

Assignment 3: BDM  
Siddhi Kate  
Student\_No. : 2982006

Python code : for creating log files.

```
In [1]: import csv
with open('Day-1.txt', "w") as my_output_file:
    with open('2019-03-01.csv', "r") as my_input_file:
        [ my_output_file.write(",".join(row)+'\n') for row in csv.reader(my_input_file)]
    my_output_file.close()
```

```
In [3]: import csv
with open('Day-2.txt', "w") as my_output_file:
    with open('2019-03-02.csv', "r") as my_input_file:
        [ my_output_file.write(",".join(row)+'\n') for row in csv.reader(my_input_file)]
    my_output_file.close()
```

```
import time
a = 1
with open('Day-1.txt', 'r') as file_1:
    with open('Day-2.txt', 'r') as file_2:
        lines_1 = file_1.readlines()
        len_1 = len(lines_1)
        lines_2 = file_2.readlines()
        len_2 = len(lines_2)
        i_file1 = 1
        j_file1 = 1000
        i_file2 = 1
        j_file2 = 1000
        while j_file1 != len_1 and j_file2 != len_2 :
            if i_file1==1 and i_file2==1:
                with open('Log/log{}.txt'.format(a), 'w') as writefile:
                    writefile.write(' '.join(line for line in lines_1[i_file1:j_file1]))
                i_file1 = j_file1
                j_file1 += 1000
                print('creating file log{}.txt'.format(a))
                a += 1

                with open('Log/log{}.txt'.format(a), 'w') as writefile:
                    writefile.write(' '.join(line for line in lines_2[i_file2:j_file2]))
```

```
                with open('Log/log{}.txt'.format(a), 'w') as writefile:
                    writefile.write(' '.join(line for line in lines_2[i_file2:j_file2]))
                i_file2 = j_file2
                j_file2 += 1000
                print('creating file log{}.txt'.format(a))
                a += 1
                time.sleep(5)
            else:
                with open('Log/log{}.txt'.format(a), 'w') as writefile:
                    writefile.write(' '.join(line for line in lines_1[i_file1:j_file1]))
                i_file1 = j_file1
                j_file1 += 1000
                print('creating file log{}.txt'.format(a))
                a += 1

                with open('Log/log{}.txt'.format(a), 'w') as writefile:
                    writefile.write(' '.join(line for line in lines_2[i_file2:j_file2]))
                i_file2 = j_file2
                j_file2 += 1000
                print('creating file log{}.txt'.format(a))
                a += 1
                time.sleep(5)
```

creating file log1.txt

## Key space generation:

```
cqlsh> CREATE KEYSPACE ASSIGNMENT_03
... WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1
};
cqlsh> use ASSIGNMENT_03;
cqlsh:assignment_03> CREATE TABLE QUESTION_01(
... TIME TEXT,
... PACKAGE TEXT,
... COUNT INT,
... PRIMARY KEY(TIME, PACKAGE)
... );
cqlsh:assignment_03> CREATE TABLE QUESTION_02( TIME TEXT, PACKAGE TEXT, COUNT IN
T, PRIMARY KEY(TIME, PACKAGE) );
cqlsh:assignment_03> CREATE TABLE QUESTION_03( TIME TEXT, COUNTRY TEXT, COUNT IN
T, PRIMARY KEY(TIME, COUNTRY) );
cqlsh:assignment_03> CREATE TABLE QUESTION_04( TIME TEXT, PACKAGE TEXT, COUNT IN
T, PRIMARY KEY(TIME, PACKAGE) );
cqlsh:assignment_03> describe tables

question_01 question_02 question_03 question_04
```

## Streaming application for queries and their output in Cassandra.

1. Question: 1 To calculate the number of downloads of each package.

```
In [1]: #Question 1
from pyspark.sql import Row
from pyspark.streaming import StreamingContext
import time

def save(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
            lambda row: Row(time=time, package=row[0], count=row[1])))
        df.write.format("org.apache.spark.sql.cassandra")\
            .options(table="question_01", keyspace="assignment_03")\
            .save(mode="append")
    except:
        pass

ssc = StreamingContext(sc, 5)
lines = ssc.textFileStream("file:///home/siddhi123/Log/")
words = lines.map(lambda line: line.split(","))
package_download_count = words.map(lambda x: (x[6], 1))
package_download_count = package_download_count.reduceByKey(lambda a,b: a+b)
package_download_count.pprint()
package_download_count.foreachRDD(save)
ssc.start()
```

