main_sap_rozi_jednorozi

Rozi jednorozi

29 05 2020

##Inicijalizacija podataka, ciscenje dataframe, obrada testova nad amenitiesima

```
#file odakle su funkcije i koje biblioteke se koriste
source("functions.R")
library(ggplot2)

#Ucitavanje i odabir zeljenih columna
airbnb<-read.csv("projekt.csv")
airbnb_main_frame<-as.data.frame(airbnb)

#maknut sam id jer je u izvornom stanju neupotrebljiv, par redova dolje je dodan kako ide redosljed
airbnb_bitno <- subset(airbnb_main_frame, select = -c( id,bed_type, description, first_review,  host_has



#ciscenje svih redaka koji sadrze NA vrijednosti
airbnb_bitno<- na.omit(airbnb_bitno)


#dodan id vector, treba nam za funkciju get_positions_of_empty_amenities
airbnb_bitno=get_Dataframe_With_Id_Row(airbnb_bitno)

#vector koji sadrzi sve pozicije gdje je amenities prazan tj {}
vector_of_empty_amenities=get_positions_of_empty_amenities(airbnb_bitno$amenities, airbnb_bitno$id)

#funkcija koja vraca dataframe bez redova gdje su amentites prazani
airbnb_bitno=clean_Empty_Amenities(airbnb_bitno, vector_of_empty_amenities)

#ponovno skidamo id jer kad maknemo prazne redove onda nastaju rupe
airbnb_bitno <- subset(airbnb_bitno, select = -c(id))

#refresh i dodavanje id-a nazad tako da ide 1,2,3,4.. do kraja bez rupa
airbnb_bitno=get_Dataframe_With_Id_Row(airbnb_bitno)

numOfRows=nrow(airbnb_bitno)

############################################################

#stvara listu koja sadrzi vector odvojenih amenitiesa za svaki red
#u petlji se jos stvara i vector koji govori koliko je amenitiesa u odredenom redu
list=list()
number_of_amentites=c()
```

```
vector_of_all_amenities=c()
counter=0
counter2=0
for(i in 1:numOfRows) {
  temp_vector_of_amenities=make_Vector_Of_Amentities(airbnb_bitno$amenities[i])
  list[[i]]=temp_vector_of_amenities
  number_of_amentites[i]=length(list[[i]])-1
  size_of_temp=length(temp_vector_of_amenities)
  counter2=counter2+1


  for(j in 1:size_of_temp) {
    counter=counter+1
    vector_of_all_amenities[counter]=temp_vector_of_amenities[j]
  }
}
#dodaje dodatni column koji sadrzi broj amenitiesa za taj red
airbnb_bitno <-  cbind(airbnb_bitno , number_of_amentites)

#list_of_vectors_of_amenities=list()

vector_of_unique_amenities=unique(vector_of_all_amenities)

list_of_tests=list()
#za svaki jedinstveni amenitie prolazi kroz dataframe
#pravi vector koji sadrzi true ili false vrijednost
#ovisno o tome je li stan u tom redu sadrzi taj amenitie
#na kraju spaja taj vector sa pocetnim dataframeom
#ovaj dio dodaje svih 129 amenitiea u airbnb_bitno
#zato je dataframe tako velik
count129=0
len_list=length(list)
number_of_apartments_that__have_certain_amenitie_vector=c()
for(i in 1:129) {


  temp_vector_of_amenities=rep(F, 73218)
  count129=count129+1
  print(paste("Loaded ", count129, "out of 129 amenities"))

  current_counter=0
  for (j in 1:len_list) {
    if(vector_of_unique_amenities[i] %in% list[[j]]) {
      temp_vector_of_amenities[j]=T
      current_counter=current_counter+1
    }
  }
  number_of_apartments_that__have_certain_amenitie_vector[i]=current_counter
  airbnb_bitno <-  cbind(airbnb_bitno , temp_vector_of_amenities)
  names(airbnb_bitno)[names(airbnb_bitno) == "temp_vector_of_amenities"] <- vector_of_unique_amenities[

  if(i<124) {
   test=t.test(airbnb_bitno$price~temp_vector_of_amenities, mu=0, alt="two.sided", conf=0.99)
```

```
   test$data.name=paste("price and ", vector_of_unique_amenities[i])
   list_of_tests[[i]]=test

  }

}
```

```
## [1] "Loaded  1 out of 129 amenities"
## [1] "Loaded  2 out of 129 amenities"
## [1] "Loaded  3 out of 129 amenities"
## [1] "Loaded  4 out of 129 amenities"
## [1] "Loaded  5 out of 129 amenities"
## [1] "Loaded  6 out of 129 amenities"
## [1] "Loaded  7 out of 129 amenities"
## [1] "Loaded  8 out of 129 amenities"
## [1] "Loaded  9 out of 129 amenities"
## [1] "Loaded  10 out of 129 amenities"
## [1] "Loaded  11 out of 129 amenities"
## [1] "Loaded  12 out of 129 amenities"
## [1] "Loaded  13 out of 129 amenities"
## [1] "Loaded  14 out of 129 amenities"
## [1] "Loaded  15 out of 129 amenities"
## [1] "Loaded  16 out of 129 amenities"
## [1] "Loaded  17 out of 129 amenities"
## [1] "Loaded  18 out of 129 amenities"
## [1] "Loaded  19 out of 129 amenities"
## [1] "Loaded  20 out of 129 amenities"
## [1] "Loaded  21 out of 129 amenities"
## [1] "Loaded  22 out of 129 amenities"
## [1] "Loaded  23 out of 129 amenities"
## [1] "Loaded  24 out of 129 amenities"
## [1] "Loaded  25 out of 129 amenities"
## [1] "Loaded  26 out of 129 amenities"
## [1] "Loaded  27 out of 129 amenities"
## [1] "Loaded  28 out of 129 amenities"
## [1] "Loaded  29 out of 129 amenities"
## [1] "Loaded  30 out of 129 amenities"
## [1] "Loaded  31 out of 129 amenities"
## [1] "Loaded  32 out of 129 amenities"
## [1] "Loaded  33 out of 129 amenities"
## [1] "Loaded  34 out of 129 amenities"
## [1] "Loaded  35 out of 129 amenities"
## [1] "Loaded  36 out of 129 amenities"
## [1] "Loaded  37 out of 129 amenities"
## [1] "Loaded  38 out of 129 amenities"
## [1] "Loaded  39 out of 129 amenities"
## [1] "Loaded  40 out of 129 amenities"
## [1] "Loaded  41 out of 129 amenities"
## [1] "Loaded  42 out of 129 amenities"
## [1] "Loaded  43 out of 129 amenities"
## [1] "Loaded  44 out of 129 amenities"
## [1] "Loaded  45 out of 129 amenities"
## [1] "Loaded  46 out of 129 amenities"
```

```
## [1] "Loaded  47 out of 129 amenities"
## [1] "Loaded  48 out of 129 amenities"
## [1] "Loaded  49 out of 129 amenities"
## [1] "Loaded  50 out of 129 amenities"
## [1] "Loaded  51 out of 129 amenities"
## [1] "Loaded  52 out of 129 amenities"
## [1] "Loaded  53 out of 129 amenities"
## [1] "Loaded  54 out of 129 amenities"
## [1] "Loaded  55 out of 129 amenities"
## [1] "Loaded  56 out of 129 amenities"
## [1] "Loaded  57 out of 129 amenities"
## [1] "Loaded  58 out of 129 amenities"
## [1] "Loaded  59 out of 129 amenities"
## [1] "Loaded  60 out of 129 amenities"
## [1] "Loaded  61 out of 129 amenities"
## [1] "Loaded  62 out of 129 amenities"
## [1] "Loaded  63 out of 129 amenities"
## [1] "Loaded  64 out of 129 amenities"
## [1] "Loaded  65 out of 129 amenities"
## [1] "Loaded  66 out of 129 amenities"
## [1] "Loaded  67 out of 129 amenities"
## [1] "Loaded  68 out of 129 amenities"
## [1] "Loaded  69 out of 129 amenities"
## [1] "Loaded  70 out of 129 amenities"
## [1] "Loaded  71 out of 129 amenities"
## [1] "Loaded  72 out of 129 amenities"
## [1] "Loaded  73 out of 129 amenities"
## [1] "Loaded  74 out of 129 amenities"
## [1] "Loaded  75 out of 129 amenities"
## [1] "Loaded  76 out of 129 amenities"
## [1] "Loaded  77 out of 129 amenities"
## [1] "Loaded  78 out of 129 amenities"
## [1] "Loaded  79 out of 129 amenities"
## [1] "Loaded  80 out of 129 amenities"
## [1] "Loaded  81 out of 129 amenities"
## [1] "Loaded  82 out of 129 amenities"
## [1] "Loaded  83 out of 129 amenities"
## [1] "Loaded  84 out of 129 amenities"
## [1] "Loaded  85 out of 129 amenities"
## [1] "Loaded  86 out of 129 amenities"
## [1] "Loaded  87 out of 129 amenities"
## [1] "Loaded  88 out of 129 amenities"
## [1] "Loaded  89 out of 129 amenities"
## [1] "Loaded  90 out of 129 amenities"
## [1] "Loaded  91 out of 129 amenities"
## [1] "Loaded  92 out of 129 amenities"
## [1] "Loaded  93 out of 129 amenities"
## [1] "Loaded  94 out of 129 amenities"
## [1] "Loaded  95 out of 129 amenities"
## [1] "Loaded  96 out of 129 amenities"
## [1] "Loaded  97 out of 129 amenities"
## [1] "Loaded  98 out of 129 amenities"
## [1] "Loaded  99 out of 129 amenities"
## [1] "Loaded  100 out of 129 amenities"
```

```
## [1] "Loaded  101 out of 129 amenities"
## [1] "Loaded  102 out of 129 amenities"
## [1] "Loaded  103 out of 129 amenities"
## [1] "Loaded  104 out of 129 amenities"
## [1] "Loaded  105 out of 129 amenities"
## [1] "Loaded  106 out of 129 amenities"
## [1] "Loaded  107 out of 129 amenities"
## [1] "Loaded  108 out of 129 amenities"
## [1] "Loaded  109 out of 129 amenities"
## [1] "Loaded  110 out of 129 amenities"
## [1] "Loaded  111 out of 129 amenities"
## [1] "Loaded  112 out of 129 amenities"
## [1] "Loaded  113 out of 129 amenities"
## [1] "Loaded  114 out of 129 amenities"
## [1] "Loaded  115 out of 129 amenities"
## [1] "Loaded  116 out of 129 amenities"
## [1] "Loaded  117 out of 129 amenities"
## [1] "Loaded  118 out of 129 amenities"
## [1] "Loaded  119 out of 129 amenities"
## [1] "Loaded  120 out of 129 amenities"
## [1] "Loaded  121 out of 129 amenities"
## [1] "Loaded  122 out of 129 amenities"
## [1] "Loaded  123 out of 129 amenities"
## [1] "Loaded  124 out of 129 amenities"
## [1] "Loaded  125 out of 129 amenities"
## [1] "Loaded  126 out of 129 amenities"
## [1] "Loaded  127 out of 129 amenities"
## [1] "Loaded  128 out of 129 amenities"
## [1] "Loaded  129 out of 129 amenities"
```

```
#izbacivanje testova koji nista ne znace:
#translationmissingenhostingamenity49 i translationmissingenhostingamenity50

clean_list_of_tests=list()
number_of_tests=length(list_of_tests)
counter=0
for (i in 1:number_of_tests) {
  if(list_of_tests[[i]]$data.name!="price and  translationmissingenhostingamenity50" &&
     list_of_tests[[i]]$data.name!="price and  translationmissingenhostingamenity49") {
    counter=counter+1
    clean_list_of_tests[[counter]]=list_of_tests[[i]]
  }
}
```

```r
#pregled koliko stanova ima koji amenitie

names(number_of_apartments_that__have_certain_amenitie_vector)=vector_of_unique_amenities
number_of_apartments_that__have_certain_amenitie_vector
```

```
##                      WirelessInternet                      Airconditioning
##                                 70968                                54994
##                              Kitchen                              Heating
##                                 67257                                66806
##                    Familykidfriendly                            Essentials
##                                 36939                                63855
##                             Hairdryer                                 Iron
##                                 43240                                41593
## translationmissingenhostingamenity50                               Washer
##                                 25226                                43010
##                                Dryer                          Smokedetector
##                                 42554                                61576
##                     Fireextinguisher                              Shampoo
##                                 30656                                49343
##                              Hangers                                   TV
##                                 49054                                52159
##                              CableTV                            Breakfast
##                                 24143                                 8284
##                Buzzerwirelessintercom                Carbonmonoxidedetector
##                                 16955                                47072
##              Laptopfriendlyworkspace                             Internet
##                                 43605                                44464
##                      Indoorfireplace                           Firstaidkit
##                                 9275                                 27478
##                   Elevatorinbuilding                                 Pool
##                                 6400                                 6265
##                 Freeparkingonpremises                                  Gym
##                                 23564                                7462
##                               Hottub                  Wheelchairaccessible
##                                 6318                                 4836
##                              Doorman                                 Dogs
##                                 4376                                 5240
##                                Cats                               Otherpets
##                                 3569                                  375
##                    Lockonbedroomdoor                Petsliveonthisproperty
##                               17952                                 9685
##                     Privateentrance                              Hotwater
##                                 7264                                 4255
##                            Bedlinens                  Extrapillowsandblankets
##                                 4166                                 3019
##                          Coffeemaker                           Refrigerator
##                                 3545                                 4637
##                   Dishesandsilverware                       Gardenorbackyard
```

```
##                                4244                                1128
##                          Petsallowed                          Safetycard
##                               10168                               11492
##                         24hourcheckin                         SelfCheckIn
##                               18973                               11028
##                             Lockbox                     Suitableforevents
##                                5733                                4248
## translationmissingenhostingamenity49                            Elevator
##                               20366                               10773
##                           Microwave                         Cookingbasics
##                                3901                                3946
##                                Oven                               Stove
##                                3933                                4070
##                      Smokingallowed                         Outletcovers
##                                3663                                 446
##                             Bathtub                           Dishwasher
##                                3760                                2310
##                      Singlelevelhome                Luggagedropoffallowed
##                                 518                                1635
##                      Stepfreeaccess                   Wideclearancetobed
##                                 827                                 404
##                   Accessibleheightbed                       Widedoorway
##                                 348                                 560
##                Accessibleheighttoilet                      Wideentryway
##                                 260                                 301
##                     Hotwaterkettle                Babysitterrecommendations
##                                 173                                 677
##                   PackânPlaytravelcrib                           Keypad
##                                1124                                2992
##                   Freeparkingonstreet                         Smartlock
##                                  77                                1442
##                   Widehallwayclearance                           Flat
##                                 472                                 396
##               smoothpathwaytofrontdoor                Welllitpathtoentrance
##                                 396                                 769
##                  Longtermstaysallowed                   Privatelivingroom
##                                1666                                2516
##                Childrenâsbooksandtoys                 Childrenâsdinnerware
##                                1126                                 635
##                              BBQgrill                       Patioorbalcony
##                                 632                                1218
##                      Hostgreetsyou                      Ethernetconnection
##                                1195                                 690
##           Fixedgrabbarsforshowertoilet                          Highchair
##                                  67                                 635
##                   Roomdarkeningshades                       Changingtable
##                                1496                                 225
##                           Stairgates                         Windowguards
##                                 299                                 612
##                     Tablecornerguards                             Crib
##                                  88                                 442
##                               Other                       Privatebathroom
##                                 217                                  74
##                         DoormanEntry                         Babymonitor
```

```
##                                452                                133
##                            Babybath                     Fireplaceguards
##                                305                                219
##                         Gameconsole                       Firmmattress
##                                478                                115
##                          Pocketwifi                Cleaningbeforecheckout
##                                217                                268
##                  Handheldshowerhead                         Waterfront
##                                135                                115
##                   Groundflooraccess                          EVcharger
##                                 25                                 48
##          Wideclearancetoshowertoilet                    Beachessentials
##                                122                                102
##                          Beachfront                Bathtubwithshowerchair
##                                 33                                 34
##                          Lakeaccess                       WasherDryer
##                                 18                                 28
##                 Disabledparkingspot                Rollinshowerwithchair
##                                 34                                  3
##                          Airpurifier                        SkiinSkiout
##                                 14                                 13
##             Pathtoentrancelitatnight                        Firmmatress
##                                 27                                 15
##               Paidparkingoffpremises         Flatsmoothpathwaytofrontdoor
##                                  7                                 11
##          Grabrailsforshowerandtoilet                          Bodysoap
##                                  2                                  1
##                             Handsoap                          Bathtowel
##                                  1                                  1
##                     Handorpapertowel                       Toiletpaper
##                                  1                                  1
##       Wideclearancetoshowerandtoilet
##                                  1
```

```
List_of_prices_per_city=sort_Numeric_Categorical(airbnb_bitno$price, airbnb_bitno$city)
List_of_price_outliers_per_city=list()
len=length(List_of_prices_per_city)

#prebrojavanje outliera po gradovima
number_of_outliers=list()
relative_number_of_outliers=list()


for(i in 1:len) {
  num_of_apartments_per_city=length(List_of_prices_per_city[[i]])

  temp_vector=c(List_of_prices_per_city[[i]])

  Q1=quantile(temp_vector, 0.25)
  Q1=unname(Q1)
  Q3=quantile(temp_vector, 0.75)
  Q3=unname(Q3)
  IQR=Q3-Q1
```

```
  donja_granica=Q1 - 1.5*IQR
  gornja_granica=Q3 + 1.5*IQR
  print(paste("donja_granica",donja_granica))
  print(paste("gornja_granica",gornja_granica))

  temp_prices=c()
  counter=0
  for(j in 1:num_of_apartments_per_city) {

    if(temp_vector[j] >= gornja_granica || temp_vector[j] <= donja_granica){
      counter = counter + 1
      temp_prices[counter]=temp_vector[j]


    }


  number_of_outliers[[i]]=counter
  relative_number_of_outliers[[i]]=counter/num_of_apartments_per_city
  List_of_price_outliers_per_city[[i]]=temp_prices
  }
}
```

```
## [1] "donja_granica -87.5000000000001"
## [1] "gornja_granica 332.5"
## [1] "donja_granica -117.5"
## [1] "gornja_granica 486.5"
## [1] "donja_granica -125"
## [1] "gornja_granica 434.999999999999"
## [1] "donja_granica -79.9999999999997"
## [1] "gornja_granica 320"
## [1] "donja_granica -62.5"
## [1] "gornja_granica 277.5"
## [1] "donja_granica -99.9999999999999"
## [1] "gornja_granica 380"
```

