Code_EDA

Madhura Jadhav

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   2.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'stringr' was built under R version 4.1.2
```

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

```
library(ggplot2)

library(skimr)
```

```
## Warning: package 'skimr' was built under R version 4.1.3
```

```
library(summarytools)
```

```
## Warning: package 'summarytools' was built under R version 4.1.3
```

```
##
## Attaching package: 'summarytools'
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(stringr)
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

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

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:summarytools':
##
##     label, label<-
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(broom)
library(car)
```

```
## Warning: package 'car' was built under R version 4.1.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.1.3
```

```
##
## Attaching package: 'car'
```

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

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

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

```r
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.3
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.3
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.1.3
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(vtreat)
```

```
## Warning: package 'vtreat' was built under R version 4.1.3
```

```
## Loading required package: wrapr
```

```
## Warning: package 'wrapr' was built under R version 4.1.3
```

```
##
## Attaching package: 'wrapr'
```

```
## The following object is masked from 'package:car':
##
##     bc
```

```
## The following object is masked from 'package:summarytools':
##
##     view
```

```
## The following objects are masked from 'package:tidyr':
##
##     pack, unpack
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```
## The following object is masked from 'package:dplyr':
##
##     coalesce
```

```
original_data = read.csv('analysisData.csv')
dim(original_data)
```

```
## [1] 34404    91
```

Filtering the data to remove out of scope / redundant attributes

```
data=original_data[c('id','host_name', 'host_since', 'host_response_time','host_response_rate', 'host_i
s_superhost', 'host_total_listings_count', 'neighbourhood_cleansed', 'neighbourhood_group_cleansed', 'z
ipcode', 'property_type', 'room_type', 'accommodates', 'bathrooms', 'bedrooms', 'beds', 'amenities', 'p
rice','cleaning_fee','guests_included','extra_people', 'minimum_nights','maximum_nights','availability_
30','availability_60','availability_90','availability_365', 'number_of_reviews','number_of_reviews_ltm'
,'first_review','last_review','review_scores_rating', 'review_scores_accuracy', 'review_scores_cleanlin
ess', 'review_scores_checkin', 'review_scores_communication', 'review_scores_location', 'review_scores_
value', 'instant_bookable','calculated_host_listings_count','reviews_per_month')]
dim(data)
```

```
## [1] 34404    41
```

```
#library(summarytools)
print(dfSummary(data,style='grid',graph.col = T),method = 'render')
```

Data Frame Summary

data

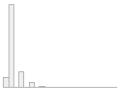
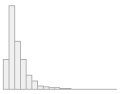
Dimensions: 34404 x 41
Duplicates: 0

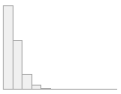
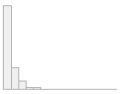
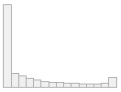
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing

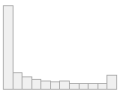
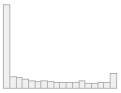
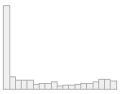
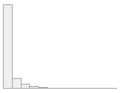
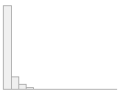
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	id [integer]	Mean (sd) : 445480 (210539.2) min ≤ med ≤ max: 100015 ≤ 437248 ≤ 898796 IQR (CV) : 338110.8 (0.5)	34404 distinct values		34404 (100.0%)	0 (0.0%)
2	host_name [character]	1. Michael 2. David 3. John 4. Alex 5. Sonder (NYC) 6. Sarah 7. Maria 8. Daniel 9. Anna 10. Mike [9246 others]	310 (0.9%) 280 (0.8%) 229 (0.7%) 195 (0.6%) 185 (0.5%) 166 (0.5%) 158 (0.5%) 156 (0.5%) 141 (0.4%) 139 (0.4%) 32445 (94.3%)		34404 (100.0%)	0 (0.0%)
3	host_since [character]	1. 2018-10-08 2. 2017-06-27 3. 2016-03-03 4. 2015-12-16 5. 2014-05-28 6. 2017-03-07 7. 2014-10-14 8. 2018-05-22 9. 2013-07-15 10. 2017-03-14 [3679 others]	201 (0.6%) 81 (0.2%) 74 (0.2%) 69 (0.2%) 64 (0.2%) 49 (0.1%) 48 (0.1%) 48 (0.1%) 47 (0.1%) 46 (0.1%) 33677 (97.9%)		34404 (100.0%)	0 (0.0%)
4	host_response_time [character]	1. (Empty string) 2. a few days or more 3. N/A 4. within a day 5. within a few hours 6. within an hour	21 (0.1%) 581 (1.7%) 8680 (25.2%) 3457 (10.0%) 5530 (16.1%) 16135 (46.9%)		34404 (100.0%)	0 (0.0%)

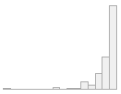
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
5	host_response_rate [character]	1. 100% 2. N/A 3. 90% 4. 80% 5. 98% 6. 50% 7. 97% 8. 94% 9. 92% 10. 93% [74 others]	18090 (52.6%) 8680 (25.2%) 1203 (3.5%) 598 (1.7%) 507 (1.5%) 322 (0.9%) 313 (0.9%) 280 (0.8%) 278 (0.8%) 271 (0.8%) 3862 (11.2%)		34404 (100.0%)	0 (0.0%)
6	host_is_superhost [character]	1. (Empty string) 2. f 3. t	21 (0.1%) 26654 (77.5%) 7729 (22.5%)		34404 (100.0%)	0 (0.0%)
7	host_total_listings_count [integer]	Mean (sd) : 7.4 (50.5) min ≤ med ≤ max: 0 ≤ 1 ≤ 1080 IQR (CV) : 1 (6.8)	75 distinct values		34383 (99.9%)	21 (0.1%)
8	neighbourhood_cleansed [character]	1. Williamsburg 2. Bedford-Stuyvesant 3. Harlem 4. Bushwick 5. Hell's Kitchen 6. East Village 7. Upper West Side 8. Upper East Side 9. Crown Heights 10. East Harlem [209 others]	2782 (8.1%) 2778 (8.1%) 1925 (5.6%) 1706 (5.0%) 1431 (4.2%) 1290 (3.7%) 1269 (3.7%) 1198 (3.5%) 1146 (3.3%) 837 (2.4%) 18042 (52.4%)		34404 (100.0%)	0 (0.0%)
9	neighbourhood_group_cleansed [character]	1. Bronx 2. Brooklyn 3. Manhattan 4. Queens 5. Staten Island	837 (2.4%) 14428 (41.9%) 14532 (42.2%) 4309 (12.5%) 298 (0.9%)		34404 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
10	zipcode [character]	1. 11211 2. 11221 3. 11206 4. 11216 5. 10002 6. 10019 7. 10009 8. 11238 9. 11222 10. 11233 [191 others]	1556 (4.5%) 1348 (3.9%) 1073 (3.1%) 1023 (3.0%) 912 (2.7%) 852 (2.5%) 815 (2.4%) 796 (2.3%) 754 (2.2%) 728 (2.1%) 24547 (71.3%)		34404 (100.0%)	0 (0.0%)
11	property_type [character]	1. Apartment 2. House 3. Townhouse 4. Condominium 5. Loft 6. Guest suite 7. Serviced apartment 8. Boutique hotel 9. Hotel 10. Other [21 others]	26942 (78.3%) 2935 (8.5%) 1250 (3.6%) 1075 (3.1%) 1048 (3.0%) 311 (0.9%) 267 (0.8%) 112 (0.3%) 94 (0.3%) 60 (0.2%) 310 (0.9%)		34404 (100.0%)	0 (0.0%)
12	room_type [character]	1. Entire home/apt 2. Hotel room 3. Private room 4. Shared room	17859 (51.9%) 3 (0.0%) 15731 (45.7%) 811 (2.4%)		34404 (100.0%)	0 (0.0%)
13	accommodates [integer]	Mean (sd) : 2.9 (1.9) min ≤ med ≤ max: 1 ≤ 2 ≤ 16 IQR (CV) : 2 (0.6)	16 distinct values		34404 (100.0%)	0 (0.0%)
14	bathrooms [numeric]	Mean (sd) : 1.1 (0.4) min ≤ med ≤ max: 0 ≤ 1 ≤ 7 IQR (CV) : 0 (0.4)	15 distinct values		34404 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
15	bedrooms [integer]	Mean (sd) : 1.2 (0.7) min ≤ med ≤ max: 0 ≤ 1 ≤ 11 IQR (CV) : 0 (0.6)	12 distinct values		34404 (100.0%)	0 (0.0%)
16	beds [integer]	Mean (sd) : 1.6 (1.1) min ≤ med ≤ max: 0 ≤ 1 ≤ 26 IQR (CV) : 1 (0.7)	20 distinct values		34360 (99.9%)	44 (0.1%)
17	amenities [character]	1. TV, Cable TV ,Wifi, Air c 2. . 3. TV, Cable TV ,Wifi, Air c 4. TV, Cable TV ,Wifi, Air c 5. TV,Wifi, Air conditioning 6. TV,Wifi, Air conditioning 7. TV, Cable TV ,Wifi, Air c 8. TV,Wifi, Air conditioning 9. Wifi, Air conditioning ,K 10. TV, Cable TV ,Wifi, Air c [32280 others]	129 (0.4%) 36 (0.1%) 23 (0.1%) 18 (0.1%) 18 (0.1%) 18 (0.1%) 16 (0.0%) 14 (0.0%) 12 (0.0%) 10 (0.0%) 34110 (99.1%)		34404 (100.0%)	0 (0.0%)
18	price [integer]	Mean (sd) : 135.1 (106) min ≤ med ≤ max: 0 ≤ 100 ≤ 999 IQR (CV) : 102 (0.8)	517 distinct values		34404 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
19	cleaning_fee [integer]	Mean (sd) : 62.2 (50.4) min ≤ med ≤ max: 0 ≤ 50 ≤ 600 IQR (CV) : 65 (0.8)	186 distinct values		29128 (84.7%)	5276 (15.3%)
20	guests_included [integer]	Mean (sd) : 1.6 (1.2) min ≤ med ≤ max: 1 ≤ 1 ≤ 16 IQR (CV) : 1 (0.8)	15 distinct values		34404 (100.0%)	0 (0.0%)
21	extra_people [integer]	Mean (sd) : 16.1 (24.7) min ≤ med ≤ max: 0 ≤ 10 ≤ 300 IQR (CV) : 25 (1.5)	104 distinct values		34404 (100.0%)	0 (0.0%)
22	minimum_nights [integer]	Mean (sd) : 5.8 (17.7) min ≤ med ≤ max: 1 ≤ 2 ≤ 1250 IQR (CV) : 3 (3)	88 distinct values		34404 (100.0%)	0 (0.0%)
23	maximum_nights [integer]	Mean (sd) : 63620.1 (11578280) min ≤ med ≤ max: 1 ≤ 365 ≤ 2147483647 IQR (CV) : 1096 (182)	248 distinct values		34404 (100.0%)	0 (0.0%)
24	availability_30 [integer]	Mean (sd) : 7.3 (9.4) min ≤ med ≤ max: 0 ≤ 3 ≤ 30 IQR (CV) : 12 (1.3)	31 distinct values		34404 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
25	availability_60 [integer]	Mean (sd) : 16.6 (19.7) min ≤ med ≤ max: 0 ≤ 8 ≤ 60 IQR (CV) : 30 (1.2)	61 distinct values		34404 (100.0%)	0 (0.0%)
26	availability_90 [integer]	Mean (sd) : 28.2 (30.9) min ≤ med ≤ max: 0 ≤ 15 ≤ 90 IQR (CV) : 54 (1.1)	91 distinct values		34404 (100.0%)	0 (0.0%)
27	availability_365 [integer]	Mean (sd) : 119 (130.5) min ≤ med ≤ max: 0 ≤ 62 ≤ 365 IQR (CV) : 239 (1.1)	366 distinct values		34404 (100.0%)	0 (0.0%)
28	number_of_reviews [integer]	Mean (sd) : 28 (47.3) min ≤ med ≤ max: 0 ≤ 9 ≤ 604 IQR (CV) : 28 (1.7)	376 distinct values		34404 (100.0%)	0 (0.0%)
29	number_of_reviews_ltm [integer]	Mean (sd) : 10.9 (16.7) min ≤ med ≤ max: 0 ≤ 3 ≤ 283 IQR (CV) : 13.2 (1.5)	144 distinct values		34404 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
30	first_review [character]	1. 2019-01-01 2. 2018-01-01 3. 2016-01-02 4. 2019-06-30 5. 2018-01-02 6. 2019-01-02 7. 2019-07-01 8. 2019-04-21 9. 2017-01-02 10. 2017-01-01 [2967 others]	150 (0.4%) 141 (0.4%) 116 (0.3%) 112 (0.3%) 104 (0.3%) 100 (0.3%) 95 (0.3%) 93 (0.3%) 92 (0.3%) 84 (0.2%) 33317 (96.8%)		34404 (100.0%)	0 (0.0%)
31	last_review [character]	1. 2019-07-21 2. 2019-07-28 3. 2019-08-04 4. 2019-07-22 5. 2019-07-31 6. 2019-06-30 7. 2019-07-29 8. 2019-08-03 9. 2019-07-30 10. 2019-07-14 [1849 others]	928 (2.7%) 822 (2.4%) 760 (2.2%) 633 (1.8%) 567 (1.6%) 521 (1.5%) 510 (1.5%) 465 (1.4%) 458 (1.3%) 452 (1.3%) 28288 (82.2%)		34404 (100.0%)	0 (0.0%)
32	review_scores_rating [integer]	Mean (sd) : 93.9 (9.1) min ≤ med ≤ max: 20 ≤ 97 ≤ 100 IQR (CV) : 8 (0.1)	55 distinct values		34404 (100.0%)	0 (0.0%)
33	review_scores_accuracy [integer]	Mean (sd) : 9.6 (0.9) min ≤ med ≤ max: 2 ≤ 10 ≤ 10 IQR (CV) : 1 (0.1)	2 : 133 (0.4%) 3 : 1 (0.0%) 4 : 79 (0.2%) 5 : 26 (0.1%) 6 : 321 (0.9%) 7 : 275 (0.8%) 8 : 1535 (4.5%) 9 : 6607 (19.2%) 10 : 25427 (73.9%)		34404 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
34	review_scores_cleanliness [integer]	Mean (sd) : 9.3 (1.1) min ≤ med ≤ max: 2 ≤ 10 ≤ 10 IQR (CV) : 1 (0.1)	2 : 168 (0.5%) 3 : 10 (0.0%) 4 : 166 (0.5%) 5 : 93 (0.3%) 6 : 621 (1.8%) 7 : 829 (2.4%) 8 : 3533 (10.3%) 9 : 10028 (29.1%) 10 : 18956 (55.1%)		34404 (100.0%)	0 (0.0%)
35	review_scores_checkin [integer]	Mean (sd) : 9.7 (0.8) min ≤ med ≤ max: 2 ≤ 10 ≤ 10 IQR (CV) : 0 (0.1)	2 : 102 (0.3%) 3 : 1 (0.0%) 4 : 50 (0.1%) 5 : 18 (0.1%) 6 : 227 (0.7%) 7 : 176 (0.5%) 8 : 929 (2.7%) 9 : 4449 (12.9%) 10 : 28452 (82.7%)		34404 (100.0%)	0 (0.0%)
36	review_scores_communication [integer]	Mean (sd) : 9.7 (0.8) min ≤ med ≤ max: 2 ≤ 10 ≤ 10 IQR (CV) : 0 (0.1)	2 : 112 (0.3%) 3 : 5 (0.0%) 4 : 49 (0.1%) 5 : 21 (0.1%) 6 : 214 (0.6%) 7 : 192 (0.6%) 8 : 952 (2.8%) 9 : 4047 (11.8%) 10 : 28812 (83.7%)		34404 (100.0%)	0 (0.0%)
37	review_scores_location [integer]	Mean (sd) : 9.6 (0.8) min ≤ med ≤ max: 2 ≤ 10 ≤ 10 IQR (CV) : 1 (0.1)	2 : 68 (0.2%) 3 : 2 (0.0%) 4 : 40 (0.1%) 5 : 10 (0.0%) 6 : 253 (0.7%) 7 : 206 (0.6%) 8 : 1822 (5.3%) 9 : 8719 (25.3%) 10 : 23284 (67.7%)		34404 (100.0%)	0 (0.0%)
38	review_scores_value [integer]	Mean (sd) : 9.4 (1) min ≤ med ≤ max: 2 ≤ 10 ≤ 10 IQR (CV) : 1 (0.1)	2 : 111 (0.3%) 3 : 6 (0.0%) 4 : 108 (0.3%) 5 : 55 (0.2%) 6 : 409 (1.2%) 7 : 373 (1.1%) 8 : 2474 (7.2%) 9 : 11229 (32.6%) 10 : 19639 (57.1%)		34404 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
39	instant_bookable [character]	1. f 2. t	21020 (61.1%) 13384 (38.9%)		34404 (100.0%)	0 (0.0%)
40	calculated_host_listings_count [integer]	Mean (sd) : 5.2 (27.5) min ≤ med ≤ max: 1 ≤ 1 ≤ 371 IQR (CV) : 1 (5.3)	52 distinct values		34404 (100.0%)	0 (0.0%)
41	reviews_per_month [numeric]	Mean (sd) : 1.4 (1.7) min ≤ med ≤ max: 0 ≤ 0.8 ≤ 66.6 IQR (CV) : 1.8 (1.2)	924 distinct values		34402 (100.0%)	2 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.0 (R (https://www.r-project.org/) version 4.1.1)
2022-05-02

Missing Values

Missing Values Columns: cleaning_fee 5276, beds 44, host_total_listings_count 21, reviews_per_month 2, host_response_time 8679, host_response_rate8679

Will drop rows with missing values <50 for a column

