

TERM PROJECT
COURSE : DSCS6020
SPRING SEMESTER 2016

SUBMITTED BY
SUPAN KAUR

CONTENTS

1. BACKGROUND
2. METHODS
3. CONCLUSION
4. REFERENCES

BACKGROUND

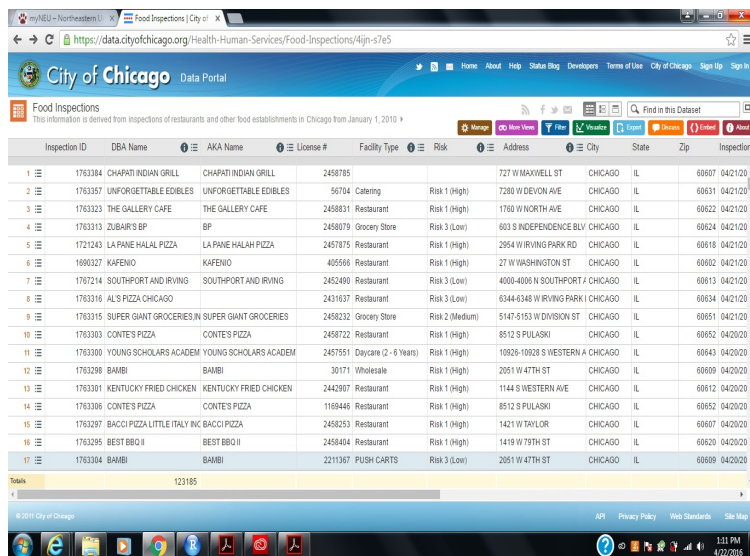
When going out to eat, sometimes a little extra information can keep you healthy . According to the Center of Disease Control, roughly 1 in 6 Americans(or 48 million people) get sick from foodborne diseases each year. Of those greater than 40% of the foodborne diseases are attributed to commercial food establishments.

Food premises are inspected on a regular basis to ensure compliance with the Food and Food Handling Establishment under the Public Health Act. Inspections determine if minimum standards and practices are being followed with respect to general food handling, storage temperature, sanitations, employee hygiene and equipment or food contact surface disinfection procedures for the specific type of processes and foods involved. The food inspection datasets are available on the web portals of almost every city for public access. As transparency is good for public health outcomes, the datasets available to public, let them better use inspection results to inform where they will eat next.

For this project, I worked on the data of Food Inspection results for city of Chicago available at the website , www.cityofchicago.org at data portal. The food inspections in Chicago are handled by Food Protection Division of the Chicago Department of Public Health(CDPH). It's a big dataset covering inspections from year 2010 till April, 2016. I downloaded the data followed by storing in SQL database and finally retrieving it by using SQL queries. These inspections covered almost every Food Establishments in the city such as restaurants, grocery stores, bakeries, convenience stores, hospitals, nursing homes, day care facilities, shelters, schools, and temporary food service events. It lists the results of inspections as well as text of violations. Inspections focus on food handling practices, product temperatures, personal hygiene, facility maintenance, and pest control. The main purpose of these inspections are to promote public health in areas of food safety and sanitation and prevent the occurrence of food-borne illness. Although there are infinite ways to query the database, depending on what information one is interested in, I would like to reveal answers about these questions.. Does the total number of failed inspections decreases each year? Which type of food establishment (restaurant, schools or day care) have the highest "failed inspections"? And the last but not the least which areas(zip) of Chicago City are the hot spots with the highest number of failed inspections? Let's begin...

