



M.Sc. in Business Analytics
Academic Year 2019-2020

Course: Big Data Systems

Professor: S.Safras

Mallios Xaralampos p2821912

Patrikalos Vangelis p2821917

Part A Redis

Loading the data

```
install.packages("redux")  
library("redux")  
modified_list <- read.csv(file.choose(),header = TRUE, sep=",")  
emails_sent<-read.csv(file.choose(),header = TRUE, sep=",")
```

Creating a connection to the local instance of REDIS

```
r <- redux::hiredis(  
  redux::redis_config(  
    host = "127.0.0.1",  
    port = "6379"))
```

```
January <- subset(modified_list,MonthID ==1 ,select=c(UserID, ModifiedListing))  
February <- subset(modified_list,MonthID ==2 ,select=c(UserID, ModifiedListing))  
March <- subset(modified_list,MonthID ==3 ,select=c(UserID, ModifiedListing))
```

Task 1.1 How many users modified their listing on January?

In order to calculate the number of users that modified their listing on January, I created a new dataset containing the data for January only and then created a bitmap for the modifications.

```
for (i in 1:nrow(January)){
r$SETBIT("ModificationsJanuary",January[i,1],January[i,2])
}
r$BITCOUNT("ModificationsJanuary")
```

9969 users modified their listings on January.

Task 1.2 How many users did NOT modify their listing on January?

```
r$BITOP("NOT","JanuaryNotModified","ModificationsJanuary")
r$BITCOUNT("JanuaryNotModified")
```

I begun by negating the bits of the January modifications so that now the 1 represents the users that did not make a modifications and counted the “1” bits. 1031 users did not modify their listing.

Task 1.3 How many users received at least one e-mail per month ? (at least one e-mail in January and at least one e-mail in February and at least one e-mail in March)?

```
library("dplyr")
emails <- emails_sent[,2:4] %>% distinct(UserID,MonthID, .keep_all = TRUE)
EmailsJanuary <- subset(emails,MonthID ==1)
EmailsFebruary <- subset(emails,MonthID ==2)
EmailsMarch <- subset(emails,MonthID ==3)
#Now inserting data in Redis
for (i in 1:nrow(EmailsJanuary)){
  r$SETBIT("EmailsSentJanuary",EmailsJanuary$UserID[i],"1")
}

for (i in 1:nrow(EmailsFebruary)){
  r$SETBIT("EmailsSentFebruary",EmailsFebruary$UserID[i],"1")
}

for (i in 1:nrow(EmailsMarch)){
  r$SETBIT("EmailsSentMarch",EmailsMarch$UserID[i],"1")
}

r$BITOP("AND","results13",c("EmailsSentJanuary","EmailsSentFebruary","EmailsSentMarch"))
r$BITCOUNT("results13")
```

First I got the distinct values per user and Month and after subsetting per month I put an “1” for each line since that signifies that the user received an email regardless of opening it. Then after inserting the data for each month on redis, I used the BITOP “and” command to see the users that received at least an email on

each of the three months and after counting them we can see that there were 2668 users.

Task 1.4 How many users received an e-mail on January and March but NOT on February?

```
r$BITOP("NOT","InvertedEmailsSentFebruary","EmailsSentFebruary")
r$BITOP("AND","results14",c("EmailsSentJanuary","InvertedEmailsSentFebruary","EmailsSentMarch"))
r$BITCOUNT("results14")
```

I inverted the bits of February and then I used the BITOP “AND” to count the cases where the user received a message on January and March but not on April. These users were 2417.

Task 1.5 How many users received an e-mail on January that they did not open but they updated their listing anyway?

