

cind110_Assignment_03

Brendan Dagys

March 10, 2018

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

This assignment makes use of data that is provided by:

["https://raw.githubusercontent.com/ansymo/msr2013-bug_dataset/master/data/v02/mozilla/bug_status.json"](https://raw.githubusercontent.com/ansymo/msr2013-bug_dataset/master/data/v02/mozilla/bug_status.json)

Installing rmongodb from Archive

```
# Download package tarball from CRAN archive

#url <- "https://cran.r-project.org/src/contrib/Archive/rmongodb/rmongodb_1.8.0.tar.gz"
#pkgFile <- "rmongodb_1.8.0.tar.gz"
#download.file(url = url, destfile = pkgFile)

# Install dependencies

#install.packages(c("plyr", "jsonlite"))

# Install package
#install.packages(pkgs=pkgFile, type="source", repos=NULL)
```

Helper functions to connect to MongoDB Server

```
library(rmongodb)
DATABASE <- "Mozilla_DB"
NS <- "Mozilla_DB.posts"
mongo_find <- function (query,sort,fields) {
  if(missing(sort)) {
    sort <- mongo.bson.from.JSON('{}')
  }
  if(missing(fields)) {
    fields <- mongo.bson.from.JSON('{}')
  }
}
```

```

# creating DB connection object with rmongodb package
m1 <- mongo.create(db=DATABASE)
#For more details, type below command on shell:
#help(mongo.create)

# send Query to btain the result set
cursor <- mongo.find(m1, NS, query,sort, fields ,limit=10L)

# print elements from result sets
while (mongo.cursor.next(cursor))
  print(mongo.cursor.value(cursor))

# close db connection, saving output in x to avoid printing the
disconnect message
x <- mongo.disconnect(m1)
}

mongo_find_count <- function (query) {

  # creating DB connection object with rmongodb package
  m1 <- mongo.create(db=DATABASE)
  #For more details, type below command on shell:
  #help(mongo.create)

  # send Query to btain result set
  print(mongo.count(m1, NS, query))

  # close db connection, saving output in x to avoid printing the
  disconnect message
  x <- mongo.disconnect(m1)
}

```

Example_1

show databases;

```

#creating connection
m1 <- mongo.create(db=DATABASE)

mongo.get.databases(m1)

## [1] "Mozilla_DB" "E_DB"

#For more details, type below command on shell:
#help(mongo.get.databases)

#disconnecting
disconnect_msg <- mongo.disconnect(m1)

```

Example_2

find one object (i.e one bug report) that exists in 'Mozilla_DB.posts' collection

```
#creating connection
m1 <- mongo.create(db=DATABASE)
mongo.find.one(m1,NS)

## _id : 7      5ab1198ec04a301062b50368
## value : 4
##    0 : 3
##      what : 2      UNCONFIRMED
##      who  : 16      111157
##      when : 16      1111425133
##
##    1 : 3
##      what : 2      RESOLVED
##      who  : 16      111157
##      when : 16      1136823088
##
##    2 : 3
##      what : 2      UNCONFIRMED
##      who  : 16      188489
##      when : 16      1136823115
##
##    3 : 3
##      what : 2      RESOLVED
##      who  : 16      188489
##      when : 16      1136823131
##
##
## b_id : 2      287144

#For more details, type below command on shell:
#help(mongo.find.one)

#disconnecting
disconnect_msg <- mongo.disconnect(m1)
```

Example_3

Find the bug report where its id equals to 501091 (i.e. b_id = 501091)

```
query <- mongo.bson.from.JSON('{ "b_id": "501091" }')

mongo_find(query)

## _id : 7      5ab1198ec04a301062b5038f
## value : 4
```

```
##      0 : 3
##          what : 2      NEW
##          who  : 16     5490
##          when : 16     1246241902
##
##      1 : 3
##          what : 2      RESOLVED
##          who  : 16     5490
##          when : 16     1246328569
##
##
## b_id : 2      501091
```