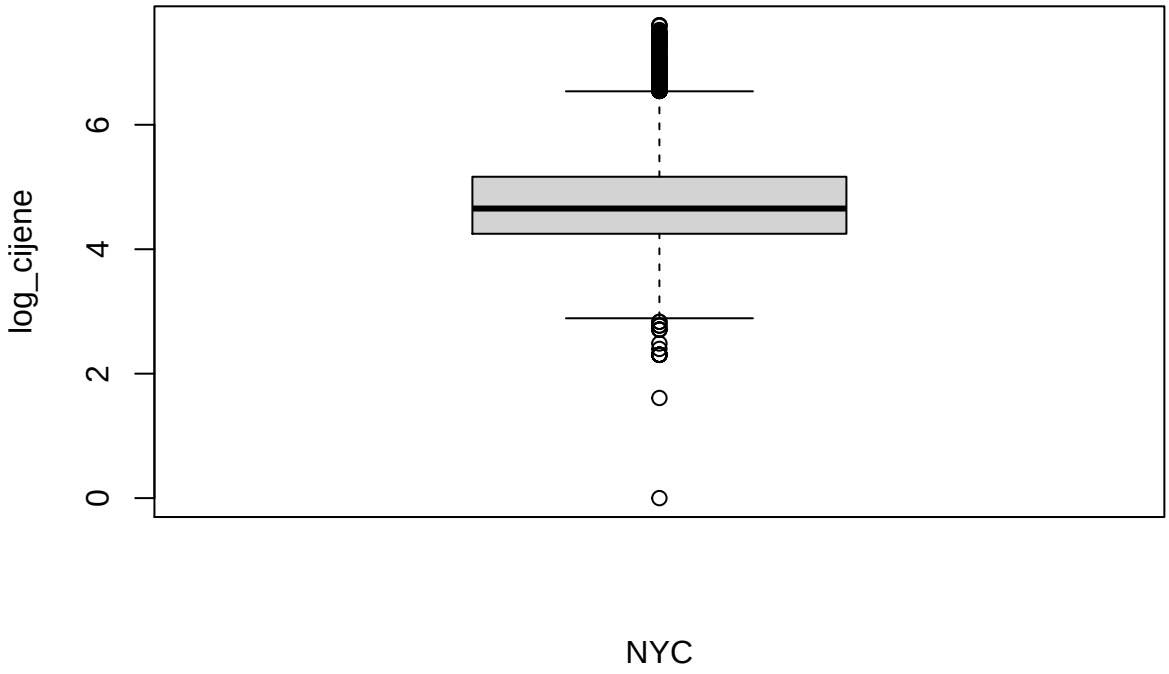
```
  names(number_of_outliers) <- unique(airbnb_bitno$city)
  names(relative_number_of_outliers) <- unique(airbnb_bitno$city)
  names(List_of_price_outliers_per_city) <- unique(airbnb_bitno$city)
  print("iz ispisa vidimo da su samo gornje granice bitne")
```

