Airbnb Anonymization

Ahmed Al-Ali

4/21/2021

Packages needed and reading in file

Data Analysis

Lets take a look at the data only first couple of rows, head() command displays first 6 rows of data we can say that data is from a relational data base

```
head(aibnb)
```

```
##     id                                             name host_id   host_name
## 1 2539               Clean & quiet apt home by the park    2787        John
## 2 2595                            Skylit Midtown Castle    2845    Jennifer
## 3 3647              THE VILLAGE OF HARLEM....NEW YORK !    4632   Elisabeth
## 4 3831                  Cozy Entire Floor of Brownstone    4869 LisaRoxanne
## 5 5022 Entire Apt: Spacious Studio/Loft by central park    7192       Laura
## 6 5099        Large Cozy 1 BR Apartment In Midtown East    7322       Chris
##   neighbourhood_group neighbourhood latitude longitude       room_type price
## 1            Brooklyn    Kensington 40.64749 -73.97237    Private room   149
## 2           Manhattan       Midtown 40.75362 -73.98377 Entire home/apt   225
## 3           Manhattan        Harlem 40.80902 -73.94190    Private room   150
## 4            Brooklyn  Clinton Hill 40.68514 -73.95976 Entire home/apt    89
## 5           Manhattan   East Harlem 40.79851 -73.94399 Entire home/apt    80
## 6           Manhattan   Murray Hill 40.74767 -73.97500 Entire home/apt   200
##   minimum_nights number_of_reviews last_review reviews_per_month
## 1              1                 9  2018-10-19              0.21
## 2              1                45  2019-05-21              0.38
## 3              3                 0                            NA
## 4              1               270  2019-07-05              4.64
## 5             10                 9  2018-11-19              0.10
## 6              3                74  2019-06-22              0.59
##   calculated_host_listings_count availability_365
## 1                              6              365
## 2                              2              355
## 3                              1              365
## 4                              1              194
## 5                              1                0
## 6                              1              129
```

Attributes

As we cann see our columns are ,colnames() command displays the name of columns in a dataset

```
colnames(aibnb)
```

```
##  [1] "id"                                "name"
##  [3] "host_id"                           "host_name"
##  [5] "neighbourhood_group"               "neighbourhood"
##  [7] "latitude"                          "longitude"
##  [9] "room_type"                         "price"
## [11] "minimum_nights"                    "number_of_reviews"
## [13] "last_review"                       "reviews_per_month"
## [15] "calculated_host_listings_count" "availability_365"
```

Data type and structure

As we can see the main strcuture of the data set , str() command shows the column data type and bunch of values from column gives a dimension of data the the first line of output

```
str(aibnb)
```

```
## 'data.frame':    48895 obs. of  16 variables:
##  $ id                            : int  2539 2595 3647 3831 5022 5099 5121 5178 5203 5238 ...
##  $ name                          : chr  "Clean & quiet apt home by the park" "Skylit Midtown Castle"
##  $ host_id                       : int  2787 2845 4632 4869 7192 7322 7356 8967 7490 7549 ...
##  $ host_name                     : chr  "John" "Jennifer" "Elisabeth" "LisaRoxanne" ...
##  $ neighbourhood_group           : chr  "Brooklyn" "Manhattan" "Manhattan" "Brooklyn" ...
##  $ neighbourhood                 : chr  "Kensington" "Midtown" "Harlem" "Clinton Hill" ...
##  $ latitude                      : num  40.6 40.8 40.8 40.7 40.8 ...
##  $ longitude                     : num  -74 -74 -73.9 -74 -73.9 ...
##  $ room_type                     : chr  "Private room" "Entire home/apt" "Private room" "Entire home/
##  $ price                         : int  149 225 150 89 80 200 60 79 79 150 ...
##  $ minimum_nights                : int  1 1 3 1 10 3 45 2 2 1 ...
##  $ number_of_reviews             : int  9 45 0 270 9 74 49 430 118 160 ...
##  $ last_review                   : chr  "2018-10-19" "2019-05-21" "" "2019-07-05" ...
##  $ reviews_per_month             : num  0.21 0.38 NA 4.64 0.1 0.59 0.4 3.47 0.99 1.33 ...
##  $ calculated_host_listings_count: int  6 2 1 1 1 1 1 1 1 4 ...
##  $ availability_365              : int  365 355 365 194 0 129 0 220 0 188 ...
```

Summary statistics

Summary statistics of data set , summary() command it shows the minimum,maximum, mean , median 1st and 3rd quantile.

```
summary(aibnb)
```

```
##        id                 name              host_id           host_name
##  Min.   :    2539   Length:48895       Min.   :     2438   Length:48895
##  1st Qu.: 9471945   Class :character   1st Qu.:  7822033   Class :character
##  Median :19677284   Mode  :character   Median : 30793816   Mode  :character
##  Mean   :19017143                      Mean   : 67620011
##  3rd Qu.:29152178                      3rd Qu.:107434423
##  Max.   :36487245                      Max.   :274321313
##
```

```
##  neighbourhood_group neighbourhood         latitude       longitude
##  Length:48895        Length:48895       Min.   :40.50   Min.   :-74.24
##  Class :character    Class :character   1st Qu.:40.69   1st Qu.:-73.98
##  Mode  :character    Mode  :character   Median :40.72   Median :-73.96
##                                         Mean   :40.73   Mean   :-73.95
##                                         3rd Qu.:40.76   3rd Qu.:-73.94
##                                         Max.   :40.91   Max.   :-73.71
##
##   room_type             price         minimum_nights    number_of_reviews
##  Length:48895       Min.   :    0.0   Min.   :   1.00   Min.   :  0.00
##  Class :character   1st Qu.:   69.0   1st Qu.:   1.00   1st Qu.:  1.00
##  Mode  :character   Median :  106.0   Median :   3.00   Median :  5.00
##                     Mean   :  152.7   Mean   :   7.03   Mean   : 23.27
##                     3rd Qu.:  175.0   3rd Qu.:   5.00   3rd Qu.: 24.00
##                     Max.   :10000.0   Max.   :1250.00   Max.   :629.00
##
##  last_review        reviews_per_month calculated_host_listings_count
##  Length:48895       Min.   : 0.010    Min.   :  1.000
##  Class :character   1st Qu.: 0.190    1st Qu.:  1.000
##  Mode  :character   Median : 0.720    Median :  1.000
##                     Mean   : 1.373    Mean   :  7.144
##                     3rd Qu.: 2.020    3rd Qu.:  2.000
##                     Max.   :58.500    Max.   :327.000
##                     NA's   :10052
##  availability_365
##  Min.   :  0.0
##  1st Qu.:  0.0
##  Median : 45.0
##  Mean   :112.8
##  3rd Qu.:227.0
##  Max.   :365.0
##
```

Deal with non sense values , zeroes , NA or empty data points

```
cbind(zeroes=
   lapply(
     lapply(aibnb, function(x) x==0),
     sum),
   Nas=lapply(
     lapply(aibnb,  is.na),
     sum),
   empty=lapply(
     lapply(aibnb,function(x) x==""),
     sum)
   )
```

```
##                               zeroes Nas   empty
## id                            0      0     0
## name                          0      0     16
## host_id                       0      0     0
## host_name                     0      0     21
```

```
## neighbourhood_group            0      0     0
## neighbourhood                  0      0     0
## latitude                       0      0     0
## longitude                      0      0     0
## room_type                      0      0     0
## price                          11     0     0
## minimum_nights                 0      0     0
## number_of_reviews              10052  0     0
## last_review                    0      0     10052
## reviews_per_month              NA     10052 NA
## calculated_host_listings_count 0      0     0
## availability_365               17533  0     0
```

1.It seems that price ,number of reviews and availability have zero values , but we know that zero for availability and and number of reviews are meaningful meanwhile, a zero for price is meaningless ,we would remove these datapoints not to introduce errors

2.Reviews per month have 10052 NA values for convienience we would replace them with zero

3.Empty values are name and host_name and last_review we would remove rows with no host and name , and change last review to NA

```
#replace NA  values
aibnb$reviews_per_month=replace_na(aibnb$reviews_per_month,0)
#replace empty values
aibnb<-aibnb[aibnb$name!="0",]
aibnb<-aibnb[aibnb$host_name!="0",]
aibnb$last_review[which(aibnb$last_review == "")]<-"NA"
#remove rows with price = 0
aibnb<-aibnb[aibnb$price!=0,]
```

Descriptive

Getting descriptive statistics into a table format using my pre defined function

```
descriptive <-function(data)
  {
   #creating a data frame for values
   descriptive_data<-data.frame(Attribute =c(),type=c(),range=c(),min=c(),max=c(),mean=c(),median=c(),ca


  for(i in colnames(data))
  {
    if(class(data[,i]) != "character")
     {
      typetemp=class(data[,i])
      rangetemp=(max(data[,i]) - min(data[,i]))
      mintemp=min(data[,i])
      maxtemp=max(data[,i])
      meantemp=mean(data[,i])
      mediantemp=median(data[,i])

      datatemp=data.frame(Attribute =i   ,
                          type=typetemp,
```

```
                          range=rangetemp,
                          min=mintemp,
                          max=maxtemp,
                          mean=meantemp,
                          median=mediantemp,
                          categories="NA"
                          )

      descriptive_data=rbind(datatemp,descriptive_data)
    }
     else
     {
       typetemp=class(data[,i])
       datatemp=data.frame(Attribute =i ,
                           type=typetemp,
                           range= "Categorical",
                           min="NA",
                           max="NA",
                           mean="NA",
                           median="NA",
                           categories=length(unique(data[,i]))
                           )
      descriptive_data=rbind(datatemp,descriptive_data)
     }


   }
  return(descriptive_data)

}

descriptive(aibnb)
```

