Restaurant Data

Eric He

Here's the restaurant data. It was taken from https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j/data on 3/2/17 with first data point having a CAMIS of 41618312 (Country Boyz Jerk Yard Restaurant). The source dataset updates in real time.

The dataset records the violations a restaurant was found to be committing during an inspection. Each row corresponds to a different violation discovered within an inspection; the columns give information on the restaurant being inspected such as name, location, phone number, and cuisine as well as the date of the inspection and the specific violation and its description. Thus, a single inspection can have many different rows, since an inspection frequently turns up many violations. Moreover, a restaurant can be inspected several times across different dates, and there are many chain restaurants with the same name, but situated in different locations across the city. Thus, a single restaurant can correspond to hundreds of rows within the data!

We are not interested in analyzing different violations, but in different restaurants. How do different restaurant classes perform in inspections? Are certain locations more prone to poor practices than others (inspection data may not be a good tool for this, since an area with harsher inspectors will have more violations recorded, but such an area may be better off because of it; domain knowledge is important here)? Are there food deserts within the city?

Our target classes of interest are the scores and grades of restaurants and restaurant inspections, but the current data is formatted by violations, not inspections or restaurants. Thus, we will have to transform the data before we begin our analysis.

There was a fatal parse error in row 17 of the originally downloaded dataset due to an unreadable character, which had to be deleted.

```
library(readr)
RestaurantData <- read_csv("RestaurantData.csv")
```

```
## Parsed with column specification:
## cols(
##   CAMIS = col_integer(),
##   DBA = col_character(),
##   BORO = col_character(),
##   BUILDING = col_character(),
##   STREET = col_character(),
##   ZIPCODE = col_integer(),
##   PHONE = col_double(),
##   `CUISINE DESCRIPTION` = col_character(),
##   `INSPECTION DATE` = col_character(),
```

```
##   ACTION = col_character(),
##   `VIOLATION CODE` = col_character(),
##   `VIOLATION DESCRIPTION` = col_character(),
##   `CRITICAL FLAG` = col_character(),
##   SCORE = col_integer(),
##   GRADE = col_character(),
##   `GRADE DATE` = col_character(),
##   `RECORD DATE` = col_character(),
##   `INSPECTION TYPE` = col_character()
## )

## Warning: 203 parsing failures.
##  row   col               expected     actual                 file
## 2160 PHONE no trailing characters _688_16    'RestaurantData.csv'
## 4023 PHONE a double               __________ 'RestaurantData.csv'
## 4636 PHONE a double               __________ 'RestaurantData.csv'
## 4865 PHONE a double               __________ 'RestaurantData.csv'
## 7334 PHONE a double               __________ 'RestaurantData.csv'
## .... ..... ...................... .......... ...................
## See problems(...) for more details.

View(RestaurantData)
```

Here's a look at the data.

```
head(RestaurantData)

## # A tibble: 6 × 18
##      CAMIS                               DBA      BORO BUILDING
##      <int>                             <chr>     <chr>    <chr>
## 1 41618312 COUNTRY BOYZ JERK YARD RESTAURANT     BRONX     1182
## 2 41710976            SWEET MAMA'S SOUL FOOD MANHATTAN      689
## 3 41569547                          DOMINO'S     BRONX     3869
## 4 50017534                NEW FOREST CAFE  BROOKLYN     7617
## 5 41105757             EL BARRIO RESTAURANT MANHATTAN      158
## 6 40718835                          TRIOMPHE MANHATTAN       49
## # ... with 14 more variables: STREET <chr>, ZIPCODE <int>, PHONE <dbl>,
## #   `CUISINE DESCRIPTION` <chr>, `INSPECTION DATE` <chr>, ACTION <chr>,
## #   `VIOLATION CODE` <chr>, `VIOLATION DESCRIPTION` <chr>, `CRITICAL
## #   FLAG` <chr>, SCORE <int>, GRADE <chr>, `GRADE DATE` <chr>, `RECORD
## #   DATE` <chr>, `INSPECTION TYPE` <chr>
```

Attach everything here.

```
colnames(RestaurantData) <- c("camis", "name", "boro", "building", "street",
"zipcode", "phone", "cdescrip", "idate", "action", "vcode", "vdescrip",
"flag", "score", "grade", "gdate", "rdate", "type")
attach(RestaurantData)
```

Create a new column which concatenates the restaurant name, borough, building, street, zipcode, and inspection date together to allow for identifying individual inspections.

```
inspection <- paste(name, boro, paste(building, street, sep = " "), zipcode,
idate, sep = ", ")
RestaurantData <- data.frame(inspection, RestaurantData)
inspectionid <- as.numeric(RestaurantData$inspection)
RestaurantData <- data.frame(inspectionid, RestaurantData)
rm(inspection, inspectionid)
```

Sort the data so that all rows corresponding to a single inspection are together.

```
RestaurantData <- RestaurantData[order(RestaurantData$inspectionid,
RestaurantData$score),]
```

Transform the data to allow for analysis.

```
RestaurantData$boro <- as.factor(RestaurantData$boro)
RestaurantData$cdescrip <- as.factor(RestaurantData$cdescrip)
RestaurantData$idate <- as.Date(RestaurantData$idate, format = "%m/%d/%y")
RestaurantData$vdescrip <- as.factor(RestaurantData$vdescrip)
RestaurantData$vcode <- as.factor(RestaurantData$vcode)
RestaurantData$grade <- as.factor(RestaurantData$grade)
```

Now that the data is properly ordered, we can build the datasets we need to perform our analysis. The first is the inspection matrix, a table where every row corresponds to a different inspection and every column corresponds to one of 93 possible violations the restaurant can commit.

