homework iv

Ayush Kumar Shah

2020-09-23

Introduction

In this report, we tidy the `nyc311` data by removing the infelicities present in it using the package `tidyr`, which is a member of the `tidyverse` package. We also introduce other related datasets, which is connectable to the nyc311 dataset.

The two additional datasets introduced in this report are:

1. Projected Population 2010-2040 - Total By Age Groups
2. 2005 - 2015 Graduation Outcomes - Department of Education

Both of these datasets were obtained from the NYC OpenData. Also, they both contain the column `Borough` which makes them connectable to the `nyc311` data.

Tidying the data

It is important to have a tidy data for easy analysis and exploration purposes. Generally, the data is untidy as it may be created for easy entry or may have higher performance. However, it is difficult to analyze such data and hence we make them tidy before analysis.

To make the data tidy, we must simply ensure that the data follows these three interrelated rules:

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.

Tidying the nyc311 data

Read the nyc311 data

We load the nyc311 data set. Then we fix the column names of the nyc311 data so that they have no spaces.

```
nyc311<-fread("311_Service_Requests_from_2010_to_Present.csv",
              na.strings=c("","NA"))
names(nyc311)<-names(nyc311) %>%
stringr::str_replace_all("\\s", ".")
# mini311<-nyc311[sample(nrow(nyc311),10000),]
# write_csv(mini311,"mini311.csv")
# sample<-fread("mini311.csv", na.strings=c("","NA"))
```

Viewing the data

Let's view the head of the nyc311 data to guess possible untidiness in the data.

```
pander(head(nyc311))
```

Table 1: Table continues below

Unique.Key	Created.Date	Closed.Date	Agency
30387854	04/14/2015 02:14:40 AM	04/14/2015 03:03:22 AM	NYPD
30388338	04/14/2015 02:10:12 AM	NA	NYPD
30395236	04/14/2015 02:03:01 AM	NA	NYPD
30394595	04/14/2015 02:02:40 AM	NA	NYPD
30390517	04/14/2015 02:00:04 AM	04/14/2015 02:47:33 AM	NYPD
30389560	04/14/2015 01:52:15 AM	04/14/2015 02:11:10 AM	NYPD

Table 2: Table continues below

Agency.Name	Complaint.Type	Descriptor
New York City Police Department	Vending	In Prohibited Area
New York City Police Department	Blocked Driveway	No Access
New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party
New York City Police Department	Noise - Street/Sidewalk	Loud Talking
New York City Police Department	Noise - Street/Sidewalk	Loud Talking
New York City Police Department	Noise - Street/Sidewalk	Loud Talking

Table 3: Table continues below

Location.Type	Incident.Zip	Incident.Address
Street/Sidewalk	10465	3775 EAST TREMONT AVENUE
Street/Sidewalk	11234	1524 RYDER STREET
Street/Sidewalk	11204	NA
Street/Sidewalk	11211	361 METROPOLITAN AVENUE
Street/Sidewalk	10025	NA
Street/Sidewalk	11205	NA

Table 4: Table continues below

Street.Name	Cross.Street.1	Cross.Street.2
EAST TREMONT AVENUE	RANDALL AVENUE	ROOSEVELT AVENUE
RYDER STREET	FLATLANDS AVENUE	AVENUE P
NA	NA	NA

Street.Name	Cross.Street.1	Cross.Street.2
METROPOLITAN AVENUE	HAVEMEYER STREET	HAVEMEYER STREET
NA	NA	NA
NA	NA	NA

Table 5: Table continues below

Intersection.Street.1	Intersection.Street.2	Address.Type	City
NA	NA	ADDRESS	BRONX
NA	NA	ADDRESS	BROOKLYN
71 STREET	16 AVENUE	INTERSECTION	BROOKLYN
NA	NA	ADDRESS	BROOKLYN
WEST 104 STREET	COLUMBUS AVENUE	INTERSECTION	NEW YORK
ST JAMES PLACE	LAFAYETTE AVENUE	INTERSECTION	BROOKLYN

