MSc in Business Analytics

Course : Big Data Systems

Professor : Spyros Safras

Assignment : Redis & MongoDB

Student : Ilias Dimos

AM : f2822102

Academic Period :

2021-2022

Tables of Context

Section 1 : Redis

Dataset Characteristics

In this task you are going to use the **"recorded actions"** dataset to generate some analytics with REDIS. We will use 2 datasets named :

- emails_sent.csv "Sets of EmailID, UserID, MonthID and EmailOpened"
- modified_listings.csv "Sets of UserID, MonthID, ModifiedListing"

The first dataset contains User IDs that have received an e-mail at least once. The second dataset contains **all** the User IDs of the classifieds provider and a flag that indicates whether the user performed a modification on his/her listing. Both datasets contain entries for the months January, February, and March.

Question 1.1

How many users modified their listing in January?

Query Code

```
January <- subset(listings,listings$MonthID==1)
February<- subset(listings,listings$MonthID==2)
March <- subset(listings,listings$MonthID==3)

for (i in 1:nrow(January)){
  if (January$ModifiedListing[i] == 1) {
    r$SETBIT("ModificationsJanuary", i,"1")
  }
}
r$BITCOUNT("ModificationsJanuary") # 9969

#### for the 1.6 question ####

for (i in 1:nrow(February)){
  if (February$ModifiedListing[i] == 1) {
    r$SETBIT("ModificationsFebruary", i,"1")
  }
}
r$BITCOUNT("ModificationsFebruary") # 10007


for (i in 1:nrow(March)){
  if (March$ModifiedListing[i] == 1) {
    r$SETBIT("ModificationsMarch", i,"1")
  }
}
r$BITCOUNT("ModificationsMarch") #9991
```

Answer

To answer the query, we created 3 datasets based on each month , and with a for loop we execute the setbit to set "1" to those who modified their listing and "0" to those who didn't. Also, for our convenience and to answer the Question 1.6 quicker ,

we precompute the modified listing not only for January but for February and March. In total **9969** modifications happened in January.

Question 1.2

How many users did **NOT** modify their listing in January?

Query Code

```
r$BITOP("NOT",'noModificationsJanuary',"ModificationsJanuary")
r$BITCOUNT('noModificationsJanuary') # 10031
```

Answer

To find the not modified listings we need to use the inversion of the previous question. The number of users that didn't modify their listing is **10031**. Its very important in this question to point out a useful conclusion. The summary of "noModificationsJanuary" and the "ModificationsJanuary" should give us a sum of 19999 but it gives us 20000.The main reason is why the "BITOP" operations happen in bits and the "BITCOUNT" in bytes (1 byte = 8 bits) . So, in order to make the last bits into 1 byte adds one more value.

Question 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)?

Query Code

```
overall_mails  <- as.data.frame(table(emails_sent$UserID,emails_sent$MonthID))
colnames(overall_mails) <- c("UserID","Month","Mails_recieved")
View(overall_mails)

January_mails <- subset(overall_mails,overall_mails$Month==1)
View(January_mails)
February_mails <-subset(overall_mails,overall_mails$Month==2)
View(February_mails)
March_mails <- subset(overall_mails,overall_mails$Month==3)
View(March_mails)

for(i in 1:nrow(January_mails)) {
  if (January_mails$Mails_recieved[i]>0){
    r$SETBIT("EmailsJanuary",i, "1")}

}

r$BITCOUNT("EmailsJanuary") # 9617

for(i in 1:nrow(February_mails)) {
  if (February_mails$Mails_recieved[i]>0){
    r$SETBIT("EmailsFebruary",i, "1")}

}

r$BITCOUNT("EmailsFebruary") #9666

for(i in 1:nrow(March_mails)) {
  if (March_mails$Mails_recieved[i]>0){
    r$SETBIT("EmailsMarch",i, "1")}

}

r$BITCOUNT("EmailsMarch") #9520


r$BITOP("AND","One at least mail in 3 months",c("EmailsJanuary","EmailsFebruary","EmailsMarch")) #2001
r$BITCOUNT("One at least mail in 3 months") #2668
```

Answer

In order to compute the answer to the Question 1.3 , we created a new dataset called "overall_mails" with the "table" function in order to calculate how many times every user received an email. After that we subset the overall_mails based on every month and we created 3 new datasets called " January_mails", "February_mails", "March_mails". We filled the "SETBIT" for every month by setting "1" where the user received at least one email. Finally, with the "BITOP AND" followed by "BITCOUNT" among the 3 months we found the users who received at least one mail. We have **2668** users that received at least one mail per month.

