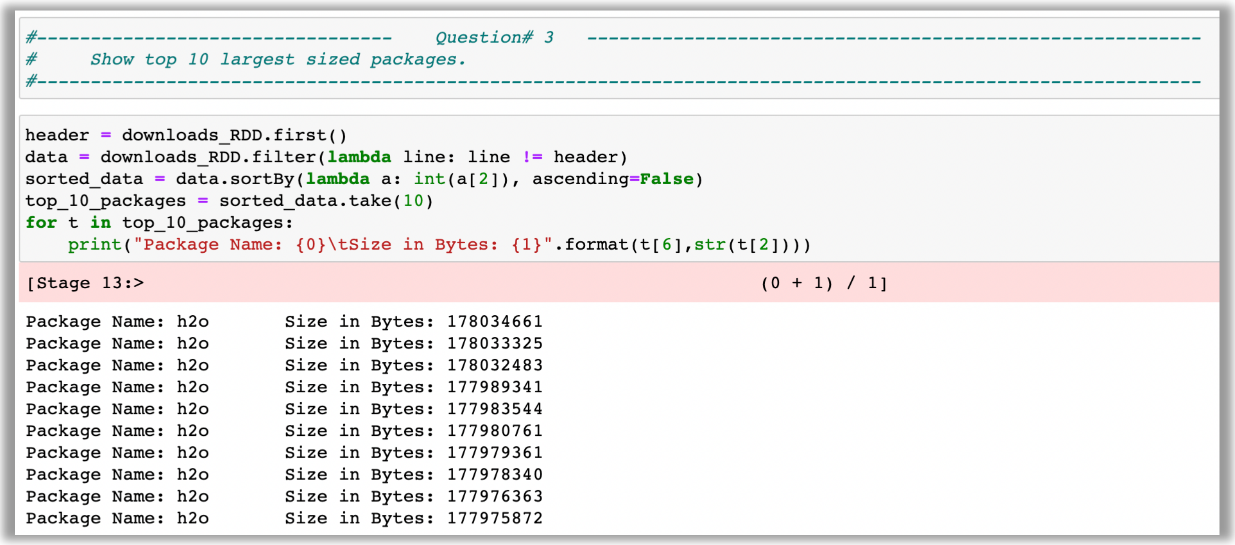
Question#2:



```
#--------------------------------     Question# 2   --------------------------------------------------------
#     List the highest number of downloads by a country.
#-----------------------------------------------------------------------------------------------------------

package_downloads_by_country = downloads_RDD.map(lambda x: (x[8], 1))
package_downloads_by_country = package_downloads_by_country.reduceByKey(lambda a,b: a+b)
package_count = package_downloads_by_country.sortBy(lambda a: a[1], ascending=False).take(1)[0]

print("{0} is the country with highest downlaods of: {1}".format(package_count[0],str(package_count[1])))
```

```
[Stage 137:>                                                          (0 + 1) / 1]

US is the country with highest downlaods of: 2090805
```

