Assignment 12A

Clint Valentine

April 16, 2017

```
rm(list = ls())

library(jsonlite)
library(mongolite)
library(stringr)
```

```
# Load in the data into one data.frame if it has not already been loaded.
if (exists('bird.strikes') == F) {
  unzip('Bird Strikes.zip')
  # Escape sep = ',' in quotations
  # Impute NA as empty strings
  # Do not consider strings factors
  bird.strikes <- read.table(
    file = 'Bird Strikes.csv',
    fill = T,
    header = T,
    na.strings = '',
    quote = '\"',
    sep = ',',
    stringsAsFactors = F
  )
}

# Replace periods with underscore, remove trailing underscore, and make lowercase.
header <- gsub('_$', '', gsub('\\.+', '_', str_trim(colnames(bird.strikes))))
header <- tolower(header)

# Replace reformatted header on bird.strike data.
colnames(bird.strikes) <- header
```

Insert the Bird Strikes.csv file into MongoDB

```
mongo <- mongo(collection='test',
               db='test',
               url='mongodb://localhost:27017',
               verbose=F)

mongo$insert(bird.strikes)
```

```
## List of 5
##  $ nInserted  : num 99404
##  $ nMatched   : num 0
##  $ nRemoved   : num 0
##  $ nUpserted  : num 0
##  $ writeErrors: list()
```

Use the export command to display the inserted file.

1. Data is first exported to a tempfile.
2. Since this ouputs many valid JSON lines but not in a valid JSON array we must preprocess the information.
3. Read into a `data.frame`.
4. Examine results by calling `dim` and `str`.

Note there is an extra column in our exported dataset which represents the unique identifer MongoDB has assigned our record.

```
tmp <- tempfile()
mongo$export(file(tmp))
exported.data <- fromJSON(sprintf("[%s]", paste(readLines(file(tmp)), collapse=",")))
```

```
print(dim(bird.strikes))
```

```
## [1] 99404    37
```

```
print(dim(exported.data))
```

```
## [1] 99404    38
```

```
str(exported.data, width=80, strict.width='cut')
```

```
## 'data.frame':    99404 obs. of  38 variables:
##  $ _id                                    :'data.frame': 99404 obs. of  1 vari..
##   ..$ $oid: chr  "58d86f5b7b149d2d5c003d42" "58d86f5b7b149d2d5c003d43" "58d86"..
##  $ aircraft_type                          : chr  "Airplane" "Airplane" "Airpl"..
##  $ airport_name                           : chr  "NEWARK LIBERTY INTL ARPT" ""..
##  $ altitude_bin                           : chr  "< 1000 ft" "Unknown" "Unkno"..
##  $ aircraft_make_model                    : chr  "B-757-200" "B-737-300" "B-7"..
##  $ wildlife_number_struck                 : chr  "2 to 10" "1" "1" "1" ...
##  $ aircraft_flight_number                 : chr  "586" NA NA NA ...
##  $ flightdate                             : chr  "1/1/2000 0:00" "1/1/2000 0:"..
##  $ record_id                              : int  200508 206593 206594 204095 2..
##  $ effect_indicated_damage                : chr  "No damage" "No damage" "No "..
##  $ aircraft_number_of_engines             : chr  "2" "2" "2" "3" ...
##  $ aircraft_airline_operator              : chr  "CONTINENTAL AIRLINES" "UNIT"..
##  $ origin_state                           : chr  "New Jersey" "N/A" "Colorado"..
##  $ when_phase_of_flight                   : chr  "Take-off run" NA "Climb" "A"..
##  $ conditions_precipitation               : chr  "Fog" NA NA NA ...
##  $ remains_of_wildlife_collected          : logi  FALSE FALSE FALSE FALSE TRUE..
##  $ remains_of_wildlife_sent_to_smithsonian: logi  FALSE FALSE FALSE FALSE FALS..
##  $ remarks                                : chr  "3 BIRDS. NO DMG" "BIRD DEBR"..
##  $ wildlife_size                          : chr  "Medium" "Medium" "Medium" ""..
##  $ conditions_sky                         : chr  "Overcast" NA NA NA ...
##  $ wildlife_species                       : chr  "Unknown bird - medium" "Unk"..
##  $ when_time_hhmm                         : int  943 NA NA NA NA 1300 NA NA 11..
##  $ when_time_of_day                       : chr  "Day" NA NA NA ...
##  $ pilot_warned_of_birds_or_wildlife      : chr  "Y" NA NA NA ...
##  $ cost_other_inflation_adj               : chr  "0" "0" "0" "0" ...
##  $ cost_repair_inflation_adj              : chr  "0" "0" "0" "0" ...
##  $ cost_total                             : chr  "0" "0" "0" "0" ...
##  $ miles_from_airport                     : chr  "0" NA NA NA ...
##  $ feet_above_ground                      : chr  "0" NA NA NA ...
##  $ location_freeform_en_route             : chr  NA "HDN-LAX" NA NA ...
```

```
##  $ effect_impact_to_flight                : chr  NA NA NA NA ...
##  $ effect_other                           : chr  NA NA NA NA ...
##  $ location_nearby_if_en_route            : chr  NA NA NA NA ...
##  $ speed_ias_in_knots                     : chr  NA NA NA NA ...
##  $ reported_date                          : chr  NA NA NA NA ...
##  $ cost_aircraft_time_out_of_service_hours: chr  NA NA NA NA ...
##  $ number_of_people_injured               : int  NA NA NA NA NA NA NA NA NA NA..
##  $ number_of_human_fatalities             : int  NA NA NA NA NA NA NA NA NA NA..
```