Question 1.4

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

Query Code

```
r$BITOP("AND","January_March",c("EmailsJanuary","EmailsMarch"))

r$BITOP("NOT","NotEmailsFebruary","EmailsFebruary")

r$BITOP("AND", "all_months_not_feb",c("NotEmailsFebruary","January_March"))


r$BITCOUNT("all_months_not_feb") #2417
```

Answer

To find the answer we created a "BITOP AND " among the January and the March to find the how many users received an email in those months. After that we performed an inversion of the "EmailsFebruary" to find how many users didn't receive an email in February and finally with the "BITOP AND" among the JanuaryMarch and in the inversion of February we compute the final answer. **2417** users received an email in January and March but not in February.

Question 1.5

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

Query Code

```
# first of all we need to see how many emails every user had received in January

jan_received_email<-as.data.frame(table(emails_sent$UserID[emails_sent$MonthID==1],emails_sent$MonthID[emails_sent$MonthID==1]))
colnames(jan_received_email)<-c("UserID","Month","How_many_Emails_Received")
View(jan_received_email)
jan_opened_email<-as.data.frame(table(emails_sent$UserID[emails_sent$EmailOpened==1],emails_sent$MonthID[emails_sent$EmailOpened==1]))
colnames(jan_opened_email)<-c("UserID","Month","How_many_Emails_opened")
View(jan_opened_email)

january_total <- merge(jan_received_email,jan_opened_email,by=c("UserID","Month"),all.x=T)
january_total$How_many_Emails_opened[is.na(january_total$How_many_Emails_opened)]<-0 # we make 0 the NA values.
# Redis counts binary data so we will make the "opened emails" 1 for open 0 for not open.
january_total$How_many_Emails_opened<-ifelse(january_total$How_many_Emails_opened>0,1,0)
View(january_total)

#When the emails have been opened Redis puts 1 else 0.
for(i in 1:nrow(january_total))
{
  if(january_total$How_many_Emails_opened[i] == 1)
  {
    r$SETBIT("EmailsOpenedJanuary",i,"1")
  }
}

r$BITCOUNT("EmailsOpenedJanuary") # 5645


r$BITOP("NOT","noEmailsOpenedJanuary","EmailsOpenedJanuary")
# Now we have to compare the outcome of the "bitop" with the modifications in January to find how many updated their list.

r$BITOP("AND","Update_but_didnt_opened", c("noEmailsOpenedJanuary","ModificationsJanuary"))

r$BITCOUNT("Update_but_didnt_opened") # 2006
```

Answer

In this question in order to answer we used a dataset for the mails of January. We found the opened and the received e-mails of January for every user and we merged them. When an email has been opened the value “1” is added otherwise the value 0. In the first “BITOP AND” the number of the emails that have been opened in January is calculated. The second “BITOP AND” is used among the first and the Modifications happened in January in order to find how many od the users update their listing but didn’t open the mail. Finally , the number of users that update listing but didn’t open the email is **2006**

Question 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 in March that they did not open but they updated their listing anyway on March?

