homework v

Nirbhay Pherwani, Dhiren Chandnani

2019-10-08

Preparing data for HW V

Preprocessing Steps for NYC 311 from HW IV (Explainations Omitted)

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
nyc311<-fread("311_Service_Requests_from_2010_to_Present.csv")
names(nyc311)<-names(nyc311) %>%
  stringr::str_replace_all("\\s", ".")

# removing unwanted columns
nyc311 <- subset(nyc311, select= -c(Landmark, Park.Borough, School.City, School.State, School.Zip, Taxi

# removing complaints with count < 60
nyc311<-nyc311 %>%
  group_by(Complaint.Type) %>%
  filter(n() >= 60)
```

```r
# making complaint values uppercase
library(dplyr)
nyc311$Complaint.Type <- toupper(nyc311$Complaint.Type)
nyc311$Complaint.Type <- gsub('/', ' ', nyc311$Complaint.Type)
nyc311$Complaint.Type <- gsub('-', ' ', nyc311$Complaint.Type)

# removing location types with count < 10
nyc311<-nyc311 %>%
  group_by(Location.Type) %>%
  filter(n() >= 10)

# removing agencies with count < 10
nyc311<-nyc311 %>%
  group_by(Agency) %>%
  filter(n() >= 10)

# splitting the community board field
nyc311<-nyc311 %>% separate(Community.Board, c("Community.Code"), sep=" ", extra = "drop")

# splitting incident address
nyc311<-nyc311 %>% separate(Incident.Address, c("Incident.Code"), sep=" ",extra = "drop")

# cleaning zipcodes
library(zipcode)
nyc311$Incident.Zip<-clean.zipcodes(nyc311$Incident.Zip)

# changing case for city
library(dplyr)
nyc311$City <- toupper(nyc311$City)

# replacing missing values with NA
nyc311 <- nyc311 %>% mutate_all(na_if,"N/A")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `Agency`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```

```r
nyc311 <- nyc311 %>% mutate_all(na_if,"")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `Agency`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```

```r
nyc311 <- nyc311 %>% mutate_all(na_if," ")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `Agency`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```

```r
nyc311 <- nyc311 %>% mutate_all(na_if,"N / A")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `Agency`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```

```
nyc311_backup <- nyc311
```

Preprocessing Steps for Housing NY Units by Building Data from HW IV (Explainations Omitted)

```
nycHousingData <-fread("https://data.cityofnewyork.us/api/views/hg8x-zxpr/rows.csv")
```

```
## Warning in require_bit64(): Some columns are type 'integer64' but
## package bit64 is not installed. Those columns will print as strange
## looking floating point data. There is no need to reload the data. Simply
## install.packages('bit64') to obtain the integer64 print method and print
## the data again.
```

```
names(nycHousingData)<-names(nycHousingData) %>%
  stringr::str_replace_all("\\s", ".")
nycHousingData[nycHousingData=="" | nycHousingData==" "]<-NA

# making borough upper case
nycHousingData$Borough <- toupper(nycHousingData$Borough)
```

Building Units

Spreading Reporting Construction Type

In this section we have spread the reporting construction type field to two different columns Preservation and New Construction to get it's presence which will later be used to find counts per zipcode for each type.

```
nycHousingData$ReportingCount <- rep(1, nrow(nycHousingData))
nycHousingData <- nycHousingData %>% spread(Reporting.Construction.Type, ReportingCount)
nycHousingData$Preservation[is.na(nycHousingData$Preservation)] <- 0
nycHousingData$`New Construction`[is.na(nycHousingData$`New Construction`)] <- 0
```

Summarize Data

In this section we have grouped the data on the basis of postcode(zipcode) and summarized it to get sum values for a number of parameters which we will be using for our analysis in the upcoming homework assignment.

```
buildingUnits<- nycHousingData %>%
group_by(Postcode) %>%
dplyr::summarise(
   TotalUnits = sum(Total.Units),
   BR1Units = sum(`1-BR.Units`),
```

```
  BR2Units = sum(`2-BR.Units`),
  BR3Units = sum(`3-BR.Units`),
  BR4Units = sum(`4-BR.Units`),
  StudioUnits = sum(`Studio.Units`),
  RentalUnits = sum(`Counted.Rental.Units`),
  OwnedUnits = sum(`Counted.Homeownership.Units`),
  PreservationUnits = sum(`Preservation`),
  ConstructionUnits = sum(`New Construction`)
 )
```

Joining both datasets

In this section we have joined the two datasets based on zipcodes.

```
buildingUnits$Postcode<-as.character(buildingUnits$Postcode) # converting df to character type
colnames(buildingUnits)[colnames(buildingUnits)=="Postcode"] <- "Incident.Zip"  # renaming field
joined_data <- nyc311 %>% inner_join(buildingUnits, by = "Incident.Zip") # joining
```

Filtering the joined data

Here we have filtered the data on the HPD type Agency as we noticed that Departement of Housing Preservation and Development (HPD) in NYC would relate more to housing based complaints and it would be interesting to perform further analysis on this data. We also selected the columns we would finally want to perform analysis on.

We will be performing further grouping, for eg: group on zipcode and get number of heating complaints for each zipcode and then see if there is a relation between the different fields we have from the second dataset.