Table 6: Table continues below

Landmark	Facility.Type	Status	Due.Date
NA	Precinct	Closed	04/14/2015 10:14:40 AM
NA	Precinct	Open	04/14/2015 10:10:12 AM
NA	Precinct	Open	04/14/2015 10:03:01 AM
NA	Precinct	Assigned	04/14/2015 10:02:40 AM
NA	Precinct	Closed	04/14/2015 10:00:04 AM
NA	Precinct	Closed	04/14/2015 09:52:15 AM

Table 7: Table continues below

Resolution.Action.Updated.Date	Community.Board	Borough
04/14/2015 03:03:05 AM	10 BRONX	BRONX
NA	18 BROOKLYN	BROOKLYN
NA	11 BROOKLYN	BROOKLYN
04/14/2015 02:10:32 AM	01 BROOKLYN	BROOKLYN
04/14/2015 02:04:59 AM	07 MANHATTAN	MANHATTAN
04/14/2015 02:11:10 AM	02 BROOKLYN	BROOKLYN

Table 8: Table continues below

X.Coordinate.(State.Plane)	Y.Coordinate.(State.Plane)	Park.Facility.Name
1033758	240162	Unspecified
1001544	164726	Unspecified
984678	164647	Unspecified
996477	199445	Unspecified
994260	229982	Unspecified
994009	190054	Unspecified

Table 9: Table continues below

Park.Borough	School.Name	School.Number	School.Region	School.Code
BRONX	Unspecified	Unspecified	Unspecified	Unspecified
BROOKLYN	Unspecified	Unspecified	Unspecified	Unspecified
BROOKLYN	Unspecified	Unspecified	Unspecified	Unspecified
BROOKLYN	Unspecified	Unspecified	Unspecified	Unspecified
MANHATTAN	Unspecified	Unspecified	Unspecified	Unspecified
BROOKLYN	Unspecified	Unspecified	Unspecified	Unspecified

Table 10: Table continues below

School.Phone.Number	School.Address	School.City	School.State
Unspecified	Unspecified	Unspecified	Unspecified
Unspecified	Unspecified	Unspecified	Unspecified
Unspecified	Unspecified	Unspecified	Unspecified
Unspecified	Unspecified	Unspecified	Unspecified
Unspecified	Unspecified	Unspecified	Unspecified
Unspecified	Unspecified	Unspecified	Unspecified

Table 11: Table continues below

School.Zip	School.Not.Found	School.or.Citywide.Complaint	Vehicle.Type
Unspecified	N	NA	NA
Unspecified	N	NA	NA
Unspecified	N	NA	NA
Unspecified	N	NA	NA
Unspecified	N	NA	NA
Unspecified	N	NA	NA

Table 12: Table continues below

Taxi.Company.Borough	Taxi.Pick.Up.Location	Bridge.Highway.Name
NA	NA	NA
NA	NA	NA
NA	NA	NA
NA	NA	NA
NA	NA	NA
NA	NA	NA

Table 13: Table continues below

Bridge.Highway.Direction	Road.Ramp	Bridge.Highway.Segment
NA	NA	NA
NA	NA	NA
NA	NA	NA
NA	NA	NA

Bridge.Highway.Direction	Road.Ramp	Bridge.Highway.Segment
NA	NA	NA
NA	NA	NA

Table 14: Table continues below

Garage.Lot.Name	Ferry.Direction	Ferry.Terminal.Name	Latitude	Longitude
NA	NA	NA	40.83	-73.82
NA	NA	NA	40.62	-73.94
NA	NA	NA	40.62	-74
NA	NA	NA	40.71	-73.96
NA	NA	NA	40.8	-73.96
NA	NA	NA	40.69	-73.96

Location
(40.8257259931145, -73.82111429330192)
(40.618794391821936, -73.93770589155426)
(40.61859442131066, -73.99845832101916)
(40.71409874640673, -73.95589458206499)
(40.79791780509379, -73.96384631347463)
(40.68832571866554, -73.96481079590191)

Checking duplicates

We check duplicates by first removing the `Unique.Key` variable since all the values are unique in the column.

