★★★★★

Paris Neighborhoods Data Analytics

Fu Chen, Liuyi Chen, Qiao Xu, Ruilin Song, Tiange Chang, Wanxin Zhang

Contents

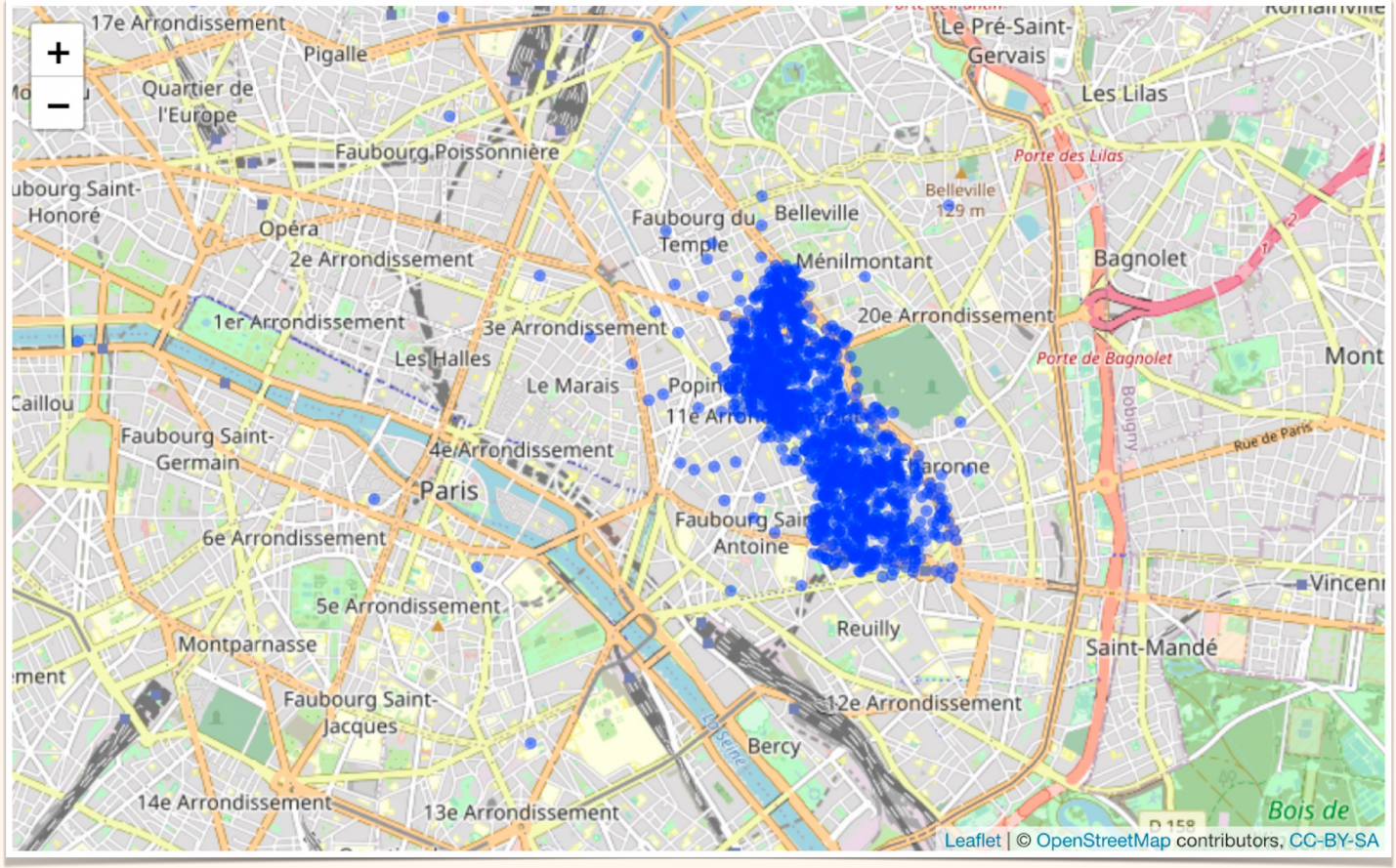
1

Introduction

Background

The 11th arrondissement of Paris is one of the most densely populated urban districts of any European city.
It is a varied and engaging area.

