AirBnb Analysis in New York

Chaoqun Yin

12/3/2018

1 Abstract

2 Introduction

2.1 Background

Several days ago when I first came to New York City for traveling, I chose Airbnb as the platform I reserved the living place. As for why to choose Airbnb, it is because that shared economy industry are changing the whole world, that is, it provides premium vacation rentals from certified owners - boutique hotels, homes, rooms, and more. Airbnb has 5+ million unique listings worldwide.

However, I am also puzzled that how to choose a reasonable residence when facing thousands of residences in New York. Therefore, I will conduct a series of analysis to dig into the Airbnb residences to help me and other people that want to find a residence in Airbnb in New York make a decision to choose the proper living place.

2.2 Previous Work

3 Data Preparing and Modeling

3.1 Data Source

I got the data from the site tomslee.net/airbnb-data-collection-get-the-data that is public to everyone. The data contains much information about the residences on some specific days in New York. Here is the data description:

- oom_id: A unique number identifying an Airbnb listing.
- host_id: A unique number identifying an Airbnb host.
- room_type: One of “Entire home/apt”, “Private room”, or “Shared room”
- borough: A subregion of the city or search area for which the survey is carried out. The borough is taken from a shapefile of the city that is obtained independently of the Airbnb web site.
- neighborhood: As with borough: a subregion of the city or search area for which the survey is carried out.
- reviews: The number of reviews that a listing has received. Airbnb has said that 70% of visits end up with a review, so the number of reviews can be used to estimate the number of visits.
- overall_satisfaction: The average rating (out of five) that the listing has received from those visitors who left a review.
- accommodates: The number of guests a listing can accommodate.
- bedrooms: The number of bedrooms a listing offers.

- price: The price (in $US) for a night stay. In early surveys, there may be some values that were recorded by month.
- minstay: The minimum stay for a visit, as posted by the host.
- latitude and longitude: The latitude and longitude of the listing as posted on the Airbnb site: this may be off by a few hundred metres. I do not have a way to track individual listing locations with
- last_modified: the date and time that the values were read from the Airbnb web site.

3.2 Overall of the data

In this section, we will load the data and check the basic information of the data. Then we will use some visualization to show how the residences' features distribute. Based on the findings observing the visualization, we choose the proper models to conduct analysis.

Prepare the data

During this part, I load the data and it has rows and columns.

```
# read the data
for(i in 1:12){
  dname = paste("ny2016_", i, sep = "")
  dfile = paste("airbnb_new_york_2016-", i, ".csv", sep = "")
  assign(dname, read.csv(dfile, header = TRUE))
  next
}

# add time to each dataset
ny2016_1$Date <- as.Date("2016-1-1")
ny2016_2$Date <- as.Date("2016-2-1")
ny2016_3$Date <- as.Date("2016-3-1")
ny2016_4$Date <- as.Date("2016-4-1")
ny2016_5$Date <- as.Date("2016-5-1")
ny2016_6$Date <- as.Date("2016-6-1")
ny2016_7$Date <- as.Date("2016-7-1")
ny2016_8$Date <- as.Date("2016-8-1")
ny2016_9$Date <- as.Date("2016-9-1")
ny2016_10$Date <- as.Date("2016-10-1")
ny2016_11$Date <- as.Date("2016-11-1")
ny2016_12$Date <- as.Date("2016-12-1")

# merge the data
ny <- rbind(ny2016_1, ny2016_2, ny2016_3, ny2016_4, ny2016_5, ny2016_6,
            ny2016_7,ny2016_8, ny2016_9, ny2016_10, ny2016_11, ny2016_12)
# add one column to record the price per person
ny$pricepp <- ny$price/ny$accommodates

ny %>%
  filter(!is.na(host_id)) -> ny
```

Data Structure

During this part, we can check the data structure and data type of each variables using `str()`.

```
str(ny)
```

```
## 'data.frame':    430783 obs. of  16 variables:
##  $ room_id             : int  3557831 7780413 1788989 9680791 8204392 7455635 1742471 2045660 960380
##  $ host_id             : num  17911953 4887492 7607092 3201897 11268108 ...
##  $ room_type           : Factor w/ 4 levels "","Entire home/apt",..: 3 4 2 3 3 2 2 3 3 2 ...
##  $ borough             : Factor w/ 5 levels "Bronx","Brooklyn",..: 3 3 3 2 2 3 4 2 2 2 ...
##  $ neighborhood        : Factor w/ 240 levels "Allerton","Arden Heights",..: 65 199 180 162 217 34 19
##  $ reviews             : int  43 4 9 1 0 11 17 7 1 2 ...
##  $ overall_satisfaction: num  4.5 5 5 4 NA 4.5 5 4.5 3 5 ...
##  $ accommodates        : num  2 4 4 2 2 2 2 2 1 4 ...
##  $ bedrooms            : num  1 1 1 1 1 0 1 1 1 2 ...
##  $ price               : num  115 69 225 75 500 110 120 55 50 350 ...
##  $ minstay             : num  2 2 3 3 1 2 3 1 1 2 ...
##  $ latitude            : num  40.7 40.7 40.7 40.7 40.7 ...
##  $ longitude           : num  -74 -74 -74 -74 -73.9 ...
##  $ last_modified       : Factor w/ 430798 levels "2016-01-21 07:02:47.986871",..: 36003 36001 35999 3
##  $ Date                : Date, format: "2016-01-01" "2016-01-01" ...
##  $ pricepp             : num  57.5 17.2 56.2 37.5 250 ...
```

Data overview

During this part, we use `summary()` to get the overview of the dataset.

```
summary(ny)
```

```
##     room_id            host_id                    room_type
##  Min.   :     105   Min.   :       43                   :     0
##  1st Qu.: 3907848   1st Qu.:  4071690   Entire home/apt:220697
##  Median : 7708956   Median : 13972066   Private room   :197044
##  Mean   : 7488534   Mean   : 22168318   Shared room    : 13042
##  3rd Qu.:10686406   3rd Qu.: 35322092
##  Max.   :16544971   Max.   :108817370
##
##           borough                   neighborhood       reviews
##  Bronx         :  6342   Williamsburg      : 38146   Min.   :  0.00
##  Brooklyn      :174455   Bedford-Stuyvesant: 27952   1st Qu.:  1.00
##  Manhattan     :209213   Harlem            : 25489   Median :  3.00
##  Queens        : 38367   Upper West Side   : 20398   Mean   : 13.75
##  Staten Island:  2406   East Village      : 19989   3rd Qu.: 15.00
##                         Bushwick          : 19643   Max.   :387.00
##                         (Other)           :279166
##  overall_satisfaction  accommodates       bedrooms         price
##  Min.   :0.00         Min.   : 1.000   Min.   : 0.00   Min.   :   10.0
##  1st Qu.:4.50         1st Qu.: 2.000   1st Qu.: 1.00   1st Qu.:   70.0
##  Median :4.50         Median : 2.000   Median : 1.00   Median :  110.0
##  Mean   :4.37         Mean   : 2.738   Mean   : 1.14   Mean   :  147.4
##  3rd Qu.:5.00         3rd Qu.: 4.000   3rd Qu.: 1.00   3rd Qu.:  175.0
##  Max.   :5.00         Max.   :16.000   Max.   :10.00   Max.   :10000.0
##  NA's   :154955       NA's   :8417     NA's   :38053
##     minstay         latitude       longitude
##  Min.   :   1    Min.   :40.50   Min.   :-74.24
##  1st Qu.:   1    1st Qu.:40.69   1st Qu.:-73.98
##  Median :   2    Median :40.72   Median :-73.96
```

```
##  Mean   :   3    Mean   :40.73   Mean   :-73.96
##  3rd Qu.:   3    3rd Qu.:40.76   3rd Qu.:-73.94
##  Max.   :1250    Max.   :40.91   Max.   :-73.71
##  NA's   :47723
##                     last_modified         Date
##  2016-01-21 07:02:47.986871:     1   Min.   :2016-01-01
##  2016-01-21 07:02:53.241423:     1   1st Qu.:2016-04-01
##  2016-01-21 07:03:21.182491:     1   Median :2016-07-01
##  2016-01-21 07:03:23.417401:     1   Mean   :2016-06-19
##  2016-01-21 07:03:25.318650:     1   3rd Qu.:2016-10-01
##  2016-01-21 07:03:32.555722:     1   Max.   :2016-12-01
##  (Other)                   :430777
##     pricepp
##  Min.   :   0.625
##  1st Qu.:  32.500
##  Median :  47.500
##  Mean   :  57.293
##  3rd Qu.:  67.500
##  Max.   :9998.000
##  NA's   :8417
```

We can see from the output that these variables may have correlation with the price per person of each listing: **room_type, borough, neighborhood, reviews, overall_satisfaction, accommodates, bedrooms**. We will explore the variables in the exploratory data analysis and based the outcome to help choose proper model to analyze the price per person.