```
data = data %>% drop_na(reviews_per_month)
data = data %>% drop_na(host_total_listings_count)
data = data %>% drop_na(beds)
```

Cleaning fees is option so absence of it is cleaning fee=0

```
data = data %>% mutate(cleaning_fee = ifelse(is.na(data$cleaning_fee),0,data$cleaning_fee))
```

We will drop host_response_time 8679, host_response_rate 8679 as we have ~25% missing data

```
data = data[ , -which(names(data) %in% c("host_response_rate","host_response_time"))]
```

Data Transformations

Converting columns with logical data type to 1 and 0

host_is_superhost,instant_bookable

```
data$host_is_superhost = as.numeric(ifelse(data$host_is_superhost == 't', 1, 0))
data$instant_bookable = as.numeric(ifelse(data$instant_bookable == 't', 1, 0))
```

New Features

host_since can be used to calculate host active duration in months

first_review & last_review can be used to calculate listing active duration in months

we can drop column last_review,first_review, host_since after calculating new features

```
#max(data$last_review) is '2020-01-03'

data$listing_active_duration = round(as.numeric(difftime(data$last_review, data$first_review, units =
"days"))/(365.25/12),2)
data$host_active_duration = round(as.numeric(difftime('2020-01-03', data$host_since, units ="days"))/(3
65.25/12),2)

data = subset(data, select = -c(host_since,first_review,last_review) )
```

Amenities can be extracted to check important individual amenity

```
#library(stringr)
data$TV = str_extract(data$amenities, "TV")
data$TV[is.na(data$TV)] <- 0
data$TV[data$TV == "TV"] <- 1

data$Elevator = str_extract(data$amenities, "Elevator")
data$Elevator[is.na(data$Elevator)] <- 0
data$Elevator[data$Elevator == "Elevator"] <- 1
```

Combine variable levels

property_type

combine levels with <1% data to ‘Others’ making total number of levels for property_type as 6

```
data %>% group_by(property_type) %>% summarise(n=n()) %>% arrange(desc(n))
```

```
## # A tibble: 31 x 2
##    property_type          n
##    <chr>              <int>
##  1 Apartment          26894
##  2 House               2924
##  3 Townhouse           1249
##  4 Condominium         1073
##  5 Loft                1048
##  6 Guest suite          311
##  7 Serviced apartment   266
##  8 Boutique hotel       111
##  9 Hotel                 94
## 10 Other                 60
## # ... with 21 more rows
```

```
data=data %>% mutate(property_type = fct_lump(property_type, prop = 0.01))
```

room_type

updating ‘Hotel room’ level as ‘Private room’ as we have just 3 ‘Hotel room’ records

```
data$room_type = fct_collapse(data$room_type, 'Private room' = c('Hotel room', 'Private room'))
```

bathrooms

updating 0.5 bathroom levels and reducing number of levels

```
data$bathrooms[data$bathrooms < 1] <- 1
data$bathrooms[ data$bathrooms == 1.5] <- 2
data$bathrooms[ data$bathrooms == 2.5] <- 3
data$bathrooms[ data$bathrooms == 3.5] <- 4
data$bathrooms[ data$bathrooms > 4] <- 4
```

bedrooms

reducing number of levels from 12 to 7

```
data$bedrooms[data$bedrooms > 6] <- 6
```

Delete erroneous data

price

delete listings with price=0,assuming it is data error

```
data= data[!(data$price==0),]
```

zipcode

clean dirty zipcode data, treat NAs with mode value

```
data$zipcode=str_replace_all(data$zipcode,"NY ","")
data$zipcode=str_replace_all(data$zipcode," ","")
data$zipcode[data$zipcode=="11385-2308"] = '11385'
data$zipcode[data$zipcode=="11103-3233"] = '11103'
data$zipcode[data$zipcode=="10003-8623"] = '10003'
data$zipcode[data$zipcode=="11413-3220"] = '11413'
data$zipcode[data$zipcode=="10065"] = '10021'
data$zipcode[data$zipcode=="11249"] = '11211'
data$zipcode <- gsub('NA','11211',data$zipcode)
```

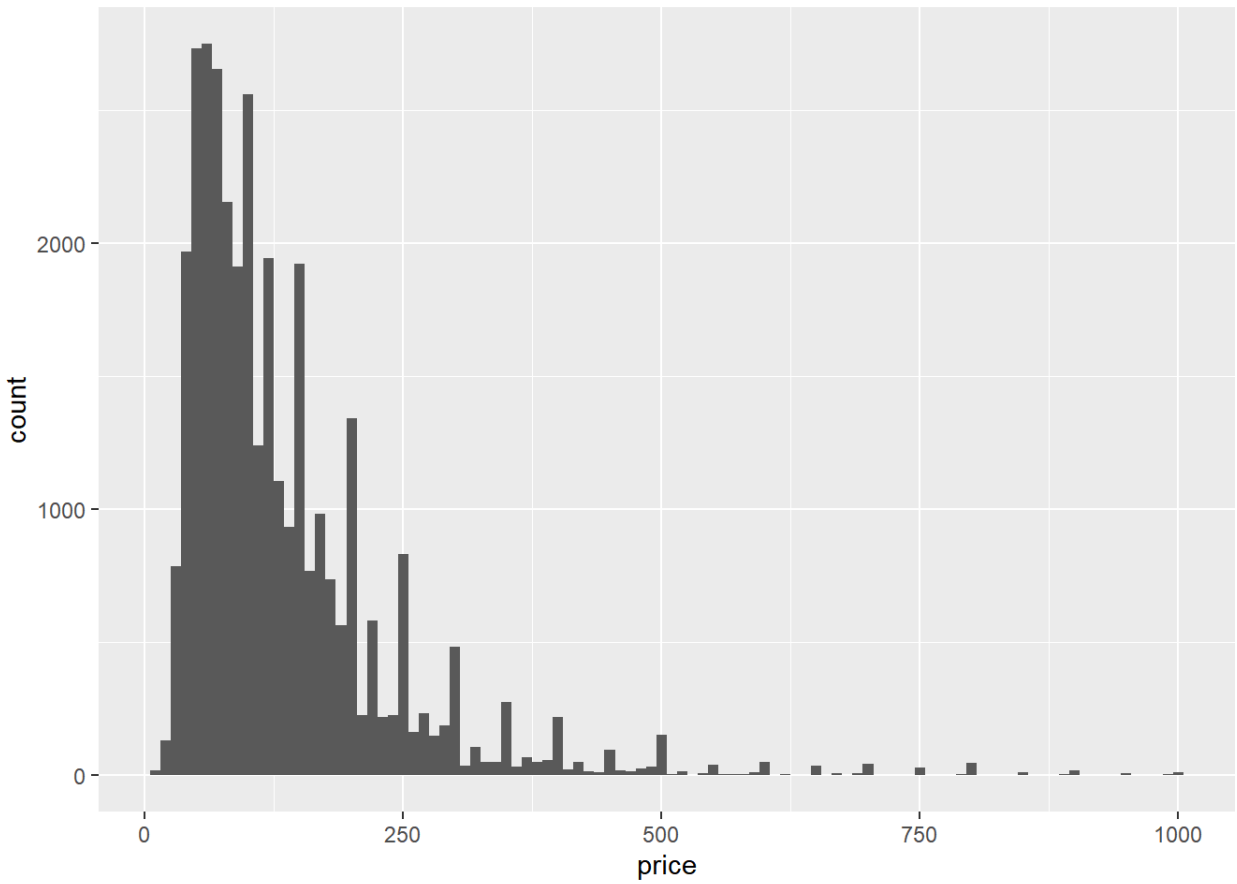
```
# odata %>%
#    filter(neighbourhood_group_cleansed=='Staten Island') %>%
#    count(zipcode, sort = TRUE)

#Brooklyn -11211
#Bronx - 10469
#Manhattan - 10002
#Queens- 11385
#SA - 10301
```

EDA

##Target Variable - price distribution

```
ggplot(data, aes(x=price)) +
  geom_histogram(binwidth = 10)
```



will be analyzing numeric variables across price with scatter plots #### will be analyzing categorical and few numeric variables across price with box plots

```
box = geom_boxplot(varwidth=T)
scatter = geom_point()
```

Analyzing Categorical Variable

categorical: neighbourhood_cleansed,neighbourhood_group_cleansed

neighbourhood_cleansed – explored neighbourhood_cleansed for each borough.. but Too many levels and no so DROP

```
# data %>%
#   filter(neighbourhood_group_cleansed == 'Brooklyn') %>%
#   ggplot(aes(x=neighbourhood_cleansed,y=price)) + box
```

```
# data %>%
#   filter(neighbourhood_group_cleansed == 'Bronx') %>%
#   ggplot(aes(x=neighbourhood_cleansed,y=price)) + box
```

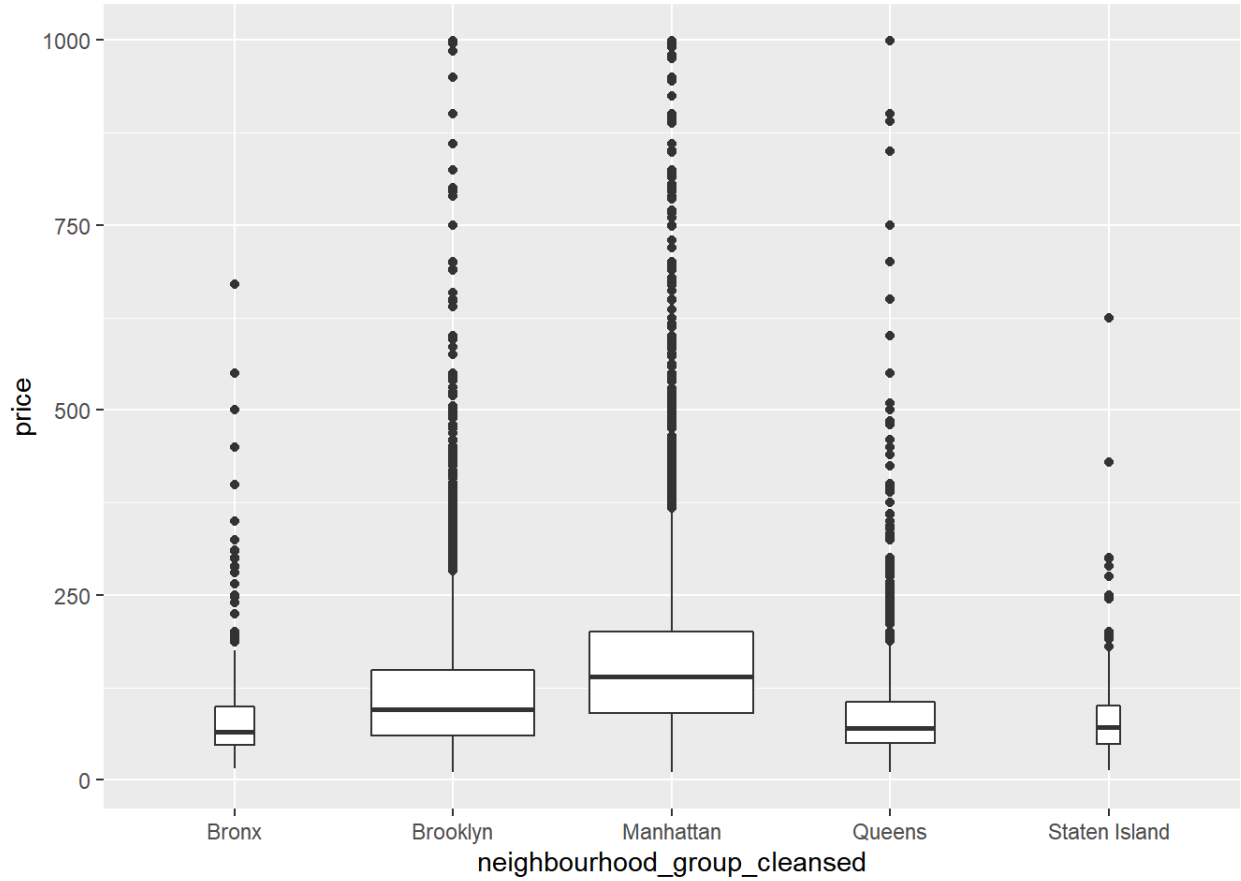
```
# data %>%
#   filter(neighbourhood_group_cleansed == 'Manhattan') %>%
#   ggplot(aes(x=neighbourhood_cleansed,y=price)) + box
```

```
# data %>%
#   filter(neighbourhood_group_cleansed == 'Queens') %>%
#   ggplot(aes(x=neighbourhood_cleansed,y=price)) + box
```

```
# data %>%
#   filter(neighbourhood_group_cleansed == 'Staten Island') %>%
#   ggplot(aes(x=neighbourhood_cleansed,y=price)) + box
```

neighbourhood_group_cleansed