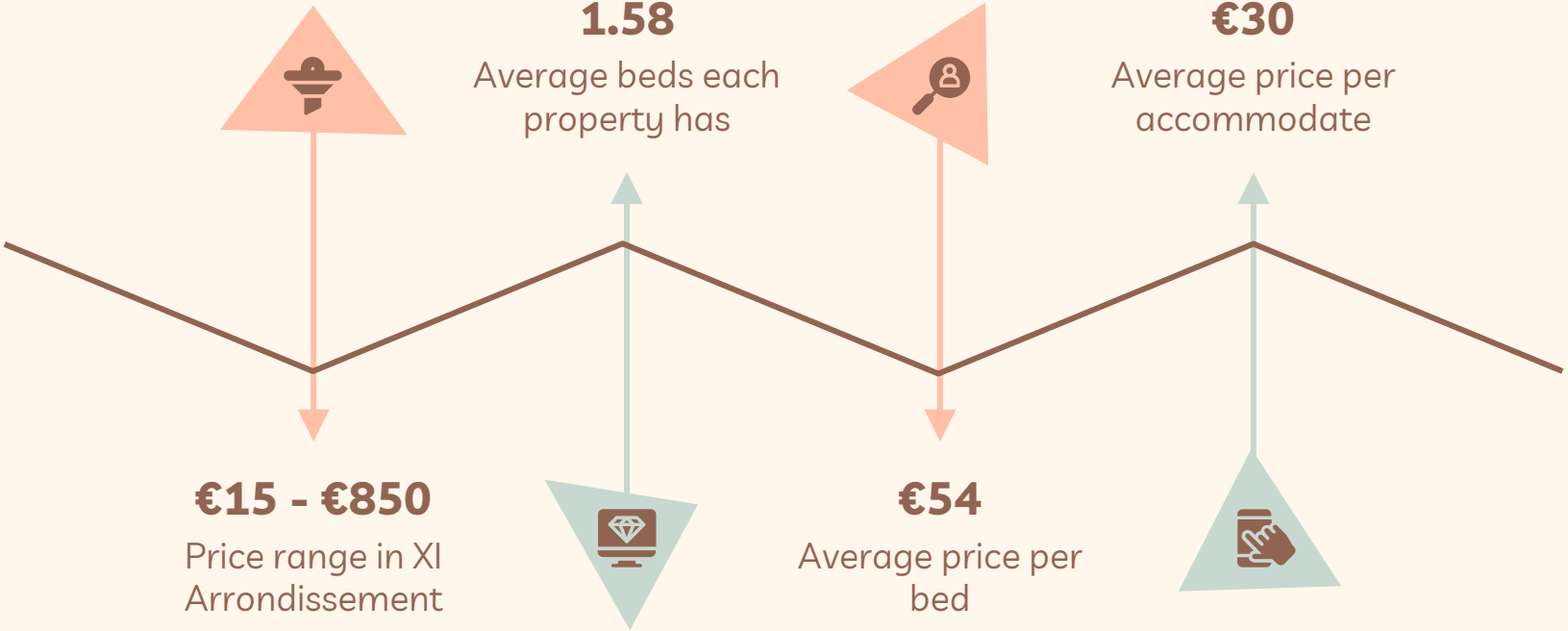
To the west lies the Place de la République, which is linked to the **Place de la Bastille**, in the east, by the sweeping, tree-lined Boulevard Richard-Lenoir, with its large markets and children's parks. The Place de la Bastille and the rue du Faubourg Saint-Antoine are full of fashionable **cafés**, **restaurants**, and **nightlife**, and they also contain a range of boutiques and galleries.


17e Arrondissement
Pigalle
Quartier de l'Europe
Faubourg Poissonnière
Opéra
2e Arrondissement
1er Arrondissement
3e Arrondissement
Les Halles
Le Marais
4e Arrondissement
Paris
6e Arrondissement
5e Arrondissement
Montparnasse
Faubourg Saint-Jacques
14e Arrondissement
13e Arrondissement
Faubourg Saint-Germain
Faubourg du Temple
Belleville
Ménilmontant
20e Arrondissement
Le Pré-Saint-Gervais
Les Lilas
Porte des Lilas
Bagnolet
Porte de Bagnolet
Reuilly
Saint-Mandé
12e Arrondissement
Bercy
Leaflet | © OpenStreetMap contributors, CC-BY-SA