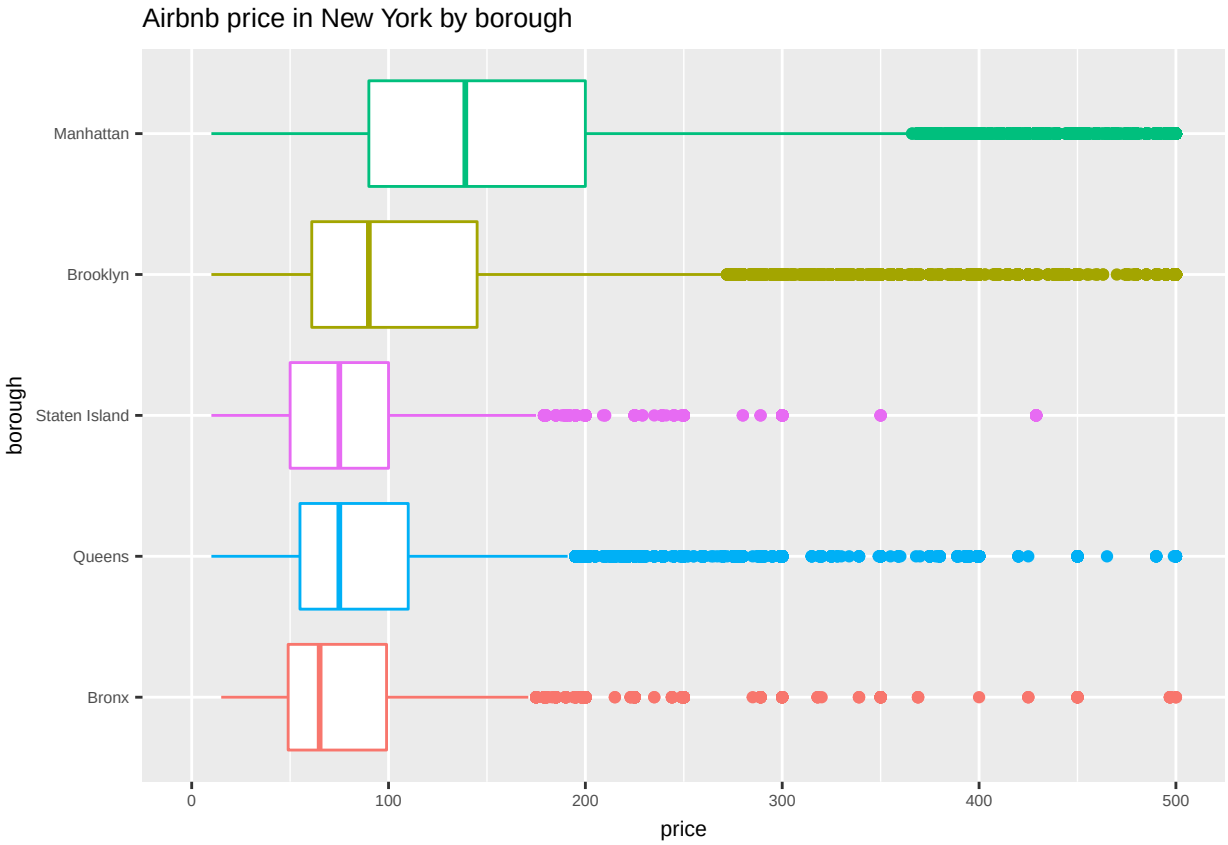
3.3 Exploratory Data Analysis

3.3.1 Location Features

The boxplot of borough and the prices

```
ggplot(ny, aes(x=reorder(borough, price, median),y = price,group = borough,
                   colour = borough)) +
  geom_boxplot() +
  ylim(0,500) +
  theme(text=element_text(size = 8),legend.position = "none") +
  labs(title="Airbnb price in New York by borough", x="borough") +
  coord_flip()
```

```
## Warning: Removed 7705 rows containing non-finite values (stat_boxplot).
```

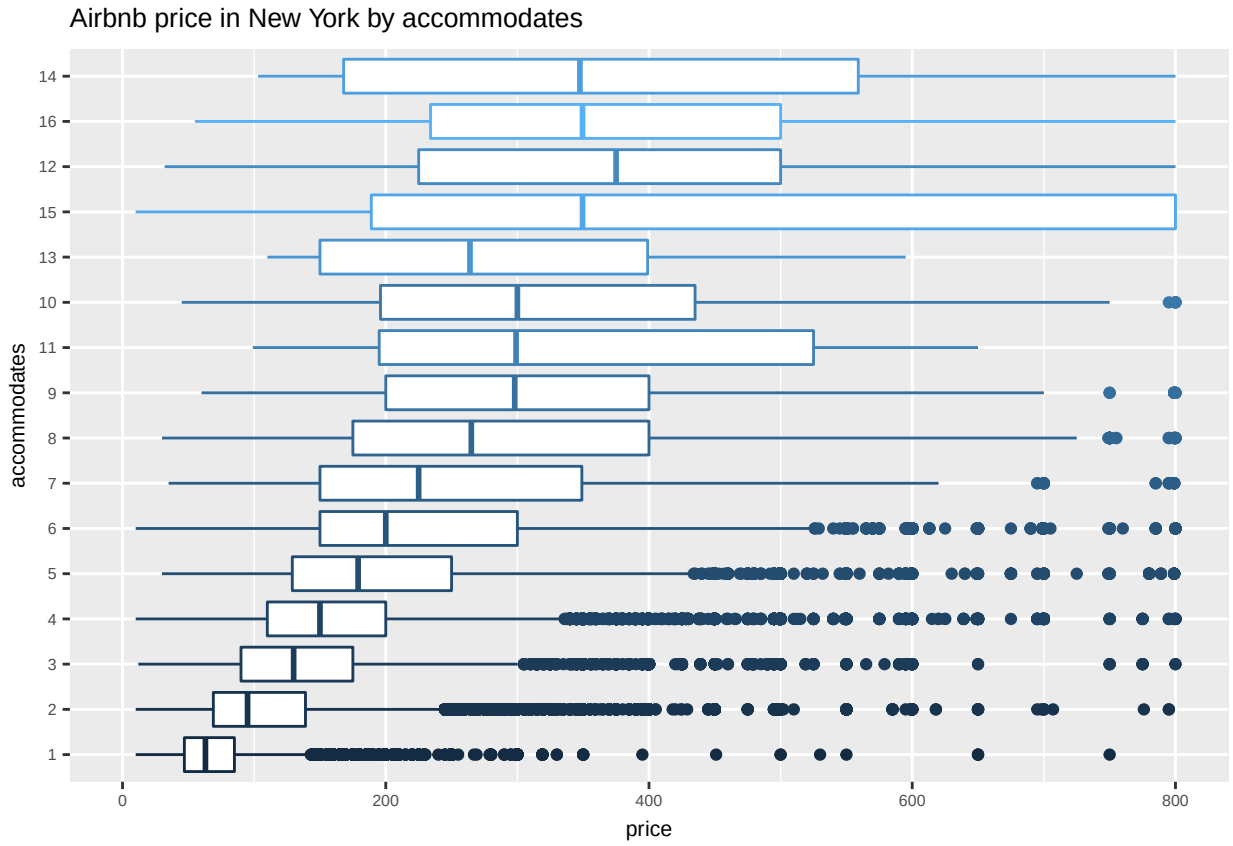


We can see from the boxplot that the median price of each borough has *obvious difference*. Manhattan borough has the highest median price rather than other borough while Bronx has the lowest median price. So we can infer from the plot that *borough* is a feature that influences the price of one property.