Aggregating the January mails to get a table with the users that didn’t open any of the emails they received. This way even if a user opened only one e-mail we will be able to classify him as a user that opened the e-mail.

```
EmailsJanuary_agg <- subset(emails_sent,MonthID ==1 ,select=c(UserID,
EmailOpened))
EmailsJanuary_agg <- aggregate(EmailsJanuary$EmailOpened, by=list(UserID =
EmailsJanuary$UserID), FUN=sum)
EmailsJanuary_agg$x <- if_else(EmailsJanuary_agg$x ==0,0,1)
```

Loading the data in Redis

```
for (i in 1:nrow(EmailsJanuary)){

r$SETBIT("EmailsOpenedJanuary",EmailsJanuary_agg$UserID[i],EmailsJanuary_agg$x[i])
}
```

Inversing the emails opened to count the cases where no mail was opened but modification was made.

```
r$BITOP("NOT","EmailsNotOpenedJanuary","EmailsOpenedJanuary")
r$BITOP("AND","results15",c("EmailsNotOpenedJanuary","ModificationsJanuary"))
r$BITCOUNT("results15")
```

7577 users updated their list without opening the e-mail they received on January.

Task 1.6 How many users received an e-mail on January that they did not open but they updated their listing anyway on January OR they received an e-mail on February that they did not open but they updated their listing anyway on February OR they received an e-mail on March that they did not open but they updated their listing anyway on March?

At this point we will have to make the same thing we did on 1.5 for February and March as well. Then we will count the users with a bitop "or" on the results bitmaps of each month.

```
EmailsFebruary_agg <- subset(emails_sent,MonthID ==2 ,select=c(UserID,
EmailOpened))
EmailsFebruary_agg <- aggregate(EmailsFebruary$EmailOpened, by=list(UserID =
EmailsFebruary$UserID), FUN=sum)
EmailsFebruary_agg$x <- if_else(EmailsFebruary_agg$x ==0,0,1)
EmailsMarch_agg <- subset(emails_sent,MonthID ==3 ,select=c(UserID,
EmailOpened))
EmailsMarch_agg <- aggregate(EmailsMarch$EmailOpened, by=list(UserID =
EmailsMarch$UserID), FUN=sum)
EmailsMarch_agg$x <- if_else(EmailsMarch_agg$x ==0,0,1)

for (i in 1:nrow(February)){
  r$SETBIT("ModificationsFebruary",February$UserID[i],February$ModifiedListing[i])
}
for (i in 1:nrow(EmailsFebruary_agg)){

r$SETBIT("EmailsOpenedFebruary",EmailsFebruary_agg$UserID[i],EmailsFebruary_a
gg$x[i])
}
r$BITOP("NOT","EmailsNotOpenedFebruary","EmailsOpenedFebruary")
r$BITOP("AND","results161",c("EmailsNotOpenedFebruary","ModificationsFebruary
"))
r$BITCOUNT("results161")

for (i in 1:nrow(March)){
  r$SETBIT("ModificationsMarch",March$UserID[i],March$ModifiedListing[i])
}
for (i in 1:nrow(EmailsMarch_agg)){

r$SETBIT("EmailsOpenedMarch",EmailsMarch_agg$UserID[i],EmailsMarch_agg$x[i])
}
r$BITOP("NOT","EmailsNotOpenedMarch","EmailsOpenedMarch")
r$BITOP("AND","results163",c("EmailsNotOpenedMarch","ModificationsMarch"))
r$BITCOUNT("results163")
```

```
r$BITOP("OR","16final",c("results15","results161","results163"))
r$BITCOUNT("16final")
```

15152 users updated their listing without opening the e-mail they received on at least 1 of the months: January, February and March.

Task 1.7 Does it make any sense to keep sending e-mails with recommendations to sellers? Does this strategy really work? How would you describe this in terms a business person would understand?

```
r$BITOP("AND","OpenedModJan",c("EmailsOpenedJanuary","ModificationsJanuary"))
r$BITOP("AND","OpenedModFeb",c("EmailsOpenedFebruary","ModificationsFebruary"))
r$BITOP("AND","OpenedModMarch",c("EmailsOpenedMarch","ModificationsMarch"))
```

```
r$BITCOUNT("OpenedModJan")
r$BITCOUNT("OpenedModFeb")
r$BITCOUNT("OpenedModMarch")
```

After calculating the percentage that proceeds to a modification to their list after receiving an e-mail we can see that the percentage that proceeds to a modifications after reading the received e-mail is too low.

