Analysis of Wildlife Strikes to Aircraft

Chandresh Lokesha

Spring 24

Practicum I CS5200

```
## Warning: package 'RMySQL' was built under R version 4.2.3
```

```
#show global variables like 'local_infile';
set global local_infile = TRUE;
```

```
# Drop the flights table if it exists
dbSendQuery(birdStrikeDBCon, "DROP TABLE IF EXISTS strikes;")
```

```
## <MySQLResult:1133928992,0,1>
```

```
dbSendQuery(birdStrikeDBCon, "DROP TABLE IF EXISTS conditions;")
```

```
## <MySQLResult:1133928992,0,2>
```

```
dbSendQuery(birdStrikeDBCon, "DROP TABLE IF EXISTS flights;")
```

```
## <MySQLResult:1133928992,0,3>
```

```
dbSendQuery(birdStrikeDBCon, "DROP TABLE IF EXISTS airports;")
```

```
## <MySQLResult:1133928992,0,4>
```

```
# Create Table airport

dbSendQuery(birdStrikeDBCon, "CREATE TABLE airports (
  aid INTEGER PRIMARY KEY,
  airportName TEXT,
  airportState TEXT,
  airportCode VARCHAR(3) DEFAULT 'ZZZ'
);
")
```

```
## <MySQLResult:12,0,5>
```

```
# Create Table Flights

dbSendQuery(birdStrikeDBCon, "CREATE TABLE flights (
  fid INTEGER PRIMARY KEY,
  date DATE,
  originAirport INTEGER,
  airlineName TEXT,
  aircraftType TEXT,
  isHeavy BOOLEAN,
  FOREIGN KEY (originAirport) REFERENCES airports(aid)
);
")
```

```
## <MySQLResult:1133928992,0,6>
```

```
dbSendQuery(birdStrikeDBCon, "CREATE TABLE conditions (
  cid INTEGER PRIMARY KEY,
  sky_condition TEXT,
  explanation TEXT
);
")
```

```
## <MySQLResult:1133928992,0,7>
```

```
dbSendQuery(birdStrikeDBCon, "CREATE TABLE strikes (
  sid INTEGER PRIMARY KEY,
  fid INTEGER,
  numbirds INTEGER,
  impact TEXT,
  damage BOOLEAN,
  altitude INTEGER CHECK (altitude >= 0),
  conditions INTEGER,
  FOREIGN KEY (fid) REFERENCES flights(fid),
  FOREIGN KEY (conditions) REFERENCES conditions(cid)
);
")
```

```
## <MySQLResult:1133928992,0,8>
```

```
# Insert sample data into the 'airports' table
dbSendQuery(birdStrikeDBCon, "INSERT INTO airports (aid, airportName, airportState, airportCode) VALUES
res <- dbExecute(birdStrikeDBCon, "SELECT * from airports")
print(res)

# Insert sample data into the 'flights' table
dbSendQuery(birdStrikeDBCon, "INSERT INTO flights (fid, date, originAirport, airlineName, aircraftType,
res <- dbExecute(birdStrikeDBCon, "SELECT * from flights")
print(res)

# Insert sample data into the 'conditions' table
dbSendQuery(birdStrikeDBCon, "INSERT INTO conditions (cid, sky_condition, explanation) VALUES (2, 'Over
res <- dbExecute(birdStrikeDBCon, "SELECT * from conditions")
```

```
print(res)

# Insert sample data into the 'strikes' table
dbSendQuery(birdStrikeDBCon, "INSERT INTO strikes (sid, fid, numbirds, impact, damage, altitude, conditi
VALUES (3, 2, 5, 'Impact1', TRUE, 10000, 2);")
res <- dbExecute(birdStrikeDBCon, "SELECT * from strikes")
print(res)

# Fetch and print data from the 'airports' table
airports_data <- dbGetQuery(birdStrikeDBCon, "SELECT * FROM airports;")
print(airports_data)

# Fetch and print data from the 'flights' table
flights_data <- dbGetQuery(birdStrikeDBCon, "SELECT * FROM flights;")
print(flights_data)

# Fetch and print data from the 'conditions' table
conditions_data <- dbGetQuery(birdStrikeDBCon, "SELECT * FROM conditions;")