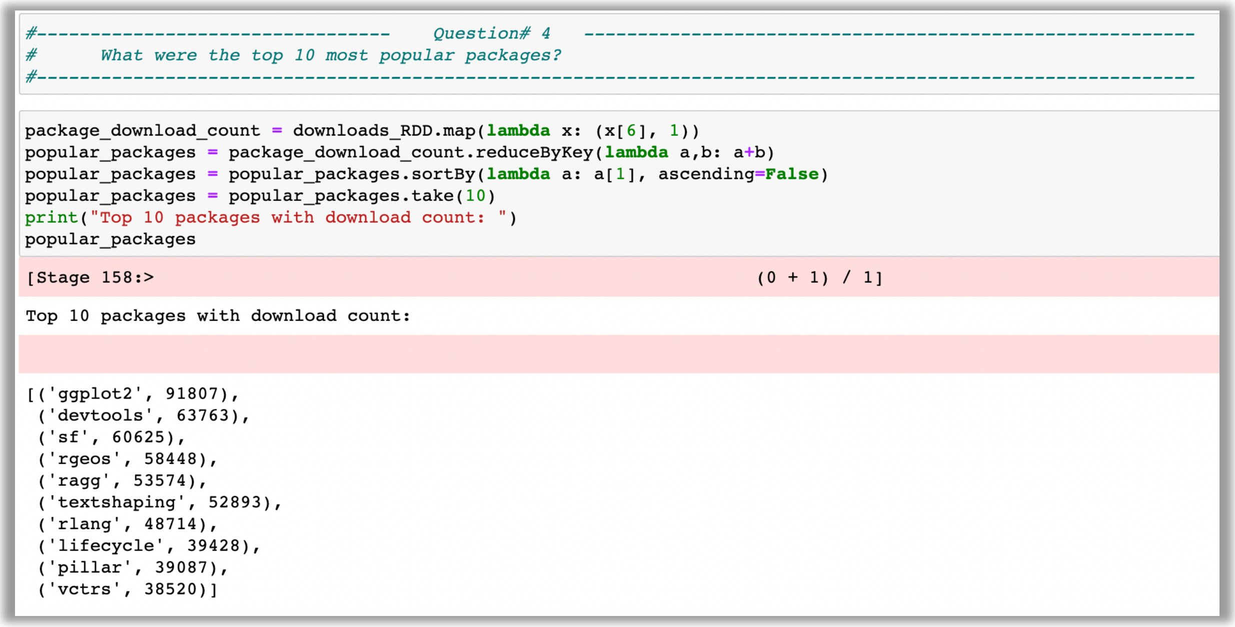
Question#3:



```
#--------------------------------     Question# 3   --------------------------------------------------------
#     Show top 10 largest sized packages.
#-----------------------------------------------------------------------------------------------------------

header = downloads_RDD.first()
data = downloads_RDD.filter(lambda line: line != header)
sorted_data = data.sortBy(lambda a: int(a[2]), ascending=False)
top_10_packages = sorted_data.take(10)
for t in top_10_packages:
    print("Package Name: {0}\tSize in Bytes: {1}".format(t[6],str(t[2])))
```

```
[Stage 13:>                                                           (0 + 1) / 1]

Package Name: h2o       Size in Bytes: 178034661
Package Name: h2o       Size in Bytes: 178033325
Package Name: h2o       Size in Bytes: 178032483
Package Name: h2o       Size in Bytes: 177989341
Package Name: h2o       Size in Bytes: 177983544
Package Name: h2o       Size in Bytes: 177980761
Package Name: h2o       Size in Bytes: 177979361
Package Name: h2o       Size in Bytes: 177978340
Package Name: h2o       Size in Bytes: 177976363
Package Name: h2o       Size in Bytes: 177975872
```

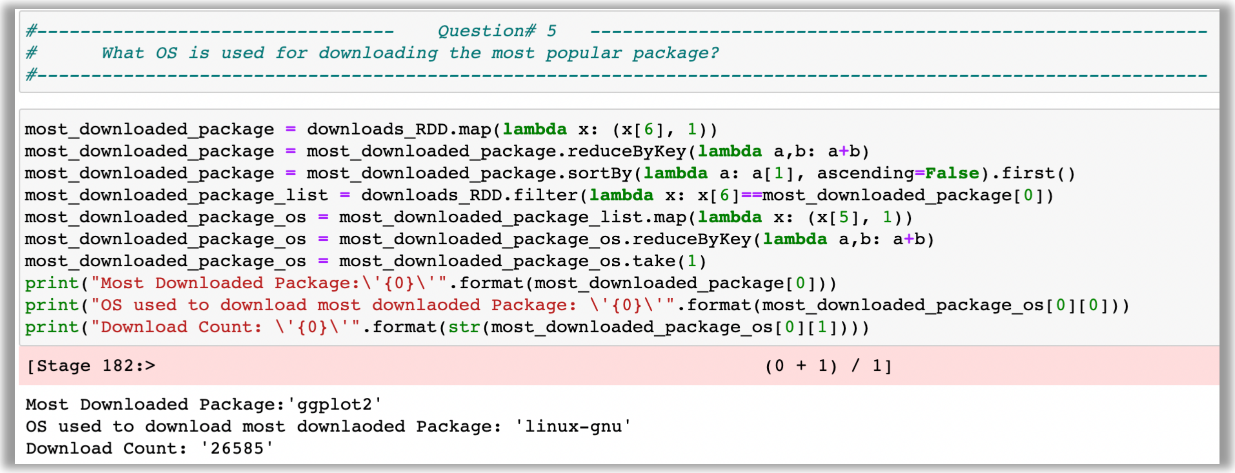
Question#4:



```
#----------------------------------    Question# 4   -------------------------------------------------------------
#      What were the top 10 most popular packages?
#-----------------------------------------------------------------------------------------------------------------

package_download_count = downloads_RDD.map(lambda x: (x[6], 1))
popular_packages = package_download_count.reduceByKey(lambda a,b: a+b)
popular_packages = popular_packages.sortBy(lambda a: a[1], ascending=False)
popular_packages = popular_packages.take(10)
print("Top 10 packages with download count: ")
popular_packages
```

```
[Stage 158:>                                                          (0 + 1) / 1]

Top 10 packages with download count:

[('ggplot2', 91807),
 ('devtools', 63763),
 ('sf', 60625),
 ('rgeos', 58448),
 ('ragg', 53574),
 ('textshaping', 52893),
 ('rlang', 48714),
 ('lifecycle', 39428),
 ('pillar', 39087),
 ('vctrs', 38520)]
```