Q1

Count the total number of objects (i.e. bug reports) in “Mozilla_DB.post” collection.

```
#creating connection
m1 <- mongo.create(db=DATABASE)

#Write your solution here.
#Help: use count function
#For more details, type below command on shell:
#help(mongo.count)
mongo.count(m1, NS)

## [1] 394878

#disconnecting
disconnect_msg <- mongo.disconnect(m1)
```

Q2

Display a list of all distinct values of the “what” attribute in “Mozilla_DB.posts” collection.

```
#creating connection
m1 <- mongo.create(db=DATABASE)

#Write your solution here.
#Help: use distinct function
#For more details, type below command on shell:
#help(mongo.distinct)
mongo.distinct(m1, NS, "value.what")

## [1] "RESOLVED"      "UNCONFIRMED" "NEW"          "VERIFIED"
      "REOPENED"
## [6] "ASSIGNED"      "CLOSED"      ""
```

```
#disconnecting  
disconnect_msg <- mongo.disconnect(m1)
```

Q3

How many objects in “Mozilla_DB.posts” collection have a status equal to “CLOSED”?
Hint: Status here is the “what” attribute

```
#Update below query  
query <- mongo.bson.from.JSON('{"value.what": "CLOSED"}')  
  
mongo_find_count(query)  
  
## [1] 928
```

Q4

Display a list of all bug reports where the timestamp (i.e. the “when” attribute) is greater than 1250739410.

```
#Update below query  
#query <- mongo.bson.from.JSON('{"value.when": {"$gt": 1250739410}}')  
  
#QUERY WHERE ALL ELEMENTS MATCH THE CONDITION:  
query <- mongo.bson.from.JSON('{"$nor": [{"value.when": {"$lte":  
1250739410}}]}')  
  
mongo_find(query)  
  
## _id : 7      5ab1198ec04a301062b50378  
## value : 4  
##    0 : 3  
##      what : 2      NEW  
##      who  : 16     359004  
##      when : 16     1343389311  
##  
##  
## b_id : 2      778284  
## _id : 7      5ab1198ec04a301062b5037a  
## value : 4  
##    0 : 3  
##      what : 2      NEW  
##      who  : 16     417288  
##      when : 16     1343389258  
##  
##    1 : 3  
##      what : 2      ASSIGNED  
##      who  : 16     417288
```

```
##          when : 16      1343813635
##
##      2 : 3
##          what : 2      RESOLVED
##          who  : 16      75935
##          when : 16      1343842875
##
##
## b_id : 2      778282
## _id : 7      5ab1198ec04a301062b5037c
## value : 4
##      0 : 3
##          what : 2      NEW
##          who  : 16      17687
##          when : 16      1321183617
##
##
## b_id : 2      702142
## _id : 7      5ab1198ec04a301062b50381
## value : 4
##      0 : 3
##          what : 2      NEW
##          who  : 16      5038
##          when : 16      1298293370
##
##      1 : 3
##          what : 2      RESOLVED
##          who  : 16      380838
##          when : 16      1298294473
##
##
## b_id : 2      635779
## _id : 7      5ab1198ec04a301062b50382
## value : 4
##      0 : 3
##          what : 2      NEW
##          who  : 16      122472
##          when : 16      1267014431
##
##
## b_id : 2      548405
## _id : 7      5ab1198ec04a301062b50383
## value : 4
##      0 : 3
##          what : 2      UNCONFIRMED
##          who  : 16      407213
##          when : 16      1298292291
##
##      1 : 3
##          what : 2      RESOLVED
```

```
##           who : 16      29552
##           when : 16     1298296167
##
##
## b_id : 2      635775
## _id : 7      5ab1198ec04a301062b50384
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who : 16      407032
##       when : 16     1298290470
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      51298
##       when : 16     1335697158
##
##
## b_id : 2      635772
## _id : 7      5ab1198ec04a301062b50387
## value : 4
##   0 : 3
##       what : 2      NEW
##       who : 16      7044
##       when : 16     1302614847
##
##
## b_id : 2      649511
## _id : 7      5ab1198ec04a301062b50385
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who : 16      407212
##       when : 16     1298291296
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      20209
##       when : 16     1298292297
##
##
## b_id : 2      635773
## _id : 7      5ab1198ec04a301062b50388
## value : 4
##   0 : 3
##       what : 2      NEW
##       who : 16      135518
##       when : 16     1325603524
##
```

```
##
## b_id : 2      715014
```