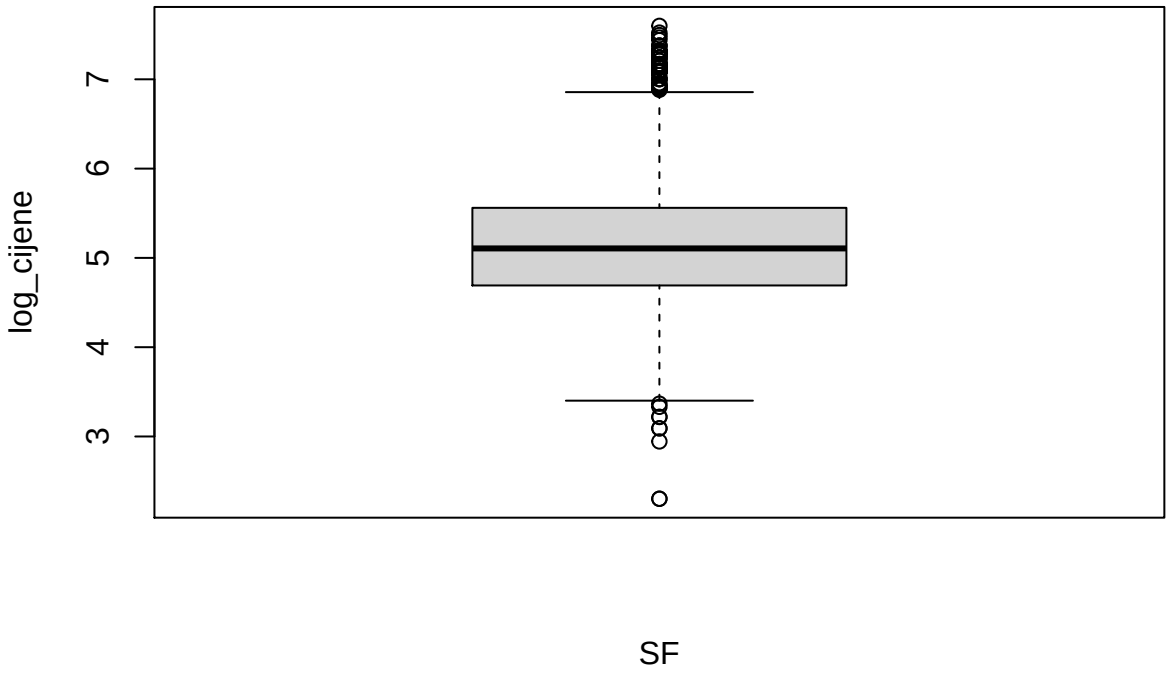
```
## [1] "iz ispisa vidimo da su samo gornje granice bitne"
```

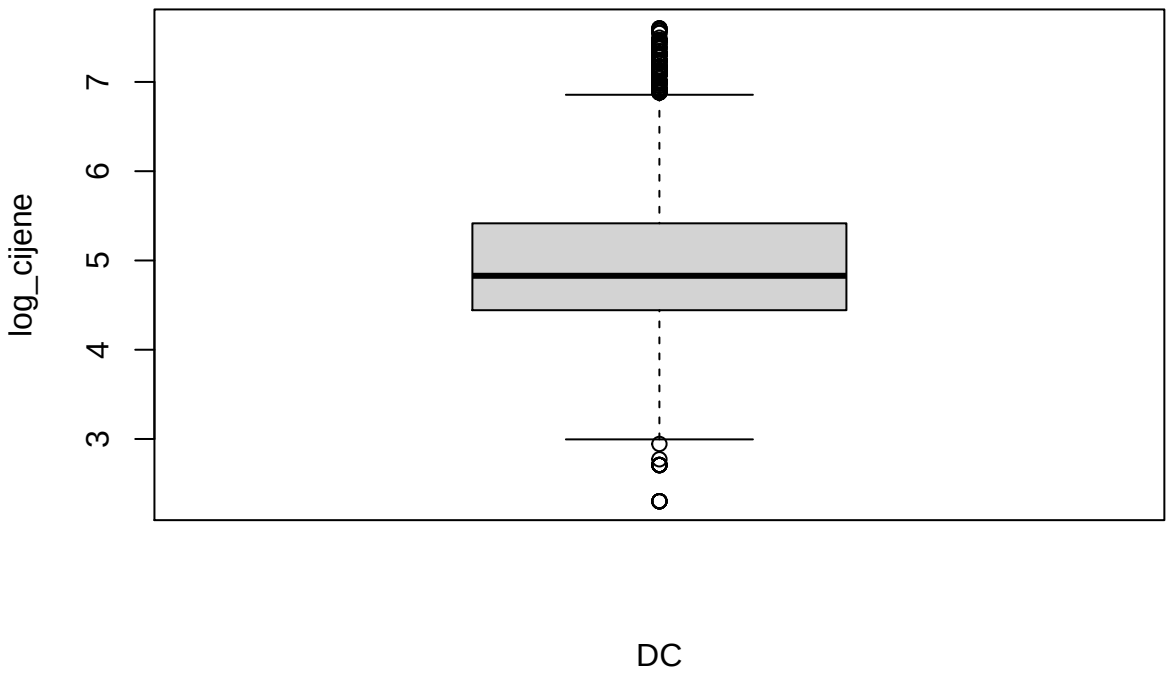
```
  print("ocekivano, razdioba cijena je nagnuta u lijevo")
```

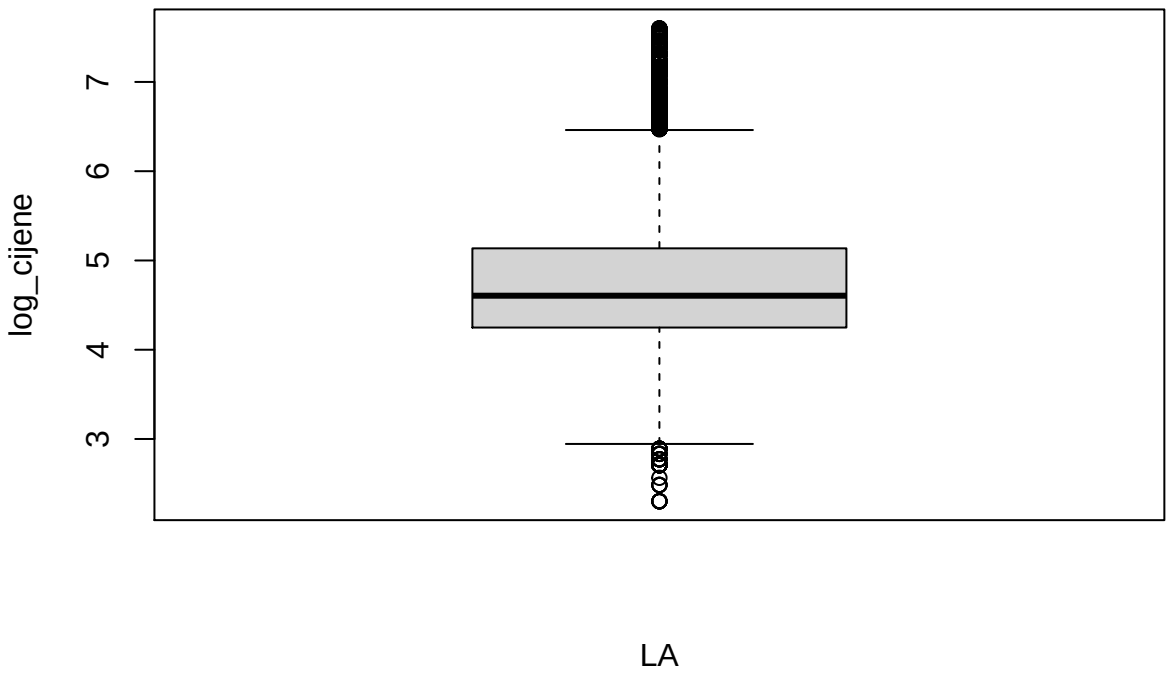
```
## [1] "ocekivano, razdioba cijena je nagnuta u lijevo"
```