Boxplot of accommodates and the prices

```
ny %>%
  na.omit(ny$accommodates) %>%
ggplot(aes(x=reorder(accommodates, price, median),y = price,group = accommodates,
                   colour = accommodates)) +
  geom_boxplot() +
  ylim(0,800) +
  theme(text=element_text(size = 8),legend.position = "none") +
  labs(title="Airbnb price in New York by accommodates", x="accommodates") +
  coord_flip()
```

```
## Warning: Removed 541 rows containing non-finite values (stat_boxplot).
```



Boxplot of the room types and the price

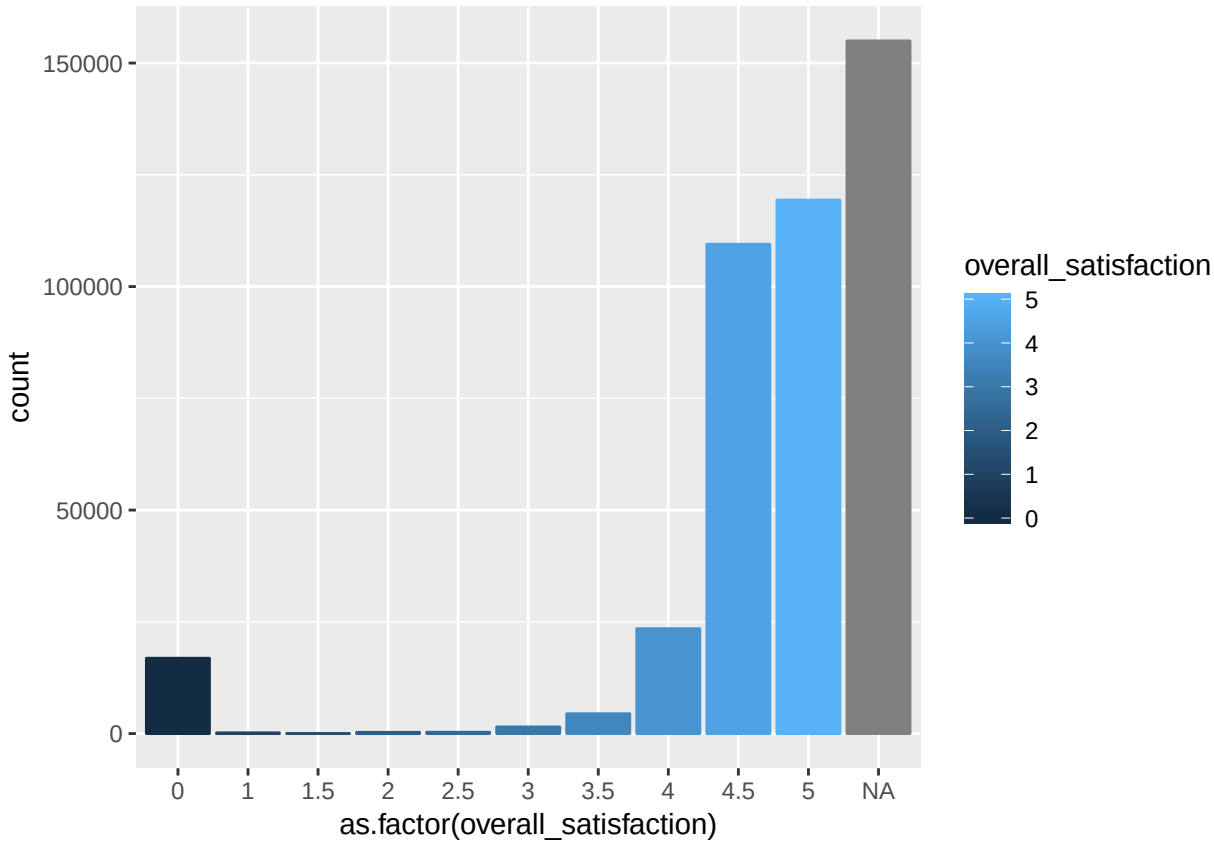
```
ny %>%
  na.omit(ny$room_type) %>%
ggplot(aes(x=reorder(room_type, price, median),y = price,group = room_type,
                    colour = room_type)) +
  geom_boxplot() +
  ylim(0,500) +
  theme(text=element_text(size = 8),legend.position = "none") +
  labs(title="Airbnb price in New York by room type", x="room_type") +
  coord_flip()
```

```
## Warning: Removed 1985 rows containing non-finite values (stat_boxplot).
```

Diagram of reviews

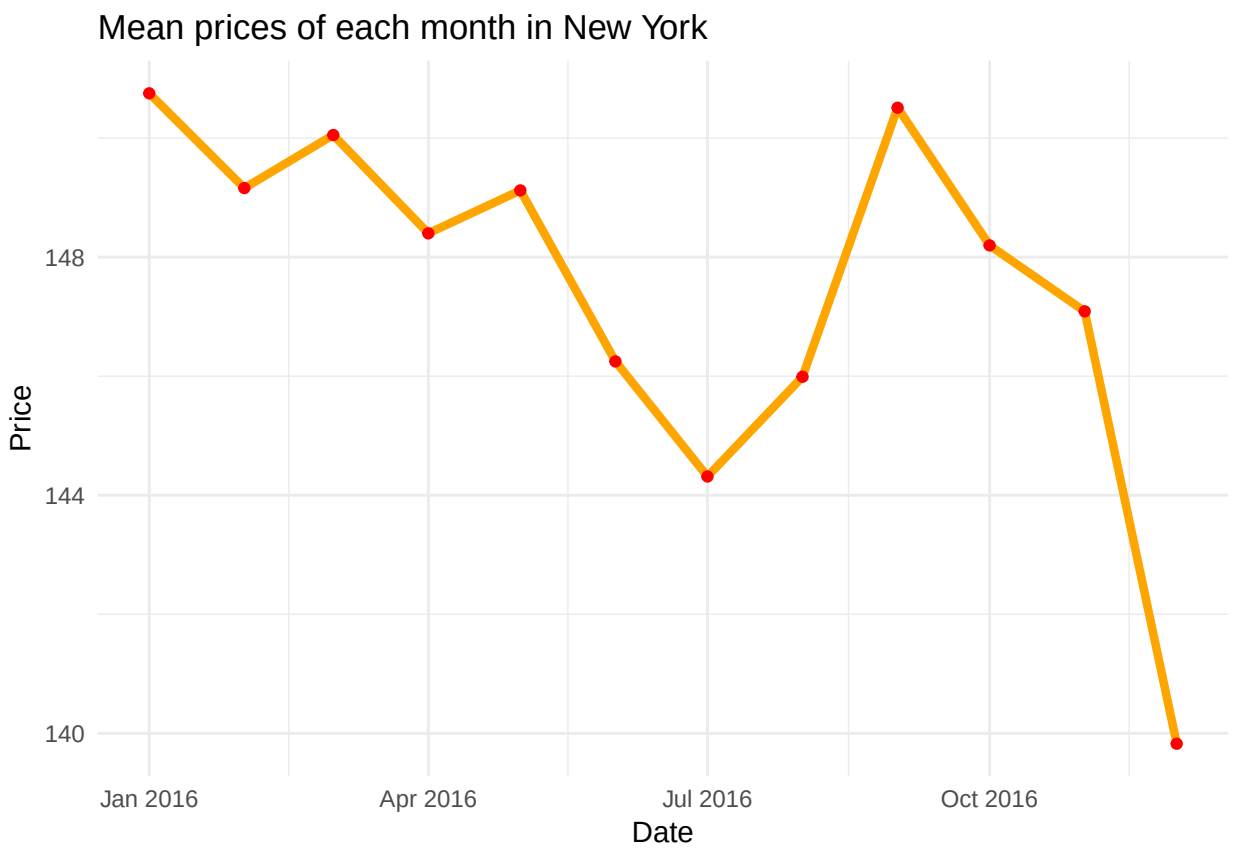
```
ggplot(data = ny, aes(x = as.factor(overall_satisfaction), col = overall_satisfaction, fill = overall_sa
  geom_histogram(stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



Mean prices of each month in New York

```
#Calculate the mean of home values
MHV <- aggregate(price ~ Date, ny, mean)

# Plot it
#Calculate the mean of home values
MHV <- aggregate(price ~ Date, ny, mean)
# Plot it
ggplot(data = MHV, aes(x = Date, y = price)) +
  geom_line(col = "orange", size = 1.5) +
  geom_point(col = "red") +
  labs(y = "Price", title = "Mean prices of each month in New York")+
  theme_minimal()
```



3.4 Model Used

The multilevel model is chosen to analyze the prices.

- At the very beginning, I check whether the month is a factor influencing the prices.

```
# create the subset for modeling
set.seed(2018)
ny %>%
  dplyr::select(Date, price, room_type, borough, neighborhood, accommodates, bedrooms) %>%
  na.omit(accommodates, bedrooms, room_type, neighborhood, borough) %>%
  sample_n(0.1 * nrow(ny))-> ny.m

# Transfer the type of the Date
#ny$Date = factor(format(Date, format = "%B"), levels = month.name)

# Fit linear regression
fit.0 <- lm(data = ny.m, price ~ Date)
summary(fit.0)
```

```
##
## Call:
## lm(formula = price ~ Date, data = ny.m)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -133.4  -71.9  -36.0   28.1 9857.8
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.540623 142.013105  -0.173    0.863
## Date          0.009838   0.008373   1.175    0.240
##
## Residual standard error: 176.4 on 43076 degrees of freedom
## Multiple R-squared:  3.205e-05,  Adjusted R-squared:  8.832e-06
## F-statistic:  1.38 on 1 and 43076 DF,  p-value: 0.24
```

- Regress **price** on **accommodates and bedrooms** and treat all other intercepts as random.