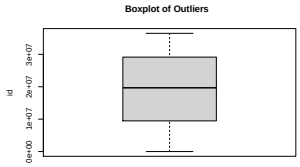
```
##                          Attribute      type              range       min
## 1                 availability_365   integer                365         0
## 2  calculated_host_listings_count   integer                326         1
## 3                reviews_per_month   numeric               58.5         0
## 4                      last_review character       Categorical        NA
## 5                number_of_reviews   integer                629         0
## 6                   minimum_nights   integer               1249         1
## 7                            price   integer               9990        10
## 8                        room_type character       Categorical        NA
## 9                        longitude   numeric            0.53143 -74.24442
## 10                        latitude   numeric 0.413270000000004  40.49979
## 11                   neighbourhood character       Categorical        NA
## 12              neighbourhood_group character       Categorical        NA
## 13                        host_name character       Categorical        NA
## 14                          host_id   integer         274318875      2438
## 15                             name character       Categorical        NA
## 16                               id   integer          36484706      2539
##          max              mean     median categories
## 1        365  112.779498404386         45         NA
## 2        327  7.14462809917355          1         NA
```

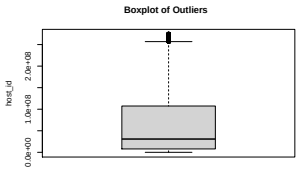
```
## 3       58.5  1.09080005727846       0.37         NA
## 4         NA                NA         NA       1765
## 5        629  23.2719908354472          5         NA
## 6       1250  7.02988707961705          3         NA
## 7      10000  152.755052778005        106         NA
## 8         NA                NA         NA          3
## 9  -73.71299 -73.9521755776941 -73.955685         NA
## 10  40.91306    40.72895268145   40.72308         NA
## 11        NA                NA         NA        221
## 12        NA                NA         NA          5
## 13        NA                NA         NA      11451
## 14 274321313  67622034.5611243 30792573.5         NA
## 15        NA                NA         NA      47895
## 16  36487245   19016793.406677 19675740.5         NA
```

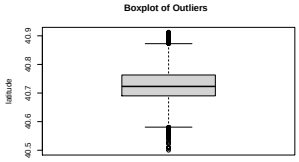
Assessing outliers

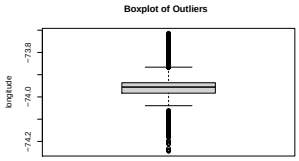
We are plotting the outliers if there are for non-categorical columns , using our predefined function

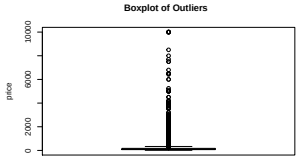
```
plot_outlier<-function(data){

  for(i in colnames(data))
    {
      if(class(data[,i]) != "character")
      {
         box= boxplot(data[,i],
                      ylab = i,
                      main = "Boxplot of Outliers  "
                      )

      }
      else
      {
        next
      }


    }
}
plot_outlier(aibnb)
```

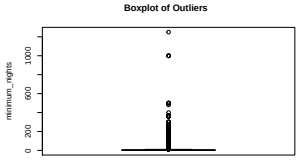


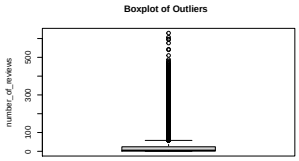

Boxplot of Outliers
host_id
0.0e+00
1.0e+08
2.0e+08

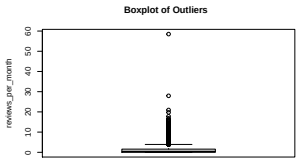

Boxplot of Outliers
latitude
40.5
40.6
40.7
40.8
40.9

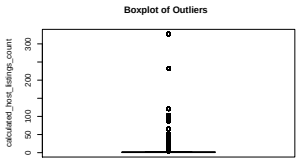

Boxplot of Outliers
longitude
−74.2
−74.0
−73.8

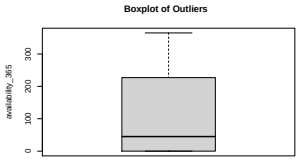

Boxplot of Outliers
price
0
2000
6000
10000


Boxplot of Outliers
minimum_nights
0
200
600
1000


Boxplot of Outliers
number_of_reviews
0
100
300
500


Boxplot of Outliers
reviews_per_month
0
10
20
30
40
50
60





As we can see from the plots in terms of data distribution calculated from the IQR of each column these points are candadite outliers but not all may pass test of checking outlines statistically roughly speaking this table summaries what I think is most suitable to be classified as a unique value , so we create a data frame and assgin unique outlier values for each column to be summarized as following

Price column it seems unusual to have a price of above 6000 dollars Minimum nights it seems unsual to have number above 800 reviews per month it seems unsual to have reviews above 25 calculated host listing above 150 seems unsual

```
outliers<-function(data){
  outlier=data.frame(Attribute = c() , values=c())
 for(i in colnames(data))
    {
      if((class(data[,i]) != c("character")))
      {
        temp=data.frame(Attribute=i,values=NA)
        outlier=rbind(outlier,temp)
      }
      else
      {
        next
      }


 }
  return(outlier)
}

out=outliers(aibnb)

#price column
out[out$Attribute == "price",]$values<-paste(unique(as.character(aibnb[aibnb$price>6000,]$price)),sep="
#Minimum nights column
out[out$Attribute == "minimum_nights",]$values<-paste(unique(as.character(aibnb[aibnb$minimum_nights
#reviews per month  column
out[out$Attribute == "reviews_per_month",]$values<-paste(unique(as.character(aibnb[aibnb$reviews_per_mo
#price column
out[out$Attribute == "calculated_host_listings_count",]$values<-paste(unique(as.character(aibnb[aibnb$c
```

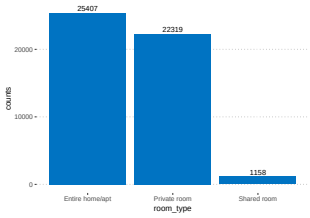
```
out<-out[!is.na(out$values),]
out
```

```
##                         Attribute                                          values
## 5                           price 6500,8000,9999,10000,7703,6419,8500,7500,6800
## 6                  minimum_nights                                     1000,1250,999
## 8               reviews_per_month                                        58.5,27.95
## 9 calculated_host_listings_count                                           232,327
```

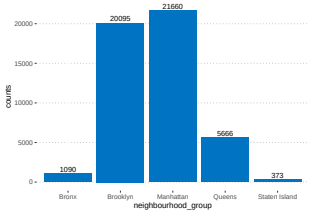
Frequencies visualization

To visiluize frequncy we would ignore specific text woth more than 10 characters as the reviews because these are not really considered categorical variables rather than strings or text manipulation as host_name and name and neighbourhood and last review

```
theme_set(theme_pubr())

#room type
df <- aibnb %>%
  group_by(room_type) %>%
  summarise(counts = n())

ggplot(df, aes(x = room_type, y = counts)) +
  geom_bar(fill = "#0073C2FF", stat = "identity") +
  geom_text(aes(label = counts), vjust = -0.3) +
  theme_pubclean()
```

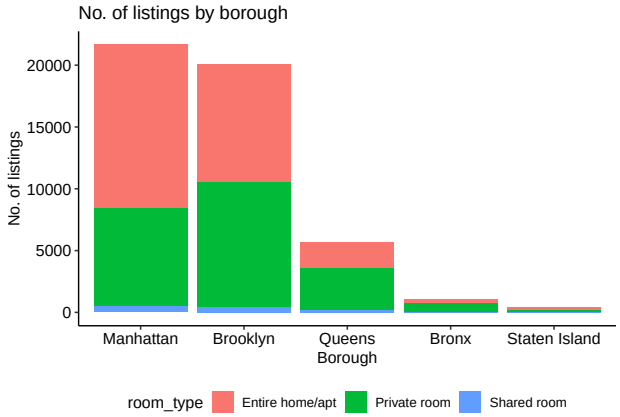



```
#neighbourhood
df <- aibnb %>%
  group_by(neighbourhood_group) %>%
  summarise(counts = n())

ggplot(df, aes(x = neighbourhood_group, y = counts)) +
  geom_bar(fill = "#0073C2FF", stat = "identity") +
  geom_text(aes(label = counts), vjust = -0.3) +
  theme_pubclean()
```