It seems that this strategy is not that effective .

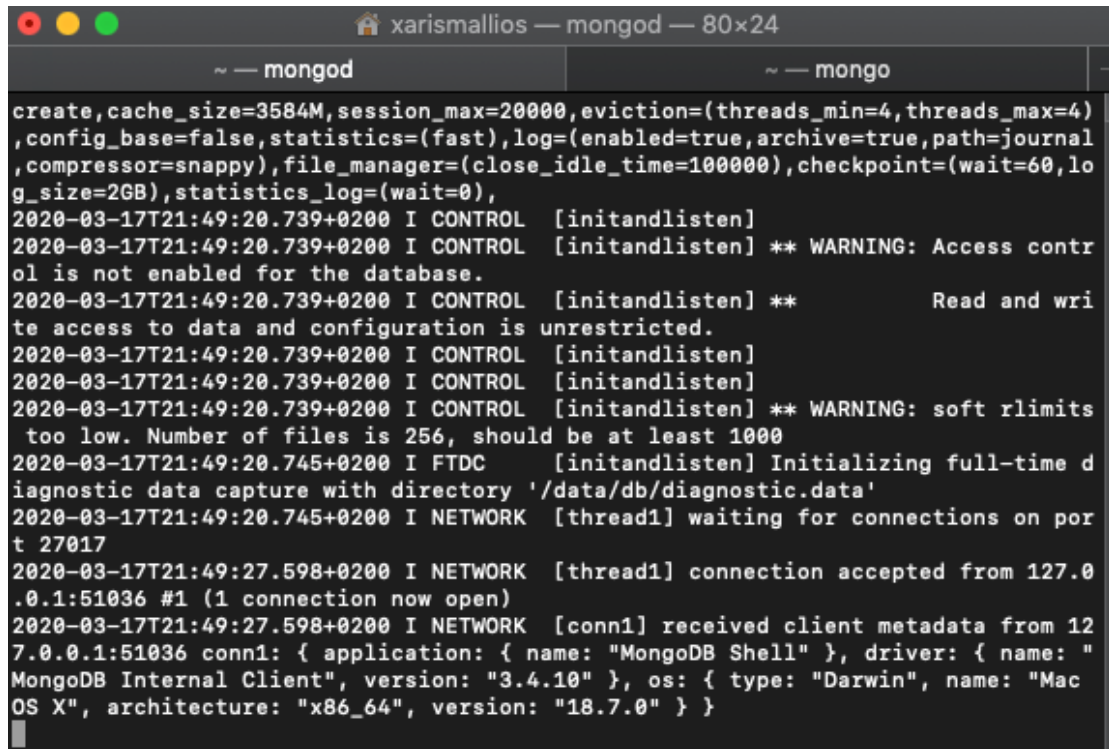
Below we can see a table with the modifications that took place after a user read the sent e-mail in comparison to the total number of sent e-mails.

For January:	24.87%	(2392/9,617)
For February:	25.64%	(2479/9,666)
For March:	24.74%	(2356/9,520)