```
fit.1 <- lmer(data = ny.m, price ~ accommodates + bedrooms + (1 | Date) + (1 | room_type) + (1 | boroug
display(fit.1)
```

```
## lmer(formula = price ~ accommodates + bedrooms + (1 | Date) +
##     (1 | room_type) + (1 | borough), data = ny.m, REML = FALSE)
##              coef.est coef.se
## (Intercept)   2.32    25.93
## accommodates 16.95     0.66
## bedrooms     50.33     1.55
##
## Error terms:
##  Groups    Name        Std.Dev.
##  Date      (Intercept)   3.15
##  borough   (Intercept)  28.28
##  room_type (Intercept)  38.79
##  Residual              157.97
## ---
## number of obs: 43078, groups: Date, 11; borough, 5; room_type, 3
## AIC = 558466, DIC = 558451.7
## deviance = 558451.7
```

- Based on model 1, include a between-group correlation between accommodates and borough.

```
fit.2 <- lmer(data = ny.m, price ~ accommodates + bedrooms + + (1 | Date) + (1 | room_type) +
                (1 + accommodates | borough), REML = FALSE)
display(fit.2)
```

```
## lmer(formula = price ~ accommodates + bedrooms + +(1 | Date) +
##     (1 | room_type) + (1 + accommodates | borough), data = ny.m,
##     REML = FALSE)
##              coef.est coef.se
## (Intercept)  10.37    27.33
## accommodates 13.71     6.66
## bedrooms     50.74     1.55
##
## Error terms:
##  Groups    Name         Std.Dev. Corr
##  Date      (Intercept)    3.16
##  borough   (Intercept)   34.55
##            accommodates  14.57   -0.72
##  room_type (Intercept)   38.18
##  Residual               156.97
## ---
## number of obs: 43078, groups: Date, 11; borough, 5; room_type, 3
```

```
## AIC = 557944, DIC = 557925.6
## deviance = 557925.6
```

- Then, I regress price on accomodates and bedrooms while including one between-group and other random effect.

```
fit.3 <- lmer(data = ny.m, price ~ accommodates + bedrooms + (1 | room_type) + (1 | Date) +
                (1 + accommodates + bedrooms | borough), REML = FALSE)
  display(fit.3)
```

```
## lmer(formula = price ~ accommodates + bedrooms + (1 | room_type) +
##     (1 | Date) + (1 + accommodates + bedrooms | borough), data = ny.m,
##     REML = FALSE)
##             coef.est coef.se
## (Intercept)   4.73    29.16
## accommodates  8.75     5.73
## bedrooms     67.77    14.15
##
## Error terms:
##  Groups    Name         Std.Dev. Corr
##  Date      (Intercept)    3.14
##  borough   (Intercept)   40.92
##            accommodates  12.32   -0.24
##            bedrooms      30.11   -0.83 -0.10
##  room_type (Intercept)   38.32
##  Residual                156.76
## ---
## number of obs: 43078, groups: Date, 11; borough, 5; room_type, 3
## AIC = 557847, DIC = 557822.8
## deviance = 557822.8
```

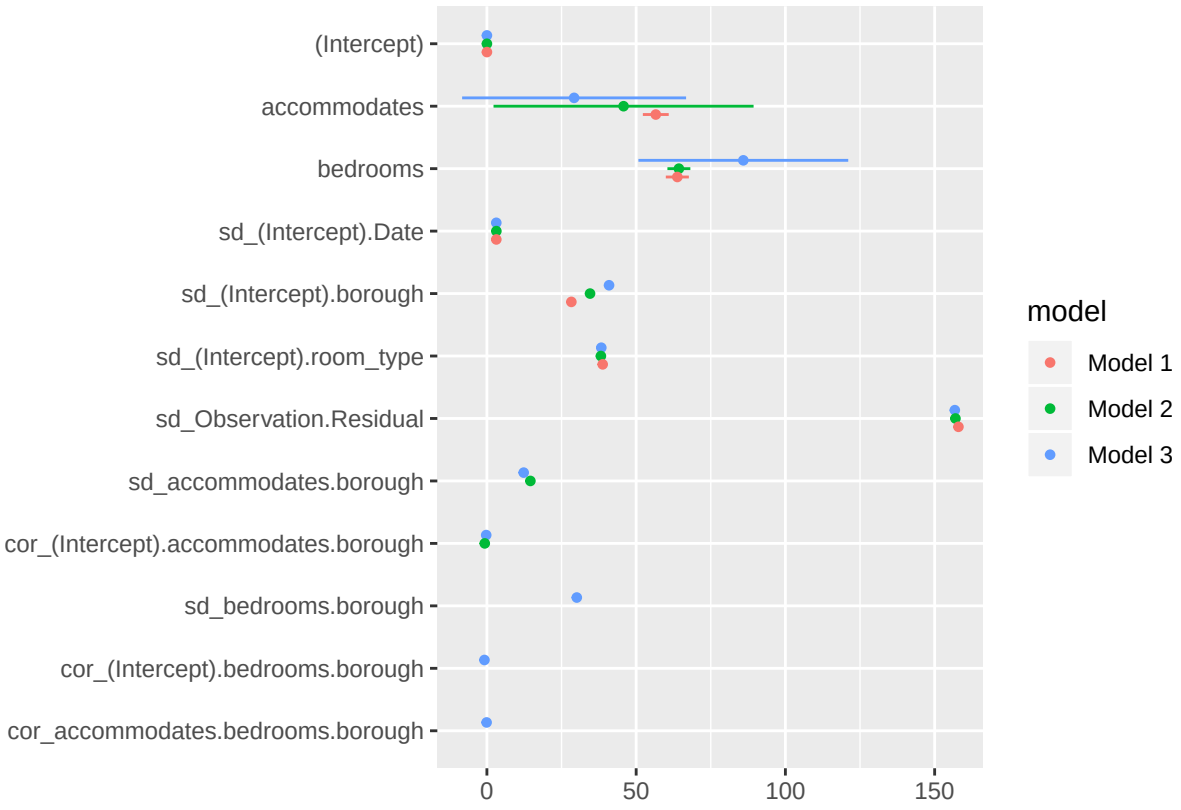
4 Result

4.1 Model Choice

Intercept plot