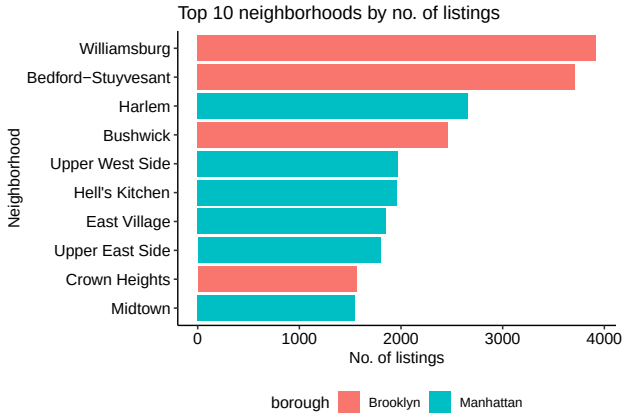


Infromative visulization

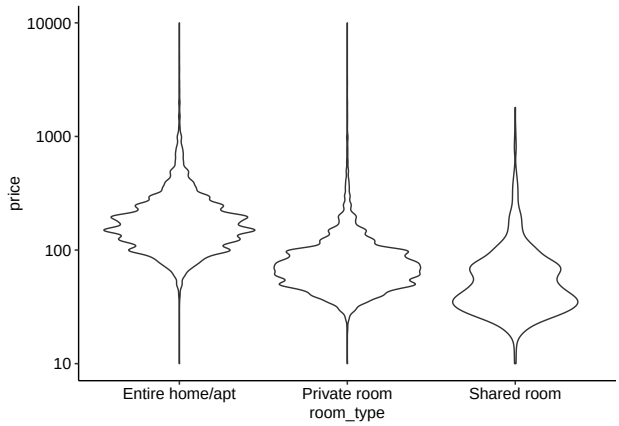
```
ggplot(aibnb, aes(x = fct_infreq(neighbourhood_group), fill = room_type)) +
    geom_bar() +
    labs(title = "No. of listings by borough",
         x = "Borough", y = "No. of listings") +
    theme(legend.position = "bottom")
```




```
aibnb%>%
    group_by(neighbourhood) %>%
    summarize(num_listings = n(),
              borough = unique(neighbourhood_group)) %>%
    top_n(n = 10, wt = num_listings) %>%
    ggplot(aes(x = fct_reorder(neighbourhood, num_listings),
               y = num_listings, fill = borough)) +
    geom_col() +
    coord_flip() +
    theme(legend.position = "bottom") +
    labs(title = "Top 10 neighborhoods by no. of listings",
         x = "Neighborhood", y = "No. of listings")
```



```
ggplot(aibnb, aes(x = room_type, y = price)) +
    geom_violin() +
    scale_y_log10()
```



Utility to preserve after anonymization

Data Utility measures The main objective of k-anonymization is privacy protection; however, it is also important that the anonymized dataset should be as useful as possible. There are various k-anonymization of a given dataset, but one having the highest utility is desirable. In privacy preserving , a data owner does not know how the published data will be analyzed by recipients, thus the data utility is measured by the quality of the anonymized dataset. We mainly focus on information loss and data truthfulness for assessing data utility, because these can cover the entire quality of the anonymized dataset in the proposed method . Information loss refers to the amount of loss caused by generalization. Data truthfulness implies that each anonymized record corresponds to a single original record. Relocated records cannot correspond to original records; thus, they are untruthful. In privacy-preserving data publishing, it is important that a published dataset is truthful. If a published dataset is not truthful, it is difficult to use the results of the data analysis, because false results may be obtained.

Utility criterion To quantify data utility, various quality metrics are proposed, such as classification metric discernibility metric (DM), loss metric (LM), and reconstruction error (RCE) . DM measures the cardinality of the equivalent class. DM considers only the number of records in the equivalent class; thus, DM does not capture information loss caused by generalization. LM can measure both the cardinality of the equivalent class and information loss. Although LM is more accurate when measuring information loss, it does not consider the data truthfulness. RCE measures the similarity between the original record and the anonymous record. This metric can reflect both information loss and data truthfulness.

Identification of Attributes

```
#Table creation of attribute types as identifiers
Iden<-function(data){
  Idenitty=data.frame(Attribute = c() , Type=c() ,Reason=c() )
 for(i in colnames(data))
    {
        temp=data.frame(Attribute=i,Type=NA,Reason=NA)
        Idenitty=rbind(Idenitty,temp)
```

```
    }
  return(Idenitty)
}

Iden_data<-Iden(aibnb)

#Function that disolays if value is unique
Check_uiq<-function(column){

  if(length(unique(column))==length(column))
  {
    ans<-" is unique identifier "
  }

  else
    ans<-" is not  unique identifier "

  return(ans)
}
```

Explicit identifiers

Potetnial EI

-id
-name
-host_id
-host_name

We quickly check if unique which means that if a certain column values are all different then this could be used as explicit identifier else no

```
id_ans<-Check_uiq(aibnb$id)
name_ans<-Check_uiq(aibnb$name)
hostid_ans<-Check_uiq(aibnb$host_id)
hostname_ans<-Check_uiq(aibnb$host_name)

print(paste("id",id_ans))
```

```
## [1] "id  is unique identifier "
```

```
print(paste("name",name_ans))
```

```
## [1] "name  is not  unique identifier "
```

```
print(paste("host_id",hostid_ans))
```

```
## [1] "host_id  is not  unique identifier "
```

```
print(paste("host_name",hostname_ans))
```

```
## [1] "host_name  is not  unique identifier "
```

it seems that only id is a unique identifier because it uniquely identifies a certain data point while others are common among multiple data points or some other values as hostname has empty strings or company names rather than individual persons if it was combination of first and last that would be unique , further study leads mt to conclude that we know that these names could easily identify a row because the text combinations is very useful so i would consider them EI too

Quasi-identifiers

potential QI -neighbourhood -neighbourhood_group -latitude
-longitude

while this variables are not unique , they are very valuable to be combined with any other varaible to infer as example name could be combined with any hotel info based on some knowledge to infer about a data point , as example if we know that the person went to a host_name in a certain neigbour hood we could definately truncate the amount of information to infer , combination of any of those with any varabile below could be a candadite to have similar utility to explicit idenitfier , these are temporal and spatial metircs

sensitive data

potetnial SD

-last_review -room_type
-price
-minimum_nights

Why are these attributes considered as SD , portioning my answer to two components first one is considered spatial which can lead to information that indeed reveals correlation between two entities , last three attributes above are considered business inrfomation which should be protected

non-sensitive data potential NSD

-number_of_reviews
-reviews_per_month
-calculated_host_listings_count
-availability_365

I consider these as NSD because the first two are similar to comparing it to political view or an opinion in other words and the rest two are facts about some sensitive data ,

Development of An Anonymization Approach

Basic Concept

The three main goals of the proposed method are as follows, the anonymized dataset should remain useful, privacy-preserving, and reliable. In other words, the anonymized dataset should not be over-generalized, and it should satisfy the privacy model (k-anonymity) and be truthful. To meet these goals, the proposed method comprises four parts,cryptographic hashing ,data masking, perturbation, and synthetic data generation, cryptographic hashing performed on unique explicit identifiers , data masking appropriate if data privacy is your main concern. Faker is a great Python library that seamlessly facilitates this process. You can easily swap out categories, labels, and other identifiers with fake values. Faker supports creating fake names, random text, addresses, etc. . . . . perturbation may involve introducing laplace mechanism to alter numerical

values , or synthetic value generation we could sample each variable from a best-fit probability distribution. This approach would mostly preserve each features original distribution and key statistics.preserve correlation between varibels hence utility preserved too

Explicit iderntifiers

```
#Hash function
hash <- function(.x, .algo = "sha256", .seed = 0, ...){
  if (!is.atomic(.x)) stop("Vector must be an atomic vector.")
  return(unname(vapply(.x, digest::digest, algo = .algo, seed = .seed, ...,character(1))))
}
```

This function uses a hashing algorithm to hash inputs ,SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the United States National Security Agency (NSA) and first published in 2001, the function checks if a column of passed data is a vector equivalent to array and then applies a digest function which hashes the inout using cryptogrpahic hash

```
#Hashing
aibnb_id_hash<-hash(aibnb$id)

#Check
length(na.omit(aibnb_id_hash)) == length(aibnb_id_hash)
```

```
## [1] TRUE
```

```
#Add
aibnb_anon<-aibnb
aibnb_anon$id<-aibnb_id_hash
```

We use the hash function to hash our id columns and then check if any value has NA which means not been hashed , checking the length of the array when we omit NA values versus original array . true means all values have been hashed successfully then we change the id columns with them

Masking and perturbation of these non-unique explicit identifiers to hide identity of users

```
#Function that perturbates entry
perturbate_string=function(column){
  pert_l=length(column)
  pet=character(pert_l)
  for( i in 0:length(column)){

    pet[i]=paste0(column[i],stri_rand_strings(1, 2, pattern = "[0-9]"))
  }
  return(pet)
}

aibnb_anon$name<-randomNames(nrow(aibnb_anon))
aibnb_anon$host_id<-perturbate_string(aibnb_anon$host_id)
aibnb_anon$host_name<-NULL
```

For the name column i use the built in function to replace the names with fake genrated names , for the host id its perturbated by adding pattern of a digit at end , hotel name is not a specific unique name but a combination of text which give a lot of unique insights if text processed , so I will completley delete them