Query Code

```
############## FOR FEBRUARY ################################################

feb_received_email<-as.data.frame(table(emails_sent$UserID[emails_sent$MonthID==2],emails_sent$MonthID[emails_sent$MonthID==2]))
colnames(feb_received_email)<-c("UserID","Month","How_many_Emails_Received")
View(feb_received_email)
feb_opened_email<-as.data.frame(table(emails_sent$UserID[emails_sent$EmailOpened==1],emails_sent$MonthID[emails_sent$EmailOpened==1]))
colnames(feb_opened_email)<-c("UserID","Month","How_many_Emails_opened")
View(feb_opened_email)

feburary_total <- merge(feb_received_email,feb_opened_email,by=c("UserID","Month"),all.x=T)
feburary_total$How_many_Emails_opened[is.na(feburary_total$How_many_Emails_opened)]<-0 # we make 0 the NA values
# redis count binary data so we will make the opened emails 1 for open 0 for not open
feburary_total$How_many_Emails_opened<-ifelse(feburary_total$How_many_Emails_opened>0,1,0)
View(feburary_total)

for (i in 1:nrow(feburary_total )){
  if (feburary_total$How_many_Emails_opened[i]==1){
    r$SETBIT("EmailsOpenedFebruary", i, "1")
  }
}

r$BITCOUNT("EmailsOpenedFebruary") # 5721


###################### FOR MARCH ################################################

mar_received_email<-as.data.frame(table(emails_sent$UserID[emails_sent$MonthID==3],emails_sent$MonthID[emails_sent$MonthID==3]))
colnames(mar_received_email)<-c("UserID","Month","How_many_Emails_Received")
View(mar_received_email)
mar_opened_email<-as.data.frame(table(emails_sent$UserID[emails_sent$EmailOpened==1],emails_sent$MonthID[emails_sent$EmailOpened==1]))
colnames(mar_opened_email)<-c("UserID","Month","How_many_Emails_opened")
View(mar_opened_email)

march_total <- merge(mar_received_email,mar_opened_email,by=c("UserID","Month"),all.x=T)
march_total$How_many_Emails_opened[is.na(march_total$How_many_Emails_opened)]<-0 # we make 0 the NA values
# redis count binary data so we will make the opened emails 1 for open 0 for not open
march_total$How_many_Emails_opened<-ifelse(march_total$How_many_Emails_opened>0,1,0)
View(march_total)

for (i in 1:nrow(march_total)){
  if (march_total$How_many_Emails_opened[i]==1){
    r$SETBIT("EmailsOpenedMarch", i, "1")
  }
}

r$BITCOUNT("EmailsOpenedMarch") # 5572
```

```
########## USING THE SAME PROCEDURE WITH THE 1.5 , WE WILL COUNT THE EMAILS NOT OPENED IN MARCH AND IN FEBRUARY ###############


r$BITOP("NOT","noEmailsOpenedFebruary", "EmailsOpenedFebruary") #1209
r$BITOP("AND","Update_but_didnt_opened_Feb", c("noEmailsOpenedFebruary","ModificationsFebruary"))#2500
r$BITCOUNT("Update_but_didnt_opened_Feb")#1981

r$BITOP("NOT","noEmailsOpenedMarch", "EmailsOpenedMarch") #1190
r$BITOP("AND","Update_but_didnt_opened_Mar", c("noEmailsOpenedMarch","ModificationsMarch")) #2500
r$BITCOUNT("Update_but_didnt_opened_Mar") # 1941


# finally with the Bitop OR we can compute the answer of the question


r$BITOP("OR","Total_mails_updated",c("Update_but_didnt_opened_Jan","Update_but_didnt_opened_Feb","Update_but_didnt_opened_Mar"))

r$BITCOUNT("Total_mails_updated") # 3510
```

Answer

To calculate the answer of the question we worked similar with the question 1.5. We made 2 new datasets for March and February and with the for loop we filled the setbit of each month with “1” if the mail has been opened and “0” if it hasn’t been opened. We computed the Modifications in March and February (in Question 1.1). Similar with the question 1.5 the first “BITOP NOT” we used it to find the mails that have been opened in every month separately. The second “BITOP AND” is used among the emails that haven’t been opened but have been modified for each month .Finally using the “BITOP OR” we compute the answer to the question. **3510** users received an email on every month that the didn’t not open it but updated their listing anyway in this Month.

Question 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 businessperson would understand?