## Output:

time	package	count
2019-05-02 20:07:50+0100	ABCExtremes	1
2019-05-02 20:07:50+0100	BH	8
2019-05-02 20:07:50+0100	BHSBVAR	1
2019-05-02 20:07:50+0100	BMA	1
2019-05-02 20:07:50+0100	BRugs	1
2019-05-02 20:07:50+0100	BalancedSampling	1
2019-05-02 20:07:50+0100	Bessel	1
2019-05-02 20:07:50+0100	BiasedUrn	1
2019-05-02 20:07:50+0100	BiocManager	1
2019-05-02 20:07:50+0100	BradleyTerry2	1
2019-05-02 20:07:50+0100	CHNOSZ	1
2019-05-02 20:07:50+0100	Cubist	1
2019-05-02 20:07:50+0100	DBI	6
2019-05-02 20:07:50+0100	DEoptimR	2
2019-05-02 20:07:50+0100	DHARMA	1
2019-05-02 20:07:50+0100	DescTools	2
2019-05-02 20:07:50+0100	DiagrammeR	1
2019-05-02 20:07:50+0100	EMT	1
2019-05-02 20:07:50+0100	EnvStats	1
2019-05-02 20:07:50+0100	Epi	1
2019-05-02 20:07:50+0100	ForeCA	1
2019-05-02 20:07:50+0100	ForecastComb	1
2019-05-02 20:07:50+0100	GA	1
2019-05-02 20:07:50+0100	GGally	1
2019-05-02 20:07:50+0100	GWLeleast	1
2019-05-02 20:07:50+0100	GeneralizedHyperbolic	1
2019-05-02 20:07:50+0100	HMM	1
2019-05-02 20:07:50+0100	HMMCont	1
2019-05-02 20:07:50+0100	HMMpa	1
2019-05-02 20:07:50+0100	Hmisc	2
2019-05-02 20:07:50+0100	ISLR	1
2019-05-02 20:07:50+0100	ISOCodes	1
2019-05-02 20:07:50+0100	ISwR	1
2019-05-02 20:07:50+0100	MASS	1
2019-05-02 20:07:50+0100	MBESS	1

2019-05-02	20:07:50+0100	GeneralizedHyperbolic	1
2019-05-02	20:07:50+0100	HMM	1
2019-05-02	20:07:50+0100	HMMCont	1
2019-05-02	20:07:50+0100	HMMpa	1
2019-05-02	20:07:50+0100	Hmisc	2
2019-05-02	20:07:50+0100	ISLR	1
2019-05-02	20:07:50+0100	ISOCodes	1
2019-05-02	20:07:50+0100	ISwR	1
2019-05-02	20:07:50+0100	MASS	1
2019-05-02	20:07:50+0100	MBESS	1
2019-05-02	20:07:50+0100	MLmetrics	1
2019-05-02	20:07:50+0100	Matrix	1
2019-05-02	20:07:50+0100	MatrixModels	2
2019-05-02	20:07:50+0100	NbClust	1
2019-05-02	20:07:50+0100	Nlcoptim	1
2019-05-02	20:07:50+0100	OpenMx	1
2019-05-02	20:07:50+0100	PMCMR	1
2019-05-02	20:07:50+0100	PerformanceAnalytics	3
2019-05-02	20:07:50+0100	R2admb	1
2019-05-02	20:07:50+0100	R6	8
2019-05-02	20:07:50+0100	RANN	1
2019-05-02	20:07:50+0100	RColorBrewer	2
2019-05-02	20:07:50+0100	RCurl	1
2019-05-02	20:07:50+0100	RGtk2Extras	1
2019-05-02	20:07:50+0100	ROCR	1
2019-05-02	20:07:50+0100	RSQLite	3
2019-05-02	20:07:50+0100	RWeka	12
2019-05-02	20:07:50+0100	RWekajars	8
2019-05-02	20:07:50+0100	Rcpp	6
2019-05-02	20:07:50+0100	RcppEigen	3
2019-05-02	20:07:50+0100	RgoogleMaps	1
2019-05-02	20:07:50+0100	Rserve	1
2019-05-02	20:07:50+0100	Rsolnp	1
2019-05-02	20:07:50+0100	SQUAREM	1
2019-05-02	20:07:50+0100	SparseM	2
2019-05-02	20:07:50+0100	TH.data	2
2019-05-02	20:07:50+0100	TeachingDemos	1
2019-05-02	20:07:50+0100	VennDiagram	1
2019-05-02	20:07:50+0100	WRS2	2
2019-05-02	20:07:50+0100	WRTDStidal	1
2019-05-02	20:07:50+0100	XML	4
2019-05-02	20:07:50+0100	abind	3
2019-05-02	20:07:50+0100	adehabitatLT	1
2019-05-02	20:07:50+0100	animation	1
2019-05-02	20:07:50+0100	argparse	1
2019-05-02	20:07:50+0100	arulesViz	1
2019-05-02	20:07:50+0100	askpass	14
2019-05-02	20:07:50+0100	assertthat	8
2019-05-02	20:07:50+0100	aws.s3	2