Part B MongoDB

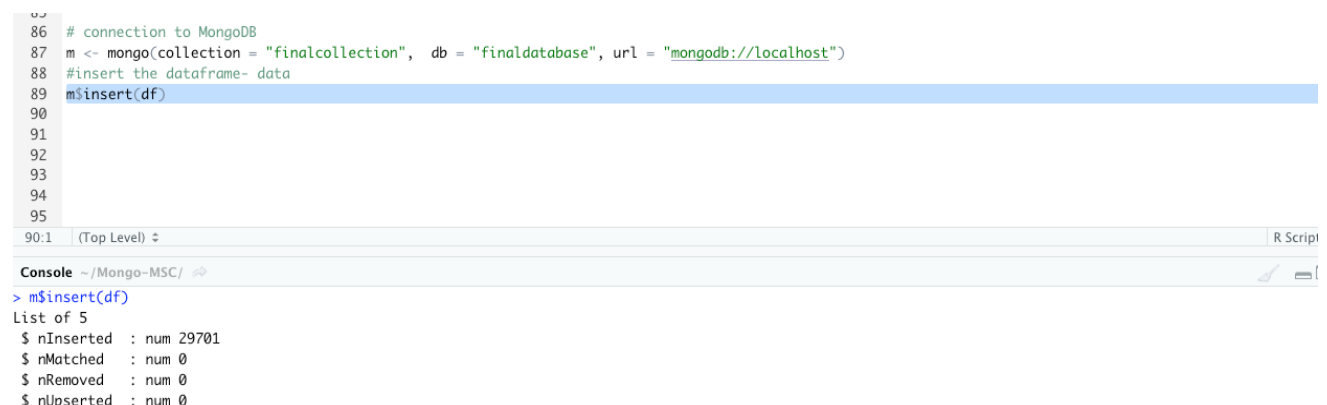
Task 2.1 Add your data to MongoDB

As a first step we initialized mongo dB server and mongo in order to able to connect with R and send the data to localhost database.



```
create,cache_size=3584M,session_max=20000,eviction=(threads_min=4,threads_max=4)
,config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal
,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,lo
g_size=2GB),statistics_log=(wait=0),
2020-03-17T21:49:20.739+0200 I CONTROL [initandlisten]
2020-03-17T21:49:20.739+0200 I CONTROL [initandlisten] ** WARNING: Access contr
ol is not enabled for the database.
2020-03-17T21:49:20.739+0200 I CONTROL [initandlisten] **          Read and wri
te access to data and configuration is unrestricted.
2020-03-17T21:49:20.739+0200 I CONTROL [initandlisten]
2020-03-17T21:49:20.739+0200 I CONTROL [initandlisten]
2020-03-17T21:49:20.739+0200 I CONTROL [initandlisten] ** WARNING: soft rlimits
too low. Number of files is 256, should be at least 1000
2020-03-17T21:49:20.745+0200 I FTDC [initandlisten] Initializing full-time d
iagnostic data capture with directory '/data/db/diagnostic.data'
2020-03-17T21:49:20.745+0200 I NETWORK [thread1] waiting for connections on por
t 27017
2020-03-17T21:49:27.598+0200 I NETWORK [thread1] connection accepted from 127.0
.0.1:51036 #1 (1 connection now open)
2020-03-17T21:49:27.598+0200 I NETWORK [conn1] received client metadata from 12
7.0.0.1:51036 conn1: { application: { name: "MongoDB Shell" }, driver: { name: "
MongoDB Internal Client", version: "3.4.10" }, os: { type: "Darwin", name: "Mac
OS X", architecture: "x86_64", version: "18.7.0" } }
```

After some cleaning, keeping only numbers on Price and Mileage and filling the NULL values with NAs we have sent the data into our database named finaldatabase and finalcollection.

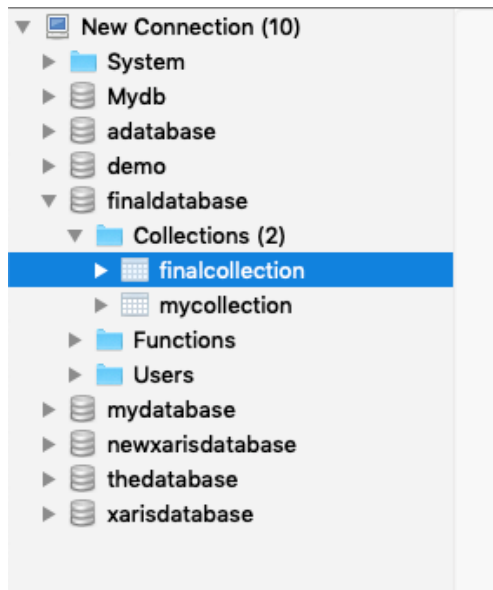


```
86 # connection to MongoDB
87 m <- mongo(collection = "finalcollection", db = "finaldatabase", url = "mongodb://localhost")
88 #insert the dataframe- data
89 m$insert(df)
90
91
92
93
94
95
```