Fetch the unique airport names from the database

```
airports <- mongo$distinct('airport_name')
cat("There are", length(airports), "unique airport names.\n\n")
```

```
## There are 1703 unique airport names.
```

```
cat("Here are the first ten:\n\n")
```

```
## Here are the first ten:
```

```
print(head(airports, 10))
```

```
##  [1] "NEWARK LIBERTY INTL ARPT"
##  [2] "UNKNOWN"
##  [3] "DENVER INTL AIRPORT"
##  [4] "CHICAGO O'HARE INTL ARPT"
##  [5] "JOHN F KENNEDY INTL"
##  [6] "CINCINNATI MUNI ARPT-LUNKEN FIELD"
##  [7] "MIAMI INTL"
##  [8] "SAN FRANCISCO INTL ARPT"
##  [9] "SALT LAKE CITY INTL"
## [10] "SOUTHWEST FLORIDA INTL ARPT"
```

Count the number of records where `origin_state` equals "New Jersey".

```
query <- '{
  "origin_state": "New Jersey"
}'

num_new_jersey <- mongo$count(query)
cat("There are", num_new_jersey, "records where `origin_state` = New Jersey.")
```

```
## There are 2936 records where `origin_state` = New Jersey.
```

Fetch the data with `conditions_precipitation` being fog and sort the data in descending order of `record_id`.

```
query <- '{
  "conditions_precipitation": "Fog"
}'

sort <- '{
  "record_id": -1
}'

records <- mongo$find(query=query, sort=sort)
cat("There are", nrow(records), "records with `conditions_precipitation` as Fog.\n\n")
```

```
## There are 878 records with `conditions_precipitation` as Fog.
```

```
cat("Are the records sorted in descending order of `record_id`?",
    all.equal(records$record_id,
              rev(sort(records$record_id))))
```

```
## Are the records sorted in descending order of `record_id`? TRUE
```

Fetch only the following columns for `aircraft_airline_operator`

Query:

`aircraft_airline_operator` = "AMERICAN AIRLINES" OR "CONTINENTAL AIRLINES"

Columns to include:

- `record_id`
- `origin_state`
- `aircraft_airline_operator`
- `airport_name`

```
query = '{
  "aircraft_airline_operator": { "$in": ["AMERICAN AIRLINES", "CONTINENTAL AIRLINES"]}
}'

fields = '{
  "_id": 0,
  "record_id": 1,
  "origin_state": 1,
  "aircraft_airline_operator": 1,
  "airport_name": 1
}'

records <- mongo$find(query=query, fields=fields)
cat("There are", nrow(records), "records for `AMERICAN AIRLINES`/`CONTINENTAL AIRLINES`.")
```

```
## There are 4684 records for `AMERICAN AIRLINES`/`CONTINENTAL AIRLINES`.
```

```
print(head(records))
```

```
##                  airport_name record_id aircraft_airline_operator
## 1    NEWARK LIBERTY INTL ARPT    200508      CONTINENTAL AIRLINES
## 2                     UNKNOWN    204787         AMERICAN AIRLINES
## 3        MINETA SAN JOSE INTL    208470         AMERICAN AIRLINES
## 4     LAFAYETTE REGIONAL (LA)    204764      CONTINENTAL AIRLINES
## 5         JOHN F KENNEDY INTL    202568         AMERICAN AIRLINES
## 6 DALLAS/FORT WORTH INTL ARPT    200470         AMERICAN AIRLINES
##   origin_state
## 1   New Jersey
## 2          N/A
## 3   California
## 4    Louisiana
## 5     New York
## 6        Texas
```