## 2. Question 2 : To find the top most downloaded package.

```
In [1]:
# Question 02
from pyspark.sql import Row
from pyspark.streaming import StreamingContext
import time

def save(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
            lambda row: Row(time=time, package=row[0], count=row[1])))
        df.write.format("org.apache.spark.sql.cassandra")\
            .options(table="question_02", keyspace="assignment_03")\
            .save(mode="append")
    except:
        pass

ssc = StreamingContext(sc, 5)
lines = ssc.textFileStream("file:///home/siddhi123/Log/")
words = lines.map(lambda line: line.split(", "))
package_download_count = words.map(lambda x: (x[6], 1))
package_download_count = package_download_count.reduceByKey(lambda a,b: a+b)
package_download_count = package_download_count.transform(lambda x: x.sortBy(lambda a: (-a[1])))
package_download_count = package_download_count.transform(lambda x: sc.parallelize(x.take(1)))
package_download_count.pprint()
package_download_count.foreachRDD(save)
ssc.start()
```

(15 rows)

```
cqlsh:assignment_03> select * from question_02;
```

time	package	count
2019-05-02 20:24:55+0100	igraph	48
2019-05-02 20:26:05+0100	Rcpp	16
2019-05-02 20:24:50+0100	data.table	32
2019-05-02 20:25:55+0100	dplyr	18
2019-05-02 20:25:20+0100	magrittr	19
2019-05-02 20:26:00+0100	tidyr	15
2019-05-02 20:25:00+0100	tibble	22
2019-05-02 20:25:35+0100	Rcpp	18
2019-05-02 20:25:15+0100	rlang	19
2019-05-02 20:25:50+0100	rlang	18
2019-05-02 20:25:25+0100	pkgconfig	20
2019-05-02 20:25:10+0100	pillar	15
2019-05-02 20:26:10+0100	tidyr	19
2019-05-02 20:25:45+0100	DAAG	17
2019-05-02 20:25:40+0100	DAAG	21
2019-05-02 20:25:05+0100	data.table	30
2019-05-02 20:25:30+0100	stringr	17