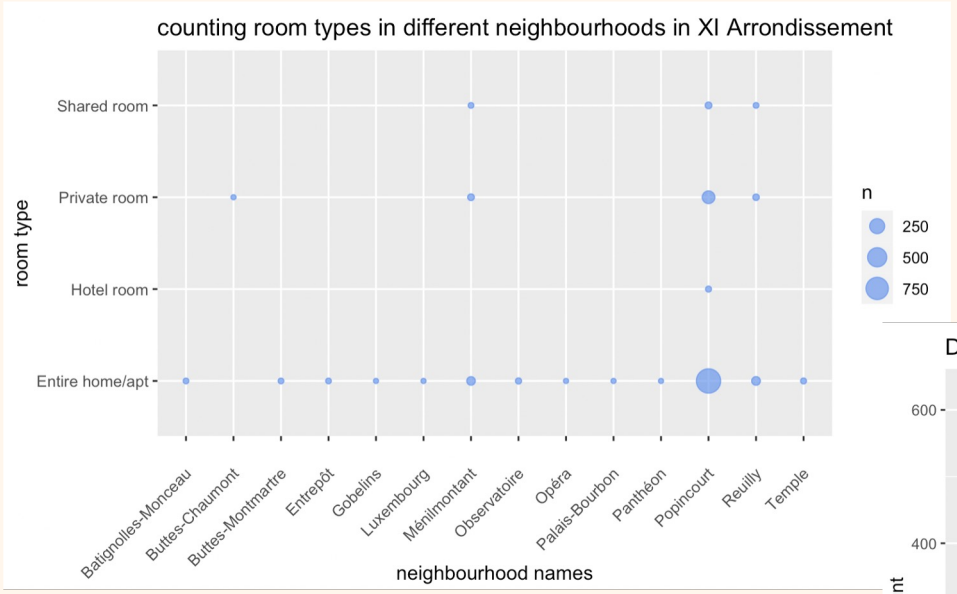
2

Data Preparation & Exploration

Quick facts

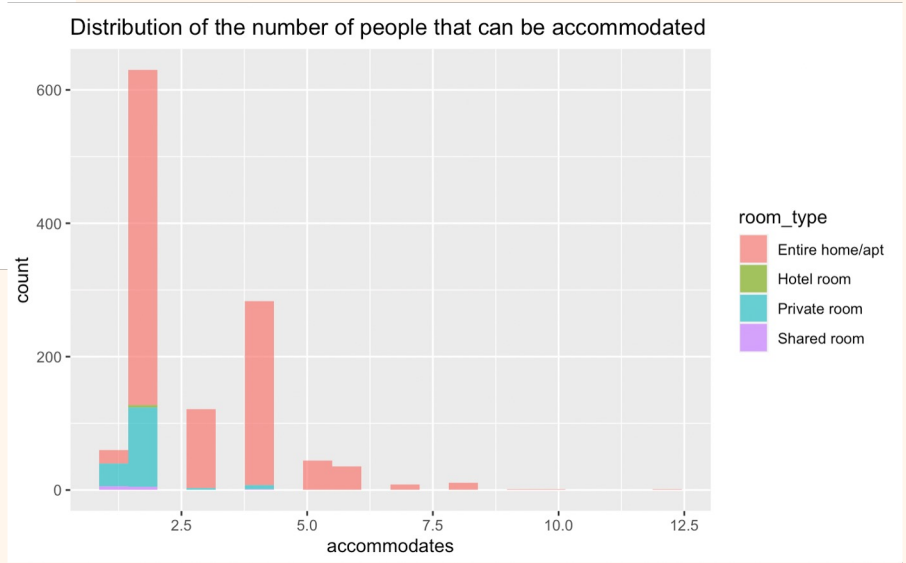

1.58
Average beds each property has
€30
Average price per accommodate
€15 - €850
Price range in XI Arrondissement
€54
Average price per bed

Room type & Accommodates



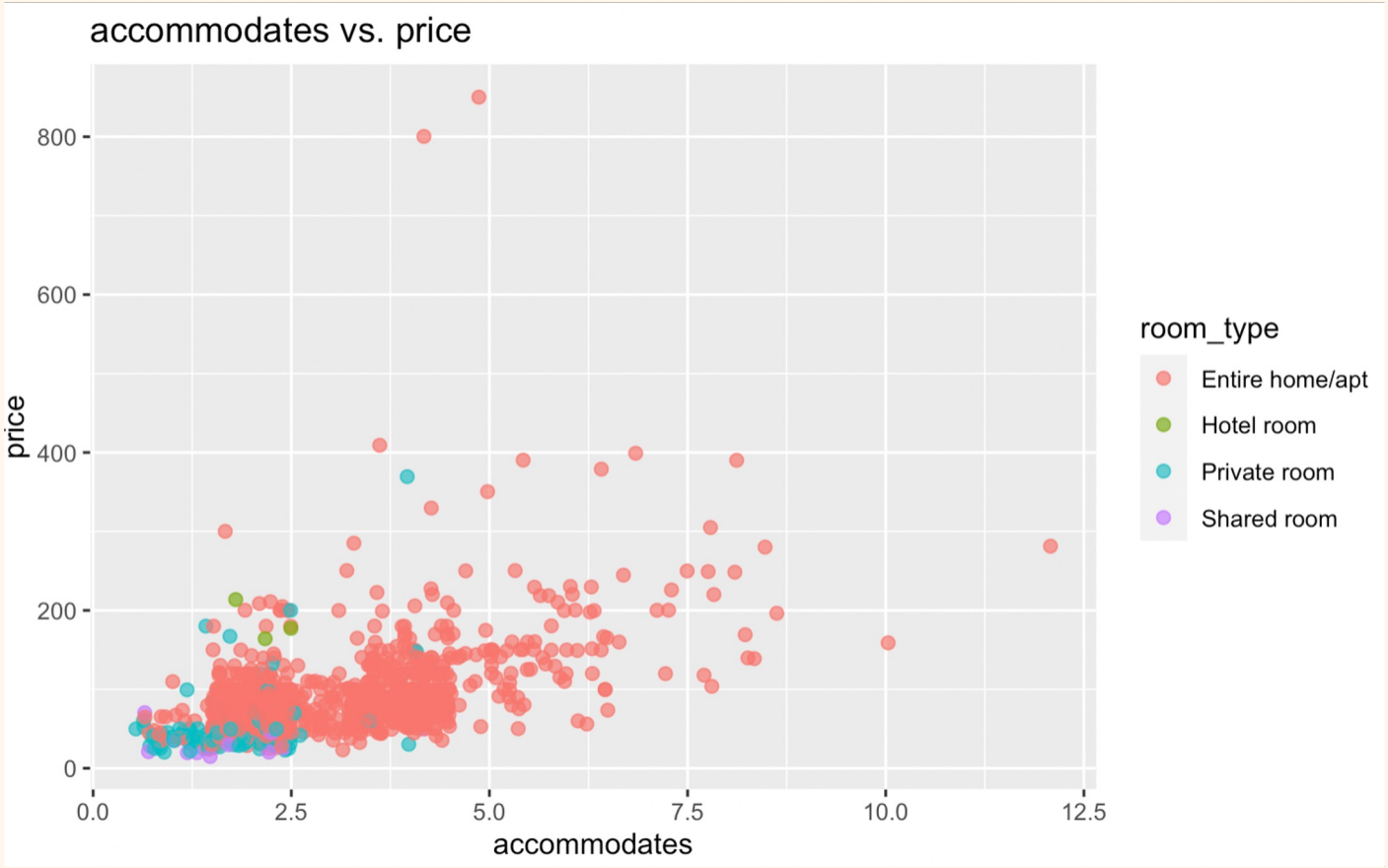
★★★★★

Almost all neighborhoods have entire home/apt.
Popincourt has all four types of listings.



Many of the properties can only accommodate 1 or 2 people, and the vast majority of the properties can only accommodate up to 4 people.

What kind of property to choose?