Q5

Find documents where attributes “when” (time-stamp) is greater than 1268734163 OR “what” (bug status) is CLOSED OR the ID of the developer is less then or equal to 310130

#Update below query

```
query <- mongo.bson.from.JSON('{ "$or": [{"value.when": {"$gt": 1268734163}}, {"value.what": "CLOSED"}, {"value.who": {"$lte": 310130}}] }')
```

```
mongo_find(query)
```

```
## _id : 7      5ab1198ec04a301062b50368
## value : 4
## 0 : 3
## what : 2      UNCONFIRMED
## who : 16      111157
## when : 16     1111425133
##
## 1 : 3
## what : 2      RESOLVED
## who : 16      111157
## when : 16     1136823088
##
## 2 : 3
## what : 2      UNCONFIRMED
## who : 16      188489
## when : 16     1136823115
##
## 3 : 3
## what : 2      RESOLVED
## who : 16      188489
## when : 16     1136823131
##
##
## b_id : 2      287144
## _id : 7      5ab1198ec04a301062b50369
## value : 4
## 0 : 3
## what : 2      UNCONFIRMED
## who : 16      277548
## when : 16     1177312569
##
## 1 : 3
## what : 2      RESOLVED
```



```
##          who : 16      29811
##          when : 16     1244276193
##
##
## b_id : 2      378467
## _id : 7      5ab1198ec04a301062b5036a
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who : 16      191715
##       when : 16     1111426531
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      39771
##       when : 16     1122963644
##
##
## b_id : 2      287146
## _id : 7      5ab1198ec04a301062b5036b
## value : 4
##   0 : 3
##       what : 2      NEW
##       who : 16      10520
##       when : 16     1177311639
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      10520
##       when : 16     1177396999
##
##
## b_id : 2      378465
## _id : 7      5ab1198ec04a301062b5036c
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who : 16      102991
##       when : 16     1177311821
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      4410
##       when : 16     1177311973
##
##   2 : 3
##       what : 2      VERIFIED
##       who : 16      76551
##       when : 16     1177406906
##
```

```
##
## b_id : 2      378466
## _id : 7      5ab1198ec04a301062b5036d
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who  : 16     191709
##       when : 16     1111425325
##
##   1 : 3
##       what : 2      RESOLVED
##       who  : 16     120380
##       when : 16     1191595993
##
##
## b_id : 2      287145
## _id : 7      5ab1198ec04a301062b5036e
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who  : 16     175240
##       when : 16     1103175907
##
##   1 : 3
##       what : 2      RESOLVED
##       who  : 16     57557
##       when : 16     1103178199
##
##   2 : 3
##       what : 2      VERIFIED
##       who  : 16     103593
##       when : 16     1103179125
##
##
## b_id : 2      274909
## _id : 7      5ab1198ec04a301062b5036f
## value : 4
##   0 : 3
##       what : 2      NEW
##       who  : 16     10520
##       when : 16     1177312844
##
##   1 : 3
##       what : 2      RESOLVED
##       who  : 16     88369
##       when : 16     1183141492
##
##   2 : 3
##       what : 2      REOPENED
##       who  : 16     88369
```

```

##           when : 16      1183414258
##
##      3 : 3
##           what : 2      RESOLVED
##           who  : 16      130269
##           when : 16      1183515504
##
##
## b_id : 2      378468
## _id : 7      5ab1198ec04a301062b50370
## value : 4
##      0 : 3
##           what : 2      NEW
##           who  : 16      39022
##           when : 16      1177313127
##
##      1 : 3
##           what : 2      RESOLVED
##           who  : 16      39022
##           when : 16      1177313592
##
##
## b_id : 2      378469
## _id : 7      5ab1198ec04a301062b50371
## value : 4
##      0 : 3
##           what : 2      UNCONFIRMED
##           who  : 16      27276
##           when : 16      994301386
##
##      1 : 3
##           what : 2      RESOLVED
##           who  : 16      25000
##           when : 16      994305787
##
##      2 : 3
##           what : 2      VERIFIED
##           who  : 16      29264
##           when : 16      994688270
##
##
## b_id : 2      89370