(17 rows)

```
cqlsh:assignment_03> select * from question_02;
```

time	package	count
2019-05-02 20:24:55+0100	igraph	48
2019-05-02 20:26:05+0100	Rcpp	16
2019-05-02 20:24:50+0100	data.table	32
2019-05-02 20:25:55+0100	dplyr	18
2019-05-02 20:25:20+0100	magrittr	19
2019-05-02 20:26:00+0100	tidyr	15
2019-05-02 20:25:00+0100	tibble	22
2019-05-02 20:25:35+0100	Rcpp	18

```

2019-05-02 20:26:50+0100 | psych | 31
2019-05-02 20:26:05+0100 | Rcpp | 16
2019-05-02 20:26:40+0100 | rlang | 35
2019-05-02 20:26:20+0100 | rlang | 18
2019-05-02 20:24:50+0100 | data.table | 32
2019-05-02 20:26:45+0100 | cowplot | 32
2019-05-02 20:28:00+0100 | rlang | 22
2019-05-02 20:27:25+0100 | assertthat | 48
2019-05-02 20:25:55+0100 | dplyr | 18
2019-05-02 20:28:15+0100 | gstat | 24
2019-05-02 20:27:45+0100 | tibble | 66
2019-05-02 20:28:05+0100 | htmlwidgets | 21
2019-05-02 20:25:20+0100 | magrittr | 19
2019-05-02 20:28:35+0100 | rlang | 20
2019-05-02 20:27:15+0100 | rlang | 16
2019-05-02 20:26:00+0100 | tidyr | 15
2019-05-02 20:27:40+0100 | rlang | 45
2019-05-02 20:25:00+0100 | tibble | 22
2019-05-02 20:25:35+0100 | Rcpp | 18
2019-05-02 20:27:10+0100 | tibble | 29
2019-05-02 20:28:40+0100 | RSQLite | 31
2019-05-02 20:26:30+0100 | htmltools | 27
2019-05-02 20:25:15+0100 | rlang | 19
2019-05-02 20:27:50+0100 | utf8 | 55
2019-05-02 20:25:50+0100 | rlang | 18
2019-05-02 20:25:25+0100 | pkgconfig | 20
2019-05-02 20:27:35+0100 | Rcpp | 64
2019-05-02 20:27:55+0100 | purrr | 25
2019-05-02 20:27:05+0100 | gsubfn | 18
2019-05-02 20:28:20+0100 | XML | 17
2019-05-02 20:28:45+0100 | ggplot2 | 31
2019-05-02 20:25:10+0100 | pillar | 15
2019-05-02 20:26:10+0100 | tidyr | 19
2019-05-02 20:27:30+0100 | dplyr | 67
2019-05-02 20:26:25+0100 | rlang | 20
2019-05-02 20:28:25+0100 | XML | 17
2019-05-02 20:25:45+0100 | DAAG | 17
2019-05-02 20:28:30+0100 | XML | 23
2019-05-02 20:25:40+0100 | DAAG | 21
2019-05-02 20:25:05+0100 | data.table | 30
2019-05-02 20:25:30+0100 | stringr | 17
2019-05-02 20:26:15+0100 | rlang | 20
2019-05-02 20:26:55+0100 | webshot | 30
2019-05-02 20:27:00+0100 | scales | 23
2019-05-02 20:26:35+0100 | Rcpp | 15
2019-05-02 20:28:50+0100 | yaml | 35

```

(49 rows)

### 3. Question 3 : To find the top 5 countries along with number of downloads.

```

In [1]: # Question 03
from pyspark.sql import Row
from pyspark.streaming import StreamingContext
import time

def save(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
            lambda row: Row(time=time, country=row[0], count=row[1])))
        df.write.format("org.apache.spark.sql.cassandra")\
            .options(table="question_03", keyspace="assignment_03")\
            .save(mode="append")
    except:
        pass

ssc = StreamingContext(sc, 5)
lines = ssc.textFileStream("file:///home/siddhi123/Log/")
words = lines.map(lambda line: line.split(","))
package_download_count = words.map(lambda x: (x[8], 1))
package_download_count = package_download_count.reduceByKey(lambda a,b: a+b)
package_download_count = package_download_count.transform(lambda x: x.sortBy(lambda a: (-a[1])))
package_download_count = package_download_count.transform(lambda x: sc.parallelize(x.take(5)))
package_download_count.pprint()
package_download_count.foreachRDD(save)

```

```
sqlsh:assignment_03> select * from question_03;
```

time	country	count
2019-05-02 20:37:35+0100	CA	86
2019-05-02 20:37:35+0100	NA	116
2019-05-02 20:37:35+0100	US	333
2019-05-02 20:37:35+0100	IN	92
2019-05-02 20:37:35+0100	US	246
2019-05-02 20:37:30+0100	CA	113
2019-05-02 20:37:30+0100	CN	115
2019-05-02 20:37:30+0100	US	321
2019-05-02 20:37:30+0100	IT	97
2019-05-02 20:37:30+0100	US	405
2019-05-02 20:37:45+0100	CA	138
2019-05-02 20:37:45+0100	NA	123
2019-05-02 20:37:45+0100	US	148
2019-05-02 20:37:45+0100	HK	159
2019-05-02 20:37:45+0100	US	256
2019-05-02 20:37:15+0100	CA	132
2019-05-02 20:37:15+0100	NA	114
2019-05-02 20:37:15+0100	US	342
2019-05-02 20:37:15+0100	IN	94
2019-05-02 20:37:15+0100	IT	202
2019-05-02 20:37:20+0100	CA	127
2019-05-02 20:37:20+0100	CN	146
2019-05-02 20:37:20+0100	ES	116
2019-05-02 20:37:20+0100	US	378
2019-05-02 20:37:20+0100	IT	129
2019-05-02 20:37:25+0100	CA	123
2019-05-02 20:37:25+0100	CN	111
2019-05-02 20:37:25+0100	NA	111
2019-05-02 20:37:25+0100	US	287