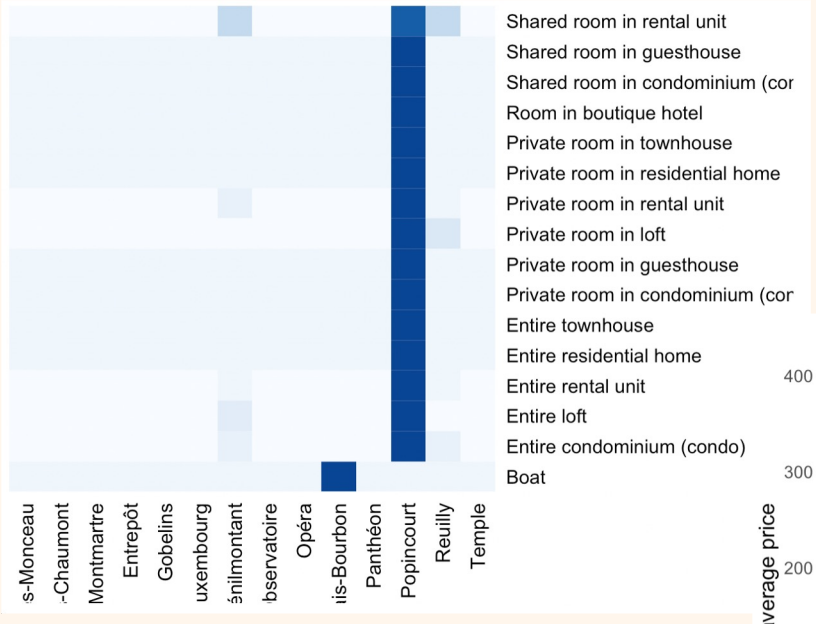


★★★★★

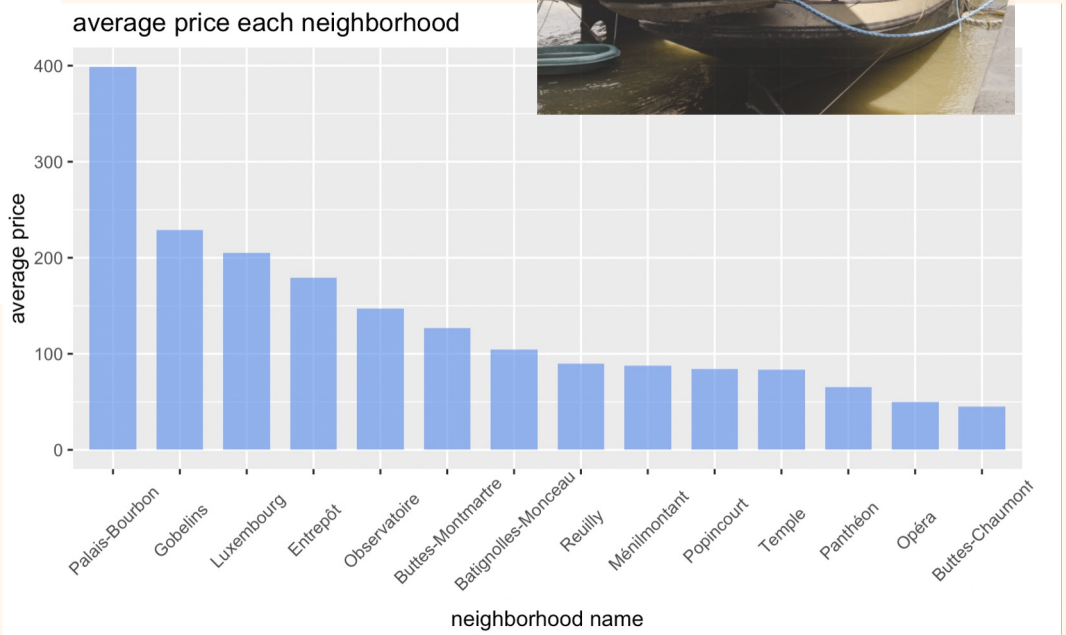
The higher the number of accommodations, the higher the price will be.

Private room and shared room are less expensive.