```

Q6

Find all bug reports that has not been CLOSED

#Update below query

```
query <- mongo.bson.from.JSON('{"value.what": {"$ne": "CLOSED"}}')
```

mongo_find(query)

```
## _id : 7      5ab1198ec04a301062b50368
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who  : 16      111157
##       when : 16      1111425133
##
##   1 : 3
##       what : 2      RESOLVED
##       who  : 16      111157
##       when : 16      1136823088
##
##   2 : 3
##       what : 2      UNCONFIRMED
##       who  : 16      188489
##       when : 16      1136823115
##
##   3 : 3
##       what : 2      RESOLVED
##       who  : 16      188489
##       when : 16      1136823131
##
##
## b_id : 2      287144
## _id : 7      5ab1198ec04a301062b50369
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who  : 16      277548
##       when : 16      1177312569
##
##   1 : 3
##       what : 2      RESOLVED
##       who  : 16      29811
##       when : 16      1244276193
##
##
## b_id : 2      378467
## _id : 7      5ab1198ec04a301062b5036a
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who  : 16      191715
##       when : 16      1111426531
##
##   1 : 3
##       what : 2      RESOLVED
```

```
##          who : 16      39771
##          when : 16     1122963644
##
##
## b_id : 2      287146
## _id : 7      5ab1198ec04a301062b5036b
## value : 4
##   0 : 3
##       what : 2      NEW
##       who : 16      10520
##       when : 16     1177311639
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      10520
##       when : 16     1177396999
##
##
## b_id : 2      378465
## _id : 7      5ab1198ec04a301062b5036c
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who : 16      102991
##       when : 16     1177311821
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      4410
##       when : 16     1177311973
##
##   2 : 3
##       what : 2      VERIFIED
##       who : 16      76551
##       when : 16     1177406906
##
##
## b_id : 2      378466
## _id : 7      5ab1198ec04a301062b5036d
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who : 16      191709
##       when : 16     1111425325
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16      120380
##       when : 16     1191595993
##
```

```
##
## b_id : 2      287145
## _id : 7      5ab1198ec04a301062b5036e
## value : 4
##   0 : 3
##       what : 2      UNCONFIRMED
##       who : 16     175240
##       when : 16    1103175907
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16     57557
##       when : 16    1103178199
##
##   2 : 3
##       what : 2      VERIFIED
##       who : 16     103593
##       when : 16    1103179125
##
##
## b_id : 2      274909
## _id : 7      5ab1198ec04a301062b5036f
## value : 4
##   0 : 3
##       what : 2      NEW
##       who : 16     10520
##       when : 16    1177312844
##
##   1 : 3
##       what : 2      RESOLVED
##       who : 16     88369
##       when : 16    1183141492
##
##   2 : 3
##       what : 2      REOPENED
##       who : 16     88369
##       when : 16    1183414258
##
##   3 : 3
##       what : 2      RESOLVED
##       who : 16     130269
##       when : 16    1183515504
##
##
## b_id : 2      378468
## _id : 7      5ab1198ec04a301062b50370
## value : 4
##   0 : 3
##       what : 2      NEW
##       who : 16     39022
```

```
##           when : 16      1177313127
##
##      1 : 3
##           what : 2      RESOLVED
##           who  : 16      39022
##           when : 16      1177313592
##
##
## b_id : 2      378469
## _id : 7      5ab1198ec04a301062b50371
## value : 4
##      0 : 3
##           what : 2      UNCONFIRMED
##           who  : 16      27276
##           when : 16      994301386
##
##      1 : 3
##           what : 2      RESOLVED
##           who  : 16      25000
##           when : 16      994305787
##
##      2 : 3
##           what : 2      VERIFIED
##           who  : 16      29264
##           when : 16      994688270
##
##
## b_id : 2      89370
```