2019-05-02	20:45:00+0100	NA	218
2019-05-02	20:45:00+0100	US	521
2019-05-02	20:37:30+0100	CA	113
2019-05-02	20:37:30+0100	CN	115
2019-05-02	20:37:30+0100	US	321
2019-05-02	20:37:30+0100	IT	97
2019-05-02	20:37:30+0100	US	405
2019-05-02	20:44:25+0100	CA	107
2019-05-02	20:44:25+0100	IN	111
2019-05-02	20:44:25+0100	JP	54
2019-05-02	20:44:25+0100	NO	67
2019-05-02	20:44:25+0100	US	223
2019-05-02	20:44:35+0100	CA	96
2019-05-02	20:44:35+0100	HK	87
2019-05-02	20:44:35+0100	NA	116
2019-05-02	20:44:35+0100	US	297
2019-05-02	20:44:35+0100	US	372
2019-05-02	20:45:50+0100	CA	117
2019-05-02	20:45:50+0100	NA	171
2019-05-02	20:45:50+0100	US	348
2019-05-02	20:45:50+0100	FR	165
2019-05-02	20:45:50+0100	US	233
2019-05-02	20:44:40+0100	CA	101
2019-05-02	20:44:40+0100	IN	112
2019-05-02	20:44:40+0100	NA	91
2019-05-02	20:44:40+0100	US	323
2019-05-02	20:44:40+0100	US	493
2019-05-02	20:47:00+0100	CA	99
2019-05-02	20:47:00+0100	NA	131
2019-05-02	20:47:00+0100	US	331
2019-05-02	20:47:00+0100	DO	66
2019-05-02	20:47:00+0100	US	493
2019-05-02	20:46:40+0100	CA	113
2019-05-02	20:46:40+0100	NA	154
2019-05-02	20:46:40+0100	US	450
2019-05-02	20:46:40+0100	NA	54
2019-05-02	20:46:40+0100	US	584
2019-05-02	20:45:20+0100	CN	107
2019-05-02	20:45:20+0100	NA	124
2019-05-02	20:45:20+0100	US	393
2019-05-02	20:45:20+0100	FR	106
2019-05-02	20:45:20+0100	US	367
2019-05-02	20:43:50+0100	CA	92
2019-05-02	20:43:50+0100	NA	88
2019-05-02	20:43:50+0100	US	589
2019-05-02	20:43:50+0100	NA	87
2019-05-02	20:43:50+0100	US	546

--MORE--

Question 4 : To find number of downloads for ggplot2 package.

```
In [1]: #Question 4
from pyspark.sql import Row
from pyspark.streaming import StreamingContext
import time

def save(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
            lambda row: Row(time=time, package=row[0], count=row[1])))
        df.write.format("org.apache.spark.sql.cassandra")\
            .options(table="question_04", keyspace="assignment_03")\
            .save(mode="append")
    except:
        pass

ssc = StreamingContext(sc, 5)
lines = ssc.textFileStream("file:///home/siddhi123/Log/")
words = lines.map(lambda line: line.split(","))
package_download_count = words.filter(lambda x: "ggplot2" in x)
package_download_count = package_download_count.map(lambda x: (x[6], 1))
package_download_count = package_download_count.reduceByKey(lambda a,b: a+b)
package_download_count.pprint()
package_download_count.foreachRDD(save)
ssc.start()
```

```
cqlsh:assignment_03> select * from question_04;
```

time	package	count
2019-05-02 20:50:10+0100	ggplot2	12
2019-05-02 20:50:00+0100	ggplot2	14
2019-05-02 20:49:55+0100	ggplot2	18
2019-05-02 20:49:45+0100	ggplot2	6
2019-05-02 20:49:50+0100	ggplot2	10
2019-05-02 20:49:40+0100	ggplot2	14
2019-05-02 20:50:05+0100	ggplot2	13