```
ggplot(data, aes(x=neighbourhood_group_cleansed,y=price)) + box
```

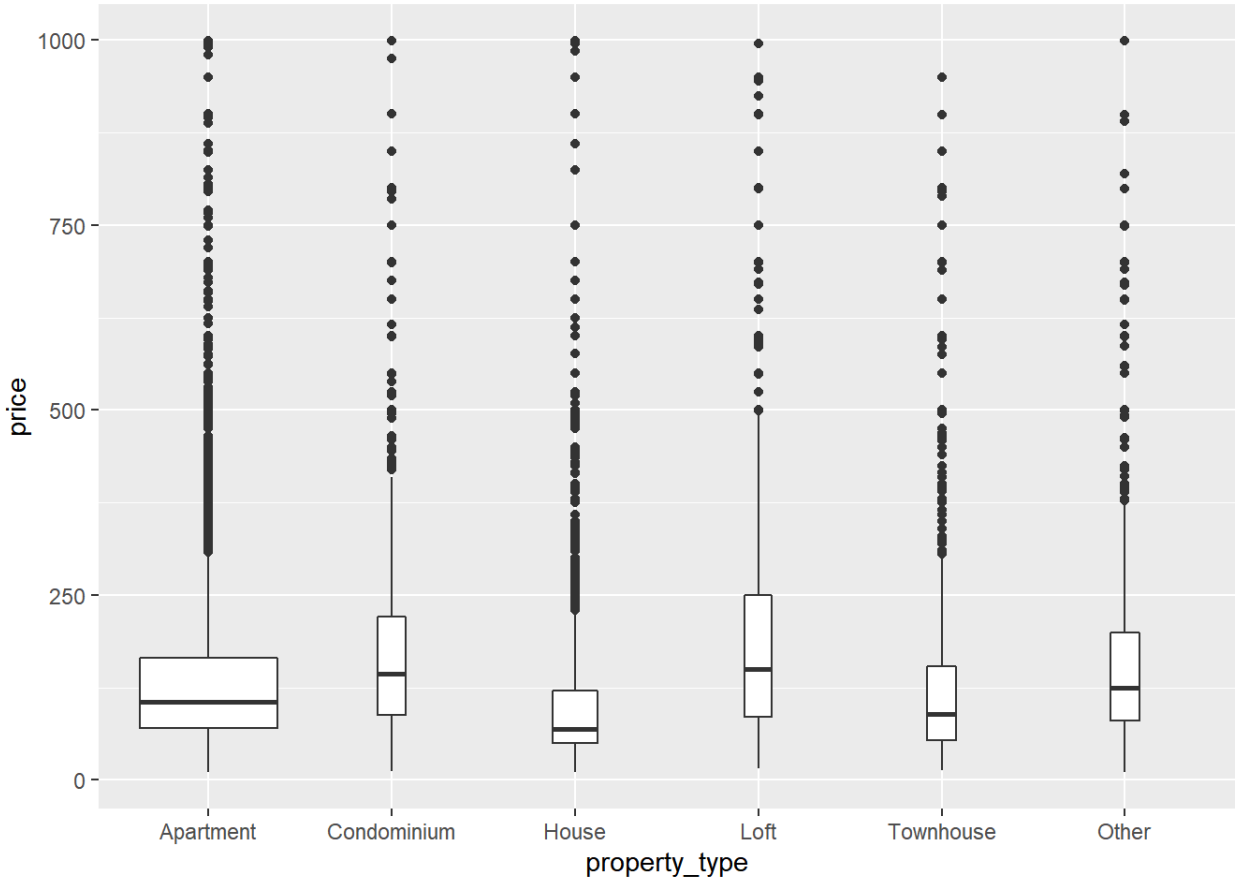


Analyzing Factor Variables

factor: property_type, room_type

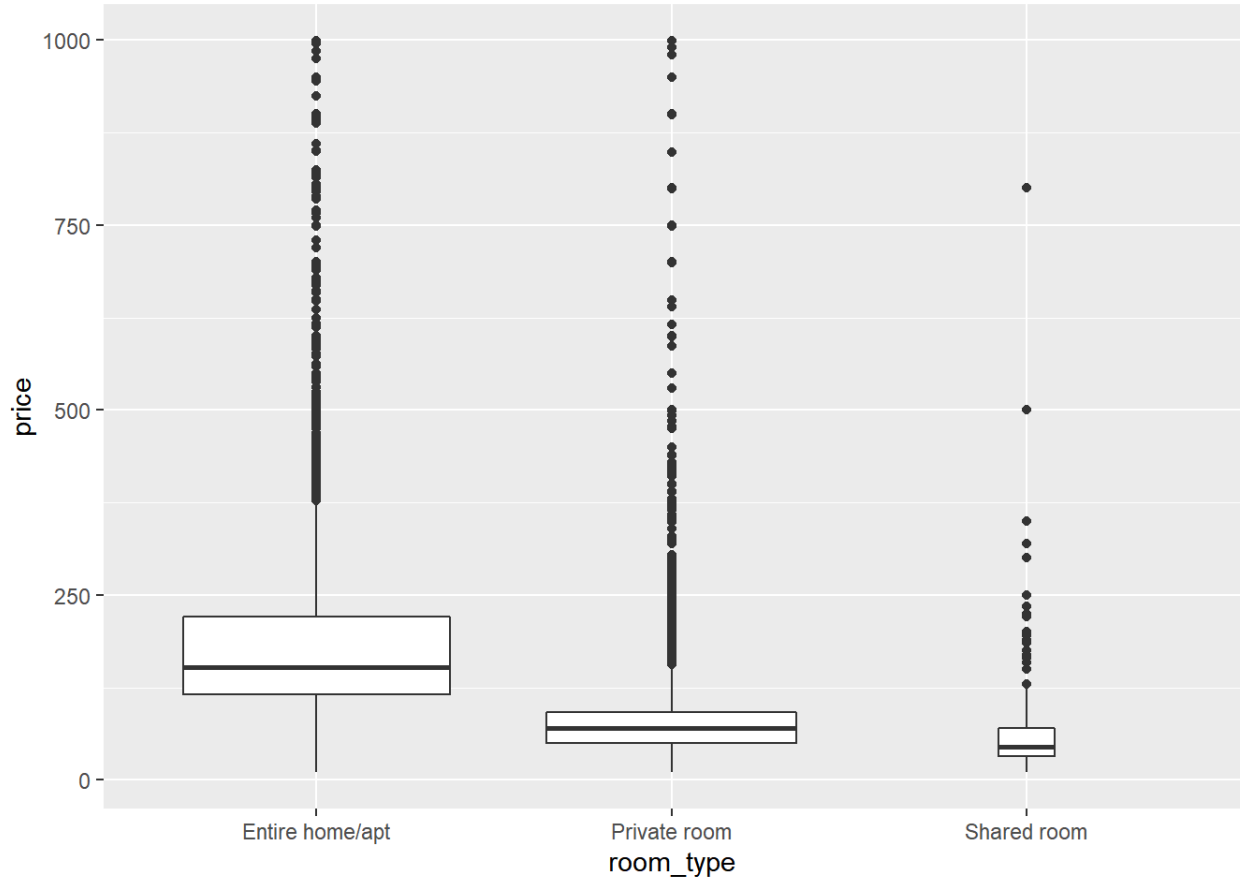
property_type

```
ggplot(data, aes(x=property_type,y=price)) + box
```



room_type

```
ggplot(data, aes(x=room_type,y=price)) + box
```

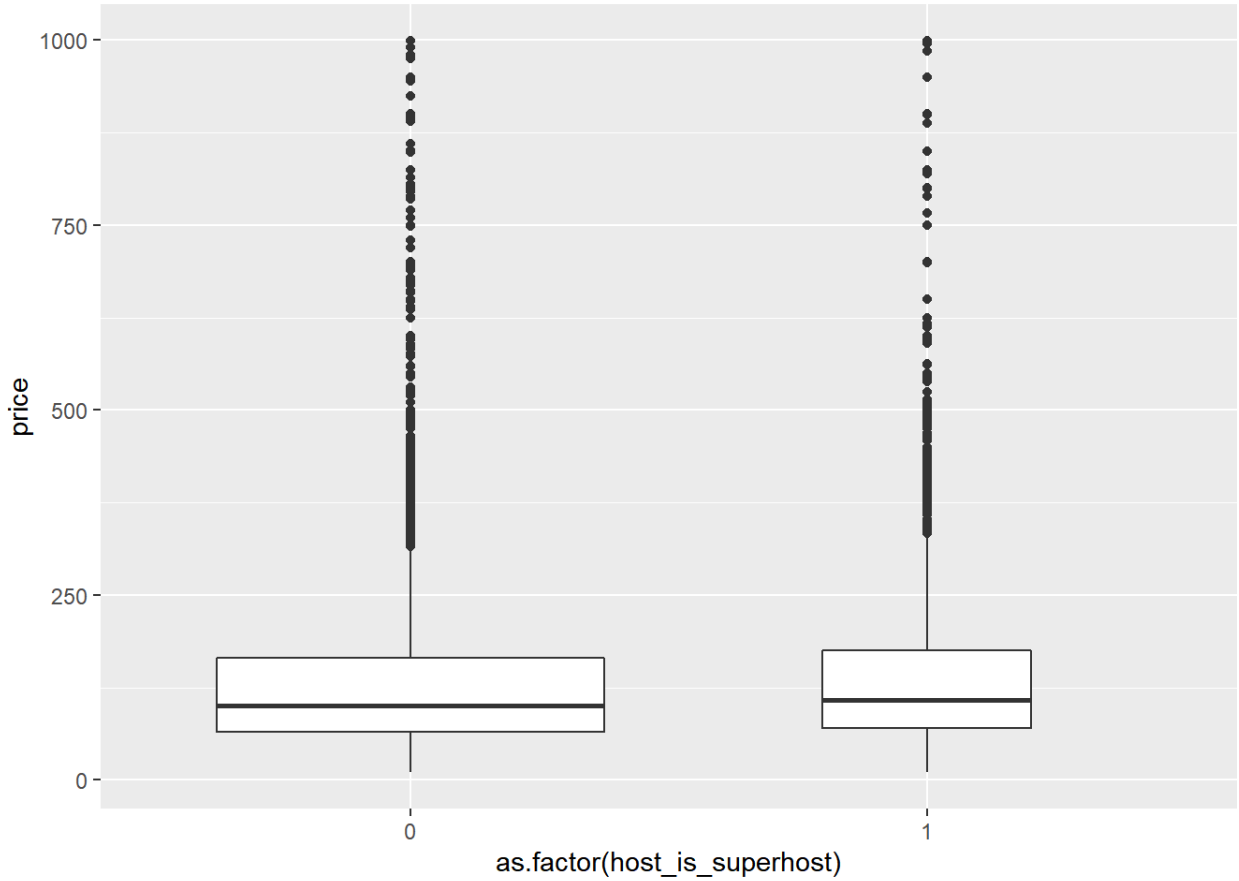


Analyzing Numeric Factor Variables

numeric factor: host_is_superhost, accommodates, bathrooms, bedrooms, beds, availability_30, review_scores_rating, review_scores_accuracy, review_scores_cleanliness, review_scores_checkin, review_scores_communication, review_scores_location, review_scores_value, instant_bookable

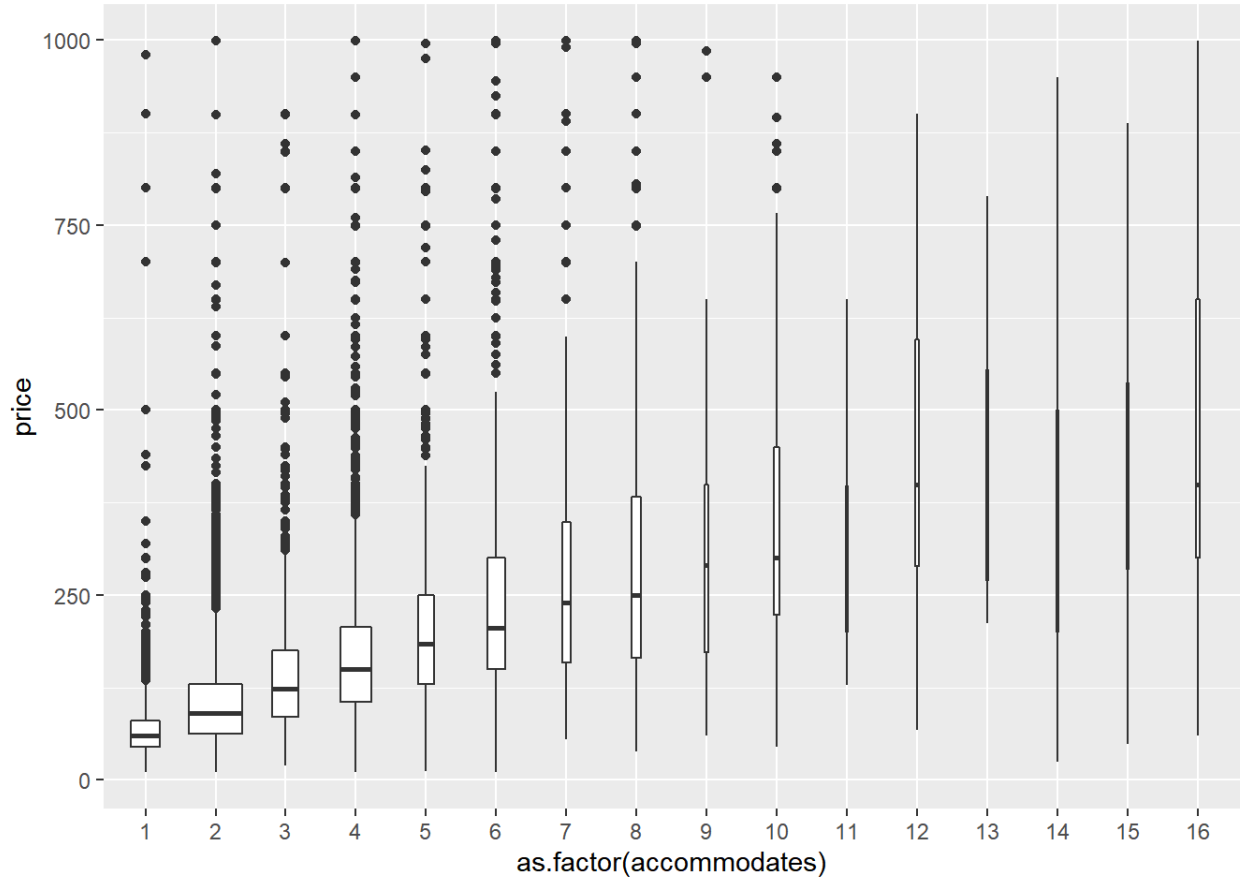
host_is_superhost

```
ggplot(data, aes(x=as.factor(host_is_superhost),y=price)) + box
```



accommodates – might convert 9 onwards as 9

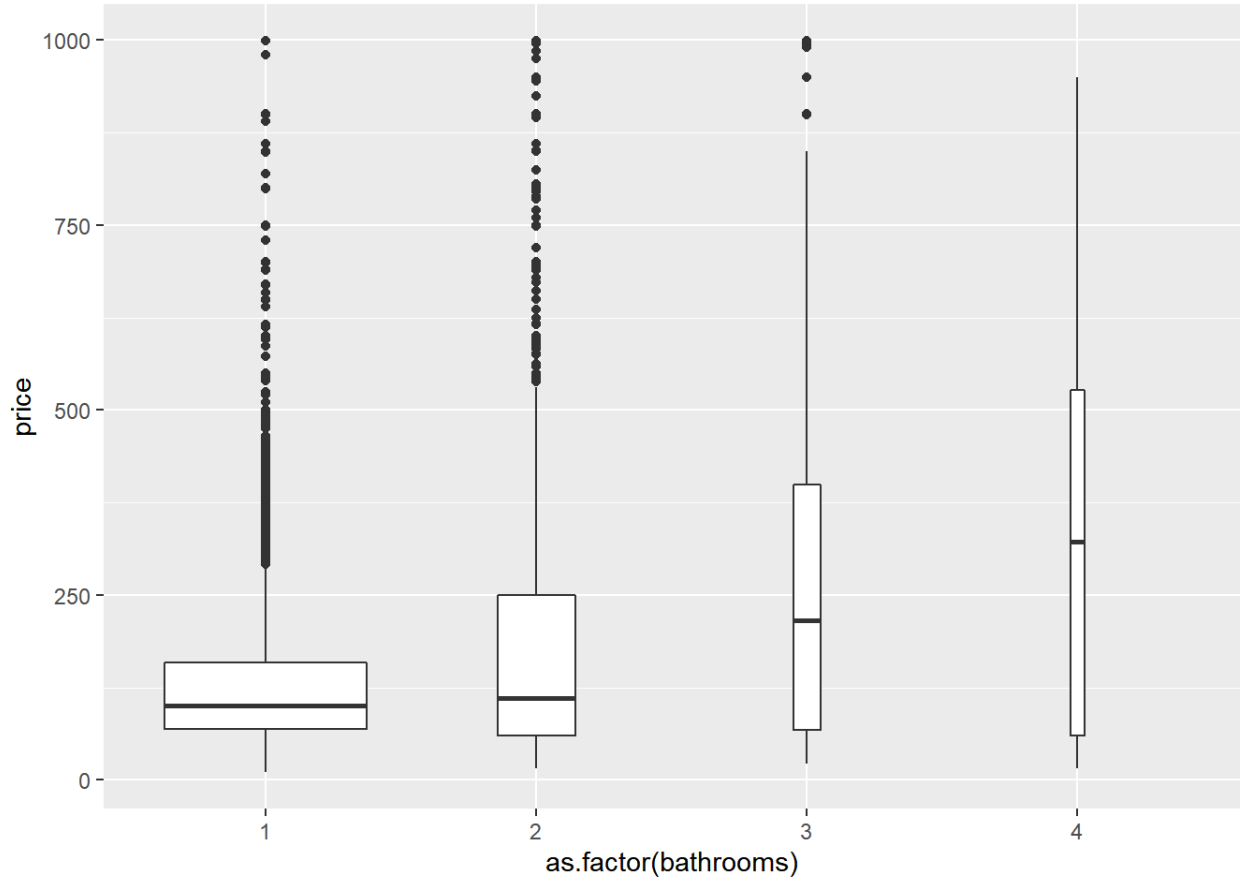
```
ggplot(data, aes(x=as.factor(accommodates),y=price)) + box
```



###

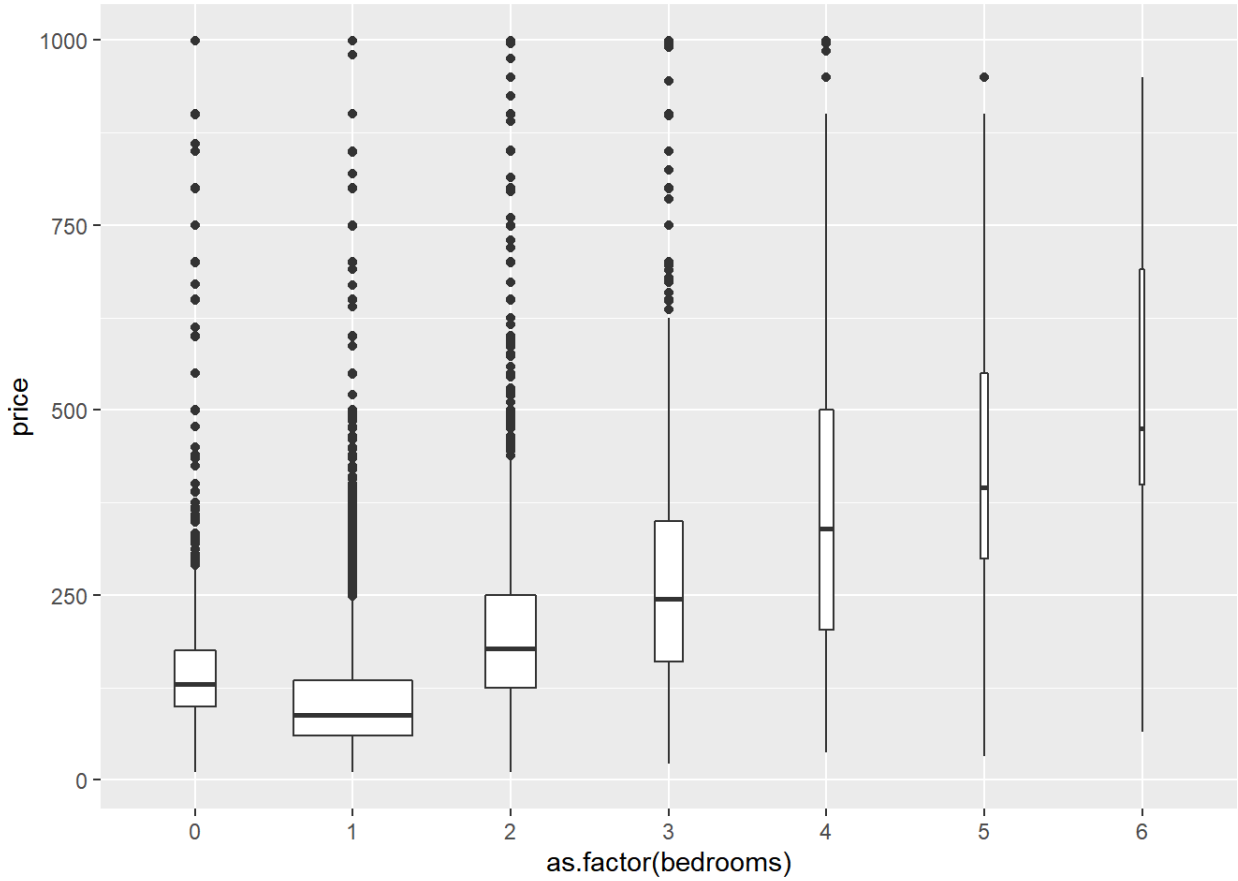
bathrooms- we have converted