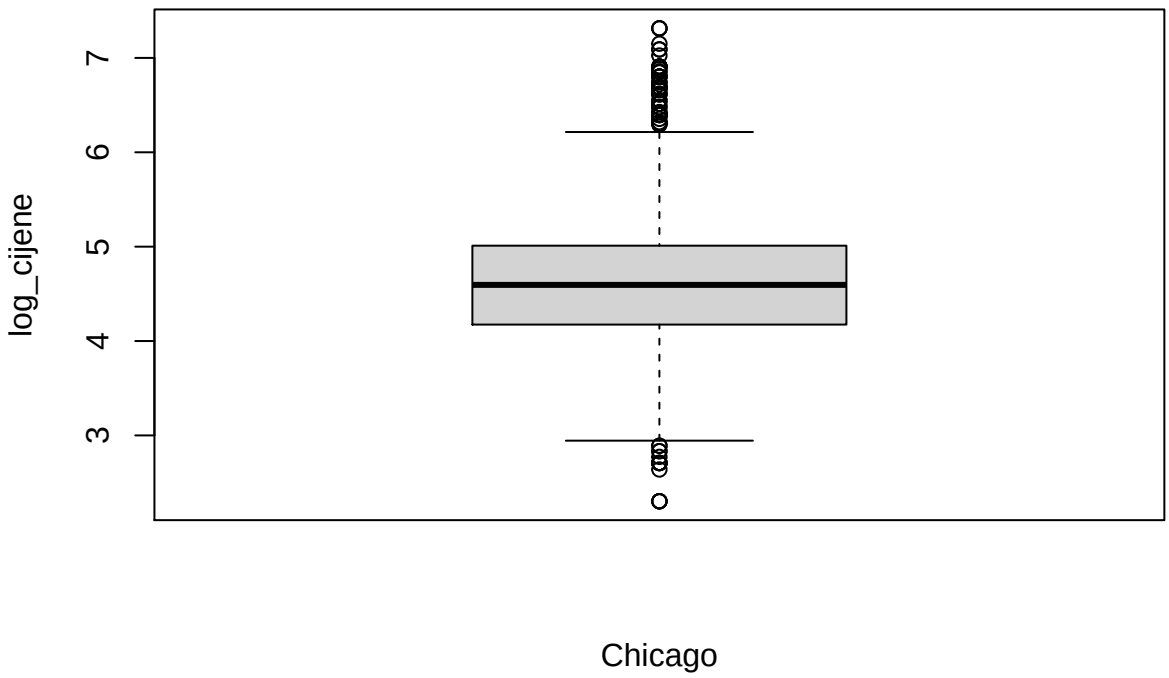
Deskriptivna statistika

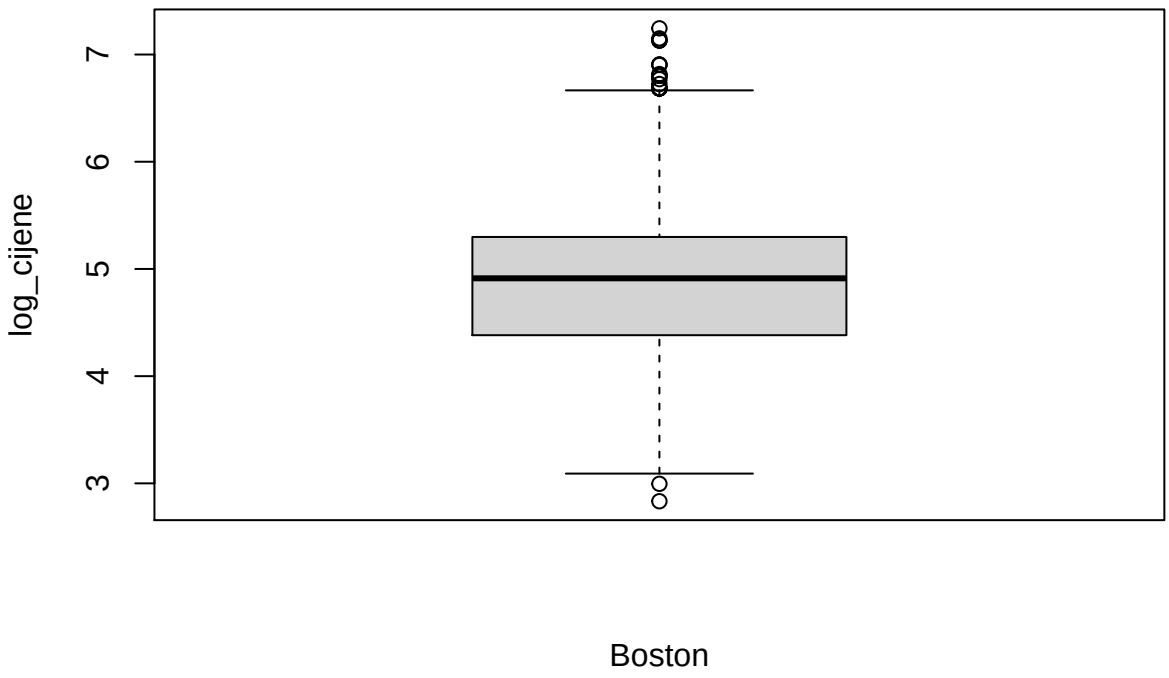


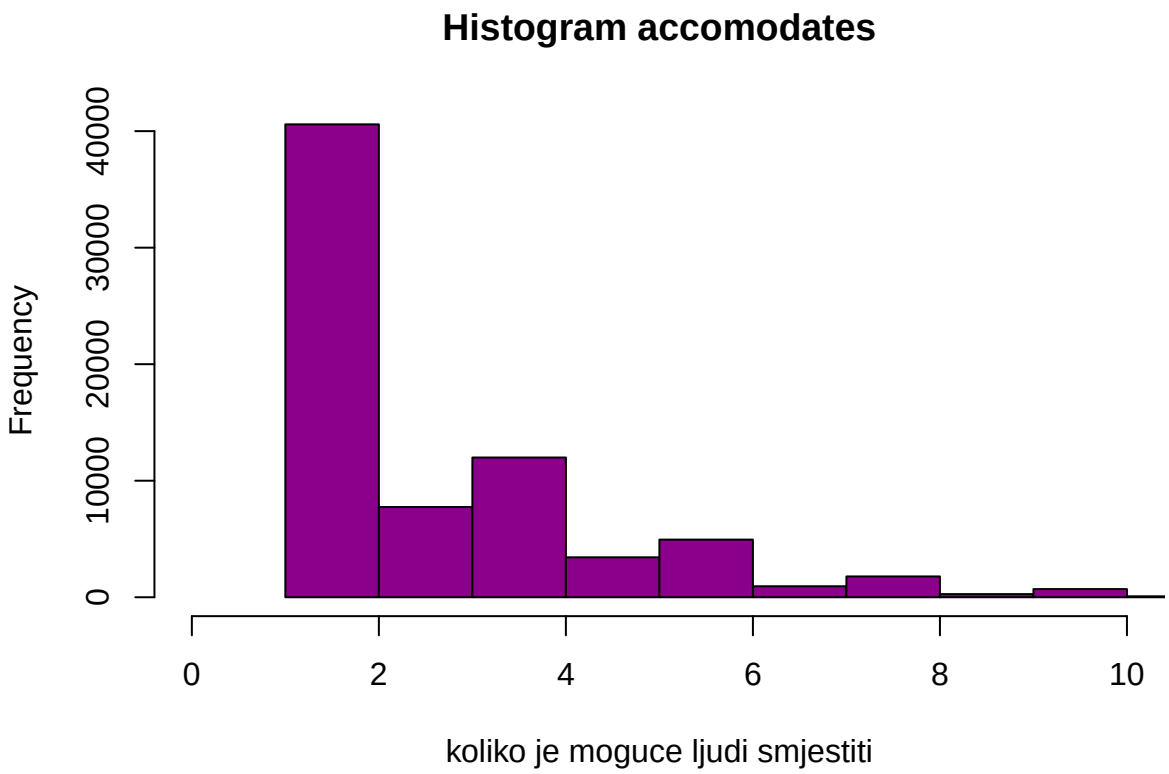


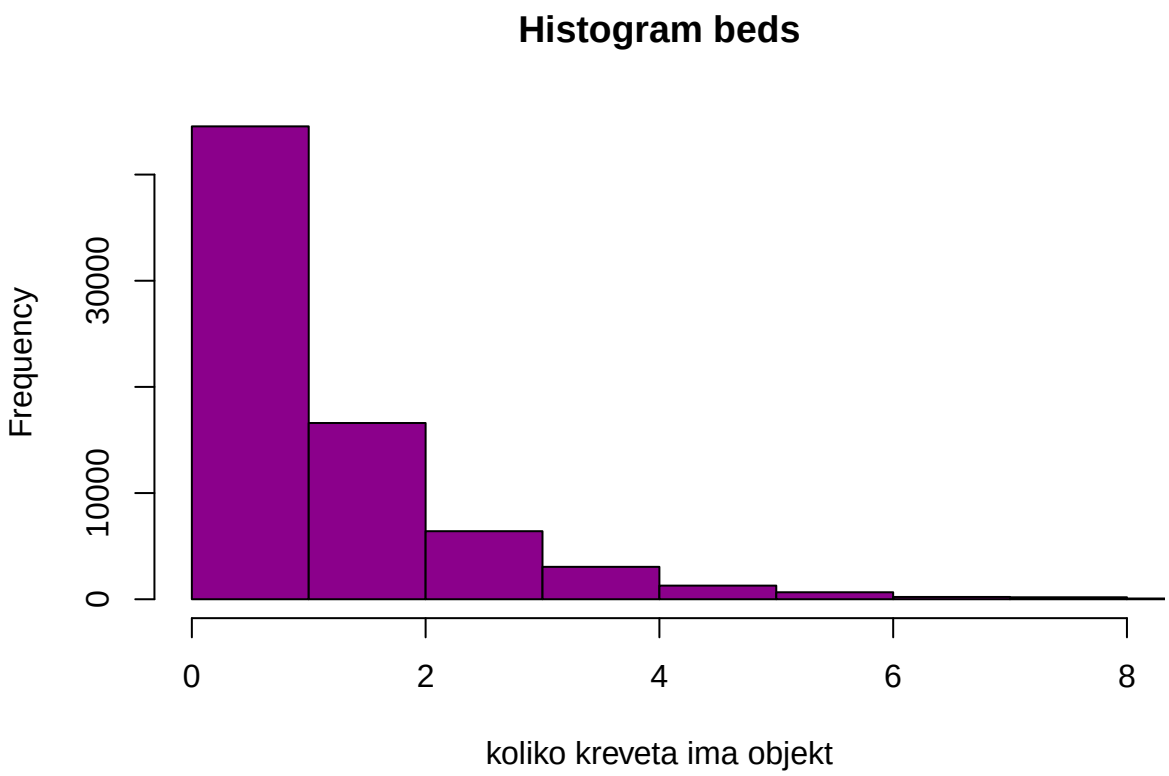

log_cijene
3
4
5
6
7
DC

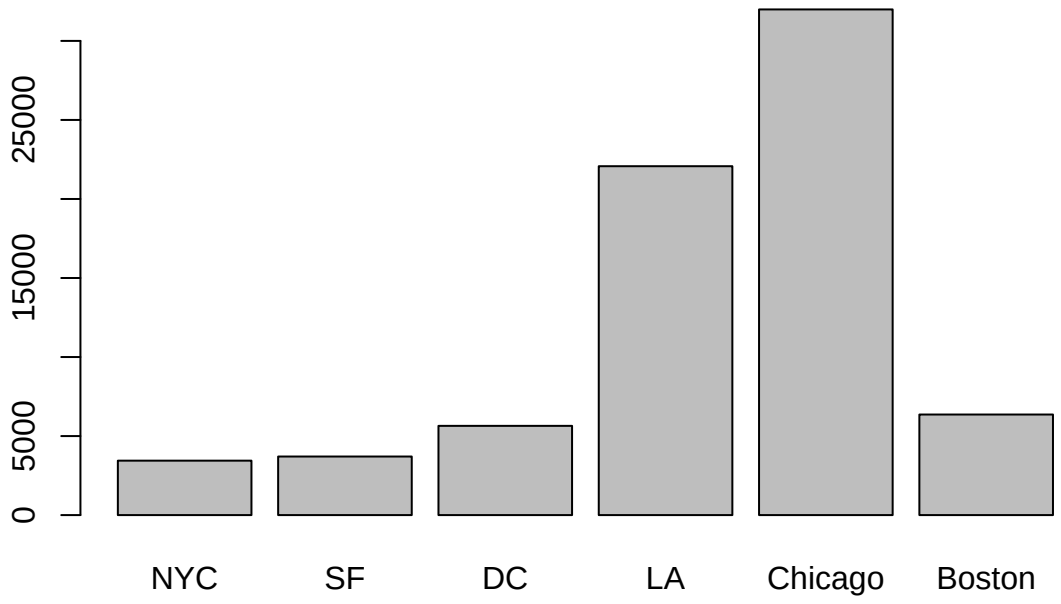

log_cijene
3
4
5
6
7
LA

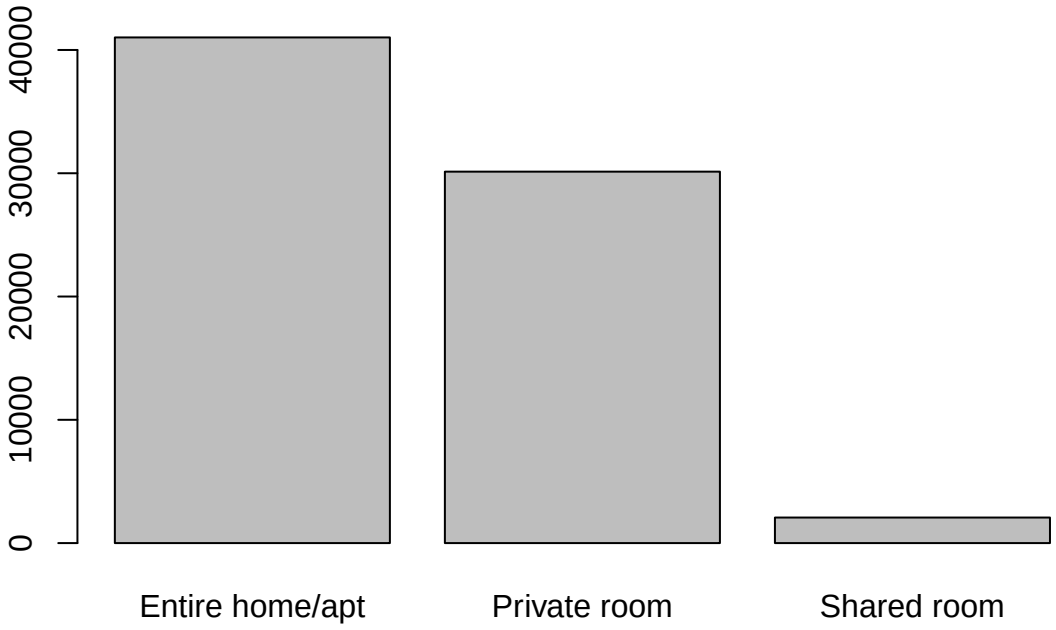

log_cijene
3
4
5
6
7
Chicago

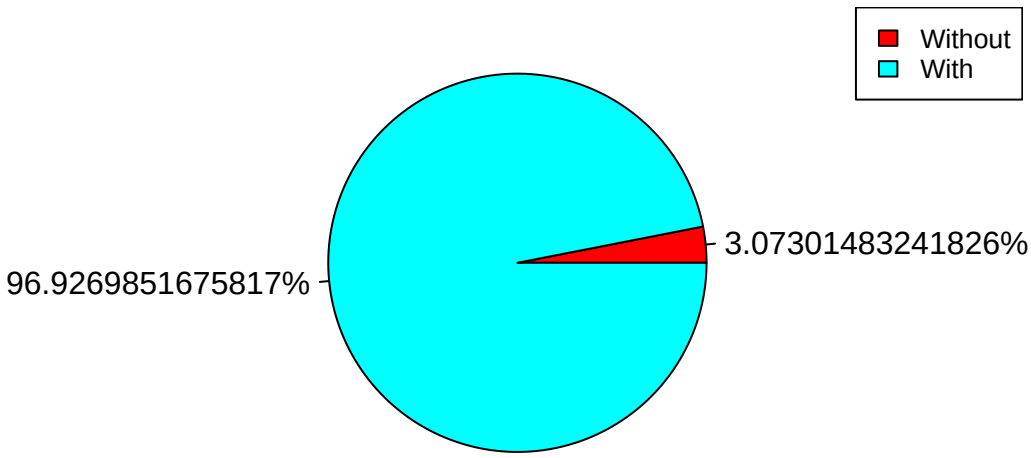

log_cijene
3
4
5
6
7
Boston


Histogram accomodates
Frequency
0
10000
20000
30000
40000
0
2
4
6
8
10
koliko je moguce ljudi smjestiti