Answer

In question 1.6 we found out that 3510 users updated their list without open their mail in these 3 months . That means that they probably didn't ever receive an email, or they updated their listing without even knowing about the suggestions of the company . So yes, in order to keep these users totally informed about the new suggestions and of course don't lose them as clients we have to keep sending them emails.

Section 2 : MongoDB

In this task you are going to use the **"bikes"** dataset in order to generate some analytics with MongoDB.

Question 2.1

Add your data to MongoDB.

Query Code

```
### 2.1 ############

#read file with paths of json files to a list
library(readr)
bikes_json <- read.csv2("C:\\Users\\elgr9\\OneDrive\\Desktop\\BIKES\\files_list.txt")



# Open a connection to MongoDB,MongoAssignment is our Collection
m <- mongo("MongoAssignment", url = "mongodb://localhost")

#Cleaning Loop,The loop reads every json file and apply the proper transformations needed to implement the first 5 mandatory questions.
for(i in 1:nrow(bikes_json)){


  json_data <-fromJSON(bikes_json[i,]) #



  json_data$ad_data$Mileage <- gsub("km","",json_data$ad_data$Mileage) # removes the km from every Mileage Value
  json_data$ad_data$Mileage <- gsub(",","",json_data$ad_data$Mileage) # also removes the , for every Mileage Value
  json_data$ad_data$Mileage <- as.numeric(json_data$ad_data$Mileage) # Making Them numeric Variables
  json_data$ad_data$Price  <- gsub("€","",json_data$ad_data$Price) # Removes the Euro symbol from the price
  json_data$ad_data$Price <- gsub("\\.","",json_data$ad_data$Price) # Removes the Dot symbol from price
  json_data$ad_data$Power <- gsub(" bhp.*", "", json_data$ad_data$Power) # Removed the bhp symbol from Power
  json_data$ad_data$Power <- as.numeric(json_data$ad_data$Power) # Making Power variable numeric


  if(json_data$ad_data$Price == "Askforprice")
  {

    json_data$ad_data$Price  <- 0


  }

  json_data$ad_data$Price <- as.numeric(json_data$ad_data$Price) # Making the Price numeric variables


  json_data <- toJSON(json_data,auto_unbox=TRUE)

  m$insert(json_data)  # inserting the json data to our MongoDB collection called "Mongo Assignment"

}
```

Answer

In the first question our main issue was the cleaning of the data. In order to answer the mandatory questions, we need to transform the “Price” variable . We removed the “€” sign and the dot “.” And also made them numeric variables in order to make calculations. To be able to answer some Optional questions we also transformed the variable “Power” by removing the “bhp” symbol and turning them in numeric variables . Also, we transformed the variable Mileage with the exact same way as we did with the “Power” variable. Finally , with the command “m$insert(json_data)” our json files are imported one by one into the collection in MongoDB. In figure 1 we can observe that the importing of the json files into the MongoDB Collection was successful.