```
ggplot(data, aes(x=as.factor(bathrooms),y=price)) + box
```



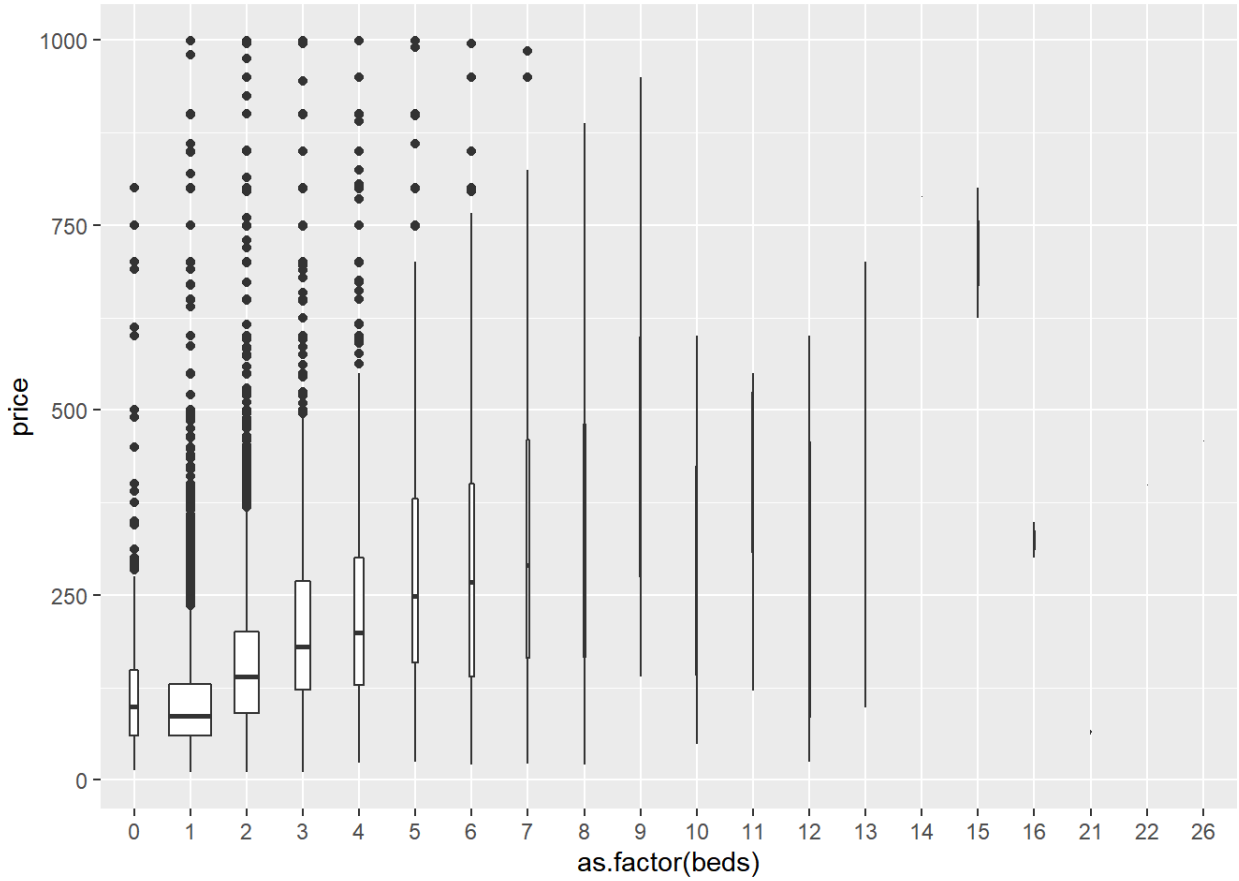
bedrooms

```
ggplot(data, aes(x=as.factor(bedrooms),y=price)) + box
```



###beds – might convert 8 onwards as 8

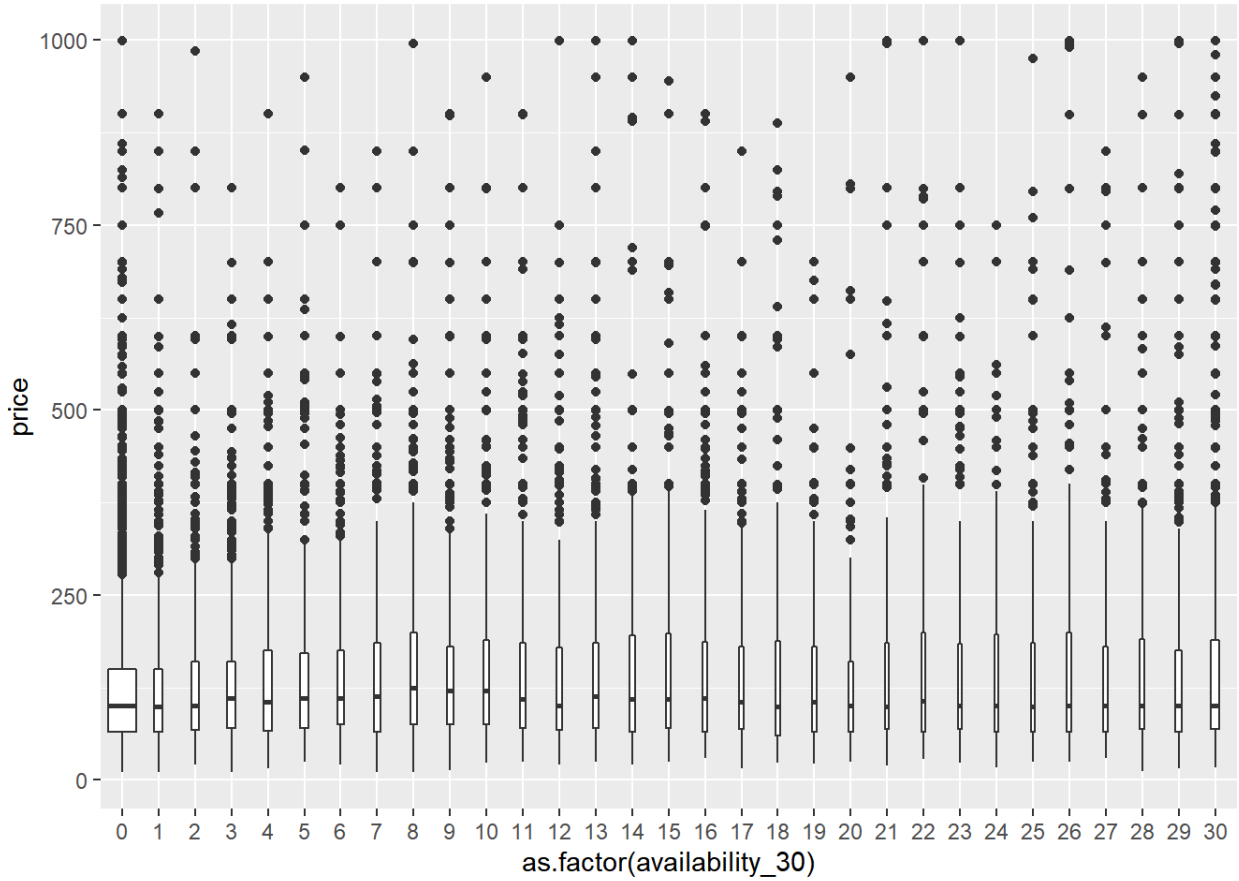
```
ggplot(data, aes(x=as.factor(beds),y=price)) + box
```



###

availability_30

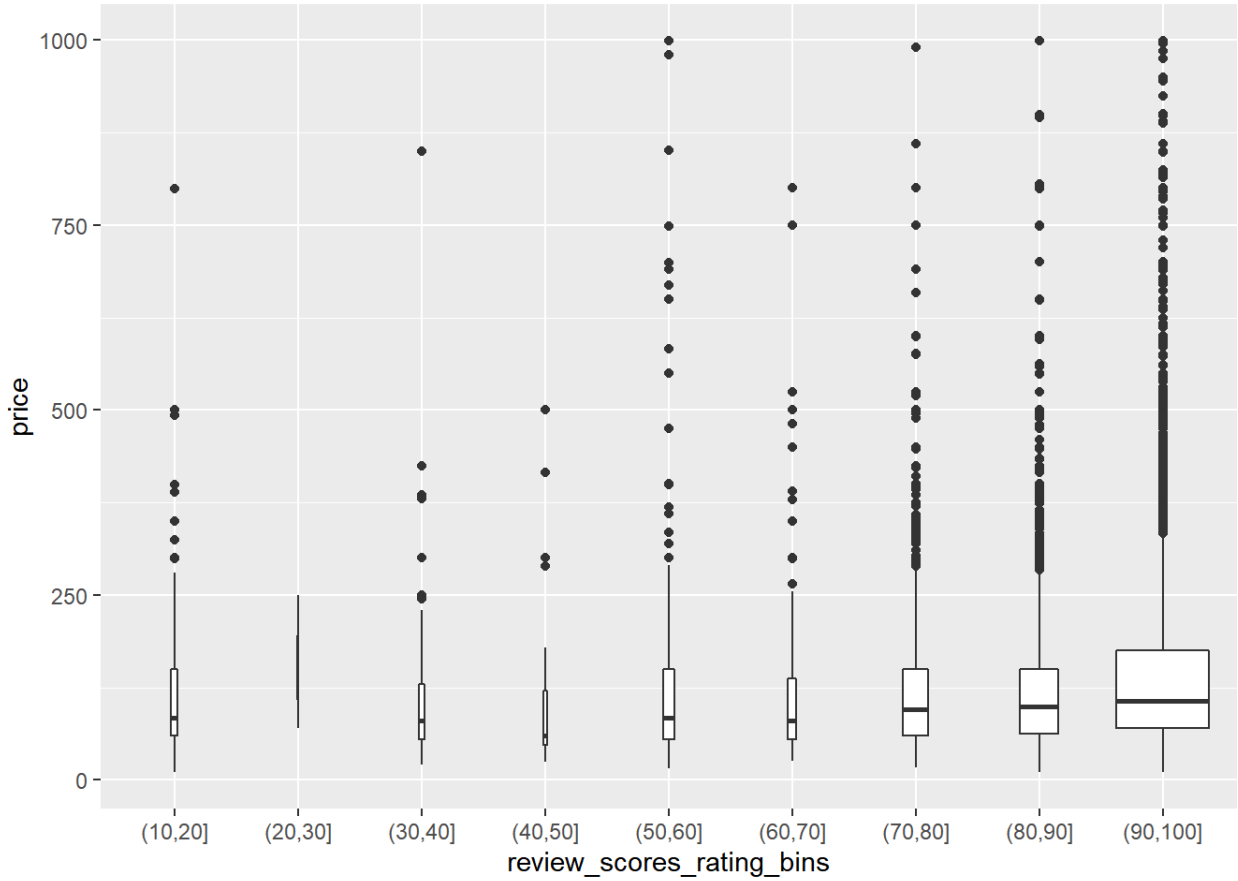
```
ggplot(data, aes(x=as.factor(availability_30),y=price)) + box
```



###

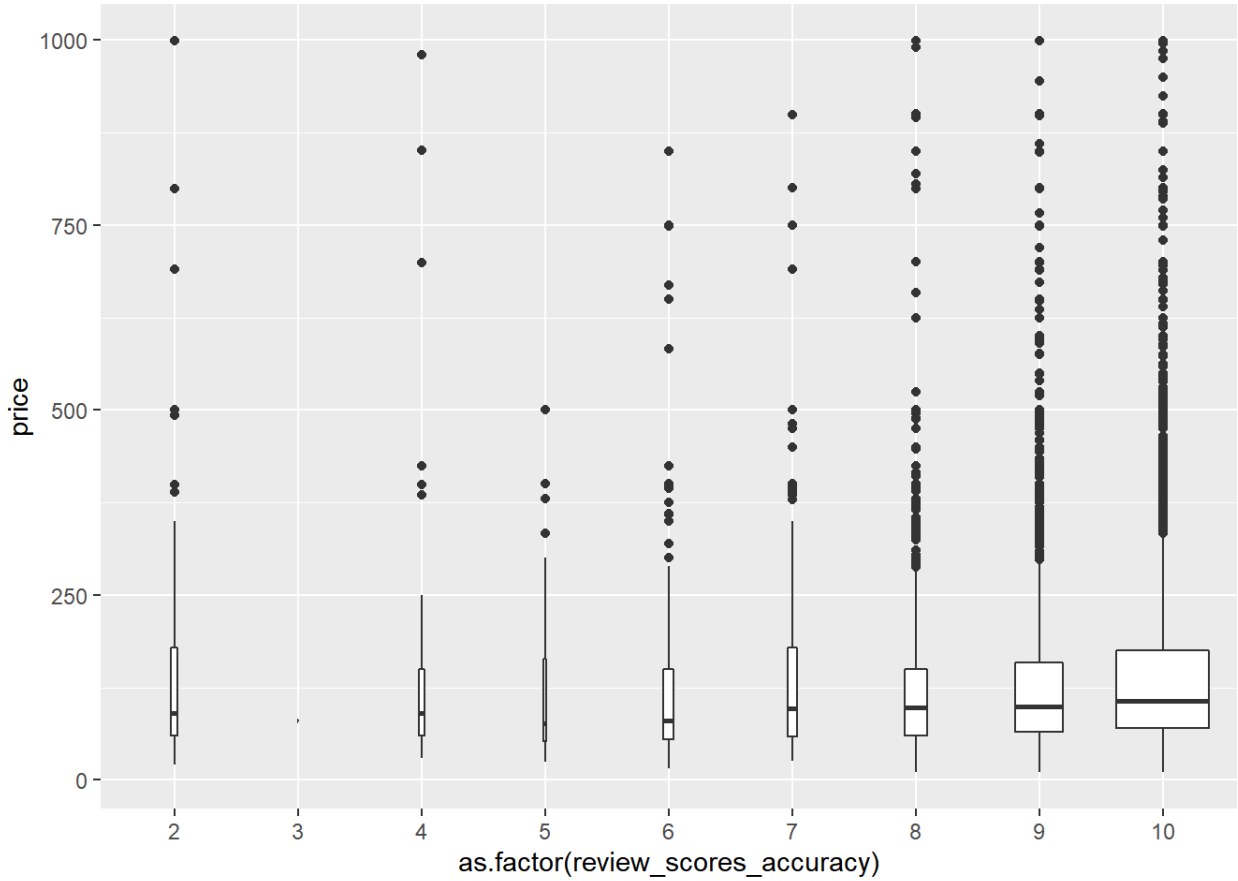
review_scores_rating