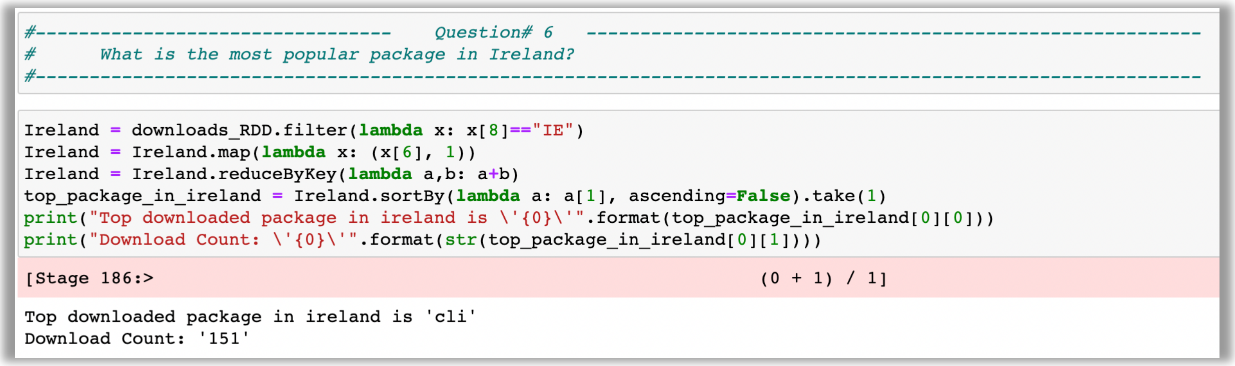
Question#5:



```
#----------------------------------    Question# 5   -------------------------------------------------------------
#      What OS is used for downloading the most popular package?
#-----------------------------------------------------------------------------------------------------------------

most_downloaded_package = downloads_RDD.map(lambda x: (x[6], 1))
most_downloaded_package = most_downloaded_package.reduceByKey(lambda a,b: a+b)
most_downloaded_package = most_downloaded_package.sortBy(lambda a: a[1], ascending=False).first()
most_downloaded_package_list = downloads_RDD.filter(lambda x: x[6]==most_downloaded_package[0])
most_downloaded_package_os = most_downloaded_package_list.map(lambda x: (x[5], 1))
most_downloaded_package_os = most_downloaded_package_os.reduceByKey(lambda a,b: a+b)
most_downloaded_package_os = most_downloaded_package_os.take(1)
print("Most Downloaded Package:\'{0}\'".format(most_downloaded_package[0]))
print("OS used to download most downlaoded Package: \'{0}\'".format(most_downloaded_package_os[0][0]))
print("Download Count: \'{0}\'".format(str(most_downloaded_package_os[0][1])))
```

```
[Stage 182:>                                                          (0 + 1) / 1]

Most Downloaded Package:'ggplot2'
OS used to download most downlaoded Package: 'linux-gnu'
Download Count: '26585'
```