(7 rows)

```
cqlsh:assignment_03> select * from question_04;
```

time	package	count
2019-05-02 20:51:20+0100	ggplot2	10
2019-05-02 20:50:20+0100	ggplot2	16
2019-05-02 20:50:15+0100	ggplot2	14
2019-05-02 20:51:05+0100	ggplot2	8
2019-05-02 20:50:10+0100	ggplot2	12
2019-05-02 20:51:10+0100	ggplot2	7
2019-05-02 20:50:40+0100	ggplot2	5
2019-05-02 20:50:00+0100	ggplot2	14
2019-05-02 20:50:45+0100	ggplot2	11
2019-05-02 20:49:55+0100	ggplot2	18
2019-05-02 20:51:50+0100	ggplot2	12
2019-05-02 20:51:45+0100	ggplot2	12
2019-05-02 20:51:00+0100	ggplot2	5
2019-05-02 20:51:25+0100	ggplot2	9
2019-05-02 20:51:30+0100	ggplot2	6
2019-05-02 20:49:45+0100	ggplot2	6
2019-05-02 20:50:55+0100	ggplot2	12
2019-05-02 20:51:35+0100	ggplot2	9
2019-05-02 20:51:15+0100	ggplot2	10
2019-05-02 20:50:35+0100	ggplot2	14
2019-05-02 20:51:55+0100	ggplot2	20
2019-05-02 20:50:25+0100	ggplot2	9
2019-05-02 20:49:50+0100	ggplot2	10
2019-05-02 20:50:30+0100	ggplot2	11
2019-05-02 20:49:40+0100	ggplot2	14
2019-05-02 20:50:05+0100	ggplot2	13
2019-05-02 20:50:50+0100	ggplot2	9
2019-05-02 20:51:40+0100	ggplot2	8

(28 rows)

```
cqlsh:assignment_03>
```



time	package	count
2019-05-02 20:52:50+0100	ggplot2	8
2019-05-02 20:51:20+0100	ggplot2	10
2019-05-02 20:50:20+0100	ggplot2	16
2019-05-02 20:50:15+0100	ggplot2	14
2019-05-02 20:51:05+0100	ggplot2	8
2019-05-02 20:50:10+0100	ggplot2	12
2019-05-02 20:51:10+0100	ggplot2	7
2019-05-02 20:50:40+0100	ggplot2	5
2019-05-02 20:50:00+0100	ggplot2	14
2019-05-02 20:50:45+0100	ggplot2	11
2019-05-02 20:52:20+0100	ggplot2	7
2019-05-02 20:52:35+0100	ggplot2	5
2019-05-02 20:49:55+0100	ggplot2	18
2019-05-02 20:51:50+0100	ggplot2	12
2019-05-02 20:52:25+0100	ggplot2	5
2019-05-02 20:52:00+0100	ggplot2	11
2019-05-02 20:52:05+0100	ggplot2	11
2019-05-02 20:51:45+0100	ggplot2	12
2019-05-02 20:51:00+0100	ggplot2	5
2019-05-02 20:51:25+0100	ggplot2	9
2019-05-02 20:51:30+0100	ggplot2	6
2019-05-02 20:52:15+0100	ggplot2	8
2019-05-02 20:53:10+0100	ggplot2	3
2019-05-02 20:49:45+0100	ggplot2	6
2019-05-02 20:53:05+0100	ggplot2	7
2019-05-02 20:50:55+0100	ggplot2	12
2019-05-02 20:51:35+0100	ggplot2	9
2019-05-02 20:53:00+0100	ggplot2	7
2019-05-02 20:51:15+0100	ggplot2	10
2019-05-02 20:50:35+0100	ggplot2	14
2019-05-02 20:52:30+0100	ggplot2	2
2019-05-02 20:51:55+0100	ggplot2	20
2019-05-02 20:52:10+0100	ggplot2	8
2019-05-02 20:50:25+0100	ggplot2	9
2019-05-02 20:49:50+0100	ggplot2	10
2019-05-02 20:50:30+0100	ggplot2	11
2019-05-02 20:52:55+0100	ggplot2	4
2019-05-02 20:49:40+0100	ggplot2	14
2019-05-02 20:50:05+0100	ggplot2	13
2019-05-02 20:50:50+0100	ggplot2	9
2019-05-02 20:51:40+0100	ggplot2	8
2019-05-02 20:52:40+0100	ggplot2	7
2019-05-02 20:52:45+0100	ggplot2	9

(43 rows)