Take a look at these columns

```
head(aibnb_anon[,c(1,2,3)])
```

```
##                                                                 id
## 1 5fbffaceef34daca3e4b12946d02ce6a0b65be40c4a1b9ed10b16a26c73c8ef3
## 2 9472e4a8d2ea5e8f87b6dd95996b5b863b15df3916f92fd056339cd6ea95e9c9
## 3 de648d5fcfd29e87f22228af61350554c6ad0b54f09d71cfa635438e3aa009bd
## 4 a47e2a26395642c8c6fe14f6d37aa4e3c140565ff9a6f0a55d2d2fb76bb34733
## 5 ddd83c1165d143e225b8de18ec79e55b77f0cd159b2f2a5b62016b4cde4d2c68
## 6 f4c886ab9ed7c55001f37dd4f8ff07fae97795f2b5e57fc968a84dc929ae18c7
##                name host_id
## 1  el-Saab, Nasreen  278795
## 2     Laugan, Jimmy  284568
## 3    Munoz, Armanda  463262
## 4     Drake, Taylor  486937
## 5      Maestas, Ana  719220
## 6 Pulling, Dominick  732218
```

Quasi iderntifiers

neighbourhood and neighbourhood_group We would form generalization approach for neighbor hood to be indicated as uptown,downtown midtown of group neighbor and then remove both columns

All main neighbors group

```
unique(aibnb$neighbourhood_group)
```

```
## [1] "Brooklyn"      "Manhattan"     "Queens"        "Staten Island"
## [5] "Bronx"
```

All manhattan neighbors

```
unique(aibnb[aibnb$neighbourhood_group=="Manhattan",]$neighbourhood)
```

```
##  [1] "Midtown"             "Harlem"              "East Harlem"
##  [4] "Murray Hill"         "Hell's Kitchen"      "Upper West Side"
##  [7] "Chinatown"           "West Village"        "Chelsea"
## [10] "Inwood"              "East Village"        "Lower East Side"
## [13] "Kips Bay"            "SoHo"                "Upper East Side"
## [16] "Washington Heights"  "Financial District"  "Morningside Heights"
## [19] "NoHo"                "Flatiron District"   "Roosevelt Island"
## [22] "Greenwich Village"   "Little Italy"        "Two Bridges"
## [25] "Nolita"              "Gramercy"            "Theater District"
## [28] "Tribeca"             "Battery Park City"   "Civic Center"
## [31] "Stuyvesant Town"     "Marble Hill"
```

All Brooklyn neighbors

```
unique(aibnb[aibnb$neighbourhood_group=="Brooklyn",]$neighbourhood)
```

```
##  [1] "Kensington"                "Clinton Hill"
##  [3] "Bedford-Stuyvesant"        "South Slope"
##  [5] "Williamsburg"              "Fort Greene"
##  [7] "Crown Heights"             "Park Slope"
##  [9] "Windsor Terrace"           "Greenpoint"
## [11] "Bushwick"                  "Flatbush"
## [13] "Prospect-Lefferts Gardens" "Prospect Heights"
## [15] "Brooklyn Heights"          "Carroll Gardens"
## [17] "Gowanus"                   "Flatlands"
## [19] "Cobble Hill"               "Boerum Hill"
## [21] "DUMBO"                     "East Flatbush"
## [23] "Gravesend"                 "East New York"
## [25] "Sheepshead Bay"            "Fort Hamilton"
## [27] "Bensonhurst"               "Sunset Park"
## [29] "Brighton Beach"            "Cypress Hills"
## [31] "Bay Ridge"                 "Columbia St"
## [33] "Vinegar Hill"              "Canarsie"
## [35] "Borough Park"              "Downtown Brooklyn"
## [37] "Midwood"                   "Red Hook"
## [39] "Dyker Heights"             "Sea Gate"
## [41] "Navy Yard"                 "Brownsville"
## [43] "Manhattan Beach"           "Bergen Beach"
## [45] "Coney Island"              "Bath Beach"
## [47] "Mill Basin"
```

All Queens neighbors

```
unique(aibnb[aibnb$neighbourhood_group=="Queens",]$neighbourhood)
```

```
##  [1] "Long Island City"    "Woodside"            "Flushing"
##  [4] "Sunnyside"           "Ridgewood"           "Jamaica"
##  [7] "Middle Village"      "Ditmars Steinway"    "Astoria"
## [10] "Queens Village"      "Rockaway Beach"      "Forest Hills"
## [13] "Elmhurst"            "Jackson Heights"     "St. Albans"
## [16] "Rego Park"           "Briarwood"           "Ozone Park"
## [19] "East Elmhurst"       "Arverne"             "Cambria Heights"
## [22] "Bayside"             "Kew Gardens"         "College Point"
## [25] "Glendale"            "Richmond Hill"       "Bellerose"
## [28] "Maspeth"             "Woodhaven"           "Kew Gardens Hills"
## [31] "Bay Terrace"         "Whitestone"          "Bayswater"
## [34] "Fresh Meadows"       "Springfield Gardens" "Howard Beach"
## [37] "Belle Harbor"        "Jamaica Estates"     "Far Rockaway"
## [40] "South Ozone Park"    "Corona"              "Neponsit"
## [43] "Laurelton"           "Holliswood"          "Rosedale"
## [46] "Edgemere"            "Jamaica Hills"       "Hollis"
## [49] "Douglaston"          "Little Neck"         "Breezy Point"
```

All Staten island neighbors

```
unique(aibnb[aibnb$neighbourhood_group=="Staten Island",]$neighbourhood)
```

```
##  [1] "St. George"                "Tompkinsville"
```

```
##  [3] "Emerson Hill"              "Shore Acres"
##  [5] "Arrochar"                  "Clifton"
##  [7] "Graniteville"              "Stapleton"
##  [9] "New Springville"           "Tottenville"
## [11] "Mariners Harbor"           "Concord"
## [13] "Port Richmond"             "Woodrow"
## [15] "Eltingville"               "Lighthouse Hill"
## [17] "West Brighton"             "Great Kills"
## [19] "Dongan Hills"              "Castleton Corners"
## [21] "Randall Manor"             "Todt Hill"
## [23] "Silver Lake"               "Grymes Hill"
## [25] "New Brighton"              "Midland Beach"
## [27] "Richmondtown"              "Howland Hook"
## [29] "New Dorp Beach"            "Prince's Bay"
## [31] "South Beach"               "Oakwood"
## [33] "Huguenot"                  "Grant City"
## [35] "Westerleigh"               "Bay Terrace, Staten Island"
## [37] "Fort Wadsworth"            "Rosebank"
## [39] "Arden Heights"             "Bull's Head"
## [41] "New Dorp"                  "Rossville"
## [43] "Willowbrook"
```

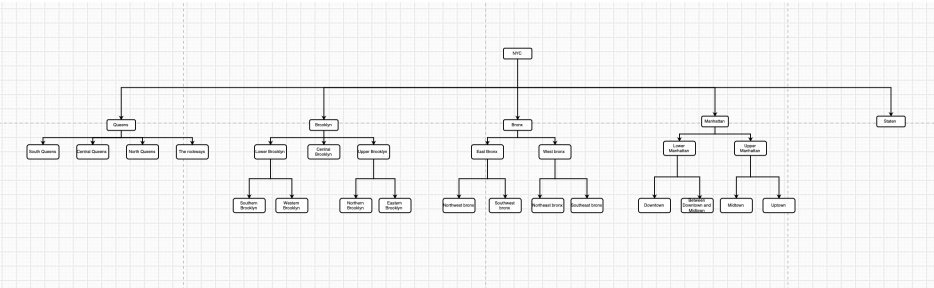
All Bronx neighbors

```
unique(aibnb[aibnb$neighbourhood_group=="Bronx",]$neighbourhood)
```

```
##  [1] "Highbridge"         "Clason Point"       "Eastchester"
##  [4] "Kingsbridge"        "Woodlawn"           "University Heights"
##  [7] "Allerton"           "Concourse Village"  "Concourse"
## [10] "Wakefield"          "Spuyten Duyvil"     "Mott Haven"
## [13] "Longwood"           "Morris Heights"     "Port Morris"
## [16] "Fieldston"          "Mount Eden"         "City Island"
## [19] "Williamsbridge"     "Soundview"          "Co-op City"
## [22] "Parkchester"        "North Riverdale"    "Bronxdale"
## [25] "Riverdale"          "Norwood"            "Claremont Village"
## [28] "Fordham"            "Mount Hope"         "Van Nest"
## [31] "Morris Park"        "Tremont"            "East Morrisania"
## [34] "Hunts Point"        "Pelham Bay"         "Throgs Neck"
## [37] "West Farms"         "Morrisania"         "Pelham Gardens"
## [40] "Belmont"            "Baychester"         "Melrose"
## [43] "Schuylerville"      "Castle Hill"        "Olinville"
## [46] "Edenwald"           "Westchester Square" "Unionport"
```

Combining both Neighborhood group and Neighbor to generalize as seen in the taximony tree

```
include_graphics("/Users/ahmed/Desktop/Screen Shot 2021-05-07 at 2.55.21 AM.png")
```