METHODS

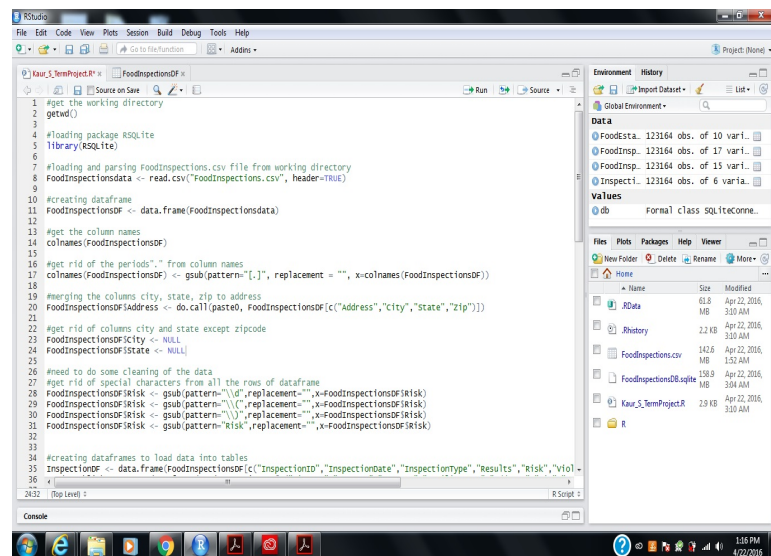
1. DATA COLLECTION: The data set for Food Inspections for the city of Chicago is available to public access at www.cityofchicago.org at data portal in various formats like csv, Excel, JSON, XLS, XLSX, XML for download. I chose the csv format of dataset for this project. Some of the manipulations that I made after uploading it in RStudio were removal of periods "." from column names, special characters from all rows, repeated entries as facility types, merging the columns city, state and zip to Address leaving zip column intact, for zip related queries and searches. Moreover I extract only years from InspectionDate column to compare total number of failed inspections from 2010 till 2015.



The screenshot shows the City of Chicago Data Portal website. The main content is a table of food inspection records. The table has columns for Inspection ID, DBA Name, AKA Name, License #, Facility Type, Risk, Address, City, State, Zip, and Inspection Date. The table is sorted by Inspection ID in descending order. The first few rows are:

Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	State	Zip	Inspection Date
1	1763384	CHAPARI INDIAN GRILL	2458785			727 W MAXWELL ST	CHICAGO	IL	60607	04/21/2015
2	1763357	UNFORGETTABLE EDIBLES	56704	Catering	Risk 1 (High)	7280 W DEVON AVE	CHICAGO	IL	60631	04/21/2015
3	1763323	THE GALLERY CAFE	2458831	Restaurant	Risk 1 (High)	1760 W NORTH AVE	CHICAGO	IL	60622	04/21/2015
4	1763313	ZUBARS BP	2458079	Grocery Store	Risk 3 (Low)	603 S INDEPENDENCE BLV	CHICAGO	IL	60624	04/21/2015

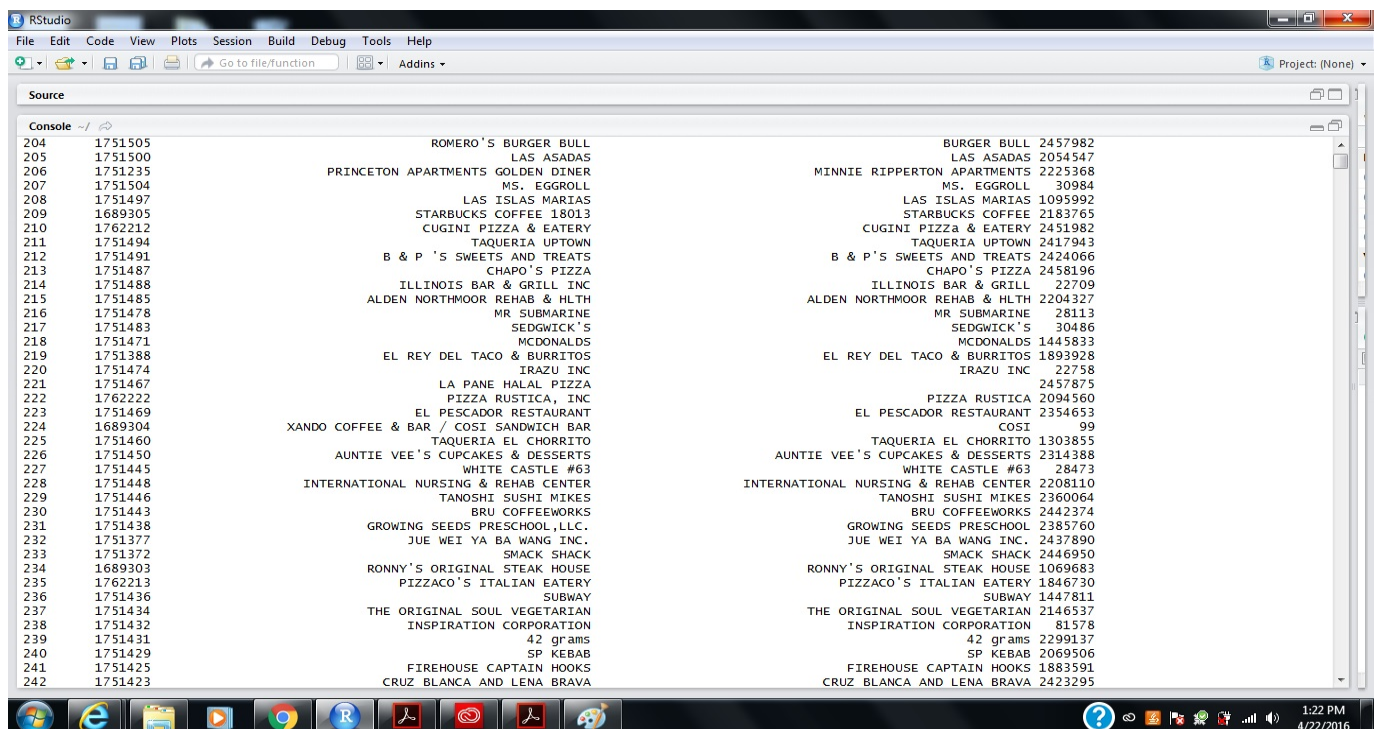
Fig1. The webpage of www.cityofchicago.org



