EDA

shoili

April 8, 2017

```r
setwd("~/Documents/airbnb_MLproj")
library(dplyr)
library(lubridate)
```

This document takes a look at the dataset. Makes the necessary joins and outlines steps to be taken for feature generation before the prediction model is built.

```r
# read data
calendar <- read.csv("calendar.csv", strip.white = T, stringsAsFactors = F)
head(calendar)
```

```
##   listing_id       date available price
## 1   12147973 2017-09-05         f
## 2   12147973 2017-09-04         f
## 3   12147973 2017-09-03         f
## 4   12147973 2017-09-02         f
## 5   12147973 2017-09-01         f
## 6   12147973 2017-08-31         f
```

```r
listings <- read.csv("listings.csv", strip.white = T, stringsAsFactors = F)
# head(listings)
reviews <- read.csv("reviews.csv", strip.white = T, stringsAsFactors = F)
# head(reviews)
```

Calendar contains the prices for all available days so we strip out the unavailable days data. We then convert the price to a number from a character field.

```r
# clean calendar table
table(calendar$available)
```

```
## 
##      f      t 
## 665853 643037
```

```r
available_listings <- calendar %>%
  filter(available == "t")

available_listings$price_num <- as.numeric(sub("\\$","", available_listings$price))
```

```
## Warning: NAs introduced by coercion
```

```r
summary(available_listings$price_num)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    11.0    85.0   150.0   192.5   250.0   999.0    2568
```

```r
available_listings <- available_listings[!is.na(available_listings$price_num), ]

rm(calendar)
```

Just read in available_listings.csv from now on instead of calendar. This file has 2898 unique listings. The dates of the listings range from 2016-09-06 to 2017-09-05. We need to generate what month/week of year the listing is for to identify seasonal trends. (done)

```
available_listings <- available_listings %>%
  mutate(month = month(date), week = isoweek(date))

write.csv(available_listings, file = "available_listings.csv")
```

The new field ‘price_num’ is the response variable for our model. Now we need to join the data from listings and reviews. Taking a look at the data contained in these.

```
colnames(listings)
```

```
##  [1] "id"                           "listing_url"
##  [3] "scrape_id"                    "last_scraped"
##  [5] "name"                         "summary"
##  [7] "space"                        "description"
##  [9] "experiences_offered"          "neighborhood_overview"
## [11] "notes"                        "transit"
## [13] "access"                       "interaction"
## [15] "house_rules"                  "thumbnail_url"
## [17] "medium_url"                   "picture_url"
## [19] "xl_picture_url"               "host_id"
## [21] "host_url"                     "host_name"
## [23] "host_since"                   "host_location"
## [25] "host_about"                   "host_response_time"
## [27] "host_response_rate"           "host_acceptance_rate"
## [29] "host_is_superhost"            "host_thumbnail_url"
## [31] "host_picture_url"             "host_neighbourhood"
## [33] "host_listings_count"          "host_total_listings_count"
## [35] "host_verifications"           "host_has_profile_pic"
## [37] "host_identity_verified"       "street"
## [39] "neighbourhood"                "neighbourhood_cleansed"
## [41] "neighbourhood_group_cleansed" "city"
## [43] "state"                        "zipcode"
## [45] "market"                       "smart_location"
## [47] "country_code"                 "country"
## [49] "latitude"                     "longitude"
## [51] "is_location_exact"            "property_type"
## [53] "room_type"                    "accommodates"
## [55] "bathrooms"                    "bedrooms"
## [57] "beds"                         "bed_type"
## [59] "amenities"                    "square_feet"
## [61] "price"                        "weekly_price"
## [63] "monthly_price"                "security_deposit"
## [65] "cleaning_fee"                 "guests_included"
## [67] "extra_people"                 "minimum_nights"
## [69] "maximum_nights"               "calendar_updated"
## [71] "has_availability"             "availability_30"
## [73] "availability_60"              "availability_90"
## [75] "availability_365"             "calendar_last_scraped"
## [77] "number_of_reviews"            "first_review"
## [79] "last_review"                  "review_scores_rating"
## [81] "review_scores_accuracy"       "review_scores_cleanliness"
## [83] "review_scores_checkin"        "review_scores_communication"
## [85] "review_scores_location"       "review_scores_value"
## [87] "requires_license"                 "license"
## [89] "jurisdiction_names"               "instant_bookable"
## [91] "cancellation_policy"              "require_guest_profile_picture"
## [93] "require_guest_phone_verification" "calculated_host_listings_count"
## [95] "reviews_per_month"
```

```
# head(listings)
```

After feature generation, join listings with available_listings by columns 'id' and 'listing_id'.

Possible features -

```
colnames(reviews)
```

```
## [1] "listing_id"    "id"            "date"          "reviewer_id"
## [5] "reviewer_name" "comments"
```

Reviews can also be joined to available_listings by listing_id.

Possible features - 1. Number of reviews a listing has. (done) 2. Avg length of review. (chars done. also have words?) 3. Avg sentiment of review. 4. fraction of reviews positive 5. fraction of reviews negative

```
review_features <- reviews %>%
  mutate(reviewlen = nchar(comments)) %>%
  group_by(listing_id) %>%
  mutate(reviewcount = n(), avgreviewlen = round(mean(reviewlen)))
```

Need to train sentiment analysis. Use python maybe?