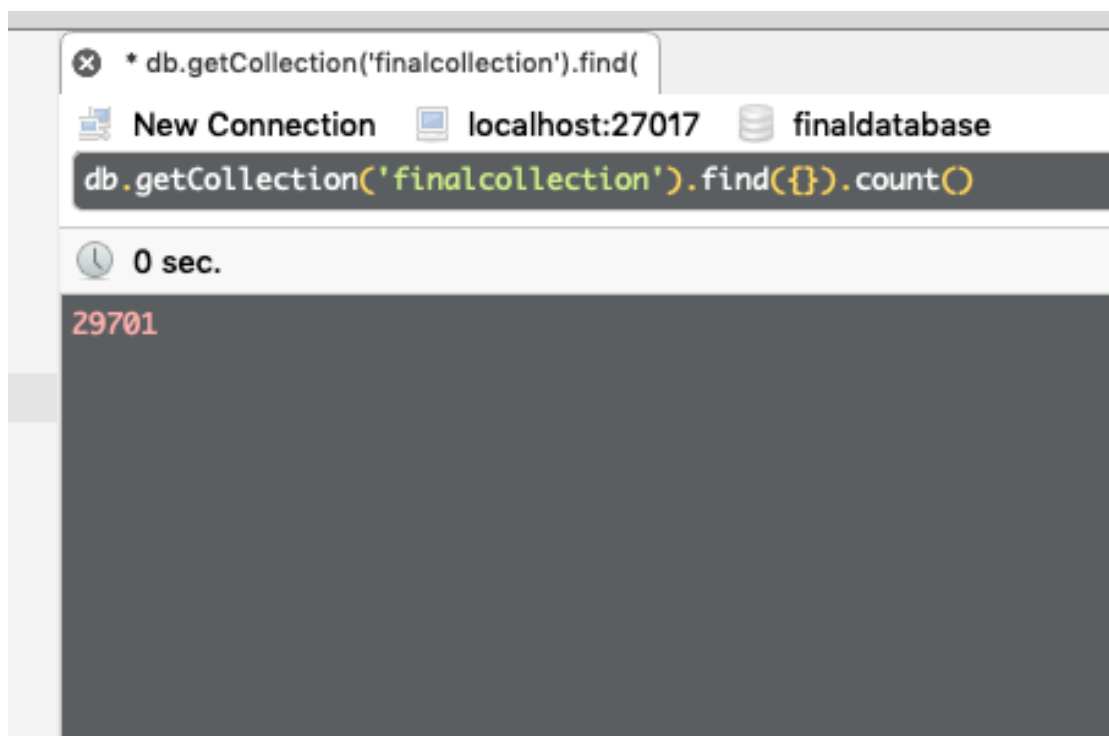
90:1 (Top Level) ↕ R Script

Console ~/Mongo-MSVC/ [🔍]

```
> m$insert(df)
List of 5
 $ nInserted : num 29701
 $ nMatched  : num 0
 $ nRemoved  : num 0
 $ nUpserted : num 0
```