Histogram beds
Frequency
0
10000
30000
0
2
4
6
8
koliko kreveta ima objekt


0
5000
15000
25000
NYC
SF
DC
LA
Chicago
Boston


0
10000
20000
30000
40000
Entire home/apt
Private room
Shared room




```
#pitanja:

#ima li neki grad znacajno veci broj outliera?
relative_number_of_outliers
```

```
## $NYC
## [1] 0.0549217
##
## $SF
## [1] 0.08299277
##
## $DC
## [1] 0.1174491
##
## $LA
## [1] 0.08349946
##
## $Chicago
## [1] 0.07638327
##
## $Boston
## [1] 0.05489399
```

```
#DC-ima najveci relativni udio outliera
#za nas skup podataka outlieri su samo visoke cijene
```

```r
#jer je razdioba cijena nagnuta na lijevo
#u iducem pitanju se vidi da od svih numerickih varijabli  iz
#pocetnog dataseta accommodates najkoreliranija sa cijenom objekta
#to svojstvo cemo koristiti i ovdje te cemo t-testom
#pokazati da objekti u DC-u u prosjeku mogu primiti vise ljudi
#u odnosi na prosjek svih objekata u svim gradovima
mean_of_all_cities_accommodates=mean(airbnb_bitno$accommodates)
vector_of_positions=get_DC(airbnb_bitno$city, airbnb_bitno$id)
dc=airbnb_bitno[vector_of_positions,]
dc <- subset(dc, select = -c(id))


dc=get_Dataframe_With_Id_Row(dc)

test=t.test(dc$accommodates, mu=mean_of_all_cities_accommodates, alt="two.sided", conf=0.99)
print(paste("svi gradovi accommodates:", mean_of_all_cities_accommodates," DC: ", test$estimate))
```

```
## [1] "svi gradovi accommodates: 3.16579256466989  DC:  3.54915854738707"
```

```r
print(paste("mean je povecan za:", ((test$estimate-mean_of_all_cities_accommodates)
                                    /mean_of_all_cities_accommodates), "posto" ))
```

```
## [1] "mean je povecan za: 0.121096368408823 posto"
```

```r
print(test)
```

```
##
##  One Sample t-test
##
## data:  dc$accommodates
## t = 13.178, df = 5644, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 3.165793
## 99 percent confidence interval:
##  3.474199 3.624118
## sample estimates:
## mean of x
##  3.549159
```

```r
#u DC-u 357 outliera su objekti sa bar 5 accommodatesa
#ukupno outliera je 663
#prema gornja_granica=Q3+1.5IQR
outlier_price=435

dc_price_per_accommodates=sort_Numeric_Numeric(dc$price, dc$accommodates)
list_numbers=list(5,6,7,8,9,10,11,12,13,14,15,16)
counter=0
big_apartments=length(list_numbers)
for (i in 1:big_apartments) {
  current=list_numbers[[i]]
  l=length(dc_price_per_accommodates[[current]])
  ll=dc_price_per_accommodates[[current]]
```

```
  for (i in 1:l) {
    if (ll[i]>outlier_price) {
      counter=counter+1
    }
  }

}
print(paste("broj outliera sa preko 5 accommodates: ",counter))
```

```
## [1] "broj outliera sa preko 5 accommodates:  357"
```

```
#Koja od varijabli je najvise korelirana sa cijenom ugostiteljskog objekta? Ima li to smisla?
  airbnb_korelacija <- subset(airbnb_bitno, select = -c(property_type,room_type,cancellation_policy,clea
  airbnb_korelacija[is.na(airbnb_korelacija)] <- 0
  airbnb_korelacija <- lapply(airbnb_korelacija, as.numeric)
```

```
## Warning in lapply(airbnb_korelacija, as.numeric): NAs introduced by coercion

## Warning in lapply(airbnb_korelacija, as.numeric): NAs introduced by coercion
```

```
  with(airbnb_korelacija, cor(price, accommodates))
```

```
## [1] 0.5215426
```

```
  with(airbnb_korelacija, cor(price, bathrooms))
```

```
## [1] 0.4612756
```

```
  with(airbnb_korelacija, cor(price, number_of_reviews))
```

```
## [1] -0.07082064
```

```
  with(airbnb_korelacija, cor(price, bedrooms))
```

```
## [1] 0.496533
```

```
  with(airbnb_korelacija, cor(price, beds))
```

```
## [1] 0.4352527
```

```
#od numerickih varijabli najkoreliranija je accomodates
#sada ide provjera za amenitiese
#uzimamo samo testove koji imaju preko 10000 stupnjeva slobode
list_of_strings_correlation=list()
list_of_tests_enough_degrees_of_freedom=list()
number_of_valid_tests=length(clean_list_of_tests)
  counter=0
  for (i in 1:number_of_valid_tests) {
```

```
    if(clean_list_of_tests[[i]]$parameter>=10000) {
      temp_string=paste(clean_list_of_tests[[i]]$data.name, clean_list_of_tests[[i]]$conf.int[1],
" ", clean_list_of_tests[[i]]$conf.int[2],": p-value ", clean_list_of_tests[[i]]$p.value )
      counter=counter+1
      list_of_tests_enough_degrees_of_freedom[[counter]]=clean_list_of_tests[[i]]
      list_of_strings_correlation[[counter]]=temp_string
      print(temp_string)
      print("\n")
    }
  }
```

```
## [1] "price and  Airconditioning -13.8858076274485   -6.63073891384078 : p-value  3.30015436643326e-13
## [1] "\n"
## [1] "price and  Familykidfriendly -73.9130410536827   -67.6782811842195 : p-value  0"
## [1] "\n"
## [1] "price and  Essentials -14.1823640534319   -3.97287234644289 : p-value  4.66916192237339e-06"
## [1] "\n"
## [1] "price and  Hairdryer -24.8664551047106   -18.3594779955358 : p-value  1.69657958458808e-65"
## [1] "\n"
## [1] "price and  Iron -26.7452051527984   -20.3407430851092 : p-value  8.78878226108424e-80"
## [1] "\n"
## [1] "price and  Washer -55.1002124451845   -49.1620938301594 : p-value  0"
## [1] "\n"
## [1] "price and  Dryer -55.9082610873432   -49.9488287751579 : p-value  0"
## [1] "\n"
## [1] "price and  Smokedetector -13.2629159723203   -4.13783204733681 : p-value  9.08334895062772e-07"
## [1] "\n"
## [1] "price and  Fireextinguisher -27.7113010167365   -21.059820700186 : p-value  2.48050789337543e-79
## [1] "\n"
## [1] "price and  Shampoo -18.5745074375982   -11.6556690614951 : p-value  2.39470685770189e-29"
## [1] "\n"
## [1] "price and  Hangers -8.81313195529614   -1.84715340653733 : p-value  8.09096642070118e-05"
## [1] "\n"
## [1] "price and  TV -69.1696470959746   -63.2910948146106 : p-value  0"
## [1] "\n"
## [1] "price and  CableTV -65.7645170753627   -58.4176533407661 : p-value  0"
## [1] "\n"
## [1] "price and  Buzzerwirelessintercom -13.114871551265   -5.66530493724311 : p-value  8.49309862608
## [1] "\n"
## [1] "price and  Carbonmonoxidedetector -17.2975757133634   -10.6013854156219 : p-value  7.6456180419
## [1] "\n"
## [1] "price and  Laptopfriendlyworkspace -22.9085230573131   -16.4419416990809 : p-value  2.836025183
## [1] "\n"
## [1] "price and  Internet -20.5973186179269   -14.0670173770801 : p-value  1.68564251671432e-42"
## [1] "\n"
## [1] "price and  Indoorfireplace -101.690046449858   -87.4047733983284 : p-value  2.10522556619178e-24
## [1] "\n"
## [1] "price and  Firstaidkit -6.82570469799039   -0.0944205949140079 : p-value  0.00809485214894145"
## [1] "\n"
## [1] "price and  Freeparkingonpremises -31.2376078829187   -23.5702265616192 : p-value  2.305330300131
## [1] "\n"
## [1] "price and  Lockonbedroomdoor 24.8993513829261   32.2280592965084 : p-value  4.07032375992866e-89
## [1] "\n"
```

```
## [1] "price and  Petsliveonthisproperty 31.9519711140054   40.002294577991 : p-value  2.8377784372756
## [1] "\n"
## [1] "price and  Petsallowed -38.6679913879877   -27.8749084186549 : p-value  2.97078775305985e-56"
## [1] "\n"
## [1] "price and  Safetycard -21.59301936911   -12.2957806666272 : p-value  6.81575550049459e-21"
## [1] "\n"
## [1] "price and  24hourcheckin -26.3254991206186   -19.100352270327 : p-value  9.02956505360116e-59"
## [1] "\n"
## [1] "price and  SelfCheckIn -20.1063834301533   -12.1862616651451 : p-value  9.96275625231173e-26"
## [1] "\n"
## [1] "price and  Elevator -31.2147272632188   -22.4294628822546 : p-value  2.57928700977233e-55"
## [1] "\n"
```

```
  print(paste("broj amenitiesa koji imaju više od 10000 stupnjeva slobode", counter))
```

```
## [1] "broj amenitiesa koji imaju više od 10000 stupnjeva slobode 27"
```

```
  print("p-value 0 indicates extremly high correlation")
```

```
## [1] "p-value 0 indicates extremly high correlation"
```

```
  print("nas izbor 5 preporuka iznajmljivacima je imati:")
```

```
## [1] "nas izbor 5 preporuka iznajmljivacima je imati:"
```

```
  print("TV, CableTV, Freeparkingonpremises, Familykidfriendly, Fireextinguisher")
```

```
## [1] "TV, CableTV, Freeparkingonpremises, Familykidfriendly, Fireextinguisher"
```

```
# ima smisla da je broj osoba koje aparatman moze primiti uskokoreliran sa brojem osoba koje ce
# rezervirati apartman pa tako i sa cijenom
# od amenitiesa je uvjek pozeljno imati tv i kabelsku, besplatan parking na posjedu
# kao i biti pirpremljen na dolazak obitelji sa djecom
# vatrogasni aparat je vazan element sigurnosti i indicira na dobro opremljen apartman









#varijable koje smo odabrali: TV, CableTV, Freeparkingonpremises, Familykidfriendly, Fireextinguisher
#kominacije varijabli koje predvidaju cijenu:
#kombinacija1:TV, CableTV,Freeparkingonpremises
#kombinacija2:TV, CableTV,Familykidfriendly
#kombinacija3:TV, CableTV,Fireextinguisher
#kombinacija4:TV, Freeparkingonpremises, Familykidfriendly
#kombinacija5:TV, Freeparkingonpremises, Fireextinguisher
#kombinacija6:TV, Familykidfriendly, Fireextinguisher
```

```
#kombinacija7:CableTV,Freeparkingonpremises, Familykidfriendly
#kombinacija8:CableTV,Freeparkingonpremises, Fireextinguisher
#kombinacija9:CableTV,Familykidfriendly, Fireextinguisher
#kombinacija10: Freeparkingonpremises, Familykidfriendly, Fireextinguisher

tv_vector=airbnb_bitno["TV"]
cabletv_vector=airbnb_bitno["CableTV"]
freeparkingonpremises_vector=airbnb_bitno["Freeparkingonpremises"]
familykidfriendly_vector=airbnb_bitno["Familykidfriendly"]
fireextinguisher_vector=airbnb_bitno["Fireextinguisher"]

comb1=c(tv_vector&cabletv_vector&freeparkingonpremises_vector)
comb2=c(tv_vector&cabletv_vector&familykidfriendly_vector)
comb3=c(tv_vector&cabletv_vector&fireextinguisher_vector)

comb4=c(tv_vector&freeparkingonpremises_vector&familykidfriendly_vector)
comb5=c(tv_vector&freeparkingonpremises_vector&fireextinguisher_vector)

comb6=c(tv_vector&familykidfriendly_vector&fireextinguisher_vector)

comb7=c(cabletv_vector&freeparkingonpremises_vector&familykidfriendly_vector)
comb8=c(cabletv_vector&freeparkingonpremises_vector&fireextinguisher_vector)
comb9=c(cabletv_vector&familykidfriendly_vector&fireextinguisher_vector)

comb10=c(freeparkingonpremises_vector&familykidfriendly_vector&fireextinguisher_vector)

airbnb_bitno <-  cbind(airbnb_bitno , comb1, comb2, comb3, comb4, comb5,
                       comb6, comb7, comb8, comb9, comb10)

list_of_combinations=list(comb1, comb2, comb3, comb4, comb5,
                          comb6, comb7, comb8, comb9, comb10)
list_of_combinations_tests=list()

for(i in 1:10) {
  test=t.test(airbnb_bitno$price~list_of_combinations[[i]], mu=0, alt="two.sided", conf=0.99)
  print(paste("comb",i, test$conf.int[1], test$conf.int[2], "p-value:",test$p.value ))
}
```

```
## [1] "comb 1 -73.5857519457325 -60.64939203991 p-value: 3.53671415775073e-152"
## [1] "comb 2 -94.4398999507401 -84.5925313939519 p-value: 0"
## [1] "comb 3 -66.6324258782835 -56.2782805915725 p-value: 5.17406591765725e-199"
## [1] "comb 4 -79.5155669601984 -68.4067384660601 p-value: 3.85088183886776e-248"
## [1] "comb 5 -58.7793161462146 -47.0641724947939 p-value: 3.15188692827505e-117"
## [1] "comb 6 -82.5500433130342 -72.7644322068185 p-value: 0"
## [1] "comb 7 -101.059367616042 -84.9641981406729 p-value: 2.16520470263699e-183"
## [1] "comb 8 -81.3691967716243 -64.6780031139225 p-value: 4.99004306303371e-108"
## [1] "comb 9 -95.6496660051461 -82.2084353494337 p-value: 3.32601447147636e-239"
## [1] "comb 10 -76.0820013337378 -62.4131291415186 p-value: 7.45437086010524e-145"
```