Average price for each property

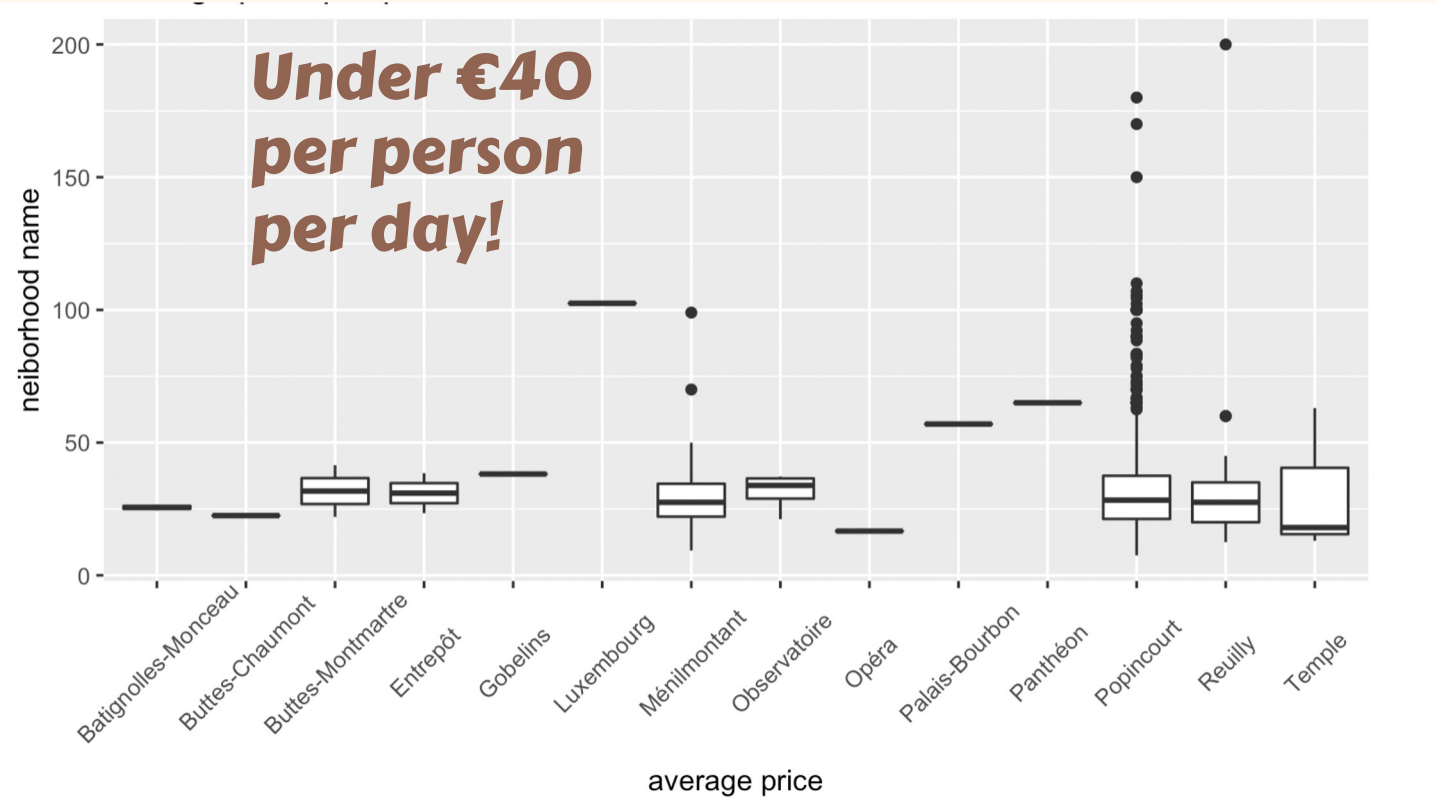


Popincourt has the highest density of most types of housing.



Neighborhood with **boats** has the highest average price.
Although **Popincourt** has the largest number of houses, its price is below average.

Average price per person



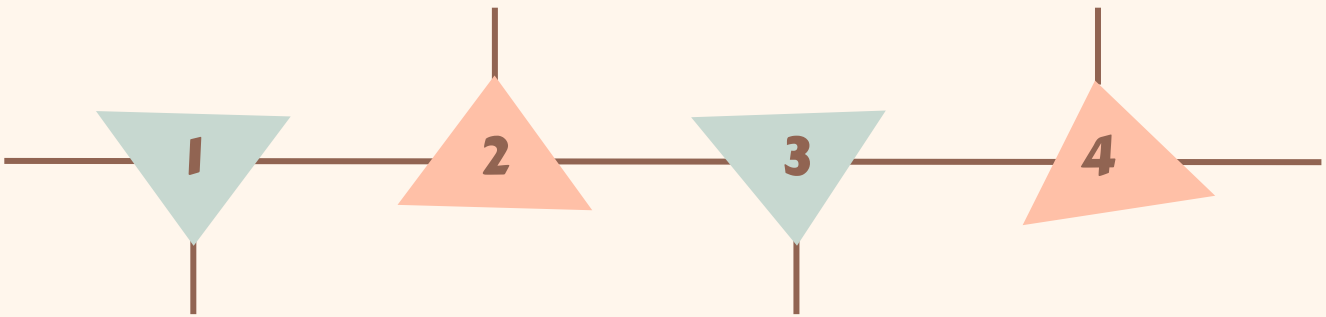
★★★★★

The median house price is mostly below 40.
In Popincourt, the median price is even lower. But there are many outliers that means one can definitely find some more expensive accommodations in this area.

3

Prediction

Data Preprocessing



Delete Variables with Missing Value > 20%

Drop Variables have no prediction power

Reduce the levels of categorical variables

Dummy the categorical variables

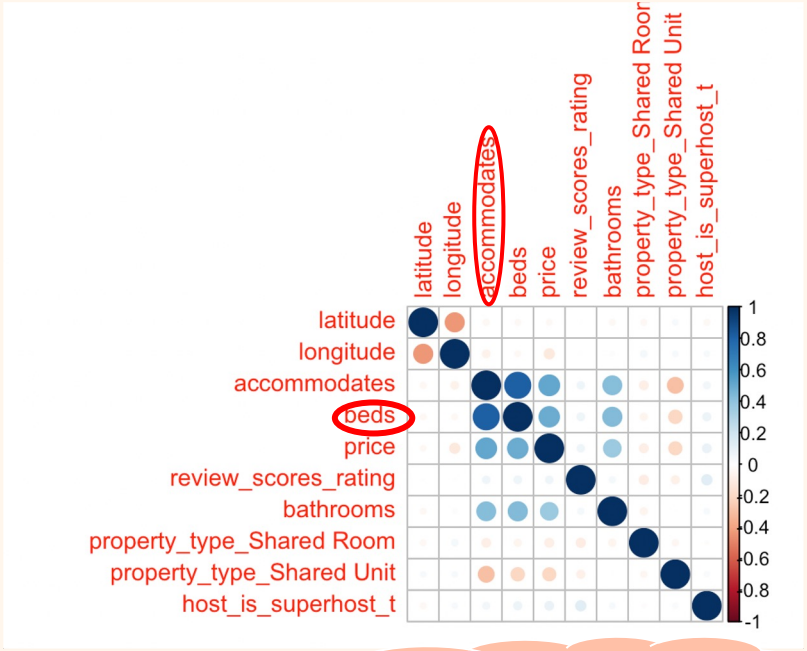
```
Entire Unit Shared Room Shared Unit
        1040          12         175
```