Since `all_equal()` takes a very long time to compare the two data frames, we simply compare the number of rows in the main and non duplicated data frames using `nrow()`.

```
## Number of rows in original nyc311 dataframe = 9124937
```

```
##
## Number of rows in non duplicated nyc311 dataframe = 8271399
```

```
##
## Duplicate observations present = TRUE
```

We can see that there are duplicate observations. So, we use the non duplicated data frame created above `nyc311nodups` in the further steps.

Remove unspecified Borough.

```
# View the Borough counts
nyc311nodups %>%
  group_by(Borough) %>%
  summarize(count = n()) %>%
  arrange(desc(count)) %>%
  pander()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

Borough	count
BROOKLYN	2288036
QUEENS	1863259
MANHATTAN	1547810
BRONX	1277987
Unspecified	859858
STATEN ISLAND	434449

```
# Remove rows with Unspecified Borough
nyc311_b <- nyc311nodups %>%
  filter(Borough != "Unspecified")
```

We can see that there is a significant number of observations with Unspecified Borough. Hence, those observations have been removed using `filter()`.

Separating Created.Date to mulitple Columns

- We separate `Created.Date` into columns `Year`, `Month`, `Day`, and `Time`. However, we do not remove the original variable `Created.Date` since it may be used in the calculation of response time later.
- We again calculate `Hours` from the `Time` variable using `POSIXlt` class.

This separation is done so that we can easily analyze different trends in the data variables later based on year, month, day or hour of day.

```
nyc311_time <- nyc311_b %>%
  separate(Created.Date, into = c("Created.Month", "Created.Day", "Created.Year"),
           sep = "/", convert = TRUE, remove = FALSE) %>%
  separate(Created.Year, into = c("Created.Year", "time", "Period"),
           sep=" ", convert = TRUE) %>%
  unite(Created.Time, time, Period, sep = " ") %>%
  mutate(Created.Hour = as.POSIXlt(Created.Time, format="%I:%M:%S %p")$hour)

nyc311_time %>%
  select(Created.Date, Created.Year, Created.Month, Created.Day, Created.Time, Created.Hour) %>%
  head(10) %>%
  pander()
```

Table 17: Table continues below

Created.Date	Created.Year	Created.Month	Created.Day
04/14/2015 02:14:40 AM	2015	4	14
04/14/2015 02:10:12 AM	2015	4	14
04/14/2015 02:03:01 AM	2015	4	14
04/14/2015 02:02:40 AM	2015	4	14
04/14/2015 02:00:04 AM	2015	4	14
04/14/2015 01:52:15 AM	2015	4	14
04/14/2015 01:47:31 AM	2015	4	14
04/14/2015 01:45:31 AM	2015	4	14
04/14/2015 01:44:15 AM	2015	4	14
04/14/2015 01:43:17 AM	2015	4	14

Created.Time	Created.Hour
02:14:40 AM	2
02:10:12 AM	2
02:03:01 AM	2
02:02:40 AM	2
02:00:04 AM	2
01:52:15 AM	1
01:47:31 AM	1
01:45:31 AM	1
01:44:15 AM	1
01:43:17 AM	1

The above table shows a sample of the data frame after applying the operations mentioned above.

Remove Columns

Redundant columns

Let's view some columns which have redundant information.