These information are from wikipedia , Now we rencode the columns as new column named location , I believe we treat the generalization for queens we stop at h=2 of tree , Islands have no further categories so they would be kept the same , rest would be evaluated into their differennt components at h=3 except for brooklyn which would be mix of h=2 and h=3,

```
#Queens
SouthQueen<-c("Jamaica","Queens Village","St. Albans","Cambria Heights","Hollis","Holliswood","Jamaica
CentralQueen<-c("Briarwood","Corona","East Elmhurst","Elmhurst","Forest Hills","Glendale","Kew Gardens"
NorthQueen<-c("Astoria","Ditmars Steinway","Jackson Heights", "Long Island City","Sunnyside","Bay Terra
Rockaways<-c("Arverne","Bayswater","Belle Harbor","Breezy Point","Edgemere","Far Rockaway","Neponsit","

#Brooklyn
SouthBrooklyn<-c("Bergen Beach" ,"Coney Island","Brighton Beach","Manhattan Beach","Sea Gate","Sheepshea
CentralBrooklyn<-c("Crown Heights","Flatbush","East Flatbush","Windsor Terrace","Prospect-Lefferts Gard
NorthBrooklyn<-c( "Bedford-Stuyvesant","Bushwick","Williamsburg","Greenpoint","Brooklyn Heights","Clint
SouthWestBrooklyn<-c( "Fort Hamilton" ,"Bensonhurst" ,  "Sunset Park"  , "Bay Ridge"  ,   "Borough Park
EasternBrooklyn<-c("Brownsville","Canarsie","East New York" ,"Cypress Hills")
#Bronx
NorthwestBronx<-c("Belmont","Fordham","Kingsbridge","Norwood","Riverdale","Spuyten Duyvil","University
SouthwestBronx<-c("Concourse" ,"Highbridge" ,"Hunts Point" ,"Longwood","Morris Heights" ,"Melrose","Mor
NortheastBronx<-c("Allerton" ,"Bronxdale"  ,"Baychester" ,"City Island","Co-op City","Eastchester","Ede
SoutheastBronx<-c("Clason Point" ,"Morris Park","Westchester Square","Pelham Bay","Unionport","Soundvie

#Manhattan
Downtown<-c("East Village" ,"Greenwich Village","NoHo","West Village","Chinatown" ,"Financial District"
Midtown<-c("Theater District","Midtown","Hell's Kitchen","Murray Hill","Gramercy","Stuyvesant Town"  )
Uptown<-c("Marble Hill","Inwood","Washington Heights","Morningside Heights","Harlem","East Harlem","Upp
BetDownMidtown<-c("Kips Bay" ,"Chelsea","Flatiron District","Lower East Side" ,"SoHo" ,"Little Italy" )
Islands<-"Roosevelt Island"
#Staten Island  No change

Staten<-unique(aibnb[aibnb$neighbourhood_group=="Staten Island",]$neighbourhood)
```

Now we rencode a new location column based on these places

```
aibnb_anon$Location="NA"
#rencode Queens
aibnb_anon[aibnb_anon$neighbourhood_group == "Queens" & aibnb_anon$neighbourhood %in% SouthQueen,]$Loca
aibnb_anon[aibnb_anon$neighbourhood_group == "Queens" & aibnb_anon$neighbourhood %in% CentralQueen,]$Lo
aibnb_anon[aibnb_anon$neighbourhood_group == "Queens" & aibnb_anon$neighbourhood %in% NorthQueen,]$Loca
aibnb_anon[aibnb_anon$neighbourhood_group == "Queens" & aibnb_anon$neighbourhood %in% Rockaways,]$Locat
#rencode Brooklyn
aibnb_anon[aibnb_anon$neighbourhood_group == "Brooklyn" & aibnb_anon$neighbourhood %in% SouthBrooklyn,]
```

```
aibnb_anon[aibnb_anon$neighbourhood_group == "Brooklyn" & aibnb_anon$neighbourhood %in% CentralBrooklyn
aibnb_anon[aibnb_anon$neighbourhood_group == "Brooklyn" & aibnb_anon$neighbourhood %in% NorthBrooklyn,]$
aibnb_anon[aibnb_anon$neighbourhood_group == "Brooklyn" & aibnb_anon$neighbourhood %in% SouthWestBrookly
aibnb_anon[aibnb_anon$neighbourhood_group == "Brooklyn" & aibnb_anon$neighbourhood %in% EasternBrooklyn
#rencode Bronx
aibnb_anon[aibnb_anon$neighbourhood_group == "Bronx" & aibnb_anon$neighbourhood %in% NorthwestBronx,]$Lo
aibnb_anon[aibnb_anon$neighbourhood_group == "Bronx" & aibnb_anon$neighbourhood %in% NortheastBronx,]$Lo
aibnb_anon[aibnb_anon$neighbourhood_group == "Bronx" & aibnb_anon$neighbourhood %in% SouthwestBronx,]$Lo
aibnb_anon[aibnb_anon$neighbourhood_group == "Bronx" & aibnb_anon$neighbourhood %in% SoutheastBronx,]$Lo

#rencode Manhattan
aibnb_anon[aibnb_anon$neighbourhood_group == "Manhattan" & aibnb_anon$neighbourhood %in% Downtown,]$Loca
aibnb_anon[aibnb_anon$neighbourhood_group == "Manhattan" & aibnb_anon$neighbourhood %in% Midtown,]$Locat
aibnb_anon[aibnb_anon$neighbourhood_group == "Manhattan" & aibnb_anon$neighbourhood %in% Uptown,]$Locati
aibnb_anon[aibnb_anon$neighbourhood_group == "Manhattan" & aibnb_anon$neighbourhood %in% BetDownMidtown
#Rencode Staten
aibnb_anon[aibnb_anon$neighbourhood_group == "Staten Island" ,]$Location<-"Staten Island"
```

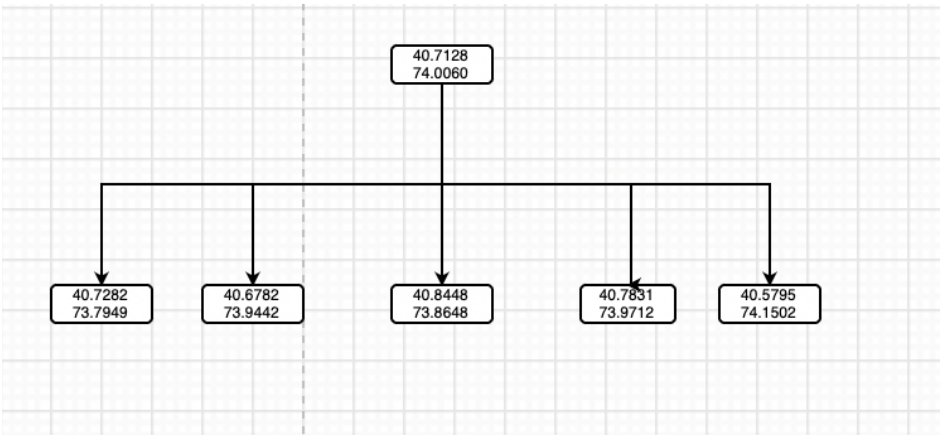
Delete old columns

```
aibnb_anon$neighbourhood=NULL
aibnb_anon$neighbourhood_group=NULL
```

Latitude and Longitude Reducing the accuracy by scaling a location to a coarser granularity

For generalization I decided to use two levels only main midpoint of nyc and Queens,Brooklyn,Bronx,Manhattan and Stane island midpoint coordinate since i already define multiple neighbor levels , note that longitude are negative suppposed to be in picture

```
include_graphics("/Users/ahmed/Desktop/Screen Shot 2021-05-07 at 2.43.19 PM.png")
```




```
#Queen Borough
aibnb_anon[str_detect(aibnb_anon$Location,c("Queen" ,"Rock")),]$latitude=40.7282
aibnb_anon[str_detect(aibnb_anon$Location,c("Queen" ,"Rock")),]$longitude= -73.7949
#Brooklyn Borough
aibnb_anon[str_detect(aibnb_anon$Location,"Brooklyn"),]$latitude=40.6782
aibnb_anon[str_detect(aibnb_anon$Location,"Brooklyn"),]$longitude= -73.9442
```

```
#Manhattan Borough
aibnb_anon[str_detect(aibnb_anon$Location,"Manhattan"),]$latitude=40.8448
aibnb_anon[str_detect(aibnb_anon$Location,"Manhattan"),]$longitude= -73.8648
#Bronx Borough
aibnb_anon[str_detect(aibnb_anon$Location,"Bronx"),]$latitude=40.7831
aibnb_anon[str_detect(aibnb_anon$Location,"Bronx"),]$longitude= -73.9712
#Stane Island Borough
aibnb_anon[str_detect(aibnb_anon$Location,"Staten Island"),]$latitude=40.5795
aibnb_anon[str_detect(aibnb_anon$Location,"Staten Island"),]$longitude= -74.1502
```

Take a look at these columns

```
head(aibnb_anon[,c(1,3,4,5,14)])
```

```
##                                                                 id host_id
## 1 5fbffaceef34daca3e4b12946d02ce6a0b65be40c4a1b9ed10b16a26c73c8ef3  278795
## 2 9472e4a8d2ea5e8f87b6dd95996b5b863b15df3916f92fd056339cd6ea95e9c9  284568
## 3 de648d5fcfd29e87f22228af61350554c6ad0b54f09d71cfa635438e3aa009bd  463262
## 4 a47e2a26395642c8c6fe14f6d37aa4e3c140565ff9a6f0a55d2d2fb76bb34733  486937
## 5 ddd83c1165d143e225b8de18ec79e55b77f0cd159b2f2a5b62016b4cde4d2c68  719220
## 6 f4c886ab9ed7c55001f37dd4f8ff07fae97795f2b5e57fc968a84dc929ae18c7  732218
##   latitude longitude         Location
## 1  40.6782  -73.9442  CentralBrooklyn
## 2  40.8448  -73.8648 MidtownManhattan
## 3  40.8448  -73.8648  UptownManhattan
## 4  40.6782  -73.9442    NorthBrooklyn
## 5  40.8448  -73.8648  UptownManhattan
## 6  40.8448  -73.8648 MidtownManhattan
```

