**MySQL and MongoDB data sets**

**1) Querying MySQL**

The following creates an R dataframe that shows rates of tuburculosis infection by country.

**The following analysis requires the RMySQL, Hmisc, and plyr libraries, along with dependencies.**

library(RMySQL)

## Loading required package: DBI

library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##

## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':

##

## format.pval, round.POSIXt, trunc.POSIXt, units

library(plyr)

##

## Attaching package: 'plyr'

## The following objects are masked from 'package:Hmisc':

##

## is.discrete, summarize

**The following demonstrates a simple select query using RMySQL along with some basic R-side refinement of the retrieved dataset.**

library("DBI")

con <- dbConnect(RSQLite::SQLite(), ":memory:")

dbWriteTable(con, "mtcars", mtcars)

rs <- dbSendQuery(con, "SELECT \* FROM mtcars WHERE cyl = 4;")

dbFetch(rs)

dbClearResult(rs)

dbDisconnect(con)

**First we calculate the total drat**

**Find mean for drat**

**Find count of rows using SQL-commands**

**2 Querying Mongo**

**2.1 MongoDB Connection**

conn\_mongo <- mongoDbConnect("flightsData")

airlines\_mongo <- dbGetQuery(conn\_mongo, "airlines", "",0,16)

airports\_mongo <- dbGetQuery(conn\_mongo, "airports", "",0,1397)

flights\_mongo <- dbGetQuery(conn\_mongo, "flights", "",0,336776)

planes\_mongo <- dbGetQuery(conn\_mongo, "planes", "",0,3322)

weather\_mongo <- dbGetQuery(conn\_mongo, "weather", "",0,8719)

**2.2 Mongo Query Examples**

**2.2.1 XID Mongo Unique Identifier**

xid<-dbGetQuery(conn\_mongo, 'airports', '{"name": "Jekyll Island Airport"}')

xid$X\_id

## [1] "5717f71eea7e80e167b74cee"

### 2.2.2 Frequency Barplot Example

**Idea borrowed from** [***https://github.com/renkun-ken/pipeR-tutorial/blob/master/Examples/dplyr.Rmd***](https://github.com/renkun-ken/pipeR-tutorial/blob/master/Examples/dplyr.Rmd)**. The data can be treated as if it were any other R list.**

flights\_mongo %>%

mutate(speed = distance / air\_time) %>%

group\_by(carrier) %>%

summarize(mean\_speed = mean((distance/air\_time)\*60,na.rm = TRUE)) %>%

arrange(desc(mean\_speed)) %>%

with(barplot(mean\_speed,names.arg = carrier,

main = "Average flight speed", xlab="Airline", ylab="Miles per hour"))