```
nyc311_time %>%
  select(Street.Name, Incident.Address, Latitude, Longitude, Location,
         Facility.Type, Location.Type, Borough, Park.Borough, Community.Board) %>%
  head() %>%
  pander()
```

Table 19: Table continues below

Street.Name	Incident.Address	Latitude	Longitude
EAST TREMONT AVENUE	3775 EAST TREMONT AVENUE	40.83	-73.82
RYDER STREET	1524 RYDER STREET	40.62	-73.94
NA	NA	40.62	-74
METROPOLITAN AVENUE	361 METROPOLITAN AVENUE	40.71	-73.96

Street.Name	Incident.Address	Latitude	Longitude
NA	NA	40.8	-73.96
NA	NA	40.69	-73.96

Table 20: Table continues below

Location	Facility.Type	Location.Type	Borough
(40.8257259931145, -73.82111429330192)	Precinct	Street/Sidewalk	BRONX
(40.618794391821936, -73.93770589155426)	Precinct	Street/Sidewalk	BROOKLYN
(40.61859442131066, -73.99845832101916)	Precinct	Street/Sidewalk	BROOKLYN
(40.71409874640673, -73.95589458206499)	Precinct	Street/Sidewalk	BROOKLYN
(40.79791780509379, -73.96384631347463)	Precinct	Street/Sidewalk	MANHATTAN
(40.68832571866554, -73.96481079590191)	Precinct	Street/Sidewalk	BROOKLYN

Park.Borough	Community.Board
BRONX	10 BRONX
BROOKLYN	18 BROOKLYN
BROOKLYN	11 BROOKLYN
BROOKLYN	01 BROOKLYN
MANHATTAN	07 MANHATTAN
BROOKLYN	02 BROOKLYN

Columns with very few data

Let's view the counts of the non empty values in each column. We only display the columns which have counts less than 65% of the total observations.

```
non_na_count <- data.frame(colSums(!is.na(nyc311_time)))
colnames(non_na_count) <- "Non.NA.Count"
non_na_count %>%
  arrange(Non.NA.Count) %>%
  filter(Non.NA.Count < 0.65 * nrow(nyc311_time)) %>%
  pander()
```

	Non.NA.Count
Ferry.Direction	1
School.or.Citywide.Complaint	2768
Garage.Lot.Name	4038
Ferry.Terminal.Name	7593
Vehicle.Type	7678
Landmark	8465

	Non.NA.Count
Taxi.Company.Borough	9237
Bridge.Highway.Segment	31431
Road.Ramp	31490
Bridge.Highway.Name	31944
Bridge.Highway.Direction	35352
Taxi.Pick.Up.Location	88382
Intersection.Street.2	1628437
Intersection.Street.1	1628589
School.Not.Found	2359846
Due.Date	2448873

Removing the columns

Let's remove the redundant columns, the columns with very less non null data and also the columns which are not relevant or useful.

```
nyc311_clean <-
  nyc311_time %>%
  select(-c(Street.Name, Location, Facility.Type, Resolution.Action.Updated.Date,
            `X.Coordinate.(State.Plane)`, `Y.Coordinate.(State.Plane)`,
            Park.Borough, Community.Board, Ferry.Direction, Garage.Lot.Name,
            Landmark, Ferry.Terminal.Name, Vehicle.Type, Taxi.Company.Borough,
            Bridge.Highway.Name, Road.Ramp,
            Bridge.Highway.Segment, Bridge.Highway.Direction, Taxi.Pick.Up.Location,
            Intersection.Street.1, Intersection.Street.2), -c(33:43))
```

Viewing columns of the tidied nyc311 dataset

```
pander(data.frame(colnames(nyc311_clean)))
```

colnames.nyc311_clean.
Created.Date
Created.Month
Created.Day
Created.Year
Created.Time
Closed.Date
Agency
Agency.Name
Complaint.Type
Descriptor
Location.Type
Incident.Zip
Incident.Address
Cross.Street.1
Cross.Street.2
Address.Type
City

colnames.nyc311_clean.
Status
Due.Date
Borough
Park.Facility.Name
School.Name
Latitude
Longitude
Created.Hour

Save the tidied nyc311 dataset