```
joined_data<-filter(joined_data, Agency == "HPD")
plyr::count(joined_data$Complaint.Type)   #displays the counts of unique complaints types
```

```
##                          x   freq
## 1                APPLIANCE  50084
## 2              CONSTRUCTION   4608
## 3               DOOR WINDOW  38804
## 4                  ELECTRIC 172114
## 5                  ELEVATOR    840
## 6           FLOORING STAIRS  26979
## 7                   GENERAL  30617
## 8     GENERAL CONSTRUCTION 472672
## 9            HEAT HOT WATER 244187
## 10                  HEATING 829205
## 11 HPD LITERATURE REQUEST   2439
## 12                NONCONST 244013
## 13        OUTSIDE BUILDING   1755
## 14         PAINT   PLASTER 340855
## 15           PAINT PLASTER  78429
## 16                PLUMBING 449537
## 17                  SAFETY   9441
## 18   UNSANITARY CONDITION  75688
## 19              WATER LEAK  38033
```

```r
# selection
joined_data <- select(joined_data, Agency, Complaint.Type, Descriptor, Incident.Zip, Borough, Status, C
```

Data Extract

```r
glimpse(joined_data)
```

```
## Observations: 3,110,300
## Variables: 21
## Groups: Agency [1]
## $ Agency            <chr> "HPD", "HPD", "HPD", "HPD", "HPD", "HPD", "H...
## $ Complaint.Type    <chr> "UNSANITARY CONDITION", "HEAT HOT WATER", "H...
## $ Descriptor        <chr> "PESTS", "ENTIRE BUILDING", "ENTIRE BUILDING...
## $ Incident.Zip      <chr> "10019", "10034", "10031", "10040", "10467",...
## $ Borough           <chr> "MANHATTAN", "MANHATTAN", "MANHATTAN", "MANH...
## $ Status            <chr> "Open", "Open", "Open", "Open", "Open", "Ope...
## $ Created.Date      <chr> "04/14/2015 12:00:00 AM", "04/14/2015 12:00:...
## $ Closed.Date       <chr> NA, NA, NA, NA, NA, NA, NA, "04/14/2015 10:2...
## $ Unique.Key        <int> 30390798, 30394103, 30395198, 30389217, 3038...
## $ TotalUnits        <int> 4274, 205, 1001, 183, 1837, 822, 3076, 3076,...
## $ BR1Units          <int> 587, 104, 370, 52, 549, 184, 758, 758, 758, ...
## $ BR2Units          <int> 246, 62, 365, 64, 563, 292, 1062, 1062, 1062...
## $ BR3Units          <int> 35, 13, 76, 38, 385, 106, 462, 462, 462, 462...
## $ BR4Units          <int> 0, 0, 14, 0, 35, 23, 28, 28, 28, 28, 28, 28,...
## $ StudioUnits       <int> 313, 5, 72, 29, 136, 100, 114, 114, 114, 114...
## $ RentalUnits       <int> 1235, 184, 840, 168, 786, 705, 1881, 1881, 1...
## $ OwnedUnits        <int> 11, 0, 57, 15, 882, 0, 1114, 1114, 1114, 111...
## $ PreservationUnits <dbl> 8, 4, 33, 4, 28, 22, 345, 345, 345, 345, 345...
## $ ConstructionUnits <dbl> 5, 1, 3, 0, 10, 9, 591, 591, 591, 591, 591, ...
## $ Latitude          <dbl> 40.76363, 40.86871, 40.82854, 40.85909, 40.8...
## $ Longitude         <dbl> -73.97959, -73.92524, -73.94146, -73.92769, ...
```

Data Dictionary

Complaint.Type - Information about the topic of the incident or condition.

Descriptor - Dependent on the complaint type. Contains more information on the incident or condition.

Incident.Zip - Incident location zip code, provided by geo validation.

Borough - Provided by the submitter and confirmed by geo validation.

Status - Status of Service Request submitted (Suggested values: Assigned, Cancelled, Closed, etc.)

Created.Date - Date of creation of service request (Format: MM/DD/YY HH:MM:SS AM/PM)

Closed.Date - Date on which the SR was closed (Format: MM/DD/YY HH:MM:SS AM/PM)

Unique.Key - A key to uniquely identify each service request (SR)

TotalUnits - The Total Units field indicates the total number of units in each zip.

BR1Units - Total number of 1-BR units i.e units with 1-bedroom in each zip.

BR2Units - Total number of 2-BR units i.e units with 2-bedroom in each zip.

BR3Units - Total number of 3-BR units i.e units with 3-bedroom in each zip.

BR4Units - Total number of 4-BR units i.e units with 4-bedroom in each zip.

StudioUnits - Total number of Studio units in each zip.

RentalUnits - Rental Units are the units in the building, counted toward the Housing New York plan, where assistance has been provided to landlords in exchange for a requirement for affordable units for each zip.

OwnedUnits - Owned Units are the units in the building, counted toward the Housing New York Plan, where assistance has been provided directly to homeowners for each zip.

PreservationUnits - The Preservation units field contains the number of 'Preservation' type building for each zip.

ConstructionUnits - The Construction units field contains the number of 'New Construction' type building for each zip.

Latitude - Geo based Latitude of the incident location

Longitude - Geo based Longitude of the incident location