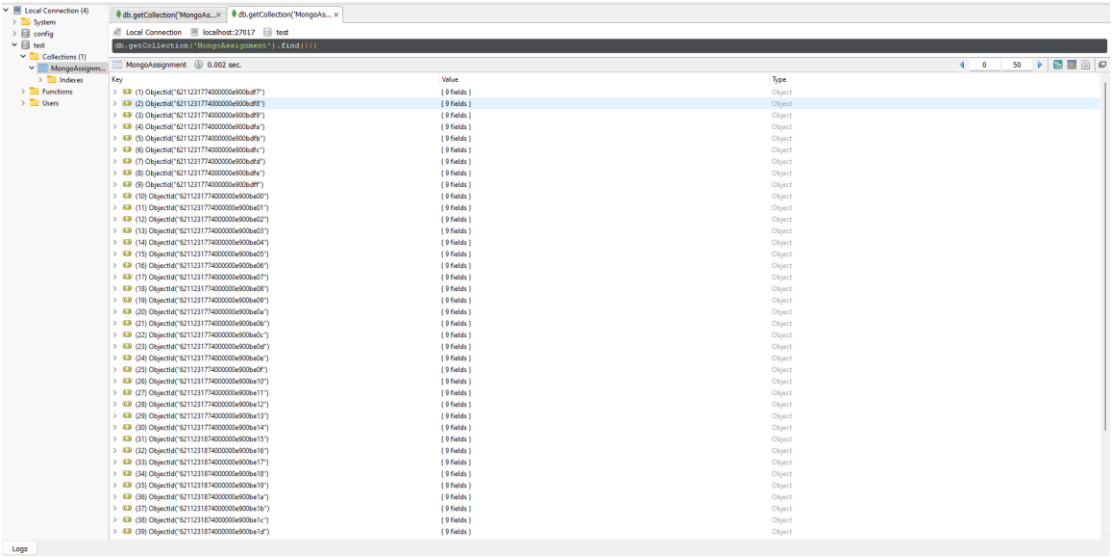
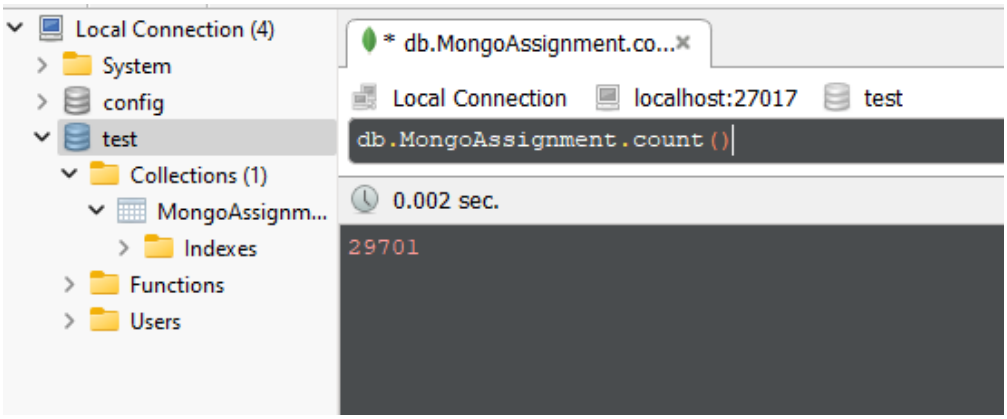


Figure 1 Successful Importing of Json data.

Question 2.2

How many bikes are there for sale?