```
data%>%
  mutate(review_scores_rating_bins = cut(review_scores_rating, breaks = c(0,10,20,30,40,50,60,70,80,90,
100))) %>%
  ggplot(aes(review_scores_rating_bins,price)) + box
```



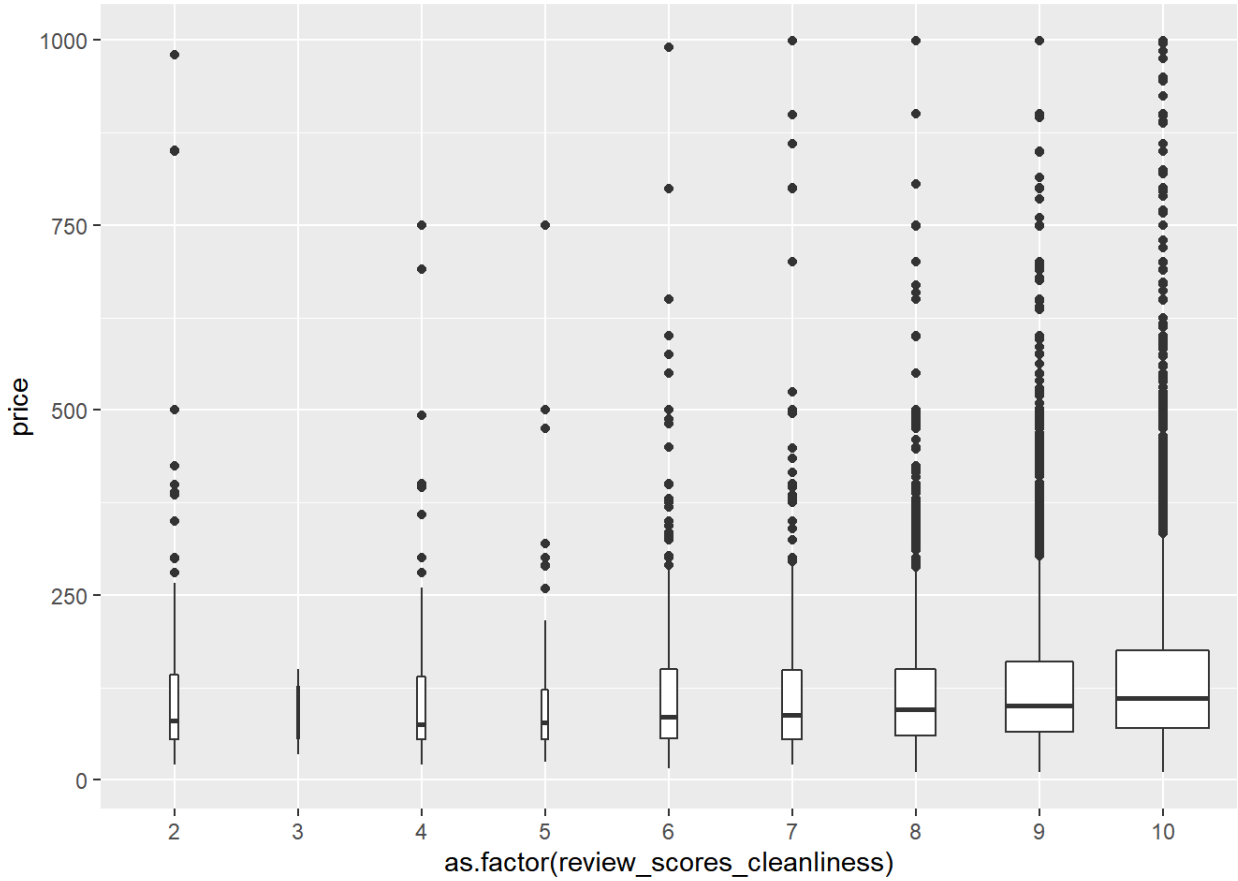
review_scores_accuracy

```
ggplot(data, aes(x=as.factor(review_scores_accuracy),y=price)) + box
```



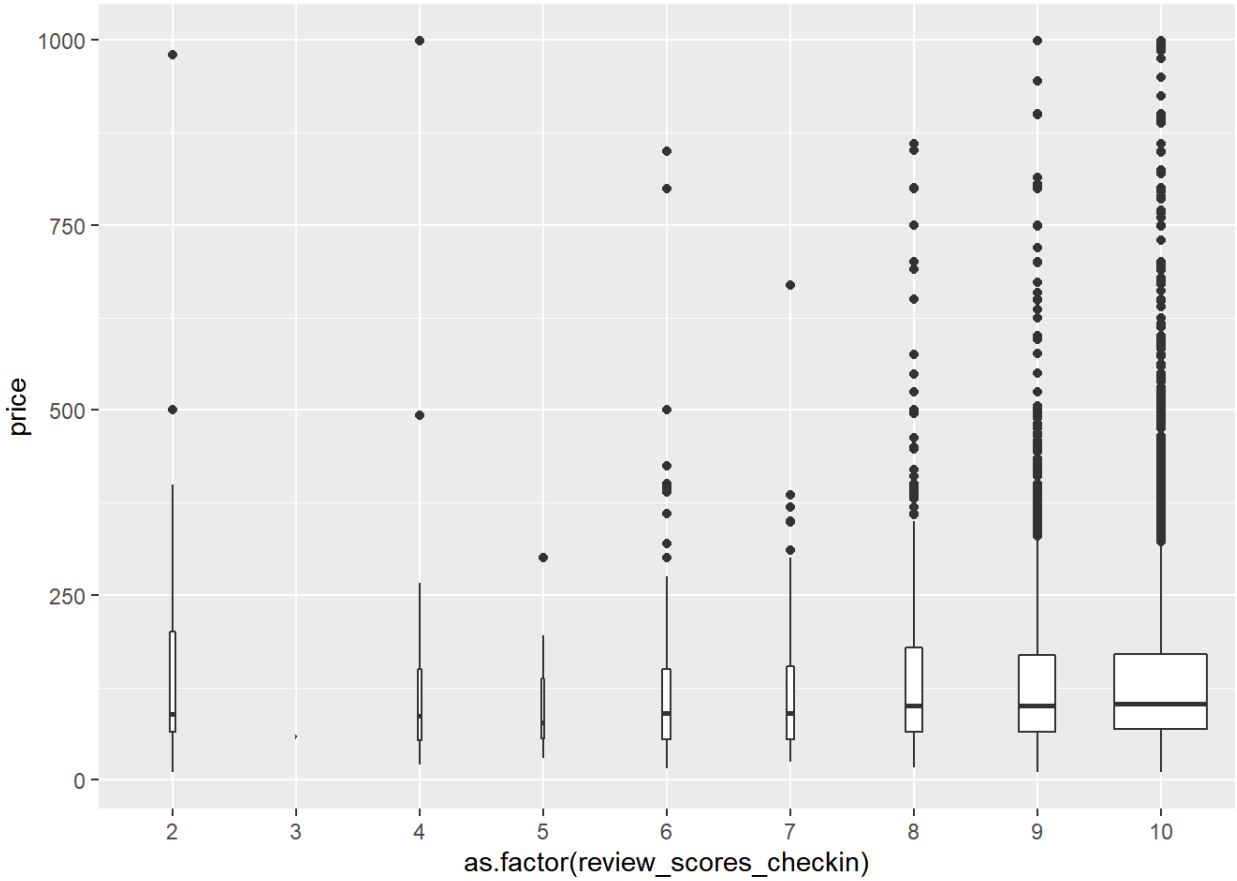
review_scores_cleanliness

```
ggplot(data, aes(x=as.factor(review_scores_cleanliness),y=price)) + box
```



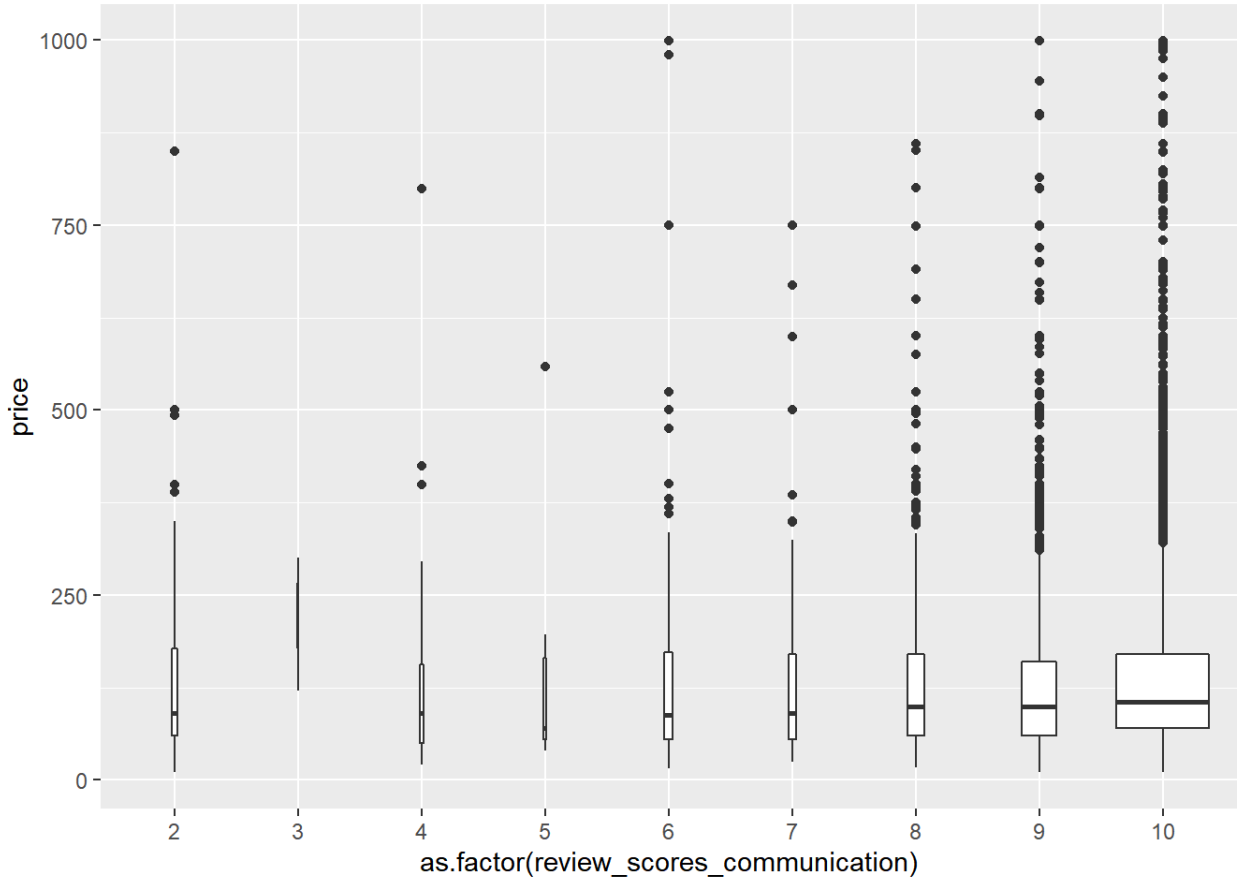
review_scores_checkin

```
ggplot(data, aes(x=as.factor(review_scores_checkin),y=price)) + box
```



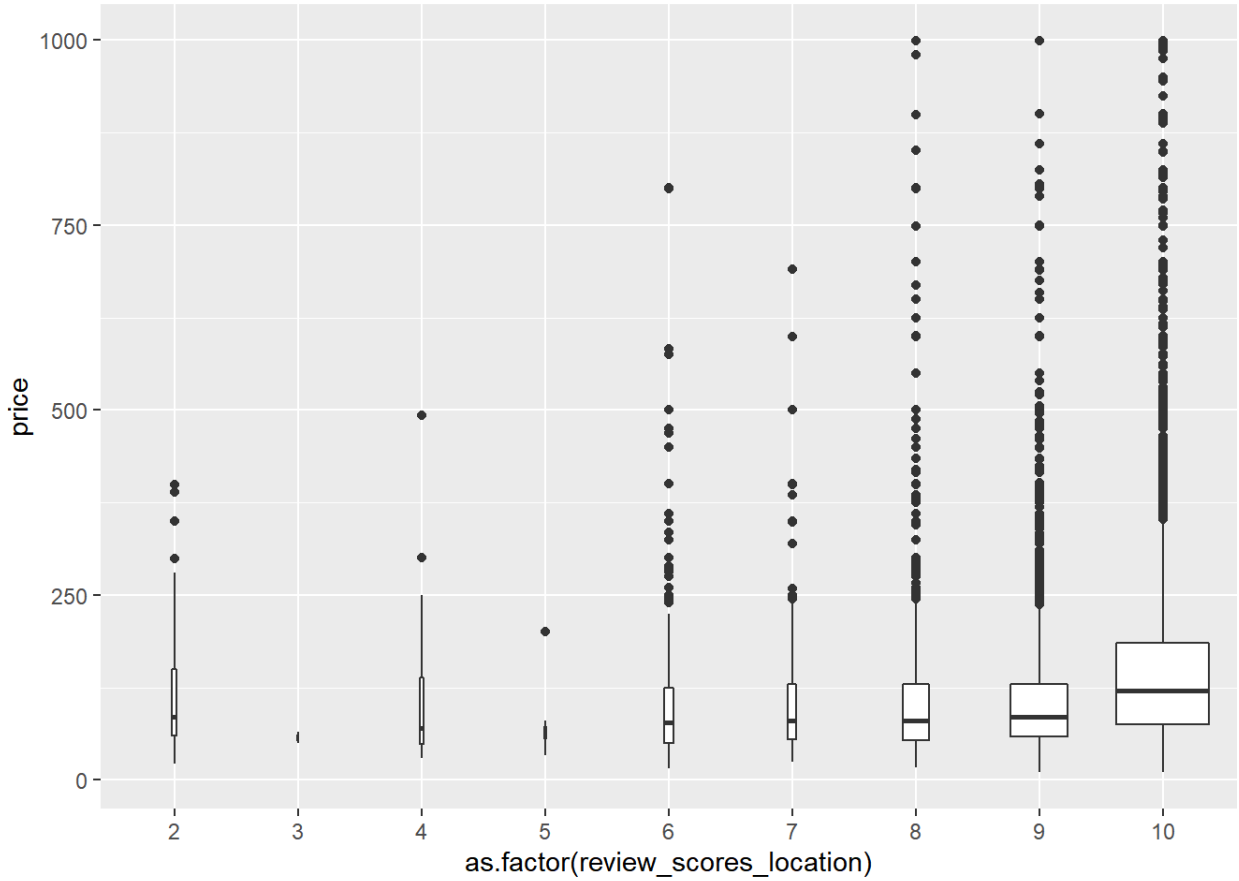
review_scores_communication

```
ggplot(data, aes(x=as.factor(review_scores_communication),y=price)) + box
```



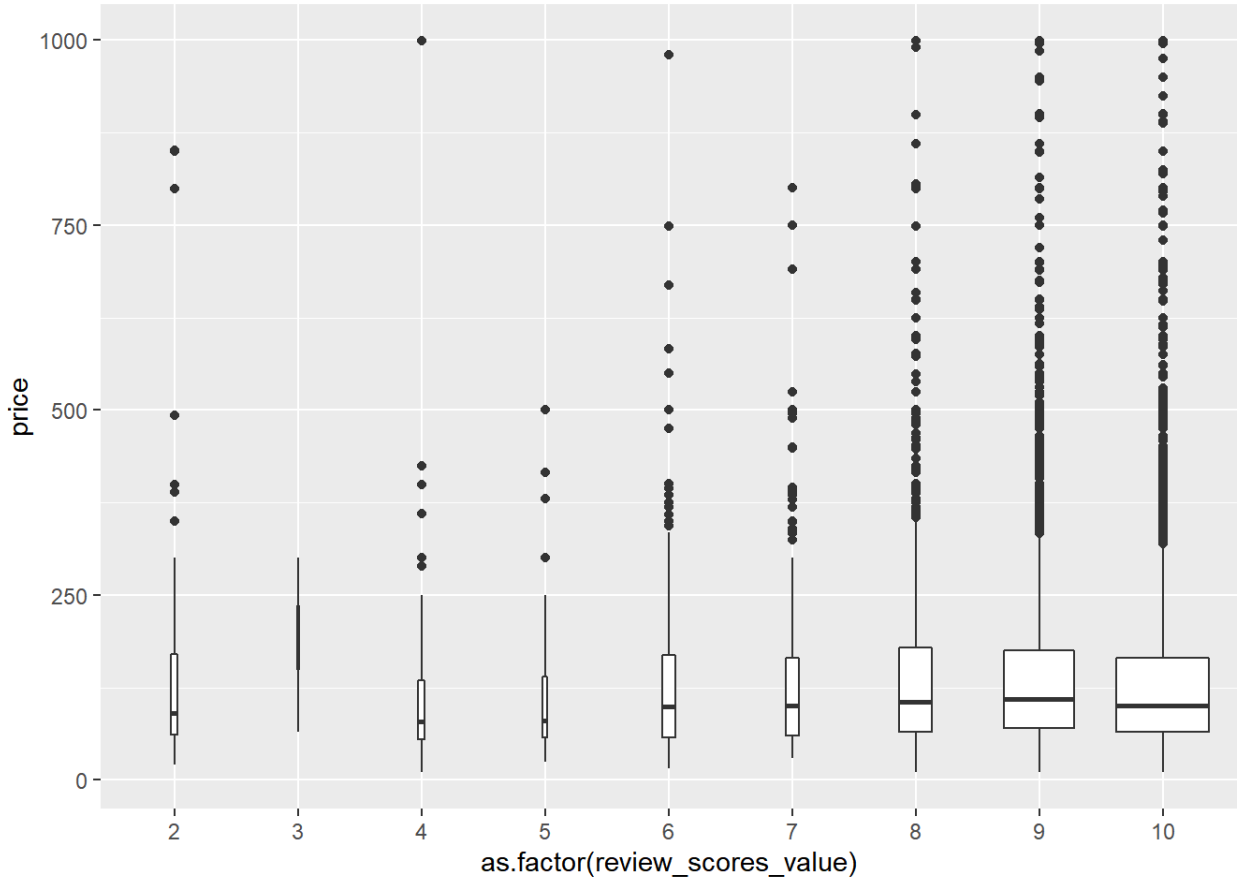
review_scores_location