print(conditions_data)

strikes_data <- dbGetQuery(birdStrikeDBCon, "SELECT * FROM strikes;")
print(strikes_data)

dbSendQuery(birdStrikeDBCon, "DELETE FROM strikes;")
dbSendQuery(birdStrikeDBCon, "DELETE FROM flights;")
dbSendQuery(birdStrikeDBCon, "DELETE FROM airports;")
dbSendQuery(birdStrikeDBCon, "DELETE FROM conditions;")
```

```
# Load CSV file into a dataframe
bds.raw <- read.csv("BirdStrikesData-V3.csv", header = TRUE, sep = ",")
print(head(bds.raw))
```

```
##      rid aircraft                    airport        model
## 1 202152 Airplane                LAGUARDIA NY    B-737-400
## 2 208159 Airplane DALLAS/FORT WORTH INTL ARPT        MD-80
## 3 207601 Airplane           LAKEFRONT AIRPORT        C-500
## 4 215953 Airplane         SEATTLE-TACOMA INTL    B-737-400
## 5 219878 Airplane                NORFOLK INTL CL-RJ100/200
## 6 218432 Airplane         GUAYAQUIL/S BOLIVAR        A-300
##                   impact     flight_date    damage           airline     origin
## 1      Engine Shut Down 11/23/2000 0:00    Damage        US AIRWAYS   New York
## 2                  None  7/25/2001 0:00    Damage AMERICAN AIRLINES      Texas
## 3                  None  9/14/2001 0:00 No damage          BUSINESS  Louisiana
## 4 Precautionary Landing   9/5/2002 0:00 No damage   ALASKA AIRLINES Washington
## 5                  None  6/23/2003 0:00 No damage   COMAIR AIRLINES   Virginia
## 6                  None  7/24/2003 0:00 No damage AMERICAN AIRLINES        N/A
##   flight_phase wildlife_size sky_conditions pilot_warned_flag altitude_ft
## 1        Climb        Medium       No Cloud                N       1,500
## 2 Landing Roll         Small     Some Cloud                Y           0
## 3     Approach         Small       No Cloud                N          50
## 4        Climb         Small     Some Cloud                Y          50
## 5     Approach         Small       No Cloud                N          50
## 6 Take-off run         Small       No Cloud                N           0
```

```
##   heavy_flag
## 1        Yes
## 2         No
## 3         No
## 4        Yes
## 5         No
## 6         No
```

```
print(colnames(bds.raw))
```

```
##  [1] "rid"               "aircraft"          "airport"
##  [4] "model"             "impact"            "flight_date"
##  [7] "damage"            "airline"           "origin"
## [10] "flight_phase"      "wildlife_size"     "sky_conditions"
## [13] "pilot_warned_flag" "altitude_ft"       "heavy_flag"
```

Data Cleaning and pre-processing to format the date

1. We are normalizing the flight phases to have only values take-off,landing,inflight,unknown

2. The flight_date field was transformed into

```
bds.raw$flight_date<- parse_date_time(bds.raw$flight_date, orders=c("m-d-y H:M","m/d/y H:M"))
bds.raw$flight_date <- format(bds.raw$flight_date,"%Y-%m-%d")
```

```
case5 <- which(bds.raw$airline=='')
bds.raw[case5,"airline"] <- 'unknown'

case6 <- which(bds.raw$aircraft=='')
bds.raw[case6,"aircraft"] <- 'unknown'

case7 <- which(bds.raw$airport=='')
bds.raw[case7,"airport"] <- 'unknown'
```

```
dbDisconnect(birdStrikeDBCon)
```

```
## Warning: Closing open result sets
```

```
## [1] TRUE
```