```
dwplot(list(fit.1, fit.2, fit.3), show_intercept = TRUE)
```

```
## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```



ANOVA

```
anova(fit.1, fit.2, fit.3)
```

```
## Data: ny.m
## Models:
## fit.1: price ~ accommodates + bedrooms + (1 | Date) + (1 | room_type) +
## fit.1:     (1 | borough)
## fit.2: price ~ accommodates + bedrooms + +(1 | Date) + (1 | room_type) +
## fit.2:     (1 + accommodates | borough)
## fit.3: price ~ accommodates + bedrooms + (1 | room_type) + (1 | Date) +
## fit.3:     (1 + accommodates + bedrooms | borough)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit.1  7 558466 558526 -279226   558452
## fit.2  9 557944 558022 -278963   557926 526.06      2  < 2.2e-16 ***
## fit.3 12 557847 557951 -278911   557823 102.86      3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

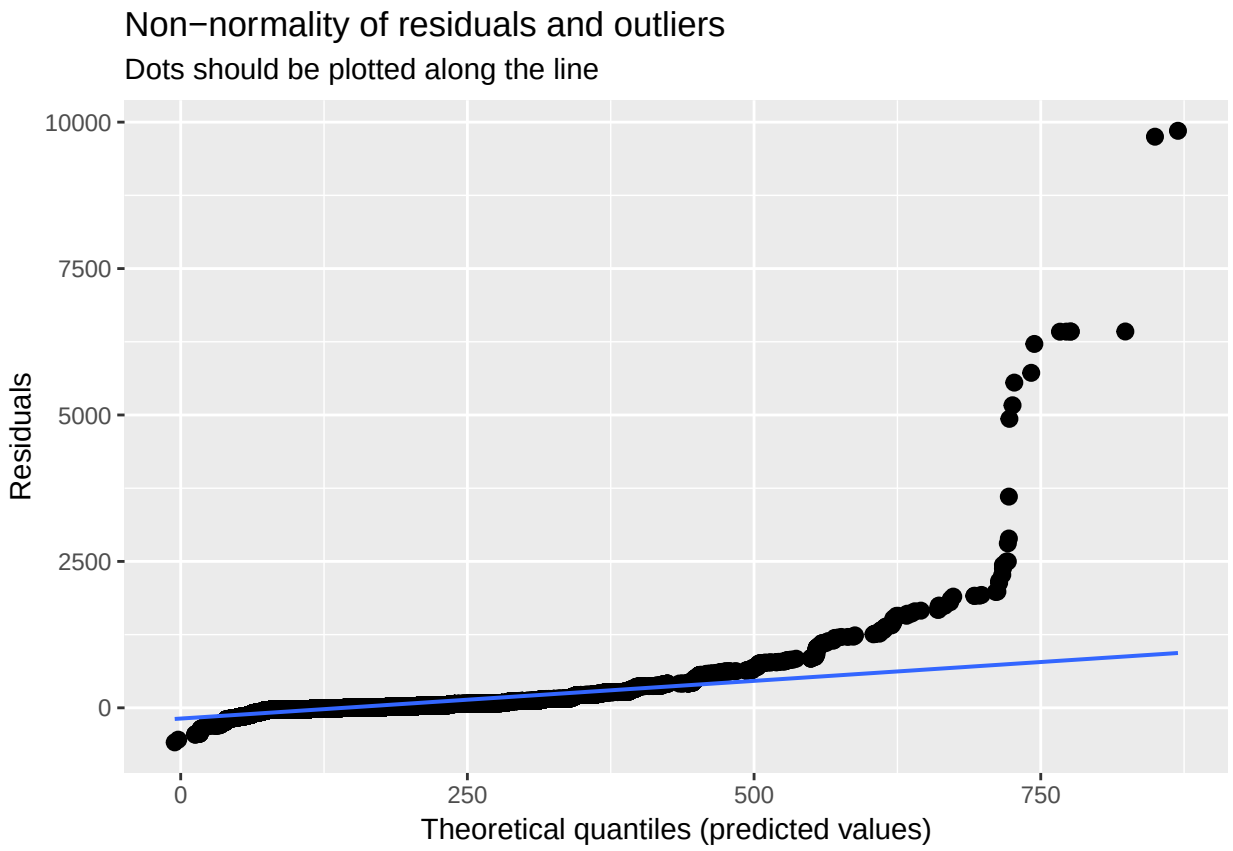
Above the

4.2 Interpretation

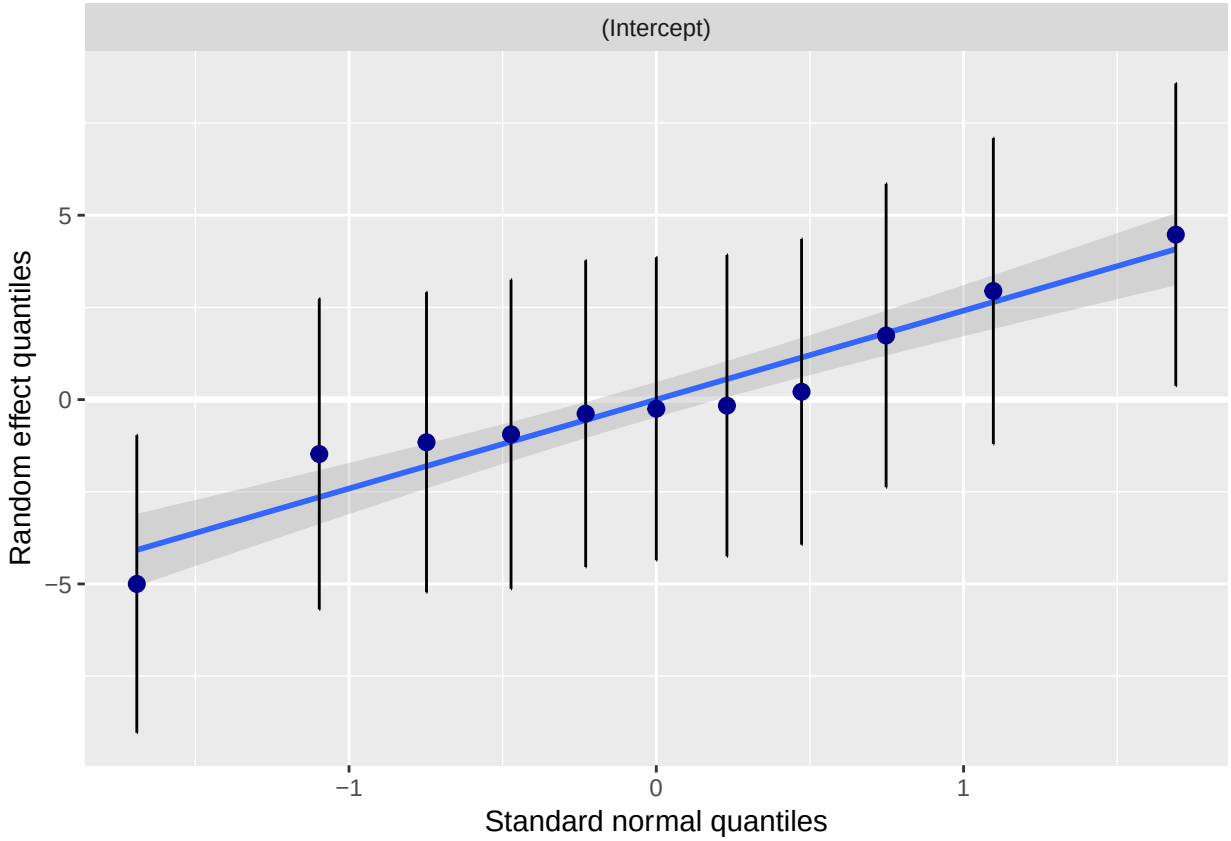
4.3 Model Check