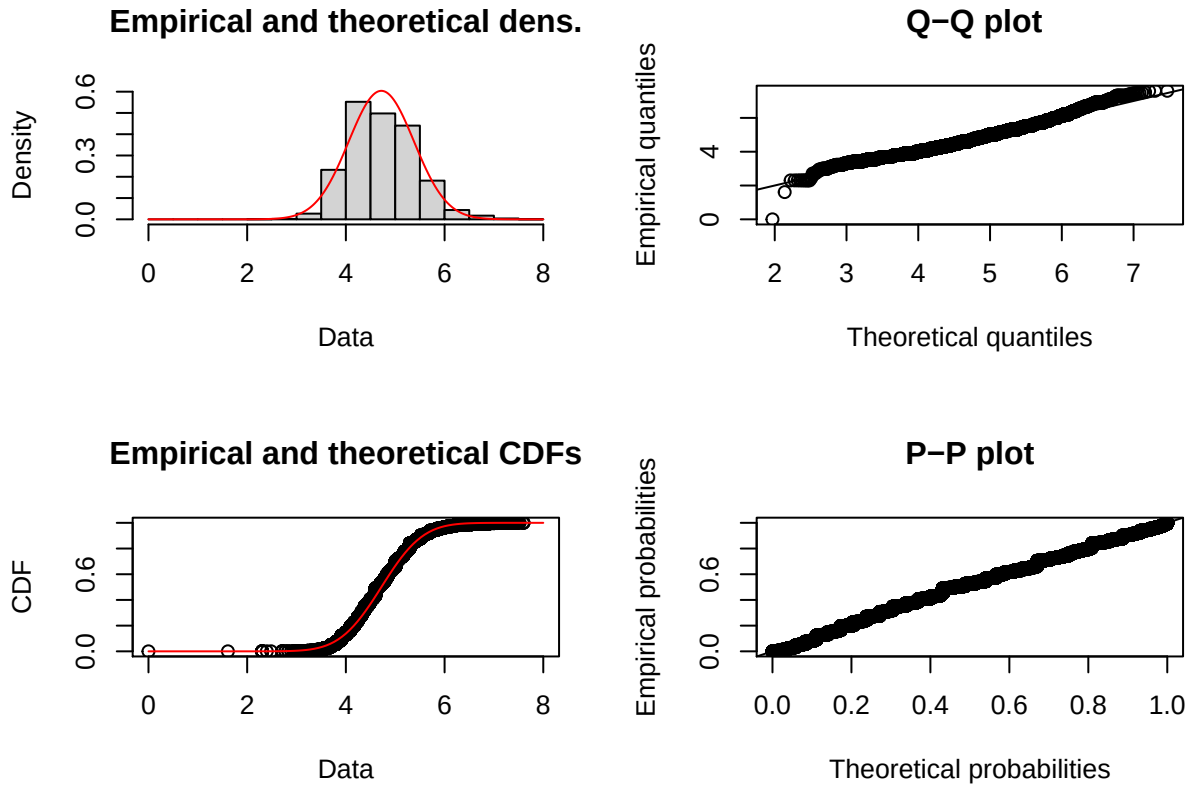
```
#odabiremo kombinaciju 2 i 6 jer imaju najmanju p vrijednost
#medutim, kako je u kombinaciji 2 Tv i CableTV koji su izuzetno korelirani
#kombinacija koja nabolje predvida cijenu po nama je
#kombinacija6:TV, Familykidfriendly, Fireextinguisher
```

```r
#Mozemo li pretpostaviti nekakvu distribuciju nad cijenama objekata za pojedine gradove?
#mozemo, noramlnu distrubuciju nad logaritmiranim cijenama, evo i grafova po gradovima
library(fitdistrplus)
```

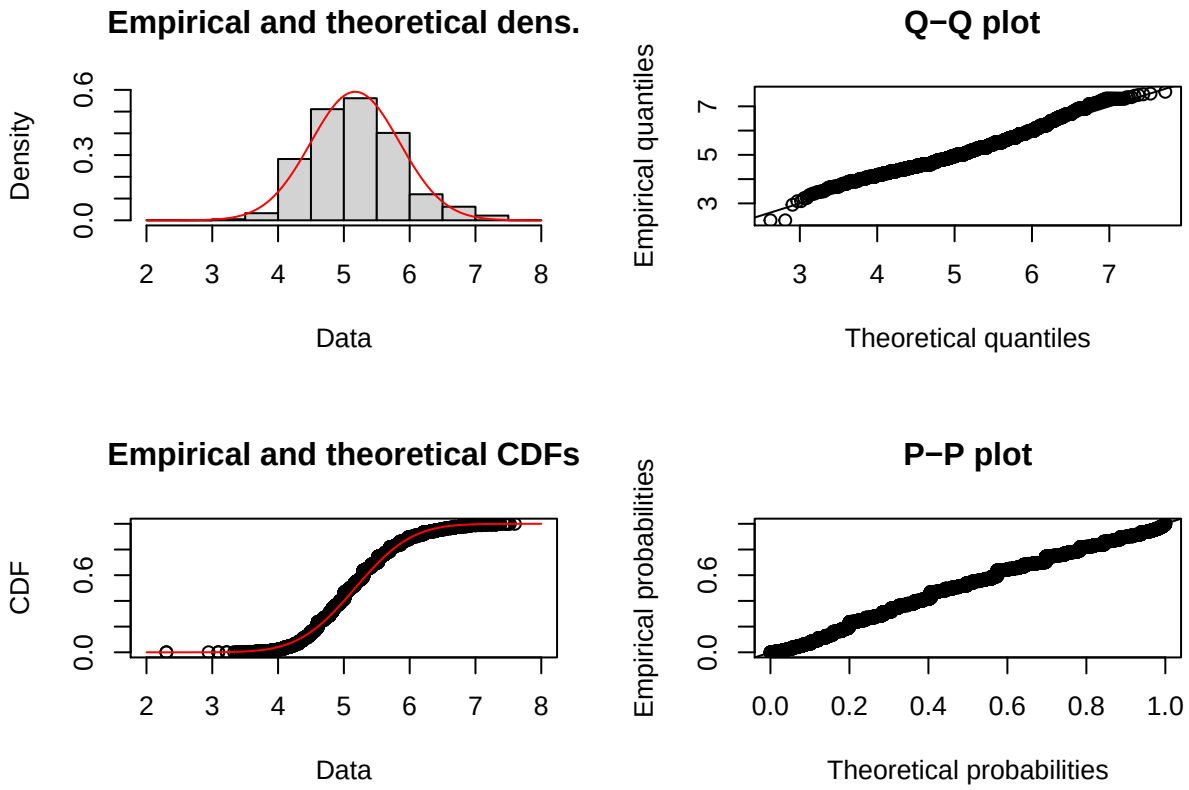
```
## Loading required package: MASS
```

```
## Loading required package: survival
```