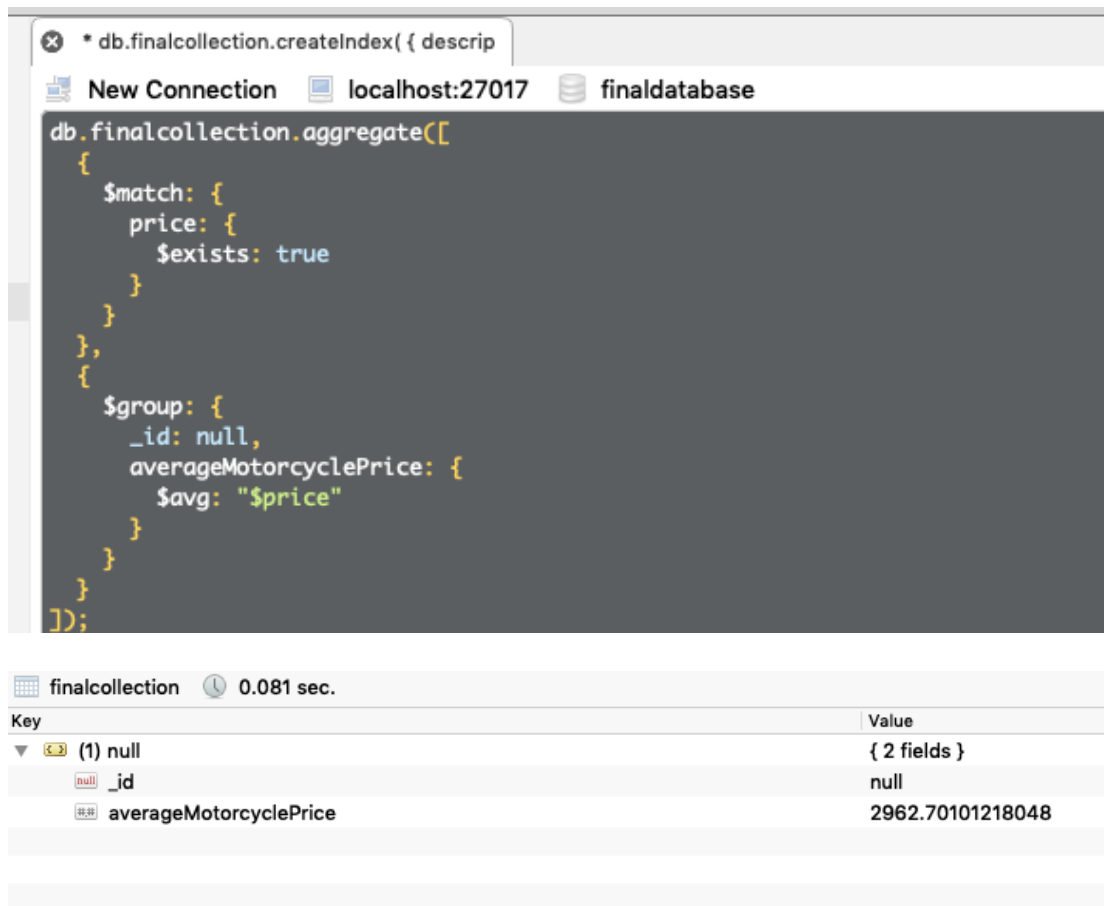


Task 2.2 How many bikes are there for sale?



The number of bikes for sale is **29701**.

Task 2.3 What is the average price of a motorcycle (give a number)? What is the number of listings that were used in order to calculate this average (give a number as well)? Is the number of listings used the same as the answer in 2.2? Why?



The screenshot shows the MongoDB Shell interface. At the top, there's a tab for a query: `* db.finalcollection.createIndex({ descrip`. Below it, the connection is set to `localhost:27017` and the database is `finaldatabase`. The main area contains the following aggregation query:

```
db.finalcollection.aggregate([
  {
    $match: {
      price: {
        $exists: true
      }
    }
  },
  {
    $group: {
      _id: null,
      averageMotorcyclePrice: {
        $avg: "$price"
      }
    }
  }
])
```

Below the query, the execution time is shown as `0.081 sec.`. The result is displayed in a table with two columns: **Key** and **Value**.

Key	Value
(1) null	{ 2 fields }
_id	null
averageMotorcyclePrice	2962.70101218048

The Average price of motorcycle is 2962 EUR.



The screenshot shows the MongoDB Shell interface. The main area contains the following count query:

```
db.getCollection('finalcollection').find({price:{$ne:null}}).count()
```

Below the query, the execution time is shown as `0.017 sec.`. The result is displayed in a large red font: **29145**.