The screenshot shows the RStudio interface with R code for cleaning and manipulating the food inspection data. The code includes comments and R commands for loading the data, creating a dataframe, and cleaning the data. The code is as follows:

```
1 #get the working directory
2 getwd()
3
4 #loading package RSQLite
5 library(RSQLite)
6
7 #loading and parsing FoodInspections.csv file from working directory
8 FoodInspectionsdata <- read.csv("FoodInspections.csv", header=TRUE)
9
10 #creating dataframe
11 FoodInspectionsDF <- data.frame(FoodInspectionsdata)
12
13 #get the column names
14 colnames(FoodInspectionsDF)
15
16 #get rid of the periods "." from column names
17 colnames(FoodInspectionsDF) <- gsub(pattern="\\.", replacement="", x=colnames(FoodInspectionsDF))
18
19 #merging the columns city, state, zip to address
20 FoodInspectionsDF$Address <- do.call(paste0, FoodInspectionsDF[c("Address", "City", "State", "Zip")])
21
22 #get rid of columns city and state except zipcode
23 FoodInspectionsDF$City <- NULL
24 FoodInspectionsDF$State <- NULL
25
26 #need to do some cleaning of the data
27 #get rid of special characters from all the rows of dataframe
28 FoodInspectionsDF$Risk <- gsub(pattern="\\d", replacement="", x=FoodInspectionsDF$Risk)
29 FoodInspectionsDF$Risk <- gsub(pattern="\\(", replacement="", x=FoodInspectionsDF$Risk)
30 FoodInspectionsDF$Risk <- gsub(pattern="\\)", replacement="", x=FoodInspectionsDF$Risk)
31 FoodInspectionsDF$Risk <- gsub(pattern="\\.", replacement="", x=FoodInspectionsDF$Risk)
32
33 #creating dataframes to load data into tables
34 InspectionDF <- data.frame(FoodInspectionsDF[c("InspectionID", "InspectionDate", "InspectionType", "Results", "Risk", "Viol")])
35
```

Fig2. Data being cleaned and manipulated in RStudio



The screenshot shows the RStudio interface with the final dataframe ready for storage in SQL database. The dataframe is displayed in the console window. The data is as follows:

InspectionID	InspectionDate	InspectionType	Results	Risk	Viol
204	1751505	ROMERO'S BURGER BULL			
205	1751500	LAS ASADAS			
206	1751235	PRINCETON APARTMENTS			
207	1751504	MS. EGGROLL			
208	1751497	LAS ISLAS MARIAS			
209	1689305	STARBUCKS COFFEE			
210	1762212	CUGINI PIZZA & EATERY			
211	1751494	TAQUERIA UPTOWN			
212	1751491	B & P'S SWEETS AND TREATS			
213	1751487	CHAPO'S PIZZA			
214	1751488	ILLINOIS BAR & GRILL INC			
215	1751485	ALDEN NORTHMOOR REHAB & HLTH			
216	1751478	MR SUBMARINE			
217	1751483	SEDGWICK'S			
218	1751471	MCDONALDS			
219	1751388	EL REY DEL TACO & BURRITOS			
220	1751474	IRAZU INC			
221	1751467	LA PANE HALAL PIZZA			
222	1762222	PIZZA RUSTICA, INC			
223	1751469	EL PESCADOR RESTAURANT			
224	1689304	XANDO COFFEE & BAR / COSI SANDWICH BAR			
225	1751460	TAQUERIA EL CHORRITO			
226	1751450	AUNTIE VEE'S CUPCAKES & DESSERTS			
227	1751445	WHITE CASTLE #63			
228	1751448	INTERNATIONAL NURSING & REHAB CENTER			
229	1751446	TANOSHI SUSHI MIKES			
230	1751443	BRU COFFEEWORKS			
231	1751438	GROWING SEEDS PRESCHOOL, LLC			
232	1751377	JUE WEI YA BA WANG INC.			
233	1751372	SMACK SHACK			
234	1689303	RONNY'S ORIGINAL STEAK HOUSE			
235	1762213	PIZZACO'S ITALIAN EATERY			
236	1751436	SUBWAY			
237	1751434	THE ORIGINAL SOUL VEGETARIAN			
238	1751432	INSPIRATION CORPORATION			
239	1751431	42 grams			
240	1751429	SP KEBAB			
241	1751425	FIREHOUSE CAPTAIN HOOKS			
242	1751423	CRUZ BLANCA AND LENA BRAVA			

Fig3. The final dataframe ready for storage in SQL database

2. DATA STORAGE :

Below is the schema and model for housing the data. I preferred traditional SQL database for this project due to its flexibility and reliability. The schema consists of two tables one is FoodEstablishment where License is the primary key and InspectionID serve as foreign key. The second table is named Inspection, and InspectionID as its primary key. These tables have one to many relationships.