```
library(sjPlot)
plot_model(fit.2, type = "diag", show.values = TRUE, value.offset = .3)
```

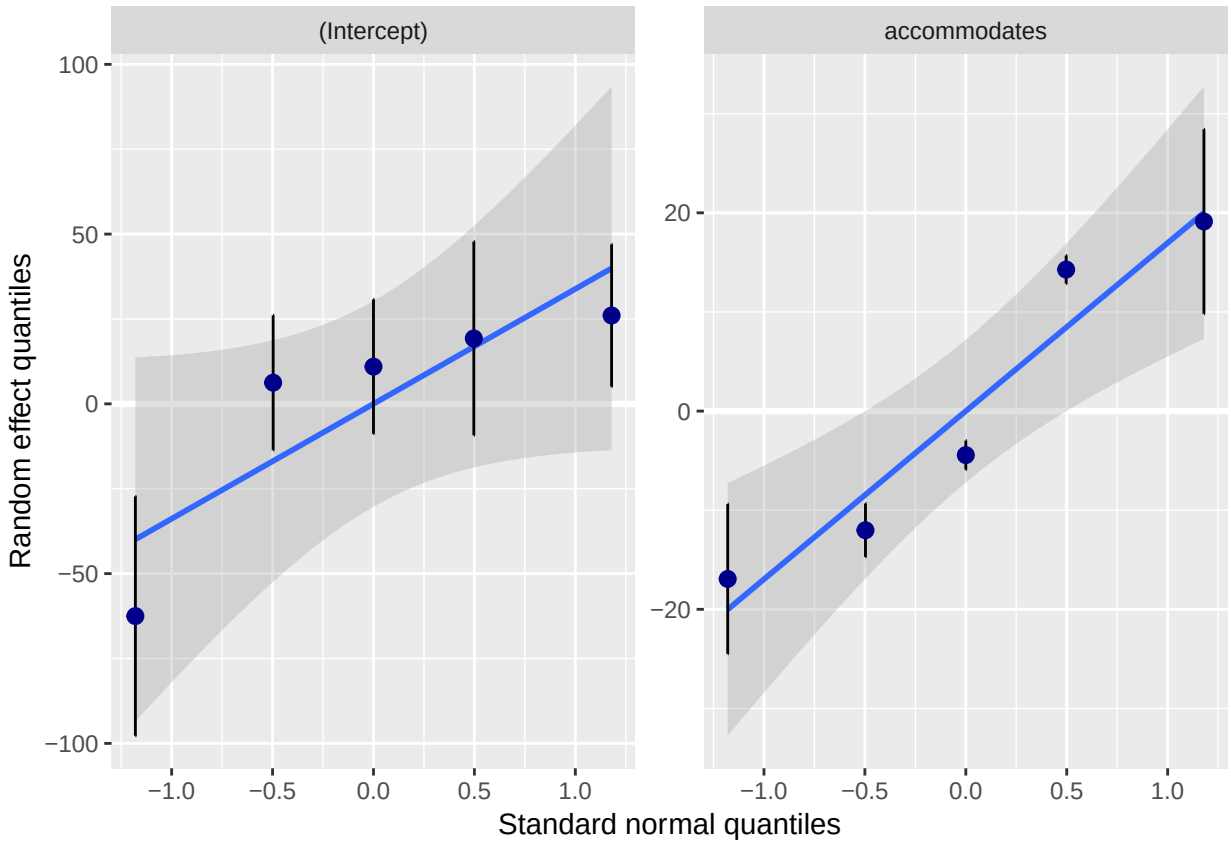
```
## [[1]]
```



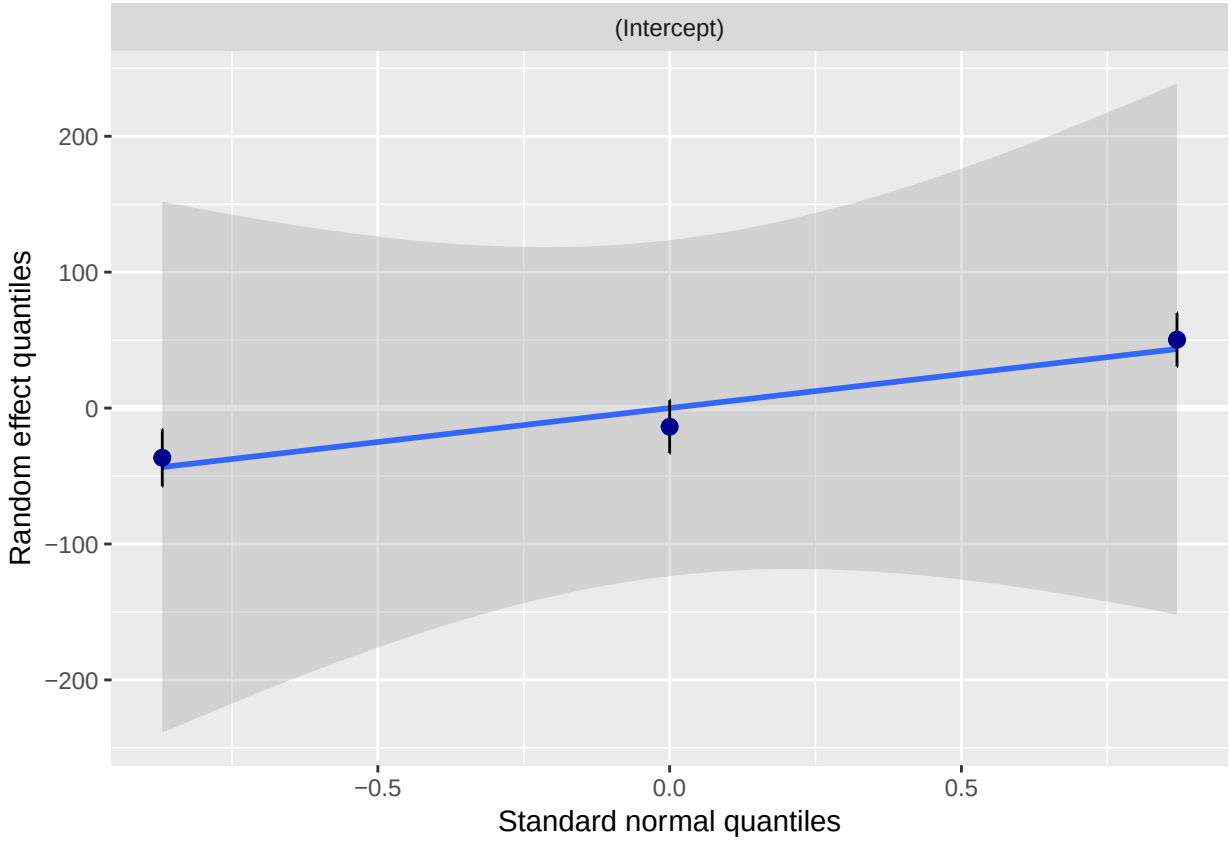

```
##
## [[2]]
## [[2]]$Date
```


(Intercept)
Random effect quantiles
5
0
−5
−1
0
1
Standard normal quantiles

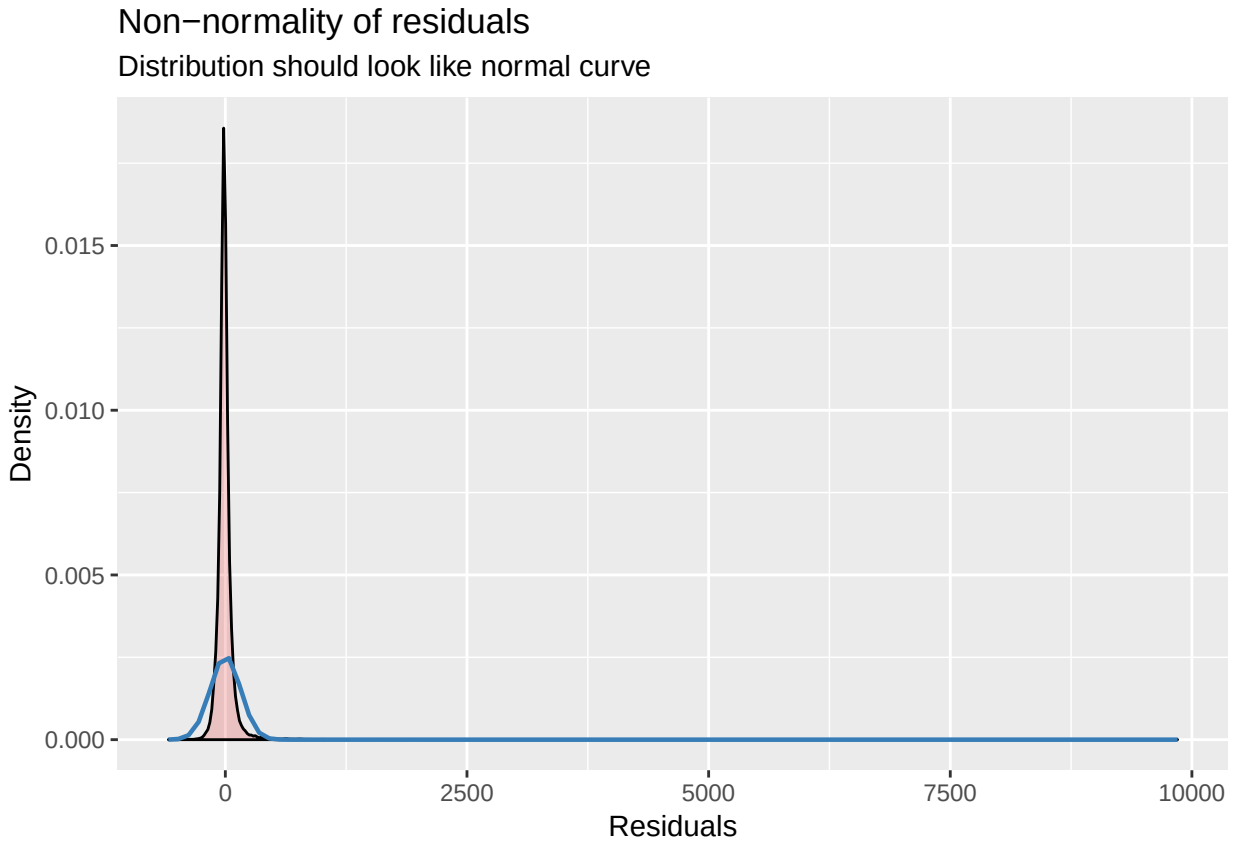
```
##
## [[2]]$borough
```


(Intercept)
accommodates
Random effect quantiles
Standard normal quantiles
100
50
0
−50
−100
20
0
−20
−1.0
−0.5
0.0
0.5
1.0

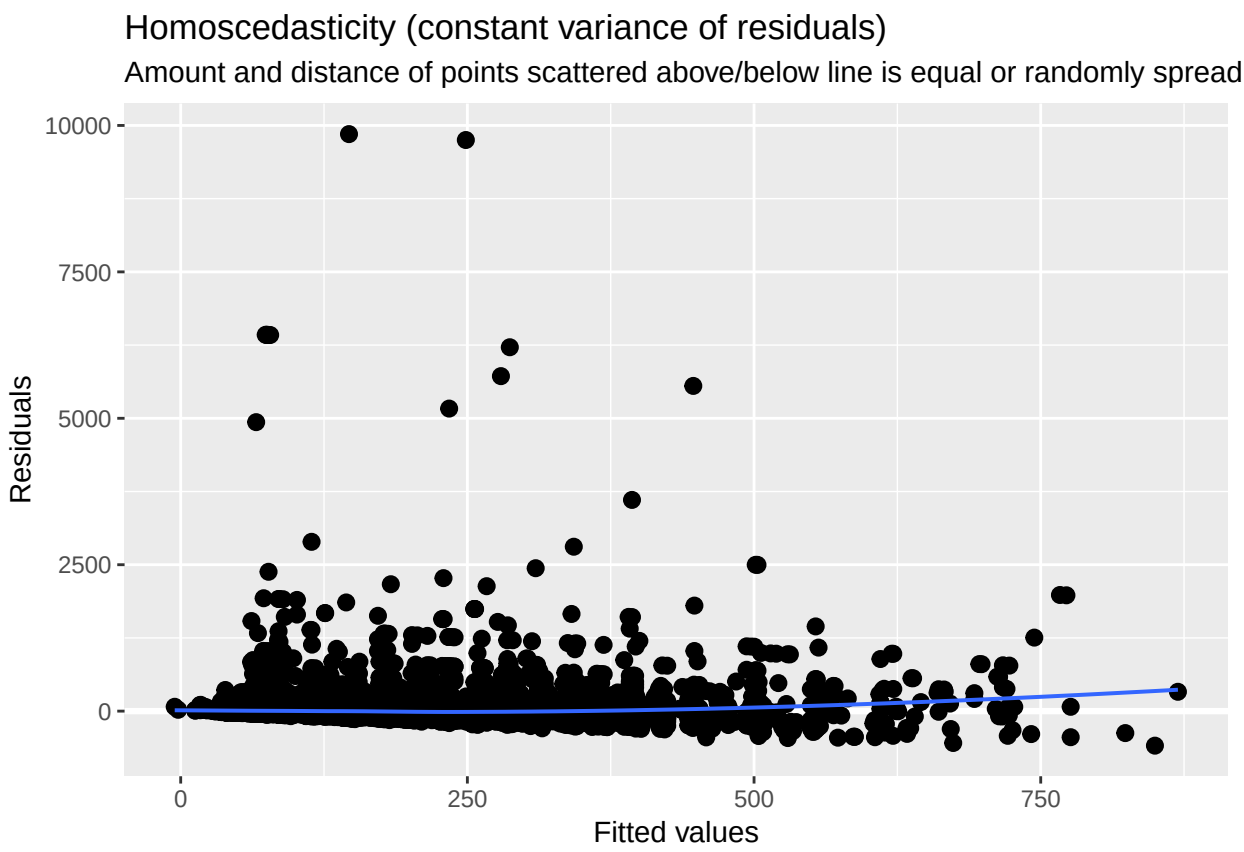
```
##
## [[2]]$room_type
```


(Intercept)
200
100
0
−100
−200
Random effect quantiles
−0.5
0.0
0.5
Standard normal quantiles