```
ggplot(data, aes(x=as.factor(review_scores_location),y=price)) + box
```



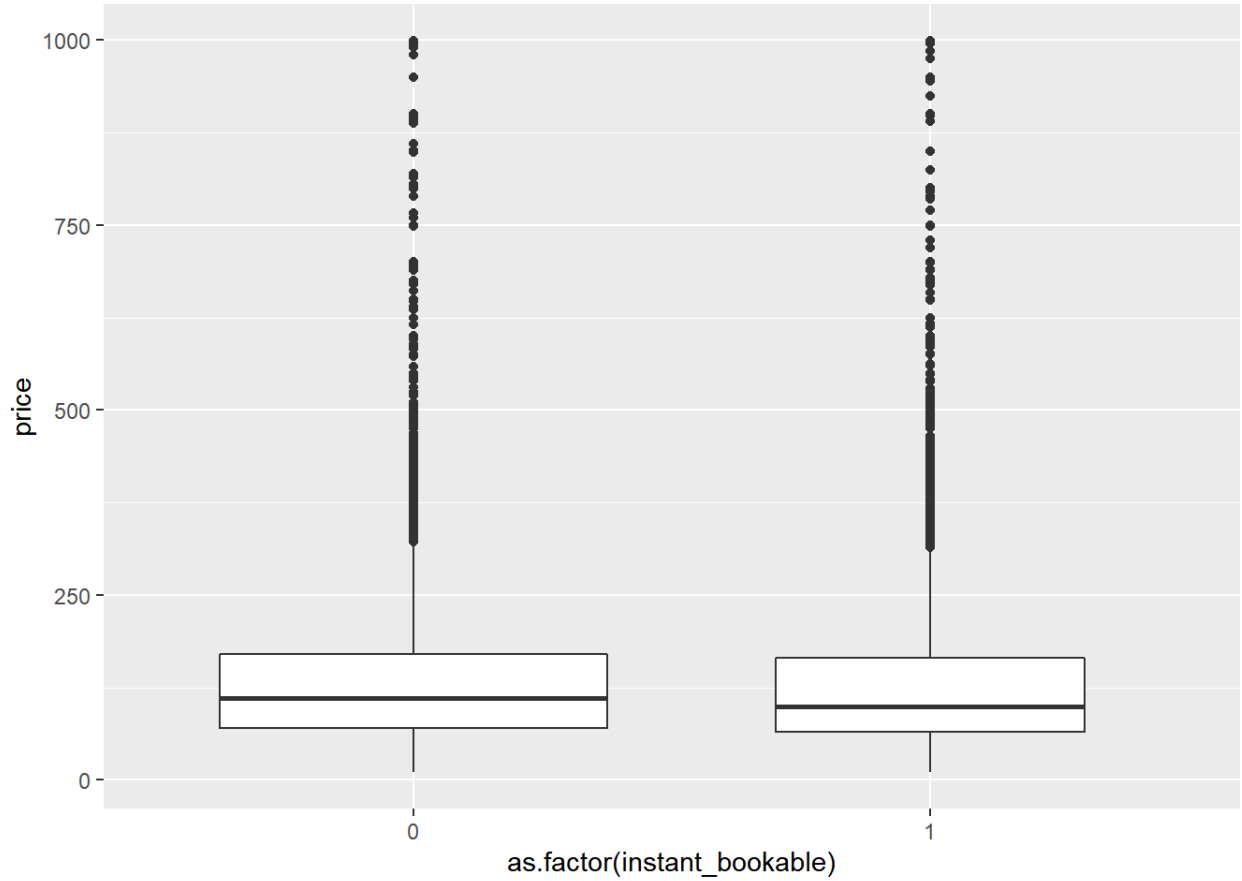
review_scores_value

```
ggplot(data, aes(x=as.factor(review_scores_value),y=price)) + box
```



instant_bookable

```
ggplot(data, aes(x=as.factor(instant_bookable),y=price)) + box
```

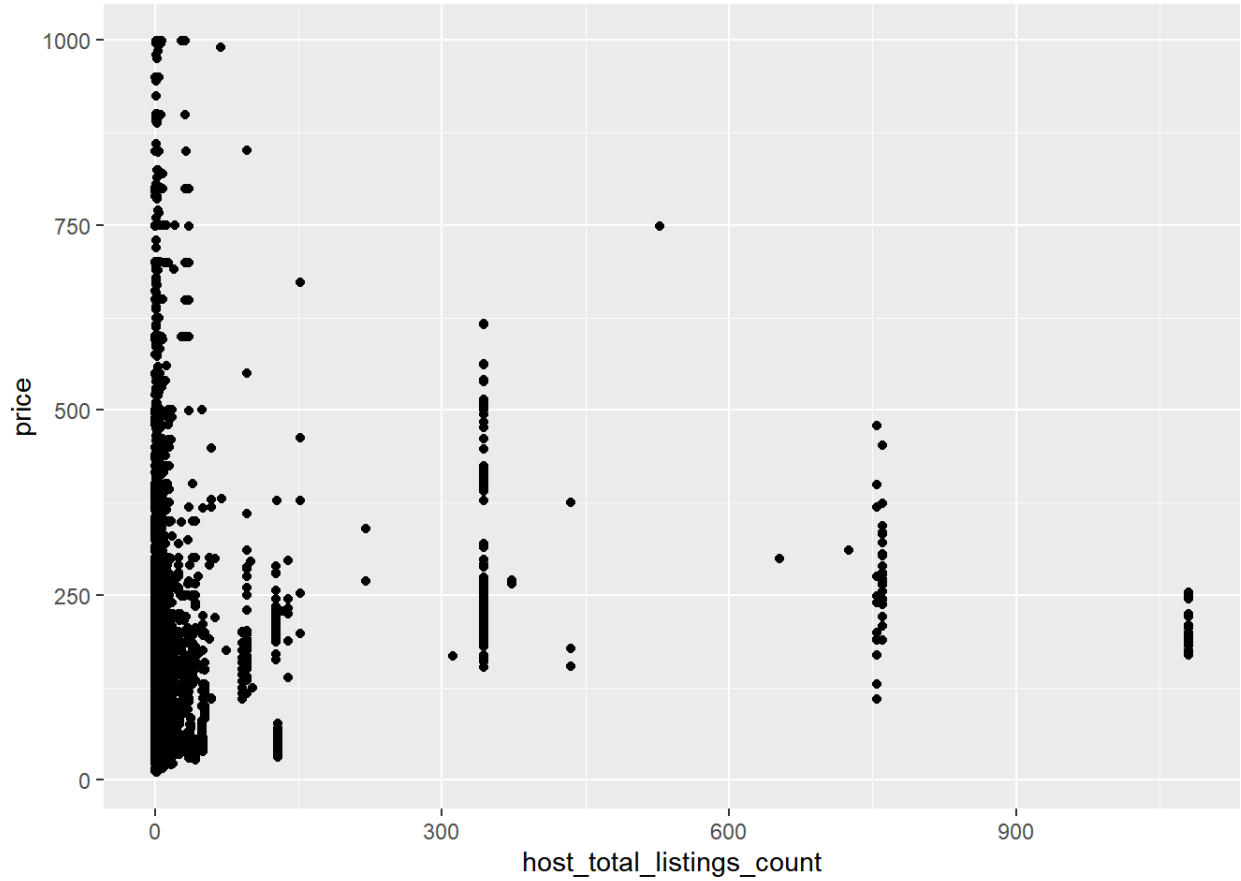


Analyzing

Numeric Variables ## numeric: host_total_listings_count, cleaning_fee, minimum_nights, maximum_nights, number_of_reviews, number_of_reviews_ltm, calculated_host_listings_count, reviews_per_month, listing_active_duration, host_active_duration

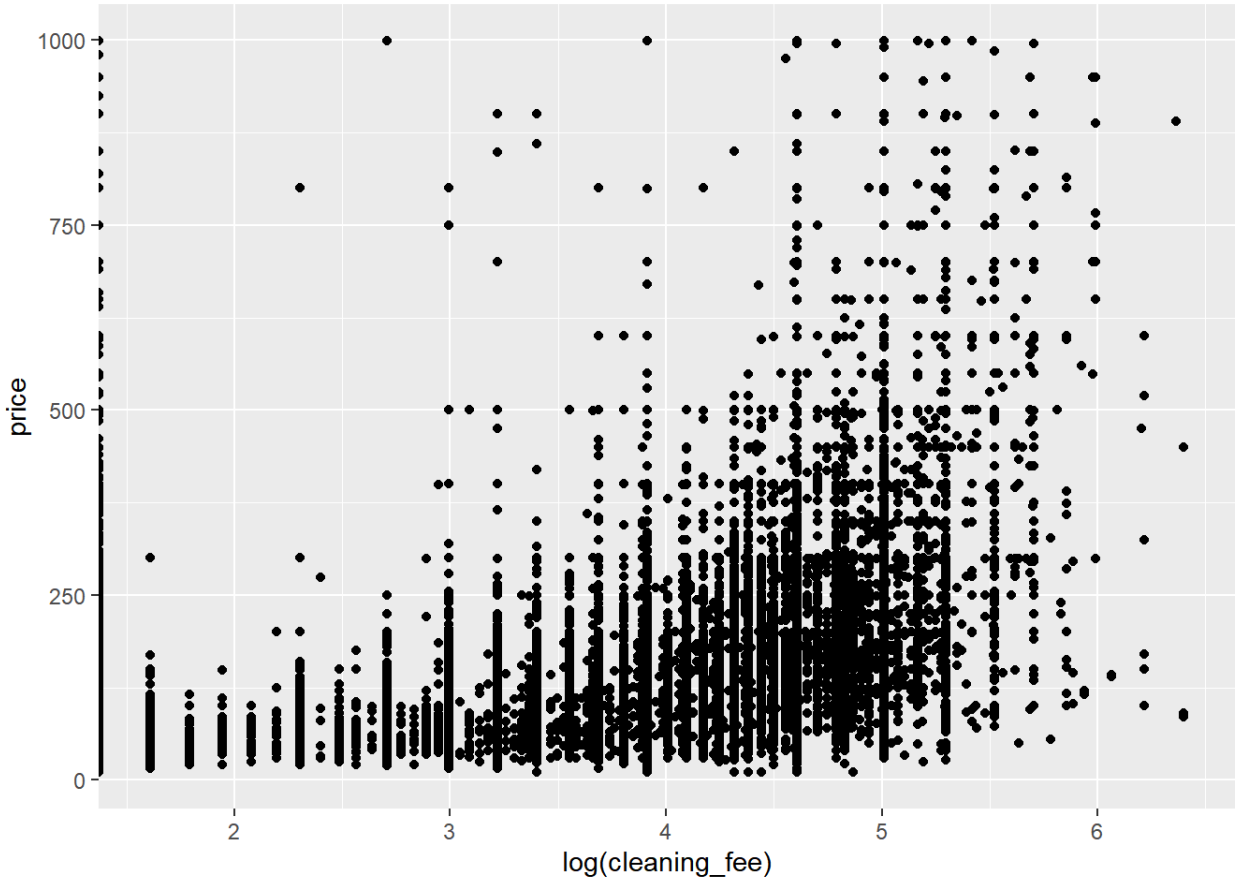
host_total_listings_count

```
ggplot(data, aes(x=host_total_listings_count,y=price)) + scatter
```



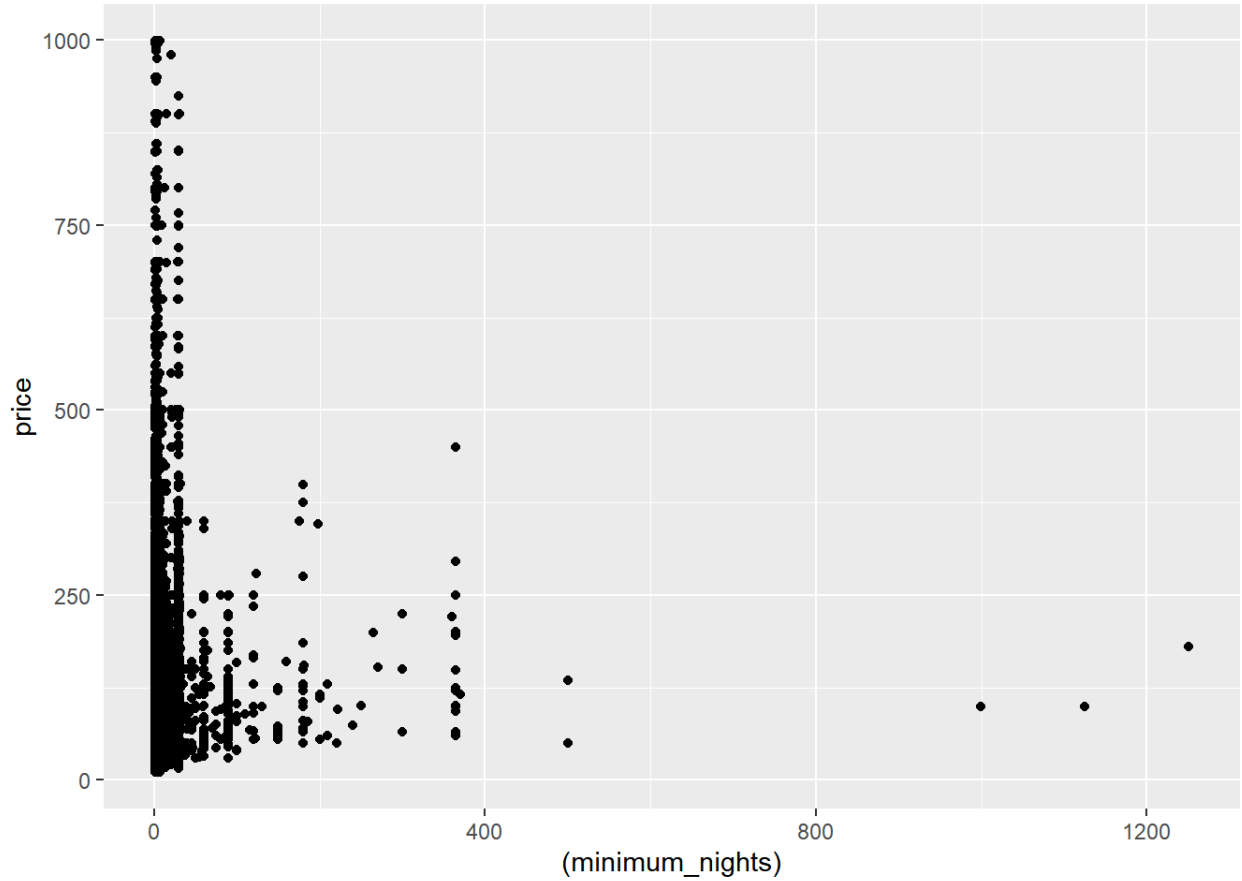
cleaning_fee