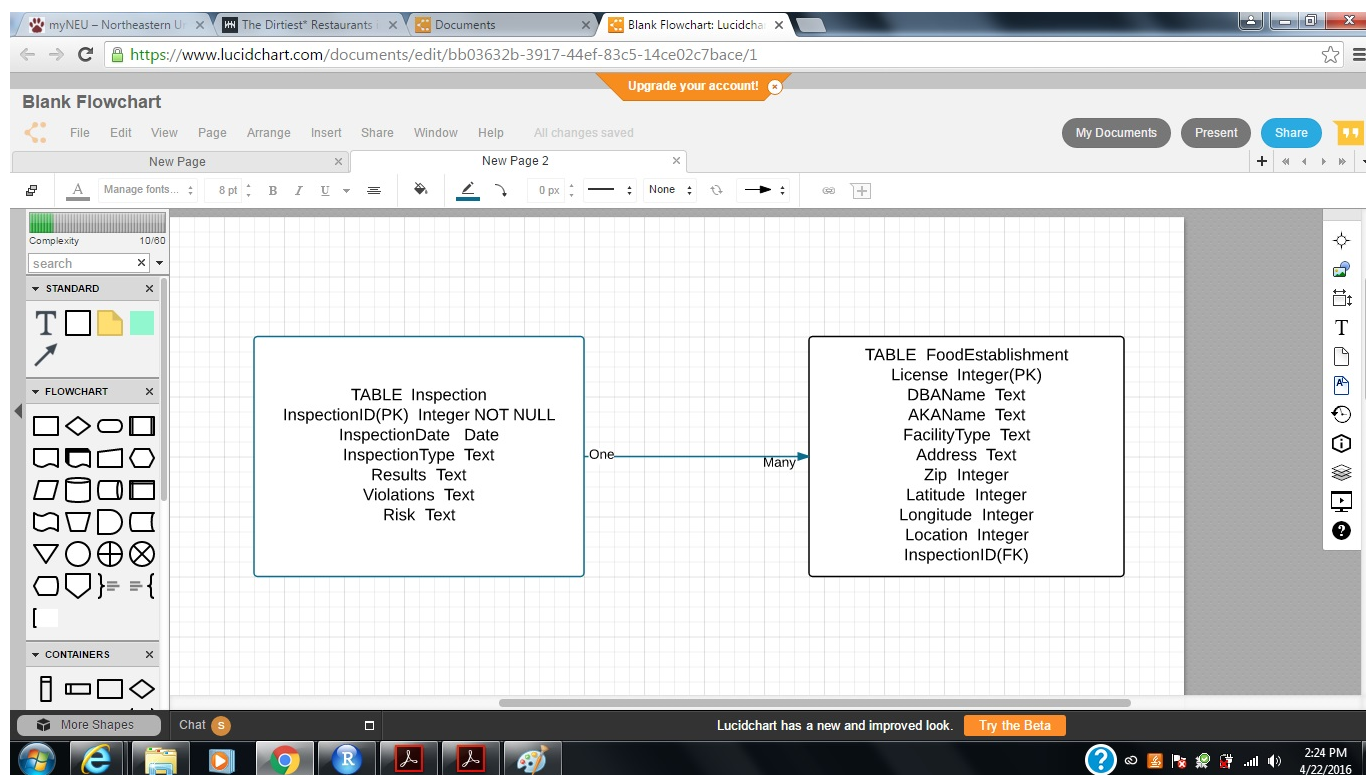


Fig3. SQL visual data model and schema where two tables have one to many relationship

```
#open a connection to SQLite and create the FoodInspectionsDB database
db <- dbconnect(SQLite(), dbname="FoodInspectionsDB.sqlite")

#In SQLite foreign key constraints are disabled by default, so they must be enabled for
#each database connection separately by turning pragma foreign_keys-on
dbSendQuery(conn = db, "pragma foreign_keys=on;")

#creating table Inspection with InspectionID as the primary key
dbSendQuery(conn = db, "CREATE TABLE Inspection (
  InspectionID INTEGER NOT NULL PRIMARY KEY,
  InspectionDate DATE,
  InspectionType TEXT,
  Results TEXT,
  Violations TEXT)
WITHOUT ROWID")

#creating table FoodEstablishment with License as primary key and InspectionID as foreign key
dbSendQuery(conn = db, "CREATE TABLE FoodEstablishment (
  License INTEGER NOT NULL PRIMARY KEY,
  DBAName TEXT,
  AKAName TEXT,
  FacilityType TEXT,
  Address TEXT,
  Zip INTEGER,
  Latitude INTEGER,
  Longitude INTEGER,
  Location INTEGER,
  InspectionID INTEGER NOT NULL,
  FOREIGN KEY(InspectionID) REFERENCES Inspection(InspectionID)
WITHOUT ROWID")

#uploading data into tables
dbWriteTable(conn = db, name = "Inspection", value = InspectionDF, row.names=FALSE, overwrite=TRUE)
```

Fig4. Creating tables in RStudio using SQLite package

3. DATA RETRIEVAL: By utilising the powerful and flexible data retrieval mechanism of SQL using "SELECT" statements, here are some of the interesting results pertaining to the project. The three scenarios that I chose to retrieve the information about are 1) comparing the total number of failed inspections from 2010 till 2015 2) the top ten facility types who failed the inspections in 2015 3) top ten areas(zip) having the highest number of failed inspections.