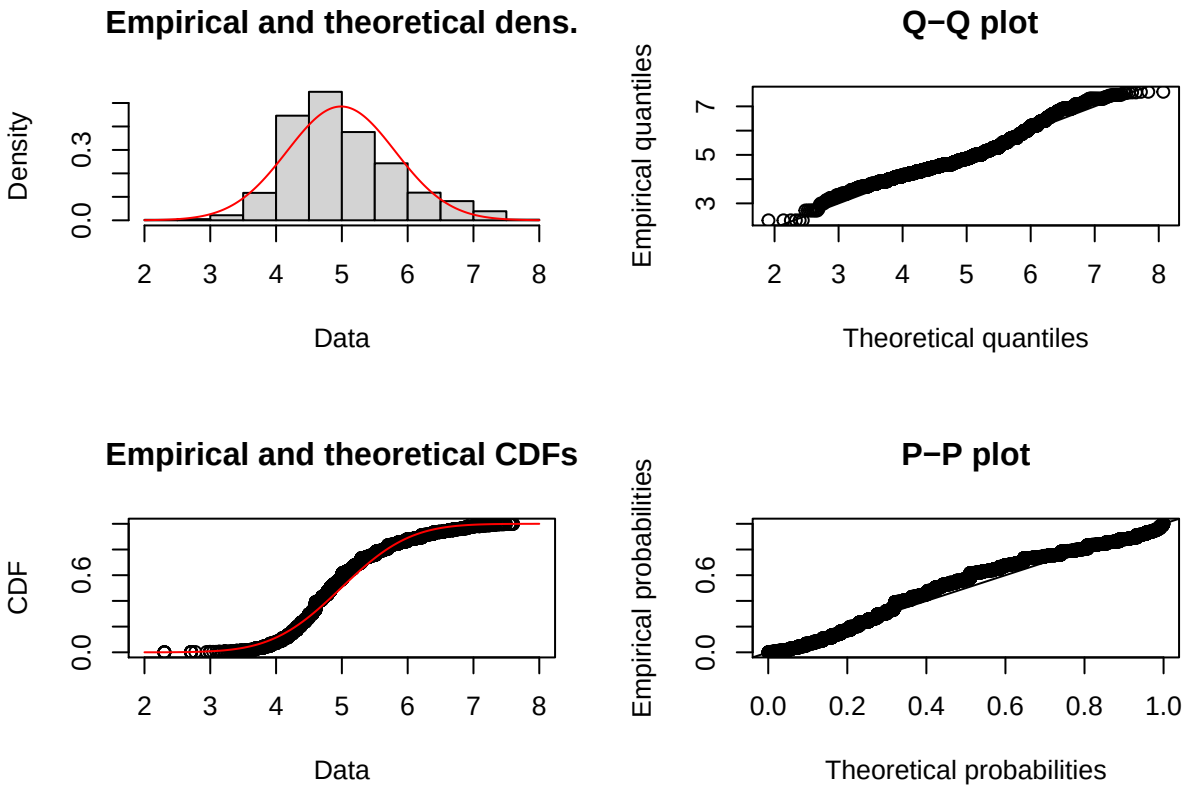
```r
#NYC
normal_dist <- fitdist(log(List_of_prices_per_city[[1]]), "norm")
plot(normal_dist)
```



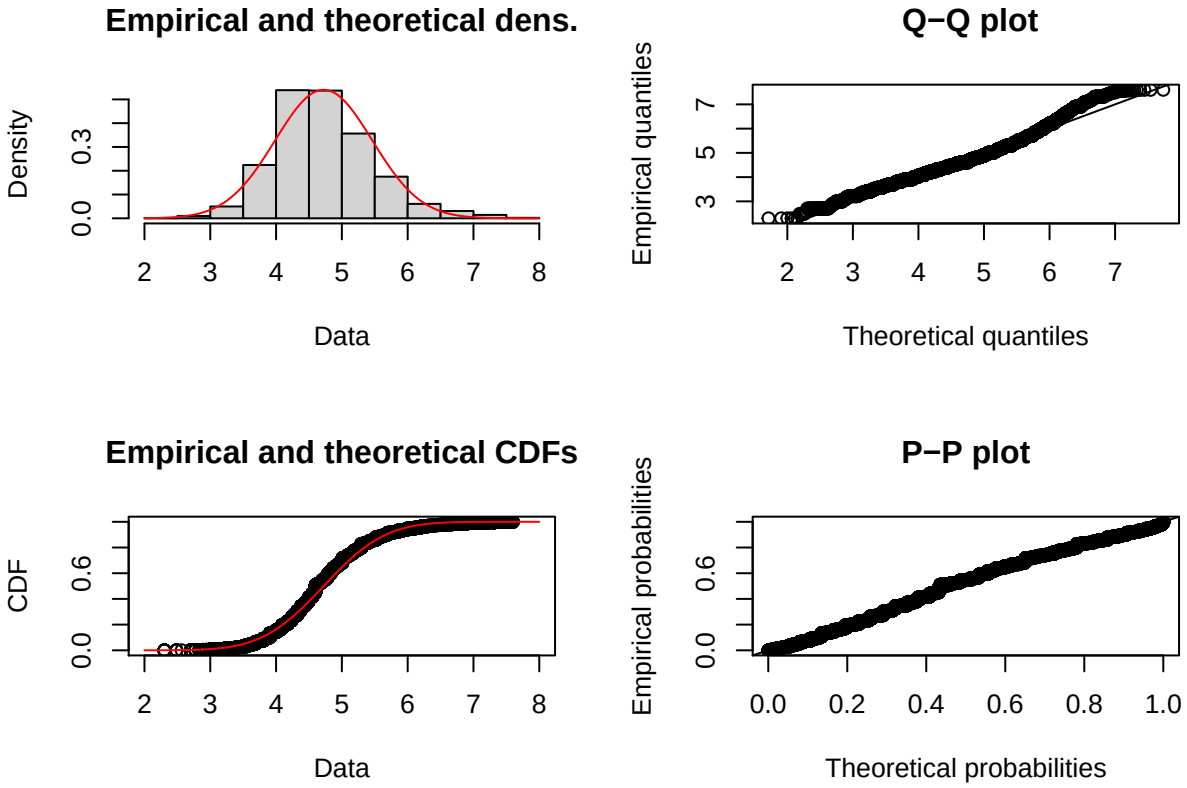

```r
#SF
normal_dist<- fitdist(log(List_of_prices_per_city[[2]]), "norm")
plot(normal_dist)
```



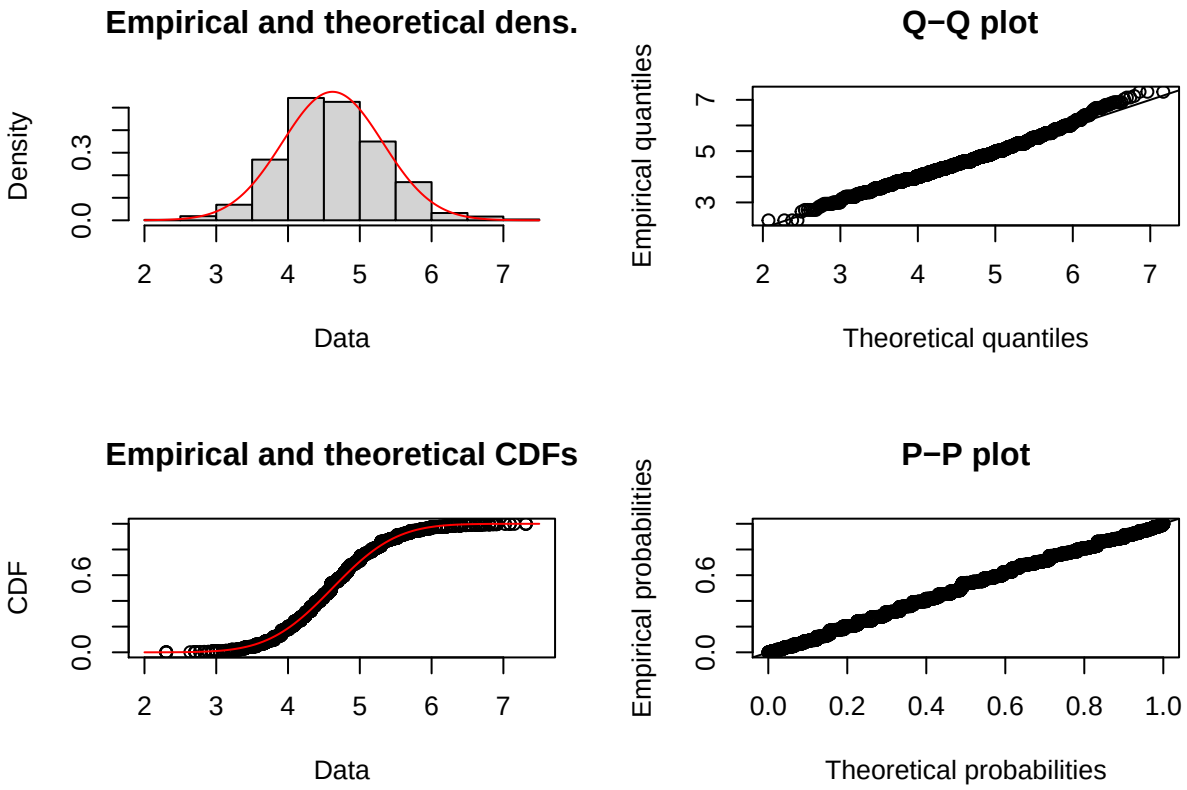

```
#DC
normal_dist <- fitdist(log(List_of_prices_per_city[[3]]), "norm")
plot(normal_dist)
```