```
ggplot(data, aes(x=log(cleaning_fee),y=price)) + scatter #+ geom_smooth(method = "lm", se=FALSE, color
="black", aes(group=1))
```



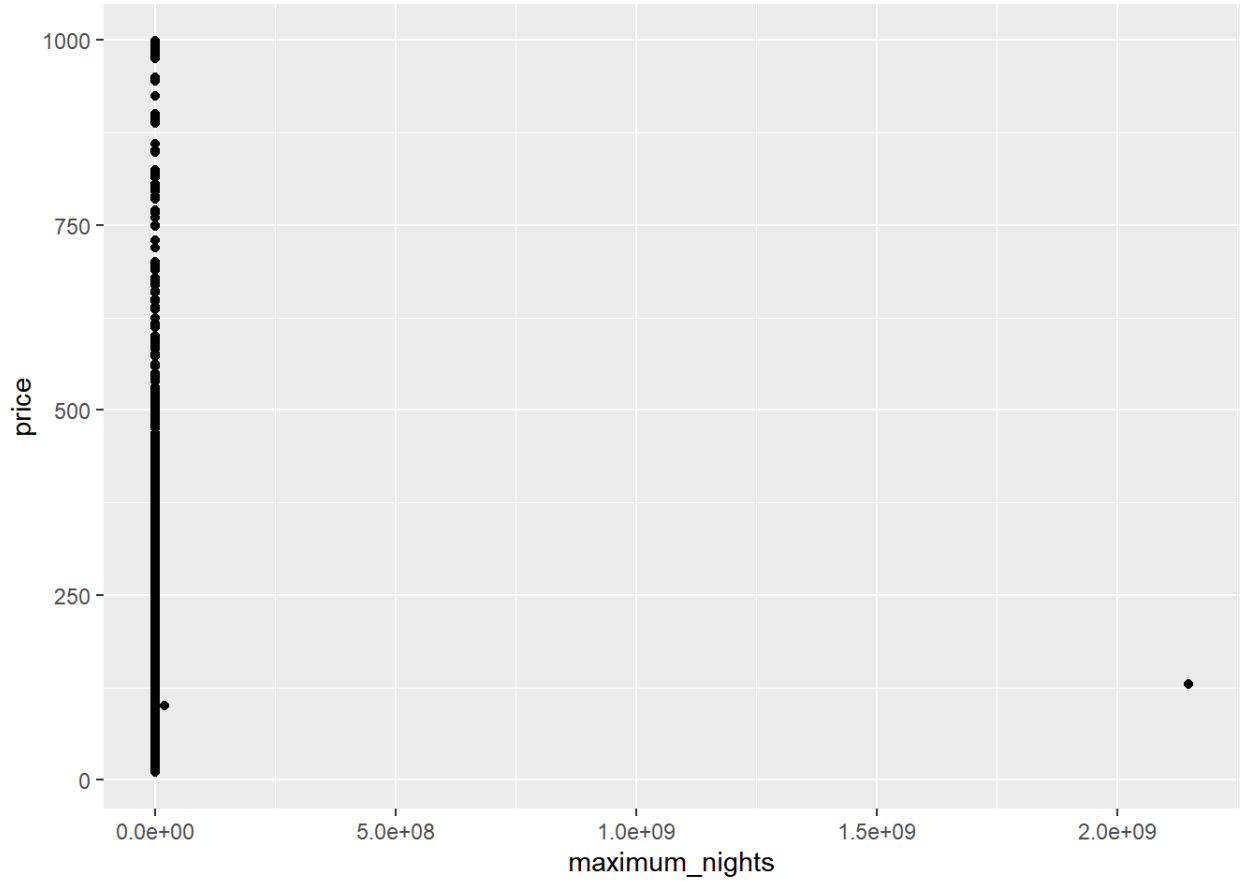
minimum_nights

```
ggplot(data, aes(x=(minimum_nights),y=price)) + scatter
```



maximum_nights

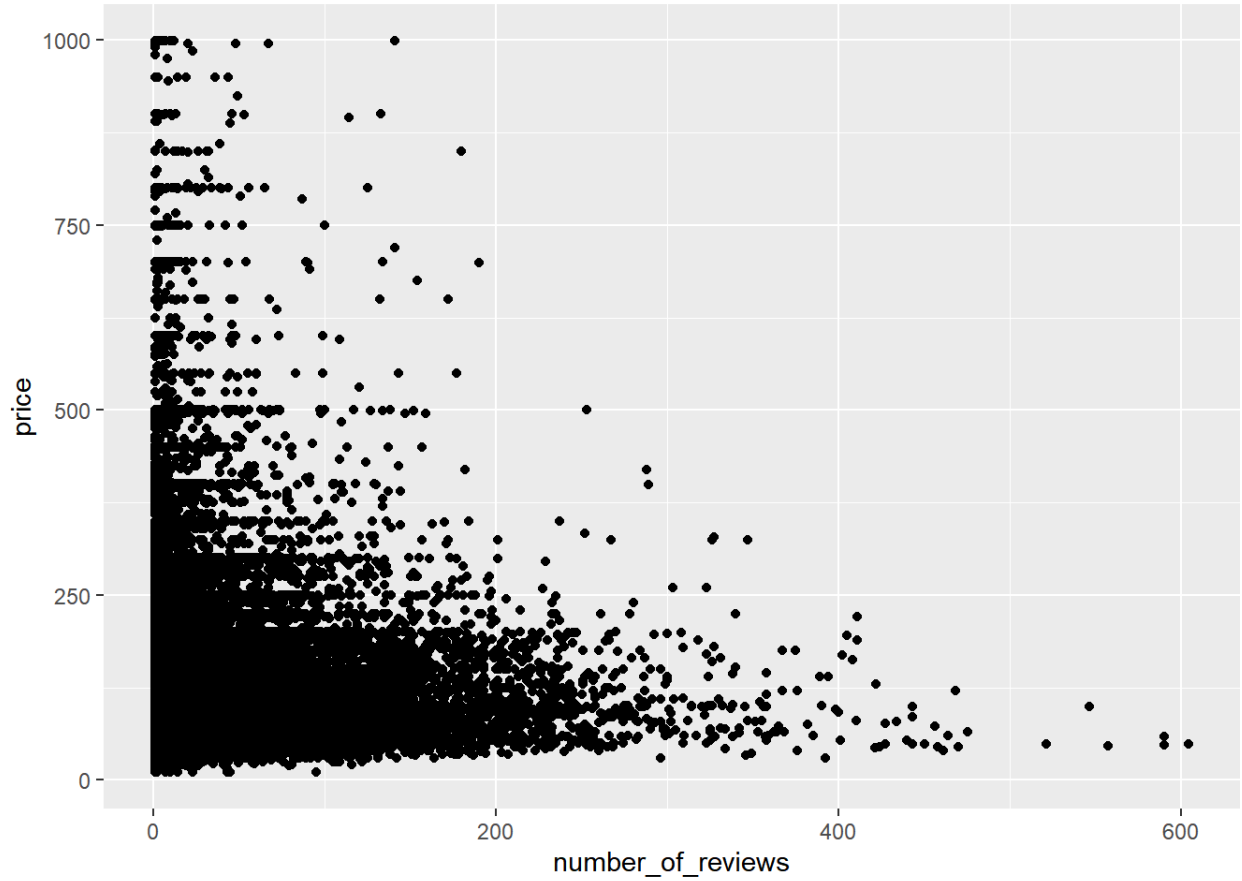
```
ggplot(data, aes(x=maximum_nights,y=price)) + scatter
```



###

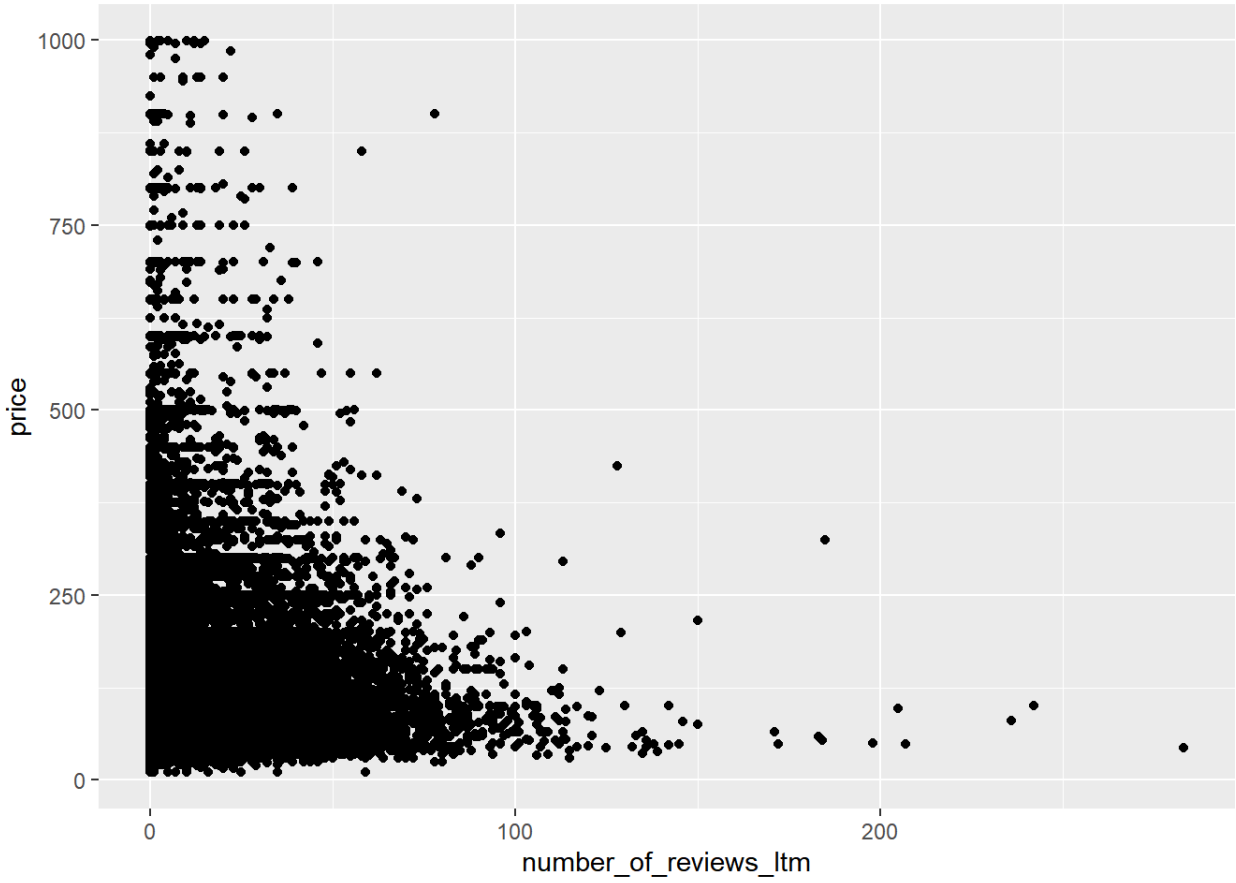
number_of_reviews

```
ggplot(data, aes(x=number_of_reviews,y=price)) + scatter
```



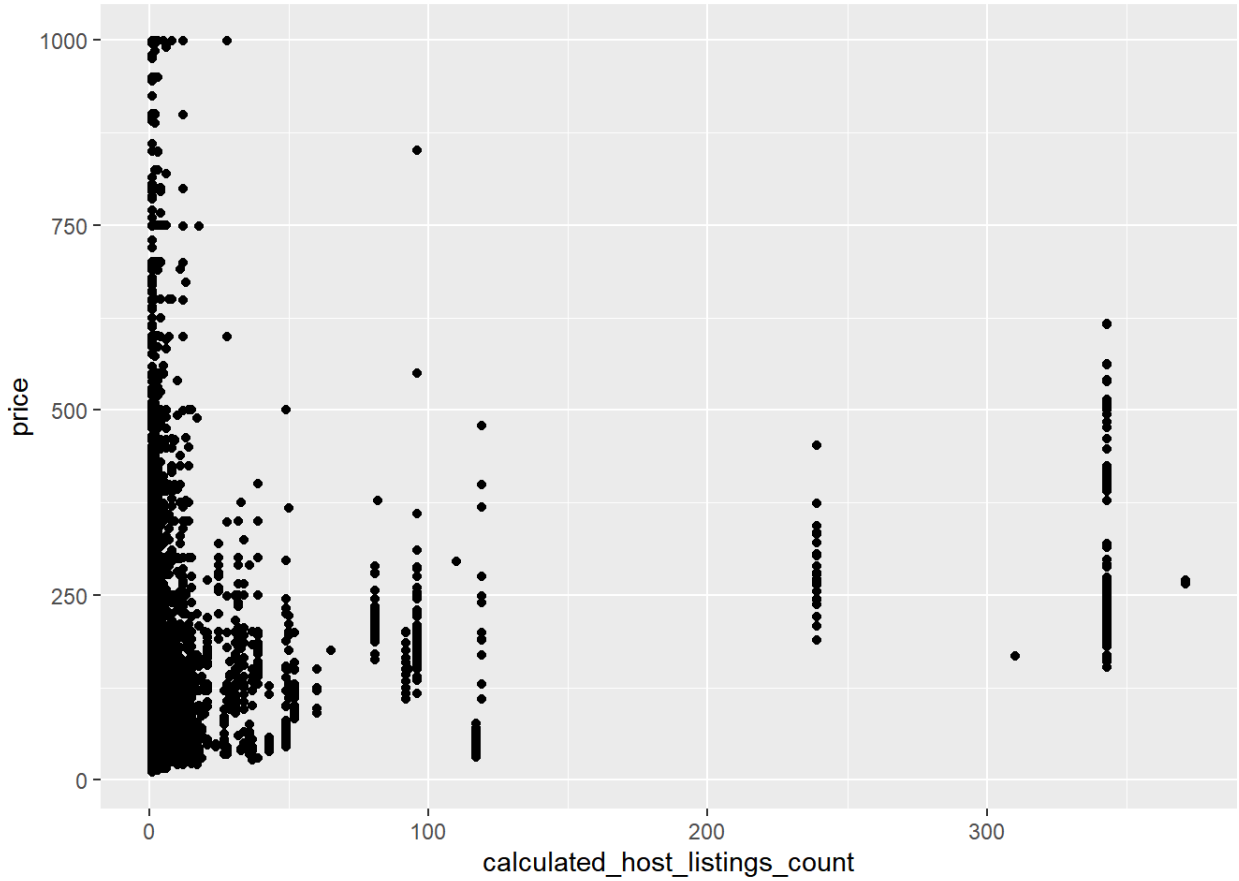
number_of_reviews_ltm

```
ggplot(data, aes(x=number_of_reviews_ltm,y=price)) + scatter
```



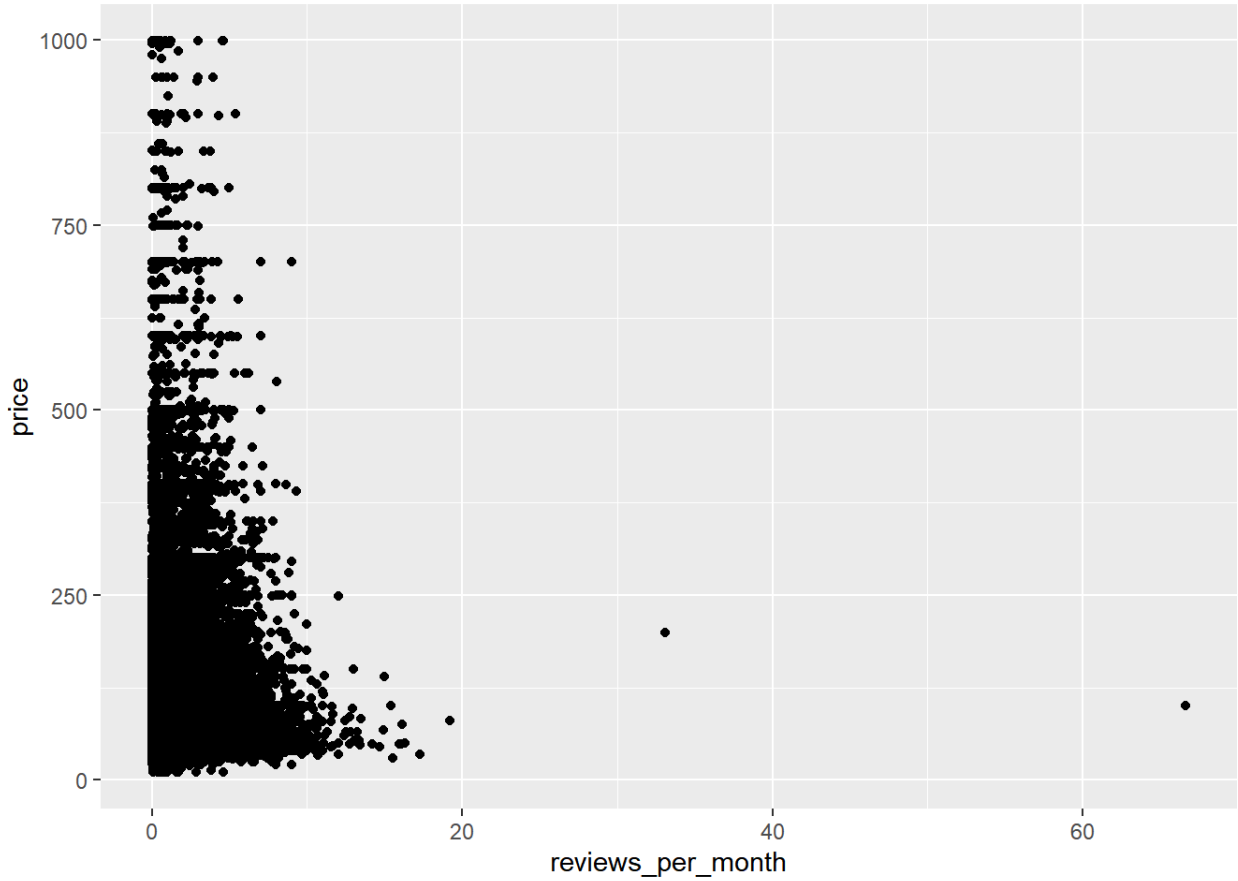
calculated_host_listings_count