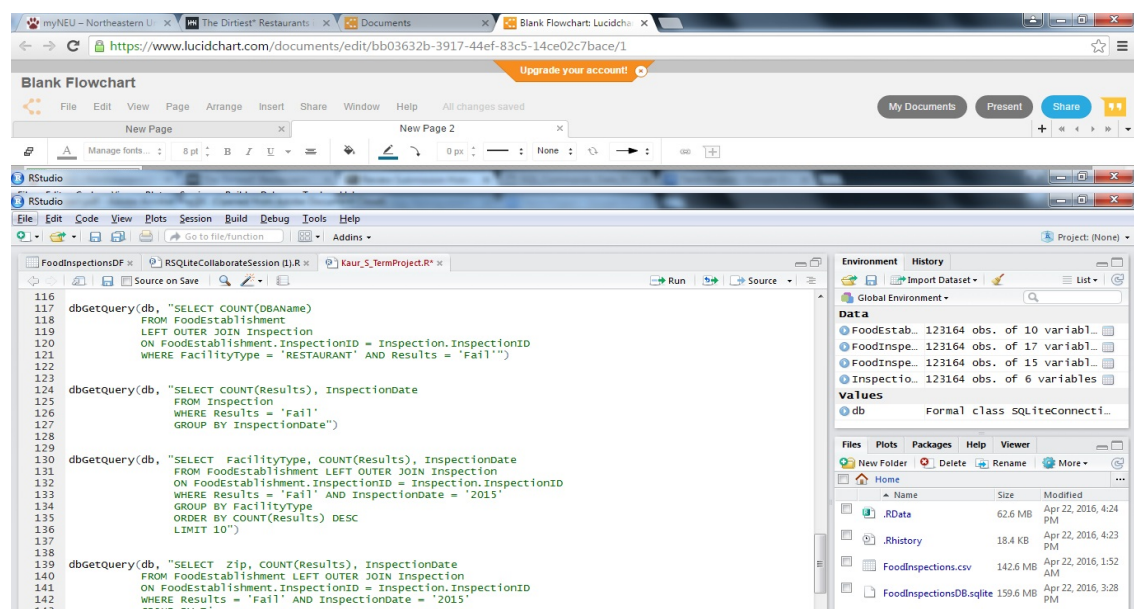


Fig5. Running the queries in RStudio

Table 1

	COUNT(Results)	InspectionDate
1	4504	2010
2	4357	2011
3	3642	2012
4	3349	2013
5	3719	2014
6	3606	2015

Table 3

Zip	COUNT(Results)	InspectionDate
1 60618	165	2015
2 60608	145	2015
3 60607	144	2015
4 60619	142	2015
5 60614	125	2015
6 60657	125	2015
7 60611	124	2015
8 60622	117	2015
9 60632	107	2015
10 60640	107	2015

Table 2

FacilityType	COUNT(Results)	InspectionDate
1 Restaurant	2478	2015
2 Grocery Store	478	2015
3 School	231	2015
4 Children's Services Facility	60	2015
5 Bakery	56	2015
6 Daycare (2 - 6 Years)	40	2015
7 Daycare Above and Under 2 Years	31	2015
8 Long Term Care	30	2015
9 Catering	23	2015
10 Liquor	23	2015

Table 1, 2 and 3 are the results pertaining to the three scenarios respectively.

CONCLUSION

The data demonstrate that the number of failed inspections from 2010 to 2015 of food establishments varied substantially over time. It's no surprise to see the restaurants at the top of the list of facility type with the highest number of failed inspections. Though this information empowers people to make more informed decisions as consumers giving them more freedom to base their decisions where to dine ,on not only the quality of the service but also on the establishments adherence to food safety practices. There is an another aspect, may be it can change the way local restaurant industry operates, since increased availability of this information can derive them to improve their food safety practices to attract customers. The specific areas of Chicago with the most failed inspections and violations is a good indication to access the involvement of environmental factors, sanitation of neighborhood etc that might be the culprits of failed inspections.

Although the dataset is easily available and convenient to download and upload in RStudio but the quality of data is little bit compromised due to missing values, repeated values, special characters etc. To begin with, I first checked all the variables and categories to examine how much messy it was and then clean up accordingly. Specifically the Food Establishment field includes mixed values like Restaurant, RESTAURANT, RESTAURANT/BAR. Restaurant/BAR, SCHOOL, School etc. That column needed lot of cleaning otherwise that would had altered the number(Counts) related to specific Facility type and other queries. Developing the schema and model for database was pretty simple and my choice of using SQL database was based on the reliability and consistency of database. Moreover the dataset in itself has InspectionID representing each unique rows, and used as primary key for one table and foreign key for another one.

I am sure there are more efficient ways to query database to get the required information which can be use to make more productive public health models and policies or increase the efficiency of food inspections protocol. Like the data can be used to predict which food establishments were most likely to be in violation of health codes, based on characteristics of previously recorded violations. It can be used to prioritize the food establishments for inspection to prevent consumers from gut bombing sickness. It would be more interesting to see the future implications of this data for public health protocol and general public access.

REFERENCES

www.cityofchicago.org

www.ncbi.nlm.nih.gov

www.washingtonpost.com

www.chicagotribune.com