Or we can go to local generalization in which we give a range of longitude and latitude but this limits the search area in which pathway privacy would be easily breached if adversary has background knowledge because range would break the preservation we did in location attribute in which certain locations would be clearly visible if inference is made

Potential SD

Last review Seemingly those are dates for last review i assume this is related to the user last visited time for the visited place i would aggregate the last review column to be have only year and month

```
aibnb_anon$last_review<-str_sub(aibnb_anon$last_review,0,7)
```

Room type Multivariate aggregation including weather a person get an entire home multiple times in a high end location could lead to profiling about him and his economic status as we can see that categories are Private or a shared room or a Home/Apartmant

```
unique(aibnb$room_type)
```

```
## [1] "Private room"    "Entire home/apt" "Shared room"
```

we aggregate the type to either be a room or an apartment

```
aibnb_anon[str_detect(aibnb_anon$room_type,"room"),]$room_type="Room"
aibnb_anon[str_detect(aibnb_anon$room_type,"apt"),]$room_type="Apartment"
```

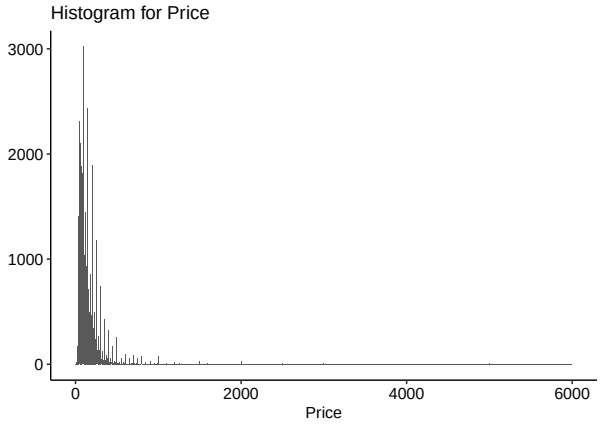
price and minimum nights Price and minim um noights reveals so much information when inferred so we are going to resample according to multivariate aggregation , i will use laplace mechanism to introduce noise to the values

studying the distribution of price and minimum nights

```
qplot(aibnb$price,
      geom="histogram",
      binwidth = 5,
      main = "Histogram for Price",
      xlab = "Price",
      xlim=c(0,6000),
      )
```

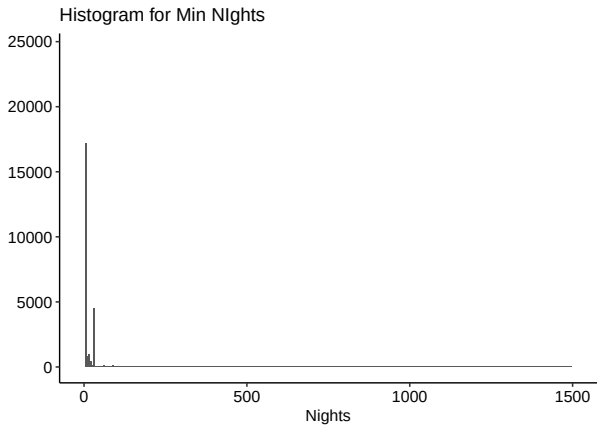
```
## Warning: Removed 16 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
qplot(aibnb$minimum_nights,
      geom="histogram",
      binwidth = 5,
      main = "Histogram for Min NIghts",
      xlab = "Nights",
      xlim=c(0,1500),
      )
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



define laplace function

```
#install.packages("LaplacesDemon")
library("LaplacesDemon")
```

```
##
## Attaching package: 'LaplacesDemon'

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

```
laplace=function(n,sensitivity,epsilon){

  return (rlaplace(n,location=0,scale=sensitivity/epsilon))

}
```

Alter price values

```
#Values excpet outliers
aibnb_anon[aibnb_anon$price<6000,]$price<-aibnb_anon[aibnb_anon$price<6000,]$price+laplace(nrow(aibnb_a
```

for minimum nights we can easily get a record knowing a person length of stay so this would be transformed into lessthan , which means ill create a new cllumn that means a certain record lived less than the value it has in this column as example a value of 50 would mean the person livied less than or equal to 50 days

first we form our boundaries by counting values

```
unique(aibnb$minimum_nights)
```

```
##   [1]   1   3  10  45   2   5   4  90   7  14  60  29  30 180   9
##  [16]  31   6  15   8  26  28 200  50  17  21  11  25  13  35  27
##  [31]  18  20  40  44  65  55 120 365 122  19 240  88 115 150 370
##  [46]  16  80 181 265 300  59 185 360  56  12  70  39  24  32 1000
##  [61] 110 270  22  75 250  62  23 1250 364  74 198 100 500  43  91
##  [76] 480  53  99 160  47 999 186 366  68  93  87 183 299 175  98
##  [91] 133 354  42  33  37 225 400 105 184 153 134 222  58 210 275
## [106] 182 114  85  36
```

it seems our range of boundary is multivariate no sepcific one

```
Bound1<-"[0,80]"
Bound2<-"(80,500]"
Bound3<-"(500,1250]"

aibnb_anon$minimum_night="NA"
aibnb_anon[aibnb_anon$minimum_nights <=80 ,]$minimum_night=Bound1
aibnb_anon[aibnb_anon$minimum_nights >80 & aibnb_anon$minimum_nights <=500 ,]$minimum_night=Bound2
aibnb_anon[aibnb_anon$minimum_nights >500 & aibnb_anon$minimum_nights <=1250 ,]$minimum_night=Bound3

#discard original column
aibnb_anon$minimum_nights=NULL
```

Dealing with outliers

```
#Outlier values
aibnb_anon[aibnb_anon$price %in% c("6500","8000","9999","10000","7703","6419","8500","7500","6800"),]
```

```
##                                                                 id
## 3775  2af2a5897e6c367570df1ad477e43a4bef67ba91342ac40e9402d81e475cdd91
## 4378  d077347c4b48b36bd29fda3951fb817cc74290a9803ad7c6c973ebb8c5194815
## 6531  a9174f68e7c2a9ef0f973bcc8b405429f2ff95d0964bb1e9634054e519ccff47
## 9152  cb9dfcd59e1c344e16bb022388b0fce7f560552597fdbd8288e7d287addf4bab
## 12343 80e86a1b05e236452107e11e3a412720ab7f70e8269e05f8cf49386e6541dd1e
## 17693 e433147a4d12191b6d36f274f930bb48f200ba3f16f1c862733f1ba0c5b154db
## 29239 9dab6fab31ed7e1733ba099a08e0cd52c06d1fb72d525fa0dd92c6acceca3f90
## 29663 d1e6b5c651c07652547df645631e0471fc317da7a8cf02c699c1439abfdce413
## 29665 7f2d3b423b7a99b149b1ee242b1a06803168ae08637e2f181021dbfc84399f32
## 30269 5df7c49713ad68ac4880f559d235665f184278459e00d6793e3a355dd0e3dd3f
## 37195 b02c682bd9be916ae65752306341848719f34c0e42d6b267a632a918e3ab05fd
## 40434 ecd8cb1dc31de5225924824c13b26d5476a9739ed70beaa7237ca98c9db3e2cb
## 42524 707c1dbbb4e1c7b7541c2f69efb70fe384728219dfe2e6163c9275dd5dbd7ce8
## 44035 e6a23ecda1109467c4a54ba119650b340f051eb9235c5e8f8a903c507b6ee8cf
## 45667 cfb778631c38ee8fc6a48921915d9c414a7e857f9f5219e67cc34fb0238cf923
## 48044 0a90886f5e66f1ca9df080e567863bf7fef21d91818330b5fae8efe048ddb8fb
##                            name     host_id latitude longitude room_type price
## 3775              Acosta, Ryan  1159835945  40.6782 -73.94420 Apartment  6500
## 4378         el-Dawood, Nabeela   117749704  40.6782 -73.94420 Apartment  8000
## 6531            el-Aziz, Azzaam   123507057  40.8448 -73.86480 Apartment  9999
## 9152             Santos, Daniel  2058283255  40.7681 -73.91651      Room 10000
## 12343            Peterson, Tony   390646405  40.8448 -73.86480      Room  9999
## 17693          Stouder, Michael   514390151  40.6782 -73.94420 Apartment 10000
## 29239              Tan, Antonia  7239039172  40.8448 -73.86480 Apartment 10000
## 29663          Mendiola, Joseph 15615877849  40.8448 -73.86480 Apartment  7703
## 29665            Durgan, Daniel 15615877879  40.8448 -73.86480 Apartment  6419
## 30269       el-Baten, Daifallah  1812845523  40.8448 -73.86480 Apartment  8500
## 37195          Wescott, Anthony  3530374323  40.8448 -73.86480      Room  6500
## 40434        el-Shaker, Tammaam   438212717  40.8448 -73.86480 Apartment  9999
## 42524           Hargers, Cylina   740774373  40.8448 -73.86480 Apartment  7500
## 44035 Matsunaga-Aragon, Abigail   375076415  40.8448 -73.86480 Apartment  6800
## 45667             Hayes, Sadyan 26253495175  40.6782 -73.94420      Room  7500
```

```
## 48044             Strock, Akira 27124866987  40.8448 -73.86480 Apartment  6500
##       number_of_reviews last_review reviews_per_month
## 3775                  0          NA              0.00
## 4378                  1     2016-09              0.03
## 6531                  1     2015-01              0.02
## 9152                  2     2016-02              0.04
## 12343                 6     2016-01              0.14
## 17693                 5     2017-07              0.16
## 29239                 0          NA              0.00
## 29663                 0          NA              0.00
## 29665                 0          NA              0.00
## 30269                 2     2018-09              0.18
## 37195                 0          NA              0.00
## 40434                 0          NA              0.00
## 42524                 0          NA              0.00
## 44035                 0          NA              0.00
## 45667                 8     2019-07              6.15
## 48044                 0          NA              0.00
##       calculated_host_listings_count availability_365               Location
## 3775                               1                0           NorthBrooklyn
## 4378                              11              365           NorthBrooklyn
## 6531                               1                0         UptownManhattan
## 9152                               1                0              NorthQueen
## 12343                              1               83 BetDownMidtownManhattan
## 17693                              1                0           NorthBrooklyn
## 29239                              1               83         UptownManhattan
## 29663                             12              146         UptownManhattan
## 29665                             12               45         UptownManhattan
## 30269                              1              251       DowntownManhattan
## 37195                              1               97         UptownManhattan
## 40434                              1              365 BetDownMidtownManhattan
## 42524                              1              364       DowntownManhattan
## 44035                              6              364 BetDownMidtownManhattan
## 45667                              2              179         CentralBrooklyn
## 48044                              1              365       DowntownManhattan
##       minimum_night
## 3775          [0,80]
## 4378          [0,80]
## 6531          [0,80]
## 9152        (80,500]
## 12343       (80,500]
## 17693         [0,80]
## 29239         [0,80]
## 29663         [0,80]
## 29665         [0,80]
## 30269         [0,80]
## 37195         [0,80]
## 40434         [0,80]
## 42524         [0,80]
## 44035         [0,80]
## 45667         [0,80]
## 48044       (80,500]
```

```
#find median per location per room type
head(aggregate(x = aibnb_anon$price,                # Specify data column
          by = list(aibnb_anon$room_type,aibnb_anon$Location),              # Specify group indicator
          FUN = median))
```

```
##     Group.1                 Group.2         x
## 1 Apartment BetDownMidtownManhattan 205.72298
## 2      Room BetDownMidtownManhattan  98.27048
## 3 Apartment         CentralBrooklyn 118.23773
## 4      Room         CentralBrooklyn  58.36946
## 5 Apartment            CentralQueen 113.88615
## 6      Room            CentralQueen  53.24200
```