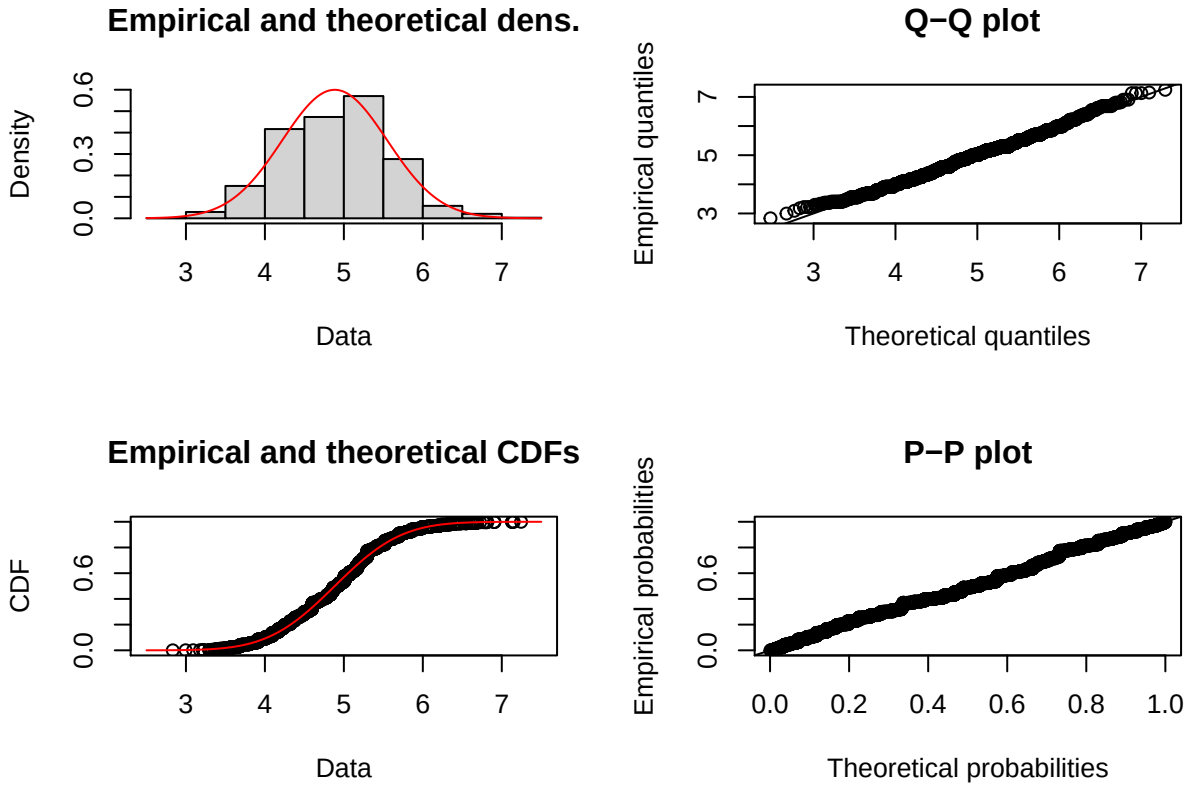

```
#LA
normal_dist <- fitdist(log(List_of_prices_per_city[[4]]), "norm")
plot(normal_dist)
```




```
#Chicago
normal_dist <- fitdist(log(List_of_prices_per_city[[5]]), "norm")
plot(normal_dist)
```




```
#Boston
normal_dist <- fitdist(log(List_of_prices_per_city[[6]]), "norm")
plot(normal_dist)
```




```
#Ima li neki grad statisticki znacajno vece cijene ugostiteljskih objekata nego neki drugi grad?


#uporedba cijena apartmana po svakom gradu sa meanom cijena svih apartmana u dataframeu
mean_of_all_cities=mean(airbnb_bitno$price)
print(paste("mean cijena apartmana u svim stanovima je", mean_of_all_cities))
```

```
## [1] "mean cijena apartmana u svim stanovima je 160.386721844355"
```

```
city_names=unique(airbnb_bitno$city)



for (i in 1:6) {
  city_prices_tests=list()
  test=t.test(List_of_prices_per_city[[i]], mu=mean_of_all_cities, alt="two.sided", conf=0.99)
  test$data.name=paste(city_names[i], "mean price")
  print(test)
}
```

```
##
##  One Sample t-test
##
## data:  NYC mean price
## t = -24.369, df = 31990, p-value < 2.2e-16
```

```
## alternative hypothesis: true mean is not equal to 160.3867
## 99 percent confidence interval:
##  140.8967 144.6233
## sample estimates:
## mean of x
##    142.76
##
##
##  One Sample t-test
##
## data:  SF mean price
## t = 26.132, df = 6361, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 160.3867
## 99 percent confidence interval:
##  220.6494 233.8329
## sample estimates:
## mean of x
##  227.2411
##
##
##  One Sample t-test
##
## data:  DC mean price
## t = 16.876, df = 5644, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 160.3867
## 99 percent confidence interval:
##  209.4978 227.1973
## sample estimates:
## mean of x
##  218.3476
##
##
##  One Sample t-test
##
## data:  LA mean price
## t = -3.8639, df = 22071, p-value = 0.0001119
## alternative hypothesis: true mean is not equal to 160.3867
## 99 percent confidence interval:
##  152.5778 158.8251
## sample estimates:
## mean of x
##  155.7015
##
##
##  One Sample t-test
##
## data:  Chicago mean price
## t = -13.658, df = 3704, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 160.3867
## 99 percent confidence interval:
##  127.1133 137.6770
## sample estimates:
## mean of x
##  132.3951
```

```
##
##
##  One Sample t-test
##
## data:  Boston mean price
## t = 2.4413, df = 3442, p-value = 0.01468
## alternative hypothesis: true mean is not equal to 160.3867
## 99 percent confidence interval:
##  160.0876 171.4332
## sample estimates:
## mean of x
##  165.7604
```

```
print("iz navedenih cijena vidimo da San Francisco ima znacajno veće cijene od prosjeka")
```

```
## [1] "iz navedenih cijena vidimo da San Francisco ima znacajno veće cijene od prosjeka"
```

```
#na interentu smo pronasli podatak da San Francisco je drugi najgusce naseljeni grad u SAD-u
#sto znacajno utjece na cijenu apartmana, kao i podatak da je izuzetno turisticki posjecen
#sto takoder dize cijenu apartmana
```

```
#Utjece li znacajno naknada za ciscenje ugostiteljskog objekta na prosjeˇcnu cijenu objekta?
cleaning_fee_test=t.test(airbnb_bitno$price~airbnb_bitno$cleaning_fee, mu=0, alt="two.sided", conf=0.99)
print(cleaning_fee_test)
```

```
##
##  Welch Two Sample t-test
##
## data:  airbnb_bitno$price by airbnb_bitno$cleaning_fee
## t = -7.2775, df = 28682, p-value = 3.489e-13
## alternative hypothesis: true difference in means is not equal to 0
## 99 percent confidence interval:
##  -15.326967  -7.313089
## sample estimates:
## mean in group False  mean in group True
##            152.0291            163.3491
```

```
#utjece, ali ne u tolikoj mjeri kao neke druge "vaznije varijable" navedene gore
#takoder iako je p-vrijendost mala, nije mala kao u vec prije navedenim varijablama
```

```
#Koje biste mjere preporucili iznajmljivacima kuca u NYC kako bi potencijalno mogli
#povecati cijenu nocenja (pod pretpostavkom da svi iznajmljivaci imaju jednaku popunjenost)?
#ocekujemo isti zakljucak kao i kada smo odabrali 5 varijabli
#za koje smo vidjeli veoma malu p-vrijednost
#a to su: TV, CableTV, Freeparkingonpremises, Familykidfriendly, Fireextinguisher
#kod je isti kao i za sve stanove
```

```r
#samo trazimo da test ima bar 5000 stupnjeva slobode umjesto dosadasnjih 10000

vector_of_positions=get_positions_of_nyc_houses(airbnb_bitno$property_type, airbnb_bitno$id)
nyc=airbnb_bitno[vector_of_positions,]
nyc <- subset(nyc, select = -c(id))


nyc=get_Dataframe_With_Id_Row(nyc)

#iz nekog razloga petlja puca za i= 116, 118, 120 i 121 pa smo skratili petlju do 115
#u vektoru number_of_apartments_that__have_certain_amenitie_vector[115:129]
#se vidi da ti amenitisi gotovo pa uopce nisu bitni
list_of_nyc_tests=list()
position_of_first_amenitie=15
for (i in 1:115) {


  test=t.test(nyc$price~nyc[,position_of_first_amenitie+i], mu=0, alt="two.sided", conf=0.99)
  test$data.name=paste("nyc price and ", vector_of_unique_amenities[i])
  list_of_nyc_tests[[i]]=test

}
clean_list_of_nyc_tests=list()
number_of_tests=length(list_of_nyc_tests)
counter=0
for (i in 1:number_of_tests) {
  if(list_of_nyc_tests[[i]]$data.name!="nyc price and  translationmissingenhostingamenity50" &&
     list_of_nyc_tests[[i]]$data.name!="nyc price and  translationmissingenhostingamenity49") {
    counter=counter+1
    clean_list_of_nyc_tests[[counter]]=list_of_nyc_tests[[i]]
  }
}


list_of_strings_correlation_nyc=list()
list_of_tests_enough_degrees_of_freedom_nyc=list()
number_of_valid_tests_nyc=length(clean_list_of_nyc_tests)
counter=0
for (i in 1:number_of_valid_tests_nyc) {
  if(clean_list_of_nyc_tests[[i]]$parameter>=5000) {
    temp_string=paste(clean_list_of_nyc_tests[[i]]$data.name, clean_list_of_nyc_tests[[i]]$conf.int[1],
    counter=counter+1
    list_of_tests_enough_degrees_of_freedom_nyc[[counter]]=clean_list_of_nyc_tests[[i]]
    list_of_strings_correlation_nyc[[counter]]=temp_string
    print(temp_string)
    print("\n")
  }
}
```

```
## [1] "nyc price and  Airconditioning -26.5783790281828 -6.63073891384078 : p-value 2.34240317858064e-
## [1] "\n"
## [1] "nyc price and  Kitchen -107.199314144537 -37.3192072675765 : p-value 1.89777975836664e-193"
## [1] "\n"
```

```
## [1] "nyc price and  Familykidfriendly -125.727630005561 -67.6782811842195 : p-value 7.55451649881953
## [1] "\n"
## [1] "nyc price and  Hairdryer -35.4352715910092 -18.3594779955358 : p-value 9.11770631474748e-12"
## [1] "\n"
## [1] "nyc price and  Iron -44.2550342557583 -20.3407430851092 : p-value 3.30185535208452e-21"
## [1] "\n"
## [1] "nyc price and  Washer -114.601209923993 -49.1620938301594 : p-value 9.03193333156248e-258"
## [1] "\n"
## [1] "nyc price and  Dryer -114.535388846736 -49.9488287751579 : p-value 2.34003507517581e-252"
## [1] "\n"
## [1] "nyc price and  Fireextinguisher -33.5220480268326 -21.059820700186 : p-value 4.84231999880826e-
## [1] "\n"
## [1] "nyc price and  Shampoo -19.1343366840287 -11.6556690614951 : p-value 0.0373133076457561"
## [1] "\n"
## [1] "nyc price and  Hangers -13.830018380501 -1.84715340653733 : p-value 0.428556142291734"
## [1] "\n"
## [1] "nyc price and  TV -114.355849776002 -63.2910948146106 : p-value 3.17872222937546e-230"
## [1] "\n"
## [1] "nyc price and  CableTV -102.224960701425 -58.4176533407661 : p-value 9.33319632320794e-119"
## [1] "\n"
## [1] "nyc price and  Carbonmonoxidedetector -27.3685475631264 -10.6013854156219 : p-value 1.361484935
## [1] "\n"
## [1] "nyc price and  Laptopfriendlyworkspace -43.6462049687999 -16.4419416990809 : p-value 1.029731758
## [1] "\n"
## [1] "nyc price and  Internet -46.6532493634283 -14.0670173770801 : p-value 1.20970011214251e-22"
## [1] "\n"
## [1] "nyc price and  Indoorfireplace -145.555467941945 -87.4047733983284 : p-value 7.87854515821582e-
## [1] "\n"
## [1] "nyc price and  Firstaidkit -7.03530703383842 -0.0944205949140079 : p-value 0.517313633485835"
## [1] "\n"
## [1] "nyc price and  Freeparkingonpremises -76.2823777754751 -23.5702265616192 : p-value 8.4658512409
## [1] "\n"
## [1] "nyc price and  Lockonbedroomdoor 48.8058365397684 32.2280592965084 : p-value 1.51093282848303e-
## [1] "\n"
## [1] "nyc price and  Petsliveonthisproperty 52.5753570099268 40.002294577991 : p-value 3.348748633761
## [1] "\n"
## [1] "nyc price and  24hourcheckin -41.5979315192266 -19.100352270327 : p-value 1.49684861834662e-15"
## [1] "\n"
```

```
#zakljucak je da: TV, CableTV, Freeparkingonpremises, Familykidfriendly, Fireextinguisher
#su stvari koje ce iznajmljivacima kuca u nyc povecati najam
```

‘‘‘