Remained Variables

```
'data.frame':   1227 obs. of  10 variables:
 $ latitude                 : num  48.9 48.9 48.8 48.9 48.9 ...
 $ longitude                : num  2.39 2.38 2.39 2.38 2.39 ...
 $ accommodates             : int  2 4 2 2 2 4 2 4 2 2 ...
 $ beds                     : num  1 2 1 1 1 2 1 1 2 1 ...
 $ price                    : int  60 140 59 50 70 90 89 84 60 60 ...
 $ review_scores_rating     : num  0 4.89 4.7 4.32 4.72 4.67 0 4.68 5 4.5
 $ bathrooms                : num  1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "problems")= tibble [5 × 4] (S3: tbl_df/tbl/data.frame)
  .. ..$ row     : int [1:5] 205 226 565 866 1054
  .. ..$ col     : int [1:5] NA NA NA NA NA
  .. ..$ expected: chr [1:5] "a number" "a number" "a number" "a number"
  .. ..$ actual  : chr [1:5] "Shared half-bath" "Half-bath" "Shared half-
 $ property_type_Shared Room: int  0 0 0 0 0 0 0 0 0 0 ...
 $ property_type_Shared Unit: int  1 0 0 0 0 0 1 0 0 0 ...
 $ host_is_superhost_t      : int  0 0 1 0 0 0 0 0 0 0 ...
```

Backward Method

Review_scores_rating ?
Host_is_super host?



multicollinearity?

Beds <-> Accommodates = 0.81

MLR Model

```
Call:
lm(formula = price ~ longitude + accommodates + bathrooms + `property_type_Shared Unit` +
    host_is_superhost_t, data = train.mul)

Residuals:
   Min     1Q Median     3Q    Max
-97.13 -22.39  -6.80  13.41 696.58

Coefficients:
                           Estimate Std. Error t value             Pr(>|t|)
(Intercept)                1749.111    545.963   3.204              0.00142 **
longitude                  -733.037    228.946  -3.202              0.00142 **
accommodates                 17.196      1.502  11.447 < 0.0000000000000002 ***
bathrooms                    33.647      6.089   5.526         0.0000000457 *
`property_type_Shared Unit`  -15.980      5.298  -3.016              0.002
host_is_superhost_t           14.751      5.150   2.864              0.004
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 46.53 on 730 degrees of freedom
Multiple R-squared:  0.337,     Adjusted R-squared:  0.3325
F-statistic: 74.22 on 5 and 730 DF,  p-value: < 0.00000000000000022
```

multicollinearity ?

Low R-squared!

Variance Inflation Factor

longitude	accommodates	bathrooms	`property_type_Shared Unit`	host_is_superhost_t
1.011009	1.432378	1.290256	1.138929	1.004491

Performance of the Model

```
accuracy(XI.lm.step.pre,train.mul$price)
```

```
                            ME     RMSE      MAE       MPE     MAPE
Test set -0.000000000000884273 46.0748 26.61292 -16.42595 34.11252
```

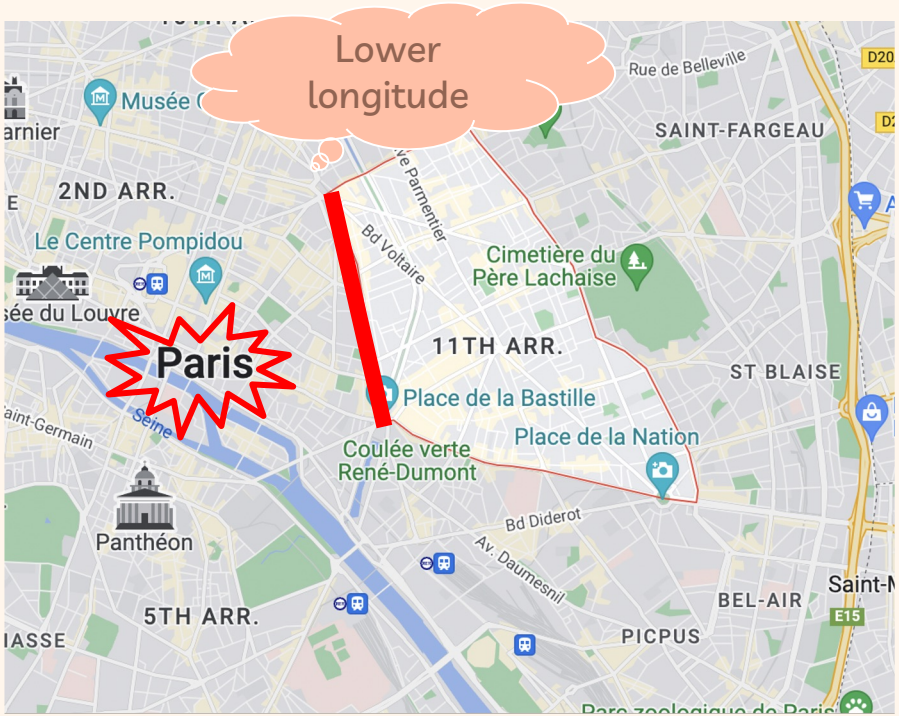
```{r}
XI.lm.step.pre1 <- predict(XI.lm.step,valid.mul)
accuracy(XI.lm.step.pre1,valid.mul$price)
```

```
                ME     RMSE      MAE       MPE     MAPE
Test set -1.825447 49.3879 26.06263 -19.36209 35.21962
```

- Difference between RMSE of training set and validation set: <20%
- MAE of both sets: Almost the SAME: <10%
- RMSE < Standard Deviation: 57

Conclusion

Price=-733.037longitude + 17.196accommodates + 33.647bathrooms-15.980property_type_Shared Unit + 14.751 host_is_superhost_t-733.037



Near the Center of Paris

Bigger House (contain more visitors)

No Shared House

Be a Super Host

4

Classification

a.