The second matrix we want is the identification matrix. Once again, each row of the matrix corresponds to a different inspection. The columns contain the relevant restaurant information.

```
inspectionMatrix <- table(RestaurantData$inspectionid, RestaurantData$vcode)
identificationMatrix <- RestaurantData[!duplicated(RestaurantData[,1]),]
#remove rows with the same inspectionid
identificationMatrix <- identificationMatrix[,-c(20, 19, 12:15)]
#remove irrelevant columns
head(inspectionMatrix)
```

```
##
##     02A 02B 02C 02D 02E 02F 02G 02H 02I 02J 03A 03B 03C 03D 03E 03F 03G
##   1   0   1   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0
##   2   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
##   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   4   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
##   5   0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0
##   6   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
##
##     04A 04B 04C 04D 04E 04F 04G 04H 04I 04J 04K 04L 04M 04N 04O 05A 05B
##   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   2   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
##   3   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0
##   4   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
```

```
##   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##
##     05C 05D 05E 05F 05H 05I 06A 06B 06C 06D 06E 06F 06G 06H 06I 07A 08A
##   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   2   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   1
##   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
##   4   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1
##   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   6   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   0   0
##
##     08B 08C 09A 09B 09C 10A 10B 10C 10D 10E 10F 10G 10H 10I 10J 15E 15H
##   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0
##   2   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   3   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
##   4   0   0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0
##   5   0   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0
##   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##
##     15I 15J 15K 15L 15S 15T 16A 16B 16C 16E 16F 18B 18C 18D 18F 20A 20B
##   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   2   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
##   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##
##     20D 20E 20F 22A 22B 22C 22E 22F 22G
##   1   0   0   0   0   0   0   0   0   0
##   2   0   0   0   0   0   0   0   0   0
##   3   0   0   0   0   0   0   0   0   0
##   4   0   0   0   0   0   0   0   0   0
##   5   0   0   0   0   0   0   0   0   0
##   6   0   0   0   0   0   0   0   0   0
```

```
head(identificationMatrix)
```

```
##        inspectionid                                                inspection
## 19457             1 ''W'' CAFE, MANHATTAN, 390 5TH AVE, 10018, 1/29/2015
## 39536             2  ''W'' CAFE, MANHATTAN, 390 5TH AVE, 10018, 1/7/2015
## 59722             3 ''W'' CAFE, MANHATTAN, 390 5TH AVE, 10018, 12/3/2015
## 70042             4 ''W'' CAFE, MANHATTAN, 390 5TH AVE, 10018, 2/23/2017
## 28371             5  ''W'' CAFE, MANHATTAN, 390 5TH AVE, 10018, 3/2/2015
## 158992            6 ''W'' CAFE, MANHATTAN, 390 5TH AVE, 10018, 7/28/2015
##           camis       name      boro building  street zipcode      phone
## 19457  50018480 ''W'' CAFE MANHATTAN      390 5TH AVE   10018 2125639444
## 39536  50018480 ''W'' CAFE MANHATTAN      390 5TH AVE   10018 2125639444
## 59722  50018480 ''W'' CAFE MANHATTAN      390 5TH AVE   10018 2125639444
## 70042  50018480 ''W'' CAFE MANHATTAN      390 5TH AVE   10018 2125639444
## 28371  50018480 ''W'' CAFE MANHATTAN      390 5TH AVE   10018 2125639444
```

```
## 158992 50018480 ''W'' CAFE MANHATTAN      390 5TH AVE   10018 2125639444
##        cdescrip      idate score grade     gdate
## 19457  American 2020-01-29    28     C 1/29/2015
## 39536  American 2020-01-07    40  <NA>      <NA>
## 59722  American 2020-12-03    12     A 12/3/2015
## 70042  American 2020-02-23    35  <NA>      <NA>
## 28371  American 2020-03-02    20  <NA>      <NA>
## 158992 American 2020-07-28    22  <NA>      <NA>
```

The inspection matrix currently has columns corresponding to different violation codes. Let's build a dictionary for the violation codes.

```
dictionary <- data.frame(RestaurantData$vcode, RestaurantData$vdescrip)
dictionary <- unique(dictionary)
dictionary <- dictionary[order(dictionary$RestaurantData.vcode),]
dictionary <- dictionary[-nrow(dictionary),]
#remove NA row
head(dictionary)
```

```
##       RestaurantData.vcode
## 759                    02A
## 1                      02B
## 1653                   02C
## 31846                  02D
## 82046                  02E
## 31722                  02F
##
RestaurantData.vdescrip
## 759
Food not cooked to required minimum temperature.
## 1
Hot food item not held at or above 140Â° F.
## 1653                 Hot food item that has been cooked and refrigerated
is being held for service without first being reheated to 1 65Â° F or above
within 2 hours.
## 31846 Precooked potentially hazardous food from commercial food processing
establishment that is supposed to be heated, but is not heated to 140Â° F
within 2 hours.
## 82046                                                  Whole frozen poultry or
poultry breasts, other than a single portion, is being cooked frozen or
partially thawed.
## 31722                                                                Meat,
fish or molluscan shellfish served raw or undercooked without prior
notification to customer.
```

We have tables for every different inspection. This data is excellent for analysis, so we can export them into csv files for further analysis.

```
write.csv(inspectionMatrix, file = "inspectionMatrix.csv")
write.csv(identificationMatrix, file = "identificationMatrix.csv")
```

```r
write.csv(dictionary, file = "dictionary.csv")
write.csv(RestaurantData, file = "cleanedRestaurantData.csv")
```