Query Code



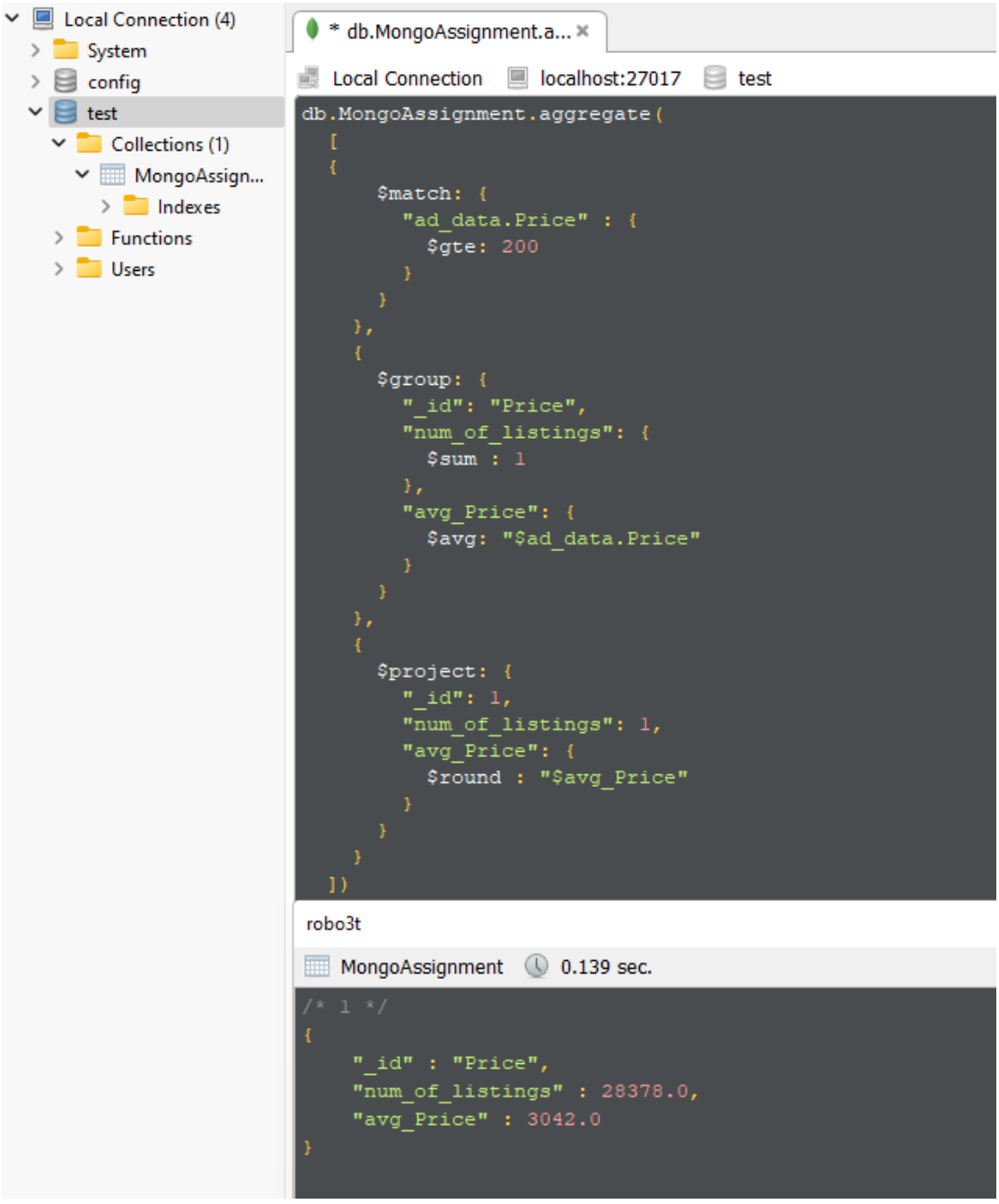
Answer

We have **29701** bikes for sale.

Question 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 using the same as the answer in 2.2? Why?