```
ggplot(data, aes(x=calculated_host_listings_count,y=price)) + scatter
```



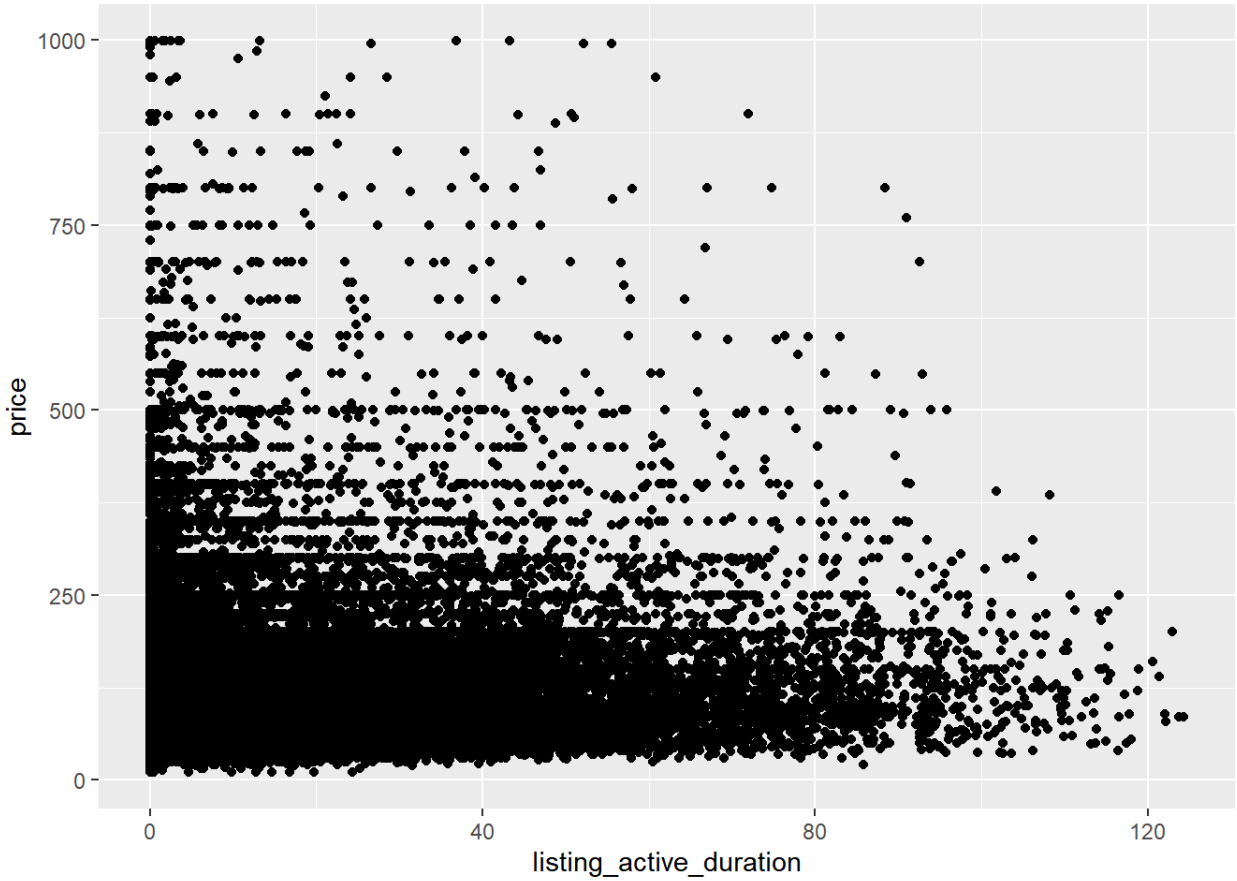
reviews_per_month

```
ggplot(data, aes(x=reviews_per_month,y=price)) + scatter
```



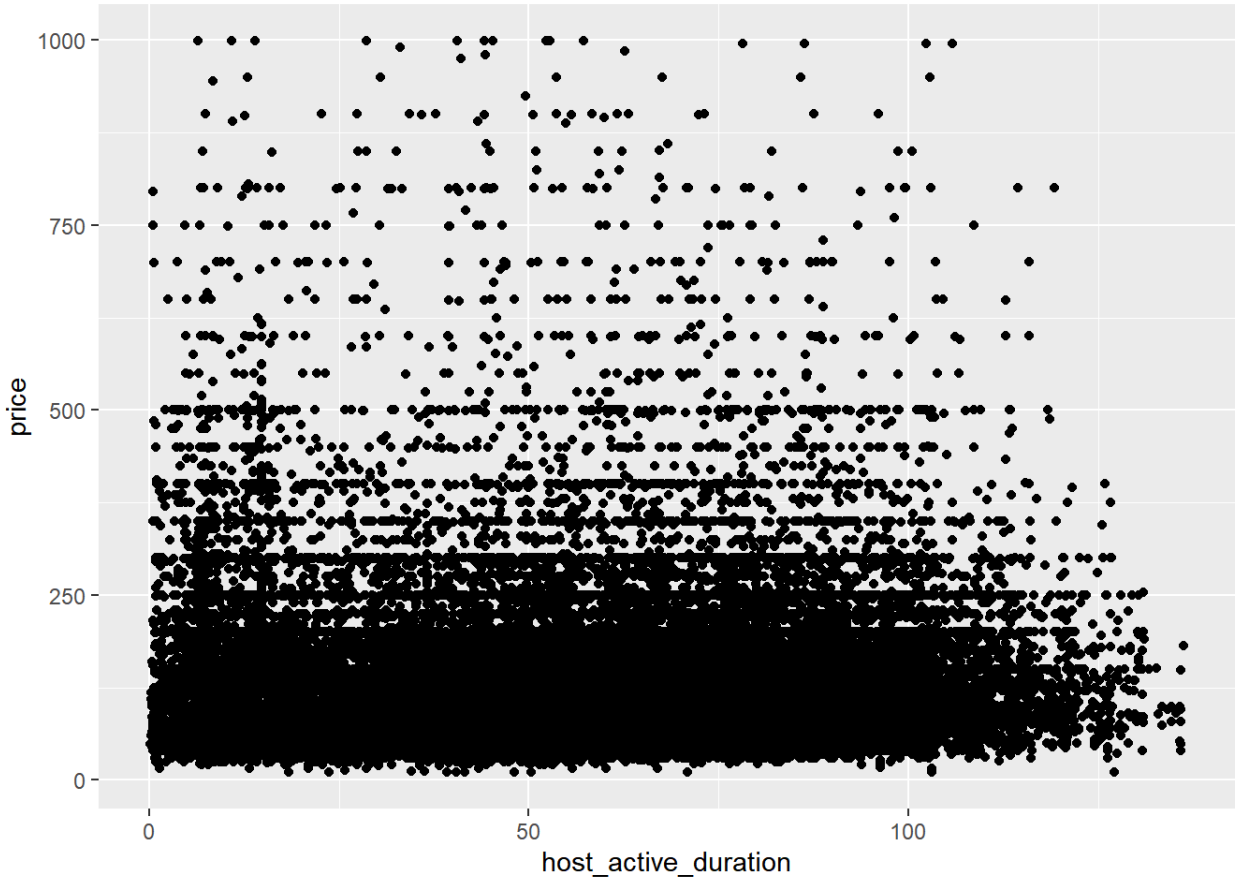
listing_active_duration

```
ggplot(data, aes(x=listing_active_duration,y=price)) + scatter
```



host_active_duration

```
ggplot(data, aes(x=host_active_duration,y=price)) + scatter
```



Correlations

```
#library(corrplot)
#numeric variables
numericVars <- which(sapply(data, is.numeric))
data_numVar <- data[, numericVars]

#correlations of all numeric variables
cor_numVar <- cor(data_numVar, use="pairwise.complete.obs")

#sorting on decreasing correlations with price
cor_sorted <- as.matrix(sort(cor_numVar[,'price'], decreasing = TRUE))
print(cor_sorted)
```

```
##                                        [,1]
## price                            1.0000000000
## accommodates                     0.5599244814
## cleaning_fee                     0.5204372469
## beds                             0.4304395718
## bedrooms                         0.4298263449
## guests_included                  0.3792144020
## bathrooms                        0.2556035195
## review_scores_location           0.1321612753
## extra_people                     0.1028865853
## calculated_host_listings_count  0.0988410625
## host_total_listings_count        0.0966659491
## availability_30                  0.0960087133
## availability_365                 0.0872138255
## review_scores_cleanliness        0.0751878409
## availability_60                  0.0679700281
## review_scores_rating             0.0625075131
## availability_90                  0.0497193478
## host_active_duration             0.0279624715
## host_is_superhost                0.0268494262
## review_scores_accuracy           0.0206434046
## listing_active_duration          0.0199215914
## review_scores_communication      0.0148947113
## review_scores_checkin            0.0089322672
## minimum_nights                   0.0068358449
## id                               0.0025195714
## maximum_nights                  -0.0002593775
## review_scores_value             -0.0131239935
## instant_bookable                -0.0189846554
## reviews_per_month               -0.0357953760
## number_of_reviews               -0.0396449764
## number_of_reviews_ltm           -0.0481008297
```

#Multivariate Filter

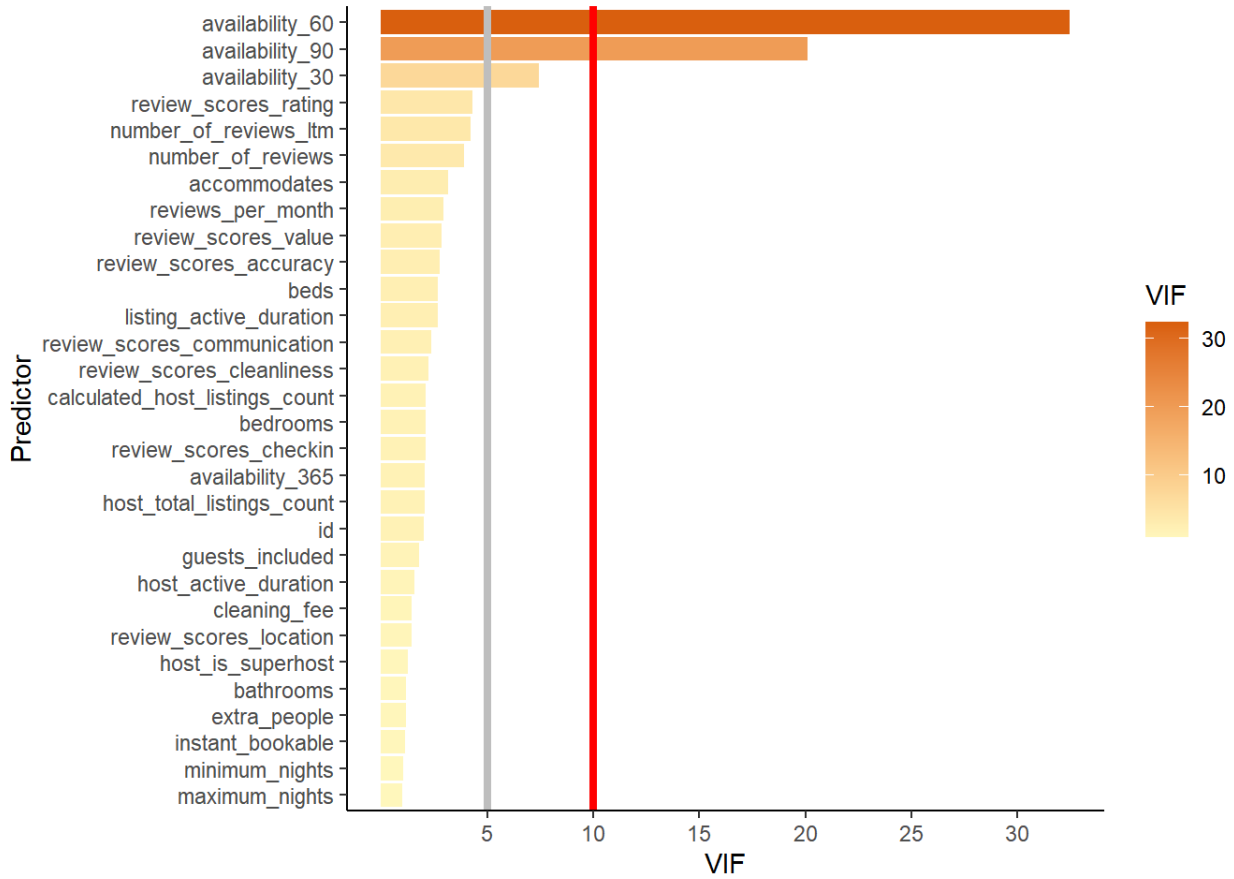
```
model = lm(price~.,data_numVar)
#library(broom)
summary(model) %>%
  tidy()
```

```
## # A tibble: 31 x 5
##    term                         estimate  std.error statistic  p.value
##    <chr>                           <dbl>      <dbl>     <dbl>    <dbl>
##  1 (Intercept)               -96.9       7.05         -13.7   6.92e-43
##  2 id                         -0.0000110 0.00000287    -3.84  1.23e- 4
##  3 host_is_superhost           1.12      1.14           0.984 3.25e- 1
##  4 host_total_listings_count   0.0651    0.0121         5.40  6.65e- 8
##  5 accommodates               21.5       0.407         52.8   0
##  6 bathrooms                  17.1       1.06          16.1   8.72e-58
##  7 bedrooms                   11.2       0.865         12.9   5.65e-38
##  8 beds                       -3.99      0.624         -6.38  1.74e-10
##  9 cleaning_fee                0.574     0.00997       57.6   0
## 10 guests_included             2.67      0.478          5.58  2.41e- 8
## # ... with 21 more rows
```

```
#library(car)
vif(model)
```

```
##                             id              host_is_superhost
##                       2.010195                       1.254679
##      host_total_listings_count                   accommodates
##                       2.049800                       3.167454
##                      bathrooms                       bedrooms
##                       1.195990                       2.104522
##                           beds                   cleaning_fee
##                       2.669752                       1.458843
##                guests_included                   extra_people
##                       1.776342                       1.155236
##                 minimum_nights                 maximum_nights
##                       1.052069                       1.000830
##                availability_30                availability_60
##                       7.458766                      32.455358
##                availability_90               availability_365
##                      20.117021                       2.052922
##              number_of_reviews          number_of_reviews_ltm
##                       3.914190                       4.224140
##           review_scores_rating         review_scores_accuracy
##                       4.289257                       2.772670
##      review_scores_cleanliness          review_scores_checkin
##                       2.224826                       2.094720
##    review_scores_communication         review_scores_location
##                       2.374831                       1.449576
##            review_scores_value               instant_bookable
##                       2.871729                       1.144593
## calculated_host_listings_count              reviews_per_month
##                       2.115325                       2.959689
##        listing_active_duration           host_active_duration
##                       2.662872                       1.594275
```

```
data.frame(Predictor = names(vif(model)), VIF = vif(model)) %>%
  ggplot(aes(x=VIF, y = reorder(Predictor, VIF), fill=VIF))+
  geom_col()+
  geom_vline(xintercept=5, color = 'gray', size = 1.5)+
  geom_vline(xintercept = 10, color = 'red', size = 1.5)+
  scale_fill_gradient(low = '#fff7bc', high = '#d95f0e')+
  scale_y_discrete(name = "Predictor")+
  scale_x_continuous(breaks = seq(5,30,5))+
  theme_classic()
```



##Write transformed data to csv file

```
write.csv(data, "clean_full_analysis_data.csv",row.names = F)
```

##Write selective features transformed data to csv file

```
data_trimmed = data[,c('price','zipcode','room_type','bedrooms','accommodates','neighbourhood_group_cle
ansed','availability_30','host_is_superhost','review_scores_rating','review_scores_location','TV','Elev
ator','cleaning_fee','property_type','minimum_nights','bathrooms')]

write.csv(data_trimmed, "clean_trimmed_analysis_data.csv",row.names = F)
```

Modelling