K-nearest Neighbors

Part I K-nearest Neighbors

Goal: Using K-nn model to predict whether a rental in our neighborhood will have a particular amenity.

Step 1: Choosing the outcome variable (The amenity)

Step 2: Choosing the numerical predictors

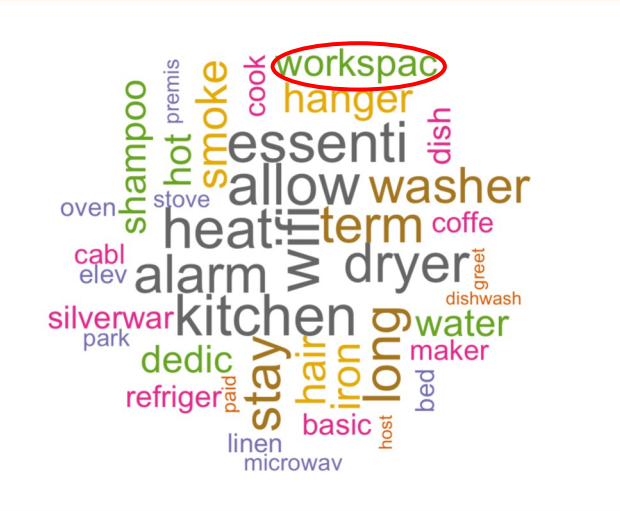
Step 3: Testing the accuracy & find the optimal k value

Step 4: Building the knn model

Step 5: Show the result

Part I K-nearest Neighbors

Why we chose “workspace”



Wordcloud of all properties

Wordcloud of properties with 75 or more reviews

We ran two Wordclouds in order to avoid amenities that all Airbnbs have, such as WiFi & Alarm.

After compared two graphs, we decided to choose “workspace”.

Because not all the Airbnbs have it, and we think people who are traveling for business will need a workspace.

Part I K-nearest Neighbors

Dropping numerical predictors with percentage difference in mean less than 5%

```
> # percentage difference in mean
> PD <- (abs(m1 - m0) / ((m1 + m0) / 2)) * 100
> PD
      latitude      longitude   accommodates      bedrooms          beds number_of_reviews
  4.293495e-05   4.202230e-02   4.230295e+00  5.390741e+00  4.394414e+00   8.368629e+01
         price minimum_nights maximum_nights     bathrooms
  1.530369e+01   4.673923e+01   2.152982e+01  3.818187e+00
```

10 numerical predictors selected

Created a training set, and a validation set.

Calculating the percentage difference in mean in the training set

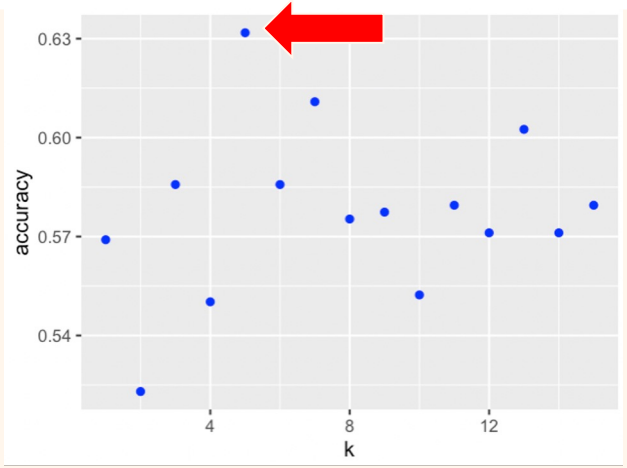
Removed 5 numerical predictors with percentage difference in mean is less than 5% to avoid overfitting.

Part I K-nearest Neighbors

Accuracy and optimal k value

```
> # Accuracy
> accuracy <- data.frame(k = seq(1, 15, 1), accuracy = rep(0, 15))
> for(i in 1:15) {
+   knn.pred <- knn(train.norm[, c(1:5)], valid.norm[, c(1:5)],
+                   cl = train.norm[, 6, drop = TRUE], k = i)
+   accuracy[i, 2] <- confusionMatrix(knn.pred,
+                                     valid.norm[, 6, drop = TRUE])$overall[1]
+ }
```

```
> accuracy
    k  accuracy
1   1 0.5690377
2   2 0.5230126
3   3 0.5857741
4   4 0.5502092
5   5 0.6317992
6   6 0.5857741
7   7 0.6108787
8   8 0.5753138
9   9 0.5774059
10 10 0.5523013
11 11 0.5794979
12 12 0.5711297
13 13 0.6025105
14 14 0.5711297
15 15 0.5794979
```



By running the accuracy test and plot, we found our optimal k value is 5, with the accuracy of 0.63

Part I K-nearest Neighbors

The knn model


```
# Predict any rental
new <- data.frame(bedrooms = 2, number_of_reviews = 80, price = 95, minimum_nights = 1,
                  maximum_nights =1000)
```

K-nearest neighbors model

```
# Compute knn
nn <- knn(train = train.norm[, c(1:5)], test = new.norm,
          cl = train.norm[, 5, drop = TRUE], k = 5)
nn
```

```
## [1] 0.65510964696682
## attr(,"nn.index")
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  335  521   54  431  277
## attr(,"nn.dist")
##           [,1]      [,2]     [,3]     [,4]     [,5]
## [1,] 0.7980145 0.9483146 1.021261 1.061863 1.116016
## Levels: 0.65510964696682
```

```
row.names(train)[attr(nn, "nn.index")]
```

```
## [1] "335" "521" "54"  "431" "277"
```