Query Code



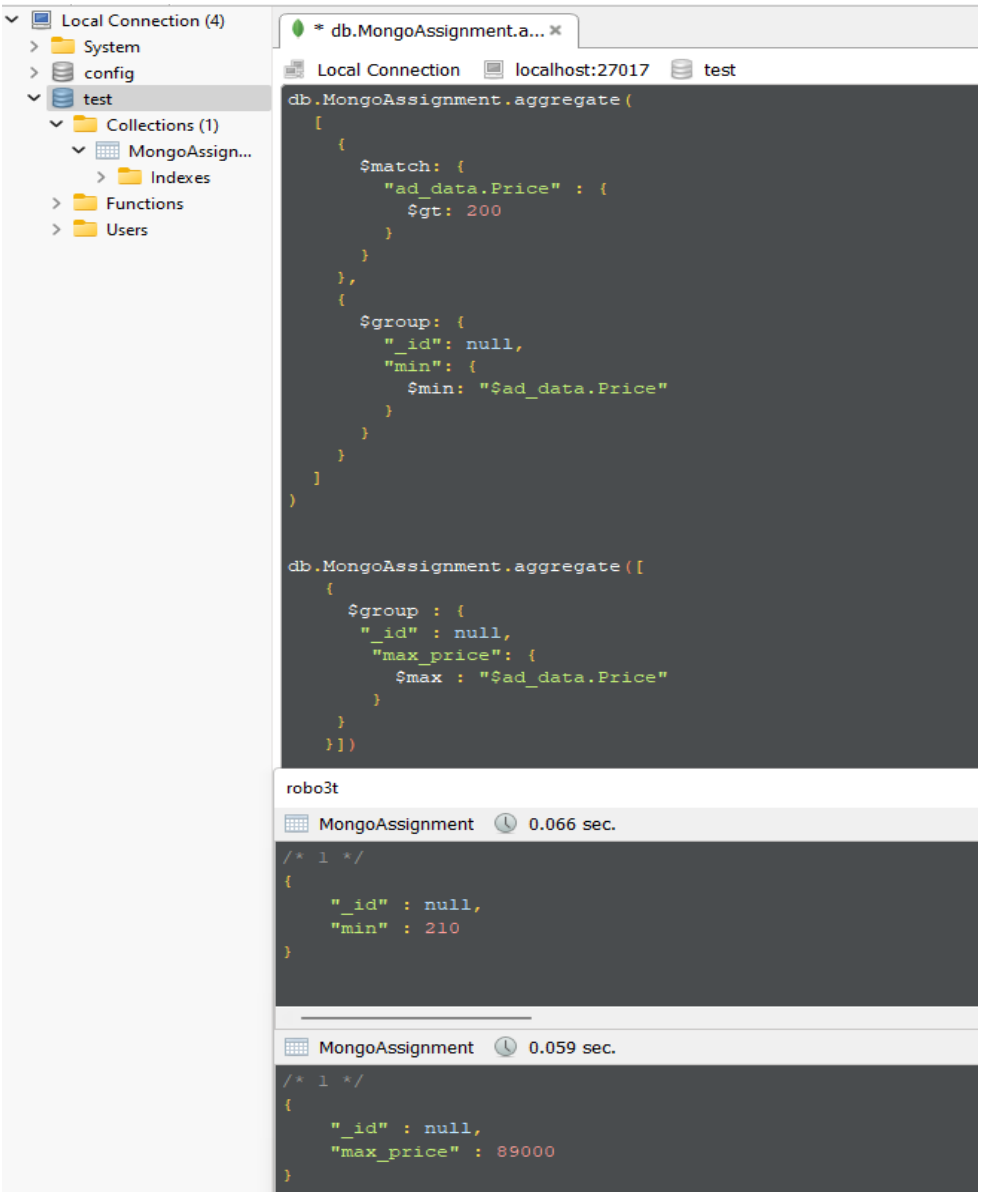
Answer

In our cleaning procedure we made “0” the Prices with value as “Askforprice”. In order to make their ad appear first on the site a lot of users set as price a very small amount of money. We consider a threshold of 200 € in order to calculate the question. Analytically , our query will compute the average price of the bikes above 200€. And that’s why our “number of listings” is **28378** instead of 29701 , because 1323 users have set a price below 200 €. The average price of the bikes with price above 200€ is **3042€.**

Question 2.4

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

Query Code



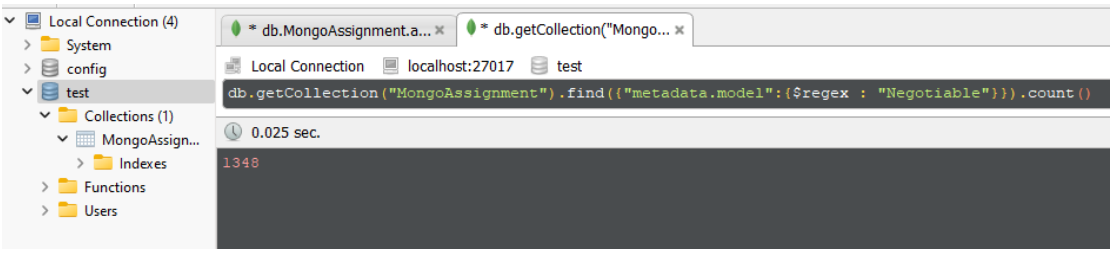
Answer

In order to compute the minimum price of the bikes we will set a threshold of 200€ again, otherwise the minimum price would be “0” ,which is fictitious. The minimum price of a motorcycle currently available in the market is **210 €** and the maximum is **89.000€.**