as we can see if we choose mean or mdeian values to replace we will lose a lot of utility

to get idea of where the values occur lets subset

```
out
```

```
##                         Attribute                                          values
## 5                           price 6500,8000,9999,10000,7703,6419,8500,7500,6800
## 6                 minimum_nights                                    1000,1250,999
## 8              reviews_per_month                                       58.5,27.95
## 9 calculated_host_listings_count                                          232,327
```

price would be balanced by counterfeit records , minimum nights is handeled by introducing range values and other two would be countierfited by fake data but within same distribution to nsure utility

Adding counterfeit record with similar bur diverse outlier values so we can lower the variance

```
for(i in 1:3000){
  aibnb_anon=rbind(aibnb_anon,data.frame(id=hash(sample(20000,1)+5000),
                   name=randomNames(1),
                   host_id=sample(500000,1)+150000,
                   latitude=sample(c(40.6782,40.8448,40.7681,40.6782),1),
                   longitude=sample(c(40.6782,40.8448,40.7681,40.6782),1),
                   room_type=sample(unique(aibnb_anon$room_type),1),
                   price=sample(c(3000:5000),1)+1500,
                   number_of_reviews=sample(unique(aibnb_anon$number_of_reviews),1),
                   last_review=sample(unique(aibnb_anon$last_review),1),
                   reviews_per_month=sample(30,1)+15,
                   calculated_host_listings_count=sample(40,1)+60,
                   availability_365=sample(unique(aibnb_anon$availability_365),1),
                   Location=sample(c("NorthBrooklyn","UptownManhattan","NorthQueen","BetDownMidtownManha
                   minimum_night=sample(unique(aibnb_anon$minimum_night),1)))

}
```

our counterfeit data are from row number 48884 to 51884, display ten only

lets observe some

```
head(aibnb_anon[c(48884:51884),])
```

```
##                                                                    id
## 48895  b4b56f7db622c536e1d38f66f173c70323d76bb0a360cf9bc93e919b9de2d266
## 110000 f03debe59dad029f37619b488c07219dbee10ceae8abf8a674347a2e4d670cc1
## 110001 411addb7b8990e4cd9582338d39346dfa90de26f5e1288b6c8ed8158db4e0f89
## 110002 d5b5c1885b9202a01a70ec5472d17a7900ac0cf9cf005f0ee59677152386f483
## 110003 f4bda28bf15ee443d8c3df3094315a75b11a8061e8ce475bc8deb9f2df1ae17e
## 110004 298a3a4a38bdfeb2ed946ef639fb510b31096de797228425ffa8c38bb40c055a
##                              name    host_id latitude longitude room_type
## 48895                 Baer, Lily 6811981480  40.8448  -73.8648      Room
## 110000             Kim, Brigitte     212184  40.7681   40.6782      Room
## 110001 Bracamontes-Avila, Jazmyen     426428  40.6782   40.7681 Apartment
## 110002           Langston, Makari     230644  40.6782   40.6782 Apartment
## 110003        Richardson, Deshaun     649314  40.6782   40.7681 Apartment
## 110004                  Le, Cathy     626797  40.6782   40.7681      Room
##             price number_of_reviews last_review reviews_per_month
## 48895    80.54621                 0          NA                 0
## 110000 6369.00000               111     2017-01                19
## 110001 4573.00000               607     2018-06                26
## 110002 6015.00000               160     2019-07                24
## 110003 5374.00000               193     2016-07                22
## 110004 5009.00000               352     2013-12                23
##        calculated_host_listings_count availability_365          Location
## 48895                               1               23  MidtownManhattan
## 110000                             66              228   CentralBrooklyn
## 110001                             82              108        NorthQueen
## 110002                             79              364 DowntownManhattan
## 110003                             67              309 DowntownManhattan
## 110004                             97              206        NorthQueen
##        minimum_night
## 48895          [0,80]
## 110000      (80,500]
## 110001      (80,500]
## 110002      (80,500]
## 110003      (80,500]
## 110004    (500,1250]
```

Measuring data utlity

creating a utility measure table

```
descriptive_data_anon<-data.frame(Attribute =c(),mean=c(),variance=c(),median=c())
for ( i in colnames(aibnb_anon)){
  if (class(i)=="numeric"){
        meantemp=mean(aibnb_anon[,i])
        vartemp=var(aibnb_anon[,i])
        medtemp=median(aibnb_anon[,i])
        datatemp=data.frame(Attribute =i,
                            mean=meantemp,
                            varaince=vartemp,
                            median=medtemp,
```

```
                            )
        descriptive_data_anon=rbind(descriptive_data_anon,datatemp)}
}
```

General utility measures for continuous variables

Statistics: mean, covariance, correlation The statistics characterizing the dataset should not change after the anonymization. Examples of such statistics are the mean, variance, and covariance and correlation structure of the most important variables in the dataset. Other statistics characterizing the data include the principal components and the loadings. In order to evaluate the information loss caused by the anonymization, one should compare the appropriate statistics for continuous variables computed from the data before and after anonymization. There are several ways to evaluate the loss of utility with respect to the changes in these statistics, for instance, by comparing means and (co-)variances in the data or comparing the (multivariate) distributions of the data. Especially changes in the correlations gives valuable information on the validity of the data for regressions.

proposed are several measures for the discrepancy between the covariance and correlation matrices. These measures are based on the mean squared error, the mean absolute error or the mean variation of the individual cells these values should have minimal change as they are very importnat utility measures

As we can see utility hasnt change by a significant amount

Visiualization before and after anonymization

```
library(tidyverse)
library(knitr)
library(ggmap)
```