```
##
##
## [[3]]
```

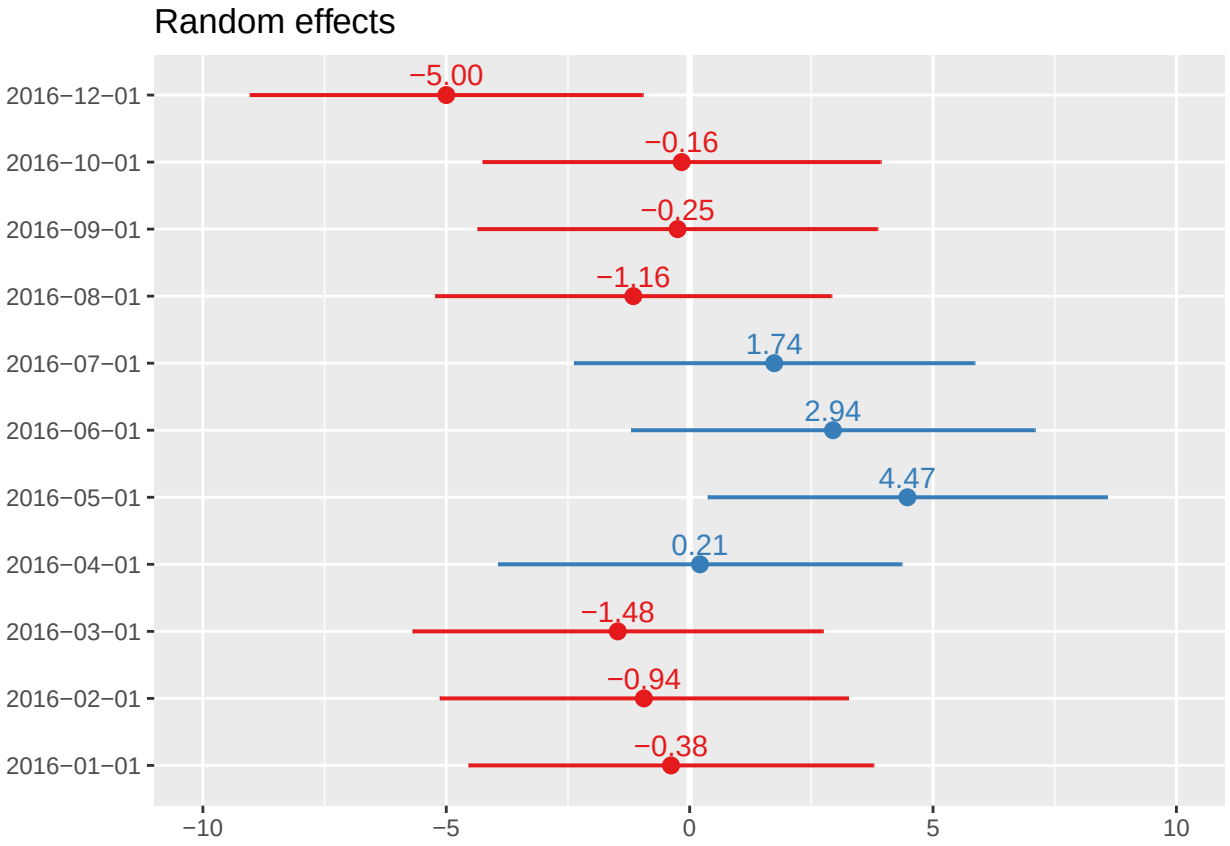

Non−normality of residuals
Distribution should look like normal curve
Density
0.015
0.010
0.005
0.000
0
2500
5000
7500
10000
Residuals