Q7

Fetch only the ID of the bugs that have been RESOLVED.

```
#Update below query and fields
query <- mongo.bson.from.JSON('{"value.what": "RESOLVED"}')

fields <- mongo.bson.from.JSON('{"b_id": 1, "_id": 0}')

mongo_find(query, fields = fields)

## b_id : 2      287144
## b_id : 2      378467
## b_id : 2      287146
## b_id : 2      378465
## b_id : 2      378466
## b_id : 2      287145
## b_id : 2      274909
## b_id : 2      378468
```

```
## b_id : 2      378469
## b_id : 2      89370
```

Q8

Insert the following document into Mozilla_DB.posts collection { "b_id" : "1111111", "value" : [{ "what" : "NEW", "who" : 6254, "when" : 1250739410 }] }

```
#creating connection
m1 <- mongo.create(db=DATABASE)

#Write your solution here.
#update below insert statement
mongo.insert(m1, NS, '{"b_id": "1111111", "value": [{"what": "NEW",
"who": 6254, "when": 1250739410}]}')

## [1] TRUE

#check if inserted
query <- mongo.bson.from.JSON('{"b_id": "1111111"}')

mongo_find(query)

## _id : 7      5aba60ed307fe89b20f74313
## b_id : 2      1111111
## value : 4
##    0 : 3
##      what : 2      NEW
##      who  : 16     6254
##      when : 16     1250739410

#disconnecting
disconnect_msg <- mongo.disconnect(m1)
```

Q9

Remove all documents with "b_id" equals to "1111111".

```
#creating connection
m1 <- mongo.create(db=DATABASE)

#Write your solution here.
#update below insert statement
mongo.remove(m1, NS, '{"b_id": "1111111"}')

## [1] TRUE

#check if deleted
query <- mongo.bson.from.JSON('{"b_id": "1111111"}')
```



```

mongo_find(query)
#disconnecting
disconnect_msg <- mongo.disconnect(m1)

```

Q10

Update all Documents with b_id equals to "501091" as below.

BEFORE:

```

{ value : 4
0 : 3
what : 2 NEW who : 16 5490 when : 16 1246241902 1 : 3
what : 2 RESOLVED who : 16 5490 when : 16 1246328569 b_id : 2 501091 }

```

AFTER:

```

{"b_id": "501091", "value": [ {"what": "NEW", "who": 6254, "when": 1250739410
}]}

```

```

#creating connection
m1 <- mongo.create(db=DATABASE)

#Write your solution here.
#update below insert statement
mongo.update(m1,NS, '{"b_id": "501091"}', '{"b_id": "501091", "value":
[{"what": "NEW", "who": 6254, "when": 1250739410}]}')

## [1] TRUE

#check if updated
query <- mongo.bson.from.JSON('{"b_id": "501091"}')

mongo_find(query)

## _id : 7      5ab1198ec04a301062b5038f
## b_id : 2      501091
## value : 4
##      0 : 3
##      what : 2      NEW
##      who : 16      6254
##      when : 16      1250739410

#disconnecting
disconnect_msg <- mongo.disconnect(m1)

```