

1 Patch MySQL configuration file and start MySQL

In your home directory add to the file `/etc/my.cnf` record for the connection of a client application:

```
[client]
socket =/Users/users/belikov/mysql/mysql.sock
```

with your socket.

Launch MySQL server with command

```
>mysqld_safe --defaults-file=/Users/users/belikov/mysql/my.cnf &
```

2 Connect to MySQL server from R

Start R and in R environment load library RMySQL

```
R version 2.9.0 (2009-04-17)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

Natural language support but running in an English locale

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

```
> library(RMySQL)
Loading required package: DBI
```

Make a connection with database "prob" as a default database

```
con <- dbConnect(MySQL(), default.file='/Users/users/belikov/mysql/my.cnf',dbname='prob',user='root')
```

put in `default.file` a full path to your default file.

Check this connection - list tables

```
> dbListTables(con)
[1] "CROSSID" "TWOMASS" "USNOA2"
```

List columns of a table:

```
dbListFields(con,"USNOA2")
[1] "ID"      "RA2000" "DEC2000" "USNOID" "ACTFLAG" "MFLAG" "BMAG"
[8] "RMAG"   "EPOCH"
```

3 Retrieve data from the database

Select coordinates from TWOMASS table:

```
que=dbSendQuery(con, "select RA2000, DEC2000 from TWOMASS")
```

Retrieve this data into R structure:

```
data1<-fetch(que,n=-1)
```

With `n` you can specify how many records should be selected, `n=-1` retrieve all records.

The same can be done with the direct execution of SQL statement:

```
>data2 <- dbGetQuery(con, "select RA2000, DEC2000 from TWOMASS")
```

which give you all results.

4 Load/unload data from R to/from database

Write a result of data selection to a new table in the database

```
> dbWriteTable(con,"TWOMASSC",data2)
[1] TRUE
```

You wrote all output of the previous request table to the new table TWOMASSC which was created on-the-fly.

```
> dbListFields(con,"TWOMASSC")
[1] "row_names" "RA2000" "DEC2000"
```

The same goes for csv-files:

```
> dbWriteTable(con,"TWOMASSBIS","2mass.csv")
[1] TRUE
```

Note that the first argument is a path to the file, and the file should be true comma-separated file, otherwise you will dump all in a single column.

Check the table

```
dbListFields(con,"TWOMASSBIS")
[1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9" "V10" "V11" "V12"
[13] "V13" "V14" "V15" "V16" "V17" "V18"
```

5 Plot the data

Read the whole table into R data frame:

```
twomass <- dbReadTable(con, "TWOMASS")
```

Check names:

```
> names(twomass)
[1] "ID" "RA2000" "DEC2000" "TWOMASSID" "JMAG" "EJMAG"
[7] "HMAG" "EHMAG" "KMAG" "EKMAG" "Qflg" "Rflg"
[13] "Bflg" "Cflg" "Xflg" "Aflg"
```

Plot CMD:

```
> plot(twomass$HMAG-twomass$KMAG,twomass$HMAG,xlab="H-K",ylab="H",ylim=c(20,5),xlim=c(-2,4))
```

Plot histogram with 10 steps:

```
>hist(twomass$HMAG, breaks=10)
```

Histogram of data2\$HMAG

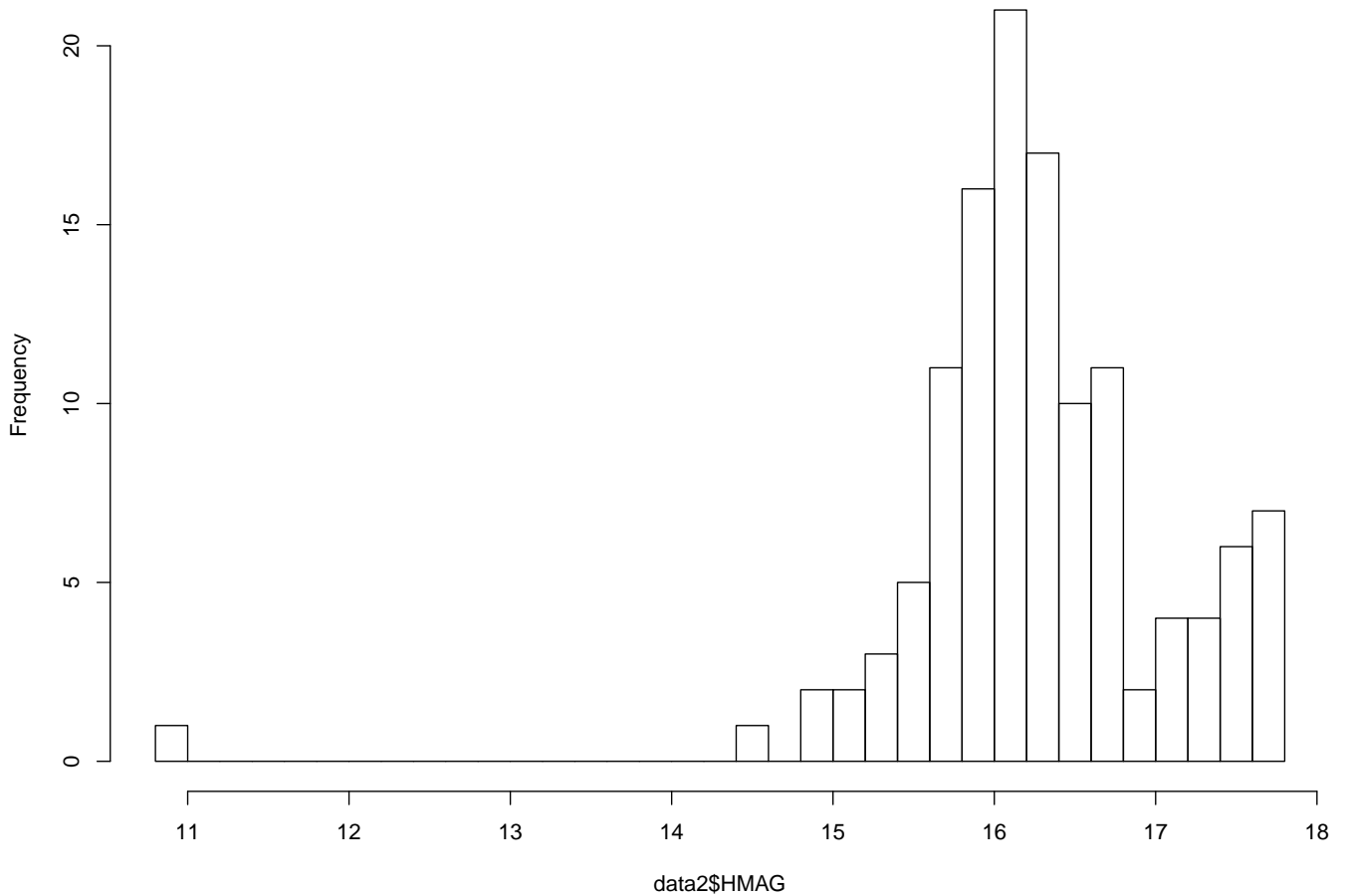


Figure 1: An example of histogram with R. Please, note that completeness limit is 16.2 mag

And, finally, search for a completeness limit for H magnitude (original 2MASS) in color interval $(H - K) > 1.0$ mag:

```
> data2 <- dbGetQuery(con, "select HMAG from TWOMASS where HMAG-KMAG > 1.0")
> names(data2)
[1] "HMAG"
> hist(data2$HMAG, breaks=30)
```

Save the plot in postscript file:

```
> postscript('hist.ps')
> hist(data2$HMAG, breaks=30)
> dev.off()
X11cairo
2
```

You can use `jpeg('hist.jpg')` instead of `postscript()`.
Do not forget to disconnect at the end:

```
>dbDisconnect(con)
```