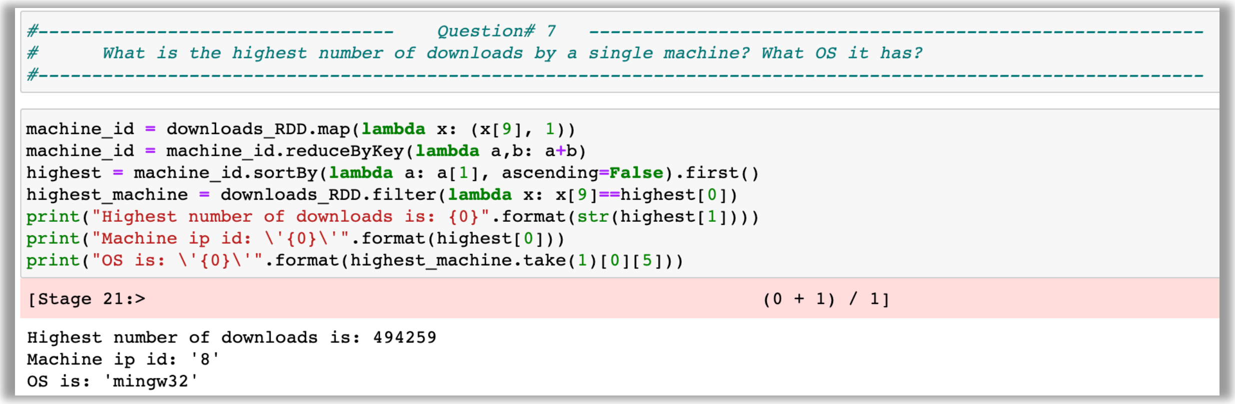
Question#6:



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

Ireland = downloads_RDD.filter(lambda x: x[8]=="IE")
Ireland = Ireland.map(lambda x: (x[6], 1))
Ireland = Ireland.reduceByKey(lambda a,b: a+b)
top_package_in_ireland = Ireland.sortBy(lambda a: a[1], ascending=False).take(1)
print("Top downloaded package in ireland is \'{0}\'".format(top_package_in_ireland[0][0]))
print("Download Count: \'{0}\'".format(str(top_package_in_ireland[0][1])))
```

```
[Stage 186:>                                                          (0 + 1) / 1]

Top downloaded package in ireland is 'cli'
Download Count: '151'
```

Question#7:

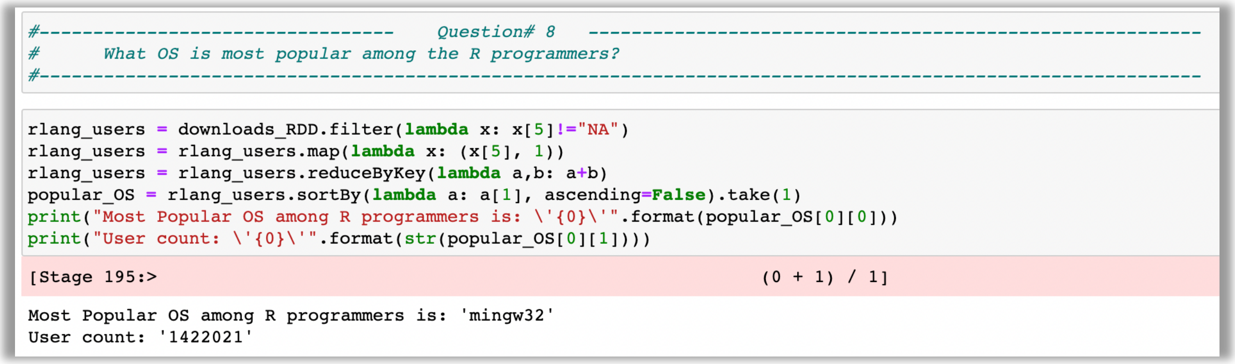


```
#----------------------------------    Question# 7   ----------------------------------------------------------
#      What is the highest number of downloads by a single machine? What OS it has?
#-------------------------------------------------------------------------------------------------------------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

```
machine_id = downloads_RDD.map(lambda x: (x[9], 1))
machine_id = machine_id.reduceByKey(lambda a,b: a+b)
highest = machine_id.sortBy(lambda a: a[1], ascending=False).first()
highest_machine = downloads_RDD.filter(lambda x: x[9]==highest[0])
print("Highest number of downloads is: {0}".format(str(highest[1])))
print("Machine ip id: \'{0}\'".format(highest[0]))
print("OS is: \'{0}\'".format(highest_machine.take(1)[0][5]))
```

```
[Stage 21:>                                                         (0 + 1) / 1]

Highest number of downloads is: 494259
Machine ip id: '8'
OS is: 'mingw32'
```

Question#8:

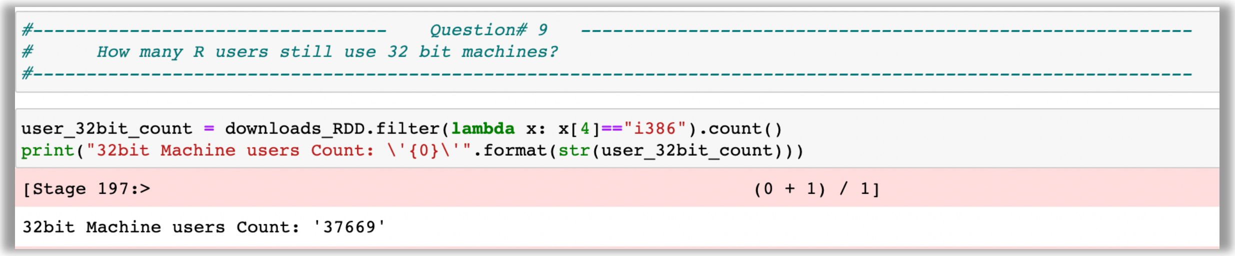


```
#----------------------------------    Question# 8   ----------------------------------------------------------
#      What OS is most popular among the R programmers?
#-------------------------------------------------------------------------------------------------------------
```

```
rlang_users = downloads_RDD.filter(lambda x: x[5]!="NA")
rlang_users = rlang_users.map(lambda x: (x[5], 1))
rlang_users = rlang_users.reduceByKey(lambda a,b: a+b)
popular_OS = rlang_users.sortBy(lambda a: a[1], ascending=False).take(1)
print("Most Popular OS among R programmers is: \'{0}\'".format(popular_OS[0][0]))
print("User count: \'{0}\'".format(str(popular_OS[0][1])))
```

```
[Stage 195:>                                                        (0 + 1) / 1]

Most Popular OS among R programmers is: 'mingw32'
User count: '1422021'
```

Question#9:

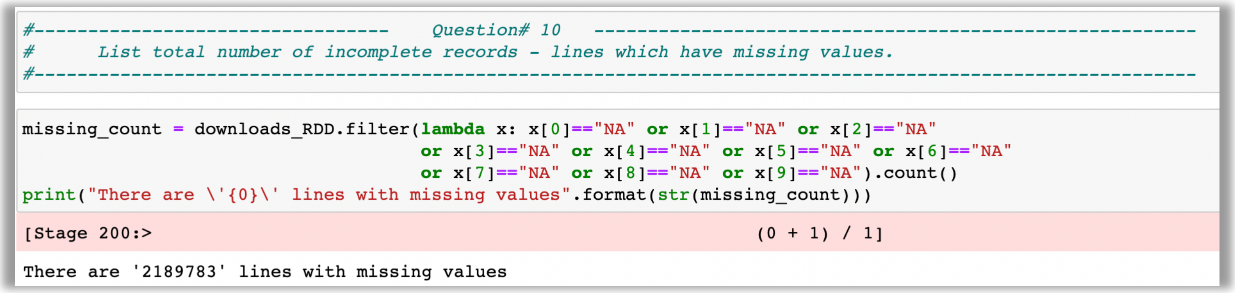


```
#----------------------------------    Question# 9   ----------------------------------------------------------
#      How many R users still use 32 bit machines?
#-------------------------------------------------------------------------------------------------------------
```

```
user_32bit_count = downloads_RDD.filter(lambda x: x[4]=="i386").count()
print("32bit Machine users Count: \'{0}\'".format(str(user_32bit_count)))
```

```
[Stage 197:>                                                        (0 + 1) / 1]

32bit Machine users Count: '37669'
```

Question#10:



```
#----------------------------------    Question# 10   ---------------------------------------------------------
#      List total number of incomplete records - lines which have missing values.
#-------------------------------------------------------------------------------------------------------------
```

```
missing_count = downloads_RDD.filter(lambda x: x[0]=="NA" or x[1]=="NA" or x[2]=="NA"
                                     or x[3]=="NA" or x[4]=="NA" or x[5]=="NA" or x[6]=="NA"
                                     or x[7]=="NA" or x[8]=="NA" or x[9]=="NA").count()
print("There are \'{0}\' lines with missing values".format(str(missing_count)))
```

```
[Stage 200:>                                                        (0 + 1) / 1]

There are '2189783' lines with missing values
```