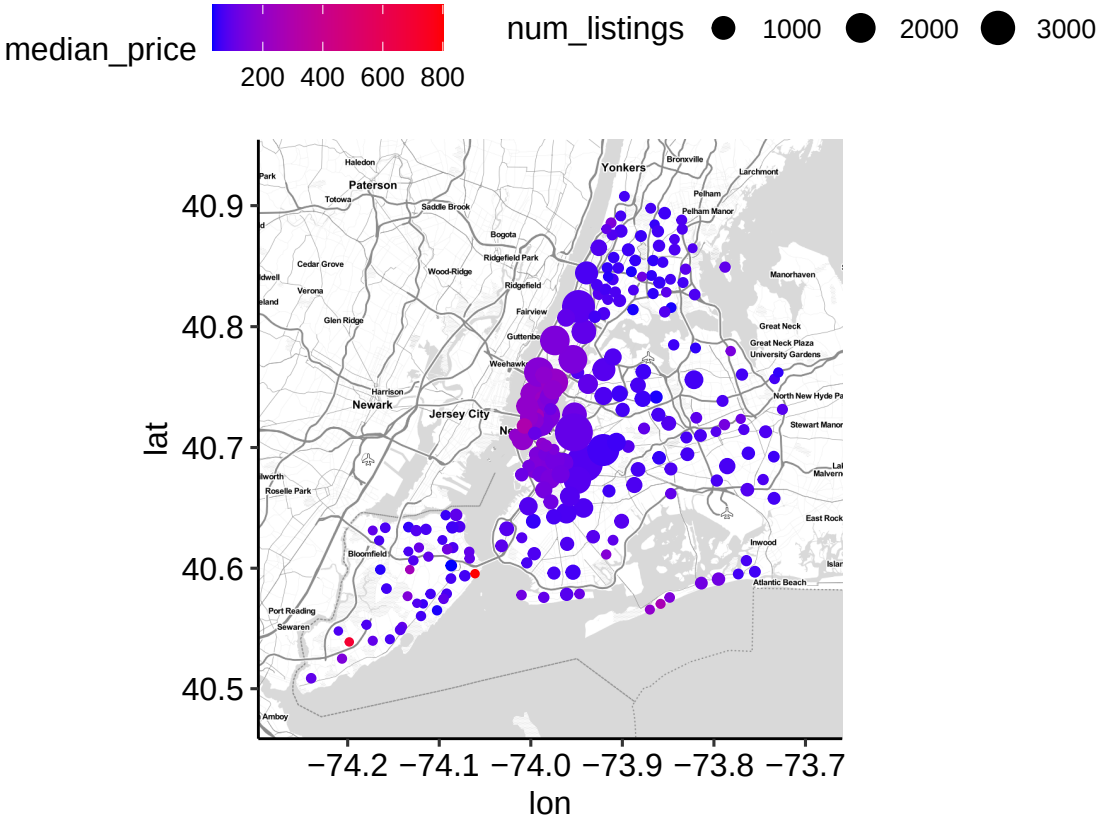
```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
df=aibnb
nhd_df <- df %>%
    group_by(neighbourhood) %>%
    summarize(num_listings = n(),
              median_price = median(price),
              long = median(longitude),
              lat = median(latitude),
              borough = unique(neighbourhood_group))


height <- max(df$latitude) - min(df$latitude)
width <- max(df$longitude) - min(df$longitude)
borders <- c(bottom  = min(df$latitude)  - 0.1 * height,
             top     = max(df$latitude)  + 0.1 * height,
             left    = min(df$longitude) - 0.1 * width,
             right   = max(df$longitude) + 0.1 * width)

map <- get_stamenmap(borders, zoom = 11, maptype = "toner-lite")
```

```
## Source : http://tile.stamen.com/toner-lite/11/601/768.png

## Source : http://tile.stamen.com/toner-lite/11/602/768.png

## Source : http://tile.stamen.com/toner-lite/11/603/768.png

## Source : http://tile.stamen.com/toner-lite/11/604/768.png

## Source : http://tile.stamen.com/toner-lite/11/601/769.png

## Source : http://tile.stamen.com/toner-lite/11/602/769.png

## Source : http://tile.stamen.com/toner-lite/11/603/769.png

## Source : http://tile.stamen.com/toner-lite/11/604/769.png

## Source : http://tile.stamen.com/toner-lite/11/601/770.png

## Source : http://tile.stamen.com/toner-lite/11/602/770.png

## Source : http://tile.stamen.com/toner-lite/11/603/770.png

## Source : http://tile.stamen.com/toner-lite/11/604/770.png

## Source : http://tile.stamen.com/toner-lite/11/601/771.png

## Source : http://tile.stamen.com/toner-lite/11/602/771.png

## Source : http://tile.stamen.com/toner-lite/11/603/771.png

## Source : http://tile.stamen.com/toner-lite/11/604/771.png
```

```
ggmap(map) +
    geom_point(data = nhd_df, mapping = aes(x = long, y = lat,
                                            col = median_price, size = num_listings)) +
    scale_color_gradient(low = "blue", high = "red")
```



Performig query "Attacks"

We would compare getting the above 5000 price for the room

from un anonmyized set

```
head(aibnb[aibnb$price>5000,c(1,2,10)])
```

```
##           id                                    name price
## 3538 2110145  UWS 1BR w/backyard + block from CP  6000
## 3721 2243699 SuperBowl Penthouse Loft 3,000 sqft  5250
## 3775 2271504    SUPER BOWL Brooklyn Duplex Apt!!  6500
## 4378 2953058                       Film Location  8000
## 6531 4737930                 Spanish Harlem Apt  9999
## 9152 7003697 Furnished room in Astoria apartment 10000
```

from anonmyized set

```
head(aibnb_anon[aibnb_anon$price>5000,c(1,2,7)])
```

```
##                                                                 id
## 3538 1ce52a0bee98b63a3e58f2e1f99b44910ea4d8d7094a6881daf9ef9b267c9e28
## 3721 5d03520db5d1f4dbd71e7695a1ecbb9b10b2351f7d559806f8749b2b7f9314e2
## 3775 2af2a5897e6c367570df1ad477e43a4bef67ba91342ac40e9402d81e475cdd91
```

```
## 4346 75b9fbe60065974ab8ad6d68397a1586bb9aa78f32020eea5bffbc9682eecbe8
## 4378 d077347c4b48b36bd29fda3951fb817cc74290a9803ad7c6c973ebb8c5194815
## 6531 a9174f68e7c2a9ef0f973bcc8b405429f2ff95d0964bb1e9634054e519ccff47
##                      name    price
## 3538         Velez, Jorge 6000.000
## 3721         Clark, Jamie 5263.798
## 3775         Acosta, Ryan 6500.000
## 4346 Rojas-Hudson, Ashley 5002.840
## 4378   el-Dawood, Nabeela 8000.000
## 6531      el-Aziz, Azzaam 9999.000
```

```
nrow(aibnb_anon[aibnb_anon$price>5000,c(1,2,7)])
```

```
## [1] 2259
```

as we can see the first query returns about 20 rows i used head not to display all the rows , and the same query on the anonymizaed data disaplsy about 2000 rows we hide the identity of the person we were searching for

we would consider searching for a person who lives in an apartment in NoHo in Manhattan , knowing he stayed about 14 nights and took an entire home

from un anonmyized set

```
aibnb[aibnb$neighbourhood=="NoHo" & aibnb$minimum_night == "14",c(1,2,5,6,11)]
```

```
##             id                              name neighbourhood_group
## 24951 19993470 Classic Manhattan Loft Apartment           Manhattan
##       neighbourhood minimum_nights
## 24951          NoHo             14
```

from anonmyized set

```
head(aibnb_anon[str_detect(aibnb_anon$Location,"Manhattan") & aibnb_anon$minimum_night == "[0,80]",])
```

```
##                                                                 id
## 2 9472e4a8d2ea5e8f87b6dd95996b5b863b15df3916f92fd056339cd6ea95e9c9
## 3 de648d5fcfd29e87f22228af61350554c6ad0b54f09d71cfa635438e3aa009bd
## 5 ddd83c1165d143e225b8de18ec79e55b77f0cd159b2f2a5b62016b4cde4d2c68
## 6 f4c886ab9ed7c55001f37dd4f8ff07fae97795f2b5e57fc968a84dc929ae18c7
## 8 5560611df9a59319278e0155c546217ba141f74ba1ec289b6657c0ae13a1b4f1
## 9 2356d61efeb10cb252aa3fcf5d3e0f11c375229589b5076d903dcb1d491a03d4
##                name host_id latitude longitude room_type     price
## 2     Laugan, Jimmy  284568  40.8448  -73.8648 Apartment 236.27484
## 3    Munoz, Armanda  463262  40.8448  -73.8648      Room 140.42768
## 5      Maestas, Ana  719220  40.8448  -73.8648 Apartment  91.54141
## 6 Pulling, Dominick  732218  40.8448  -73.8648 Apartment 205.45792
## 8     Yanito, Aaron  896790  40.8448  -73.8648      Room  86.60810
## 9  Beards, Emmanuel  749099  40.8448  -73.8648      Room  71.00188
##   number_of_reviews last_review reviews_per_month
## 2                45     2019-05              0.38
## 3                 0          NA              0.00
```

```
## 5                  9     2018-11              0.10
## 6                 74     2019-06              0.59
## 8                430     2019-06              3.47
## 9                118     2017-07              0.99
##   calculated_host_listings_count availability_365         Location
## 2                              2              355 MidtownManhattan
## 3                              1              365  UptownManhattan
## 5                              1                0  UptownManhattan
## 6                              1              129 MidtownManhattan
## 8                              1              220 MidtownManhattan
## 9                              1                0  UptownManhattan
##   minimum_night
## 2        [0,80]
## 3        [0,80]
## 5        [0,80]
## 6        [0,80]
## 8        [0,80]
## 9        [0,80]
```

```
nrow(aibnb_anon[str_detect(aibnb_anon$Location,"Manhattan") & aibnb_anon$minimum_night == "[0,80]",])
```

```
## [1] 21891
```

As we can see querying from the un anonymized we find the person directly but the anonymized data set returns about 21k enetry if an adversary want to search fro a person he can only know which part of manhattan he lived , results of rows are displayed in n row fnction returns number of rows from query