```
write_csv(nyc311_clean, 'tidied_nyc311.csv')
```

Other datasets

Read the datasets

The two additional datasets introduced in this report are:

1. Projected Population 2010-2040 - Total By Age Groups

Projected total New York City population for five intervals from 2010 through 2040 by Borough, broken down by 18 age cohorts. (Age groups may not add up to the total due to rounding.)

This dataset is introduced so that the population and age group information in the Borough can be known in correlation to the complaints in the nyc311 data.

2. 2005 - 2015 Graduation Outcomes - Department of Education

Graduation results for all students by year; cohorts of 2001 through 2011 (Classes of 2005 through 2015). Graduation Outcomes as Calculated by the New York State Education Department. The New York State calculation method was first adopted for the Cohort of 2001 (Class of 2005).

Graduates are defined as those students earning either a Local or Regents diploma and exclude those earning either a special education (IEP) diploma or GED.

This dataset is introduced so that the educational status of the people in different Borough is available for further analysis in correlation with the nyc311 data.

```
nyc_popn <- read_csv('Projected_Population_2010-2040_-_Total_By_Age_Groups.csv')
```

```
## Parsed with column specification:
## cols(
##   Borough = col_character(),
##   Age = col_character(),
##   `2010` = col_double(),
##   `2015` = col_double(),
##   `2020` = col_double(),
##   `2025` = col_double(),
##   `2030` = col_double(),
##   `2035` = col_double(),
##   `2040` = col_double()
## )
```

```
## 2005-15 graduation
nyc_grad <- read_csv('https://data.cityofnewyork.us/resource/qk7d-gecv.csv')
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   cohort_category = col_character(),
##   demographic = col_character()
## )

## See spec(...) for full column specifications.
```

Viewing the datasets

Table 24: Projected_Population 2010-2040 By AgeGroups

Borough	Age	2010	2015	2020	2025	2030	2035	2040
NYC Total	0-4	521990	535209	545778	547336	542426	540523	546426
NYC Total	15-19	539844	505783	492532	519298	535024	546062	546750
NYC Total	20-24	647483	646075	606203	591683	625253	643728	657403
NYC Total	25-29	736105	770396	763956	715824	698195	740437	762757
NYC Total	30-34	667657	707726	743916	740268	693684	675497	715486
NYC Total	35-39	592299	611239	649594	684249	682964	639237	621899

```
nyc_grad %>%
  select(1:5) %>%
  head() %>%
  pander(caption="2005-2015 Graduation Outcomes - Department of Education")
```

Table 25: 2005-2015 Graduation Outcomes - Department of Education (continued below)

cohort_year	cohort_category	demographic	total_cohort
2001	4 Year June	English Language Learner	10540
2001	5 Year June	English Language Learner	10540
2001	6 Year	English Language Learner	10540
2002	4 Year June	English Language Learner	7454
2002	5 Year June	English Language Learner	7454
2002	6 Year	English Language Learner	7454

total_grads
2791
3920
4296
1691
2354
2729

Tidying the population dataset `nycpopn`

Gathering the years

We can see that the years need to be gathered together in the `nyc_popn` data.

```
nyc_popn_tidy <-
  nyc_popn %>%
  gather('2010':'2040', key="Year", value="Population")

pander(head(nyc_popn_tidy, 10), caption = "Tidied Population data")
```

Table 27: Tidied Population data

Borough	Age	Year	Population
NYC Total	0-4	2010	521990
NYC Total	15-19	2010	539844
NYC Total	20-24	2010	647483
NYC Total	25-29	2010	736105
NYC Total	30-34	2010	667657
NYC Total	35-39	2010	592299
NYC Total	40-44	2010	571825
NYC Total	45-49	2010	570273
NYC Total	50-54	2010	546204
NYC Total	55-59	2010	479661

Conclusion

Hence, we tidied the `nyc311` data by removing duplicate values, gathering and spreading required columns like Created.Date using `tudyr` package. We also removed the redundant columns and columns with very little or irrelevant information.

Finally, we also loaded two related datasets which have important population and education status information useful for further analysis later in connection with the original `nyc311` dataset. The population dataset was also tidied by gathering the years.