```
##
## [[4]]
```

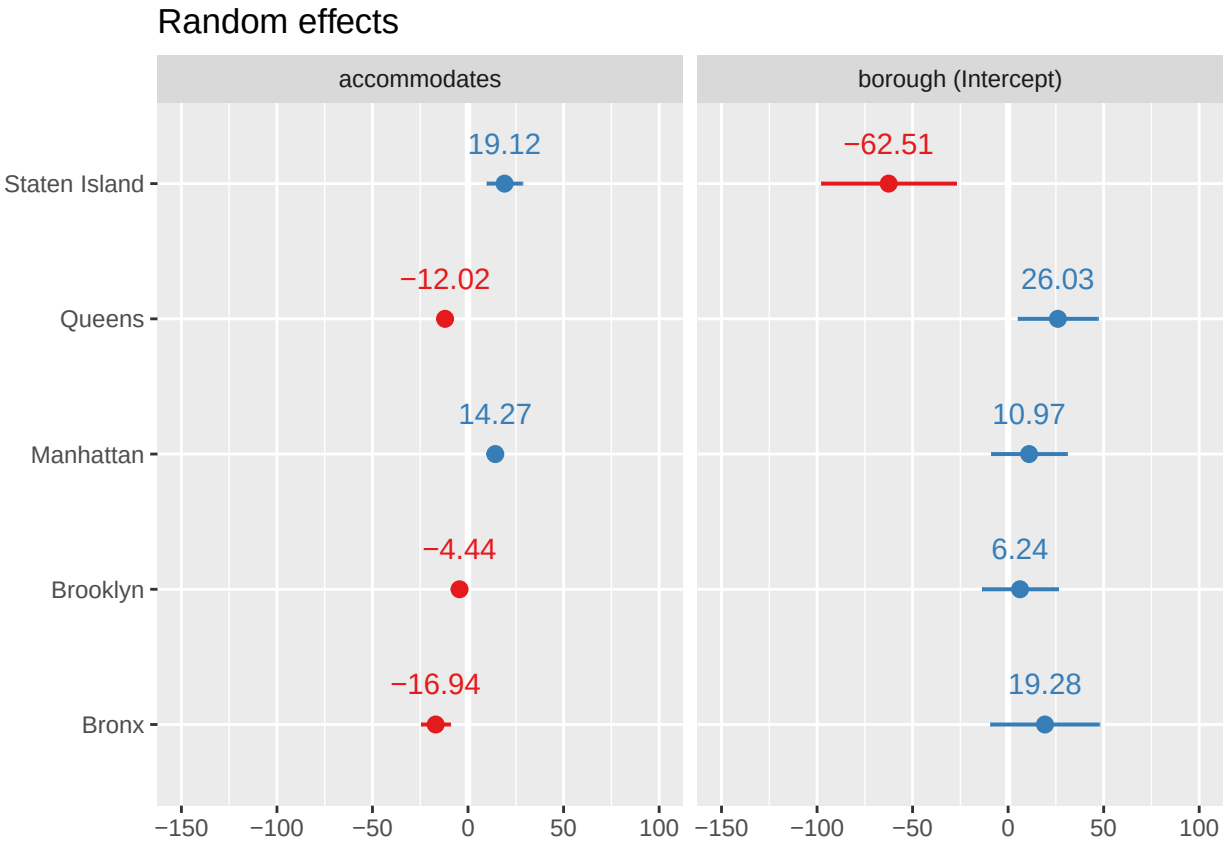



```
plot_model(fit.2, type = "re", show.values = TRUE, value.offset = .3)
```

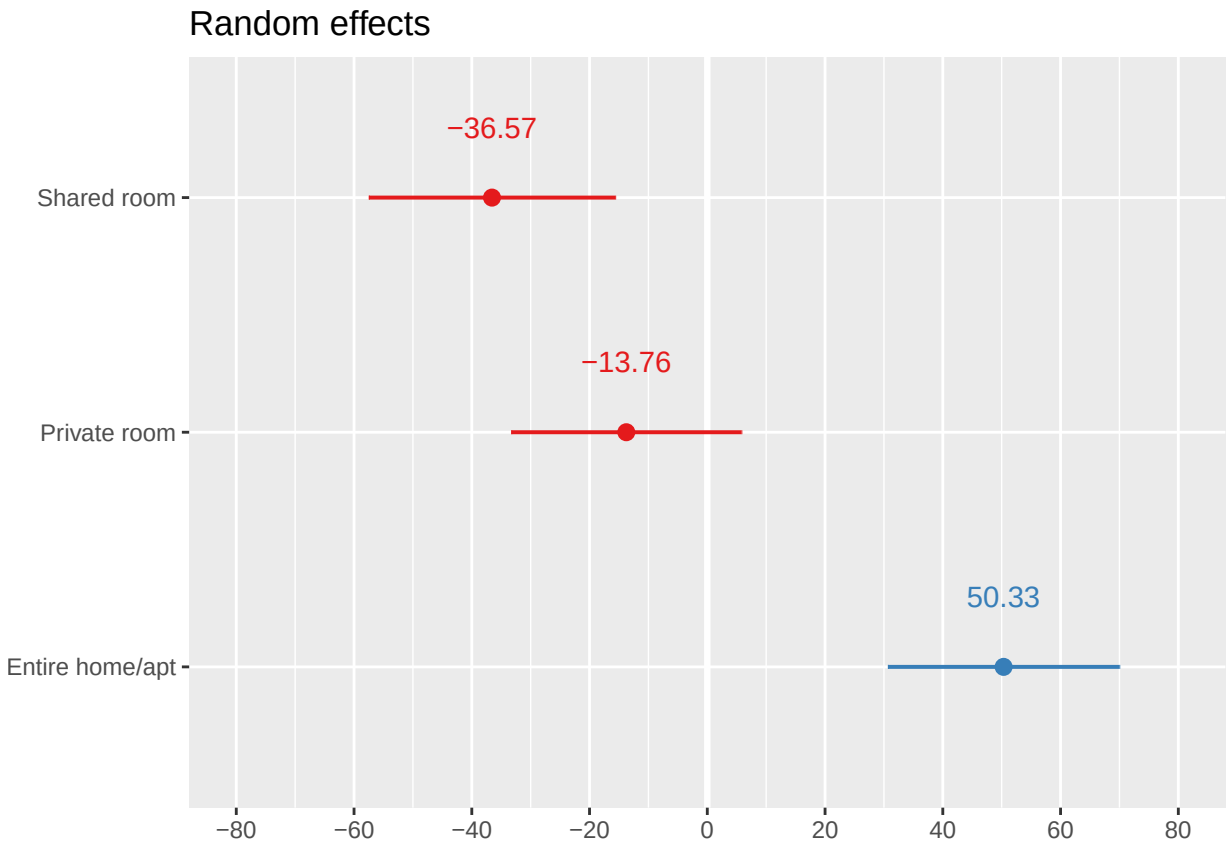
```
## [[1]]
```


Random effects
2016−12−01
2016−10−01
2016−09−01
2016−08−01
2016−07−01
2016−06−01
2016−05−01
2016−04−01
2016−03−01
2016−02−01
2016−01−01
−5.00
−0.16
−0.25
−1.16
1.74
2.94
4.47
0.21
−1.48
−0.94
−0.38
−10
−5
0
5
10

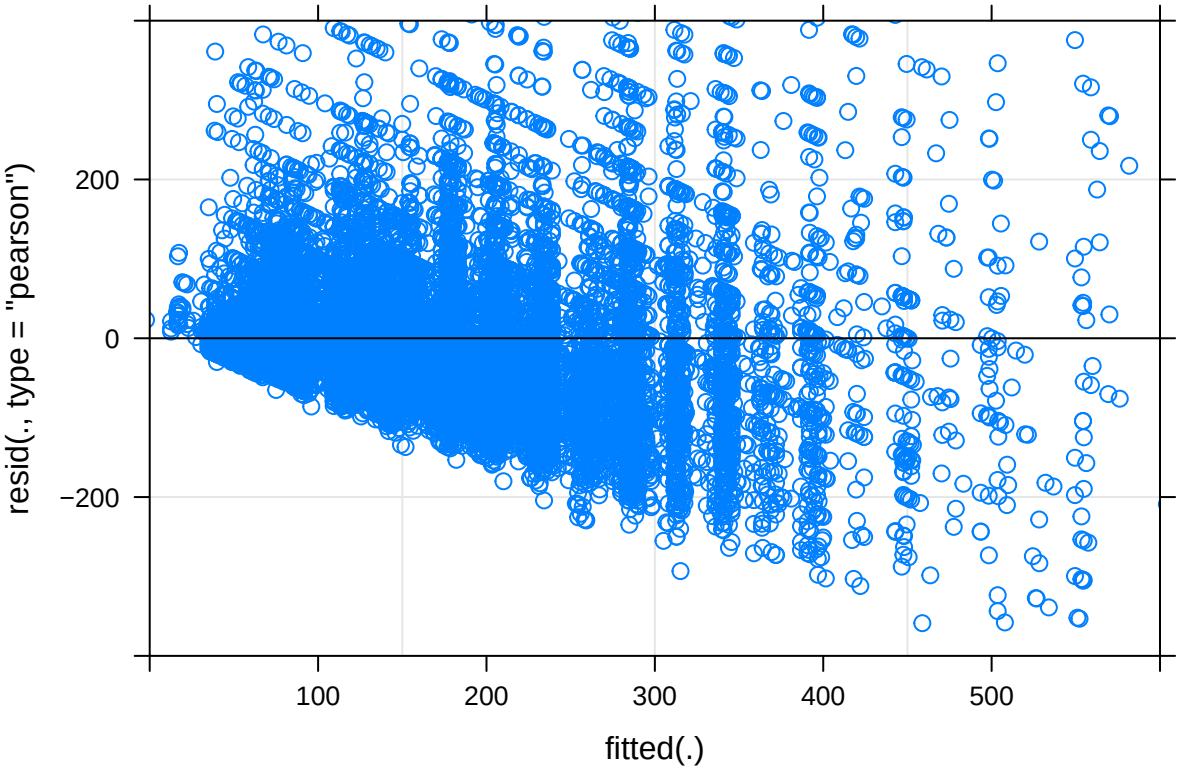
```
##
## [[2]]
```


Random effects
accommodates
borough (Intercept)
Staten Island
Queens
Manhattan
Brooklyn
Bronx
19.12
−12.02
14.27
−4.44
−16.94
−62.51
26.03
10.97
6.24
19.28
−150
−100
−50
0
50
100

```
##
## [[3]]
```




```
plot(fit.2, ylim = c(-400,400), xlim = c(0,600))
```



5 Discussion

5.1 Implication

5.2 Limitation

5.3 Future Direction

6 Acknowledgement

7 Reference

8 Appendix

The boxplot of neighborhood and the prices

It is easy to find that the *neighborhood* variable has 240 levels. Therefore, we choose the top 20 cost neighborhood to display.

```
ny %>%
  ggplot(aes(x = reorder(neighborhood,price, median),y = price,group = neighborhood,
                 colour = neighborhood)) +
  geom_boxplot() +
  ylim(0,500) +
  theme(text=element_text(size = 8),legend.position = "none") +
```

```
  labs(title="Airbnb price in New York by neighborhood", x="neighborhood") +
  coord_flip()
```

```
## Warning: Removed 7705 rows containing non-finite values (stat_boxplot).
```