Question 2.5

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

Query Code



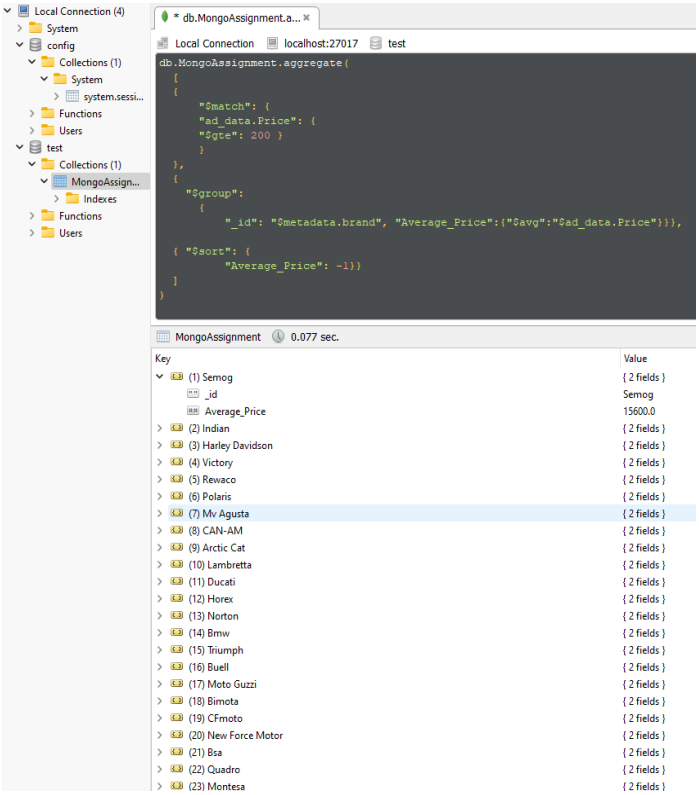
Answer

The number of listings that have a price as “Negotiable” is **1348.**

Question 2.7 (Optional)

What is the motorcycle brand with the highest average price?

Query Code



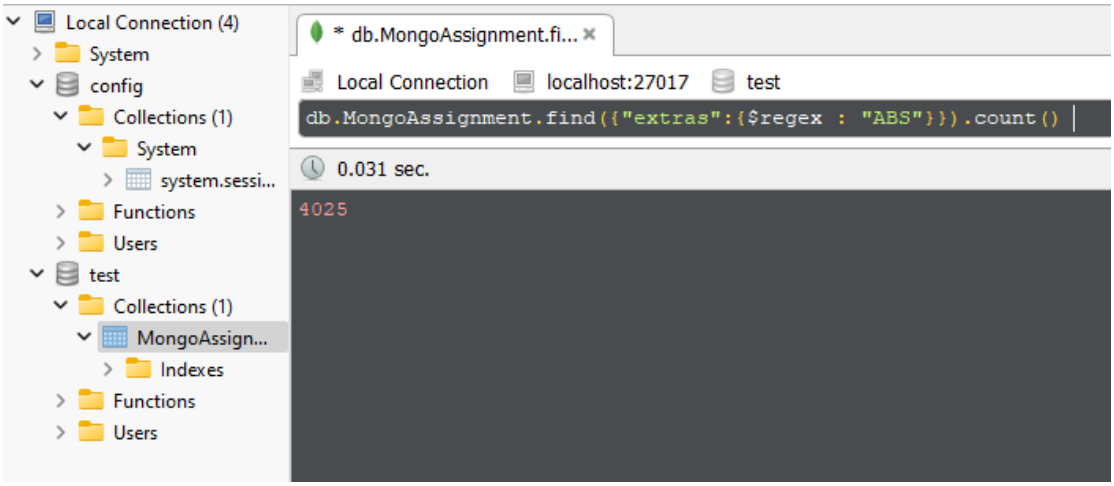
Answer

As we can see from the output of the Query code above among to all the bikes brands the one with the highest average price is the Semog Brand with **15.600** €.

Question 2.9 (Optional)

How many bikes have “ABS” as an extra?

Query Code



Answer

As we can see from the output of the Query Code the number of bikes that have “ABS” as extra is **4025**.