The number is not the same with 2.2.

Task 2.4 What is the maximum and minimum price of a motorcycle currently available in the market?

```
db.finalcollection.aggregate([
  {
    $match: {
      price: {
        $exists: true
      }
    }
  },
  {
    $group: {
      _id: null,
      minimumMotorcyclePrice: {
        $min: "$price"
      }
    }
  }
]);
```

finalcollection 0.036 sec.

Key	Value
(1) null	{ 2 fields }
_id	null
minimumMotorcyclePrice	1.0

db.finalcollection.createIndex({ descrip

New Connection localhost:27017 finaldatabase

```
db.finalcollection.aggregate([
  {
    $match: {
      price: {
        $exists: true
      }
    }
  },
  {
    $group: {
      _id: null,
      maximumMotorcyclePrice: {
        $max: "$price"
      }
    }
  }
]);
```

finalcollection 0.035 sec.	
Key	Value
(1) null	{ 2 fields }
_id	null
maximumMotorcyclePrice	89000.0

Minimum price is 1 and maximum is 89000 EUR.

Task 2.5 How many listings have a price that is identified as negotiable?

✕
db.finalcollection.createIndex({ descrip

New Connection
localhost:27017
finaldatabase

```
db.finalcollection.createIndex( { description: "text" } )
db.finalcollection.find( { $text: { $search: "συζητήσιμη" } } ).count()
```

0.003 sec.

508

There are 508 listings as “συζητήσιμη”.

Task 2.6 For each Brand, what percentage of its listings is listed as negotiable?

✕
db.finalcollection.aggregate([{ "\$grou

New Connection
localhost:27017
finaldatabase

```
db.finalcollection.aggregate([
  { "$group": {
    "_id": "$brand",
    "totalCount": { "$sum": 1 },
    "matchedCount": {
      "$sum": {
        "$cond": [{ "$ne": [{ "$indexOfCP": [ "$description", "Συζητήσιμη" ] }, -1] }, 1, 0]
      }
    }
  } },
  { "$addFields": {
    "percentage": {
      "$cond": {
        "if": { "$ne": [ "$matchedCount", 0 ] },
        "then": {
          "$multiply": [
            { "$divide": [ "$matchedCount", "$totalCount" ] },
            100
          ]
        }
      }
    }
  } }
])
```

```

    }
  },
  "else": 0
}
},
}},
{ "$sort": { "percentage": -1 } }
})

```

Key	Value
(1) Dayang	{ 4 fields }
_id	Dayang
totalCount	30.0
matchedCount	1.0
percentage	3.333333333333333
(2) Derbi	{ 4 fields }
_id	Derbi
totalCount	86.0
matchedCount	2.0
percentage	2.32558139534884
(3) Harley Davidson	{ 4 fields }
_id	Harley Davidson
totalCount	309.0
matchedCount	1.0
percentage	0.323624595469256
(4) Kawasaki	{ 4 fields }
_id	Kawasaki
totalCount	1953.0
matchedCount	5.0
percentage	0.256016385048643
(5) Daytona	{ 4 fields }
_id	Daytona
totalCount	393.0
matchedCount	1.0
percentage	0.254452926208651

Task 2.7 What is the motorcycle brand with the highest average price?

```

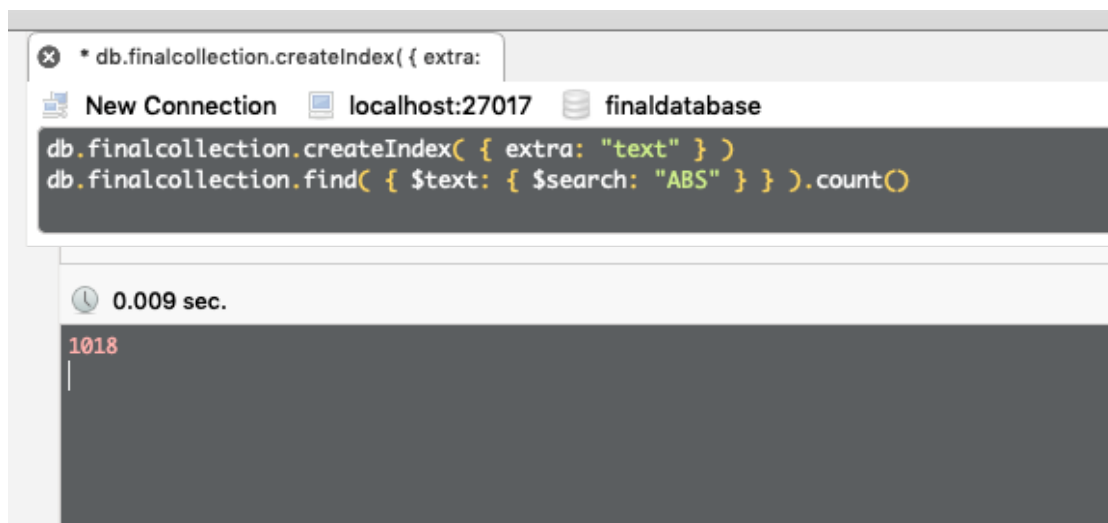
* db.finalcollection.aggregate( { $grou
New Connection localhost:27017 finaldatabase
db.finalcollection.aggregate(
{ $group: { "_id": "$brand", "avg_price": { $avg: "$price" } } },
{ $sort: { "avg_price": -1 } },
{ $limit: 1 }
);

```

Key	Value
(1) Semog	{ 2 fields }
_id	Semog
avg_price	15600.0

Average price of Semog is 15600 which the highest.

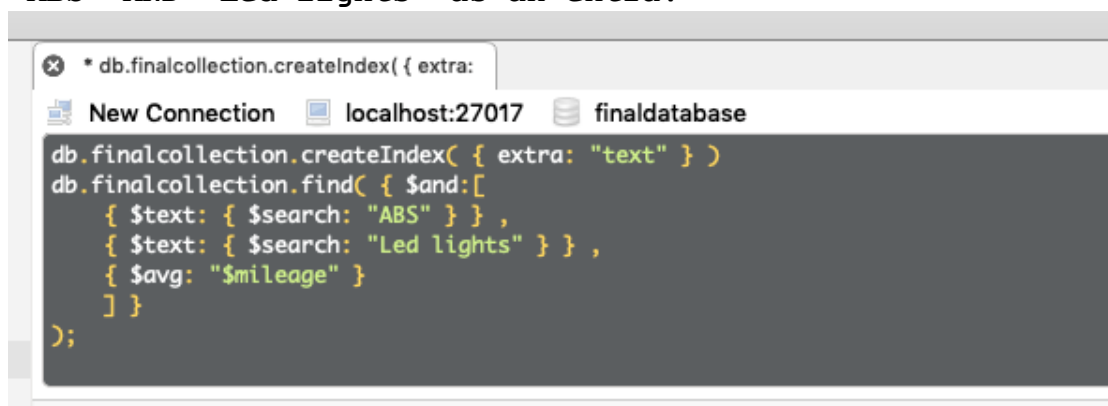
Task 2.9 How many bikes have "ABS" as an extra?



```
* db.finalcollection.createIndex( { extra:
New Connection localhost:27017 finaldatabase
db.finalcollection.createIndex( { extra: "text" } )
db.finalcollection.find( { $text: { $search: "ABS" } } ).count()
0.009 sec.
1018
```

There are 1018 bikes with ABS.

Task 2.10 What is the average Mileage of bikes that have "ABS" AND "Led lights" as an extra?



```
* db.finalcollection.createIndex( { extra:
New Connection localhost:27017 finaldatabase
db.finalcollection.createIndex( { extra: "text" } )
db.finalcollection.find( { $and:[
  { $text: { $search: "ABS" } } ,
  { $text: { $search: "Led lights" } } ,
  { $avg: "$mileage" }
] }
);
```