We made up a rental with the above data

The five nearest neighbors are shown below

Part I K-nearest Neighbors

Result

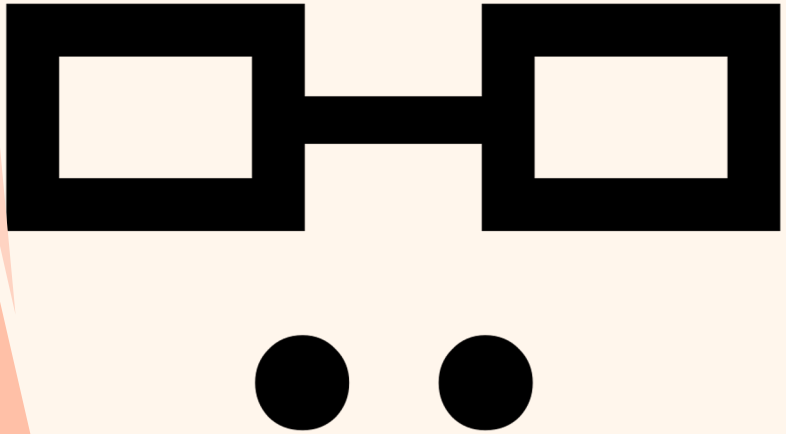

```
# Predict any rental
new <- data.frame(bedrooms = 2, number_of_reviews = 80, price = 95, minimum_nights = 1,
                  maximum_nights =1000)
```

```
> # Predicted classification for my rental
> PC <- XI4[c(335, 521, 54, 431, 277), 1:6]
> PC
# A tibble: 5 × 6
  bedrooms number_of_reviews price minimum_nights maximum_nights factor$df
     <dbl>             <dbl> <dbl>          <dbl>          <dbl> <fct>
1        1                25   120              2             15 0
2        1                 9    90              1           1125 1
3        1                 5    69              3           1125 1
4        1                48    65              2           1124 1
5        1                 0    55            365           1125 0
```

Based on our knn model with the optimized k value, 3 out of 5 Airbnbs have “workspace”

b.

Naïve Bayes



Part II Naive Bayes

Goal: **Predicting** whether **instantly bookable.**

Dependent variable: instant_bookable

What about the input variables?

Independent variables **trade off!**

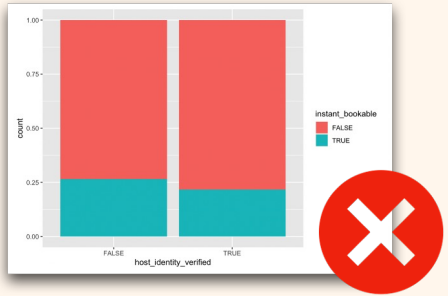
3 rounds Independent variables trade off

1st
by common sense
(number and factor variables)

Inputs	Data type
accommodates	number
availability_30	number
bathrooms_text	text(to number)
bedrooms	number
beds	number
has_availability	boolean
host_has_profile_pic	boolean
host_identity_verified	boolean
host_is_superhost	boolean
maximum_nights	number
minimum_nights	number
neighbourhood_cleansed	text
number_of_reviews_ltm	number
price	number
property_type	text
review_scores_accuracy	number
review_scores_location	number
review_scores_rating	number
room_type	text

19 Inputs? Crazy?

2nd
Dropping **low-impact** variables



Inputs
bathrooms_text
beds
host_has_profile_pic
host_is_superhost
minimum_nights
neighbourhood_cleansed
property_type
review_scores_rating
room_type

9 Inputs. Seems reasonable now

3rd

Remove variables that affect the **accuracy** of the model

Inputs	Data type	Description	Bin
beds	number	beds number	factor it
host_has_profile_pic	boolean	picture or not	factor it
host_is_superhost	boolean	superhost or not	factor it
property_type	text	property type	factor it
review_scores_rating	number	1~5	1~5
room_type	text	property type	factor it

6 inputs to build the model

```
book.nb <- naiveBayes(instant_bookable ~
                       beds
                      +host_has_profile_pic
                      +host_is_superhost
                      +property_type
                      +room_type
                      +review_scores_rating, data = train.df)
```

The **performance**
of our Naive Bayes

Train set accuracy 0.777	<	**Valid set accuracy 0.7809**	>	Instant_bookable = FALSE 0.777

Our model is slightly **more accurate** in predicting **new data**
than directly classified as most dependent variable.

Our model has **the ability to predict**
whether a particular rental will be instantly bookable.

Let’s give it a try!

1. Bed? 2. Host Picture? 3. Superhost?
4. Property type? 5. Room type? 6. Rating?

Lonely boy to travel there. Only need **1 bed**.
A **host picture** will give him confidence. Host is a **superhost**.
He needs **entire loft** and **private room** with **a rating close to 5**.

Let’s give it a try!

1. Bed? 2. Host Picture? 3. Superhost?
4. Property type? 5. Room type? 6. Rating?

Lonely boy to travel there. Only need **1 bed**.
A **host picture** will give him confidence. Host is a **superhost**.
He needs **entire loft** and **private room** with **a rating close to 5**.

```
## [1] FALSE
## Levels: FALSE TRUE
```

```
##          FALSE      TRUE
## [1,] 0.5960615 0.4039385
```

Unfortunately, this ‘great’ house is **not** instant bookable.

c.

Classification Tree

Part III Classification Tree

```r
XI2 <- subset(XI_Arrondissement, select = c(review_scores_rating, host_identity_verified,
                                            neighbourhood_cleansed, latitude, longitude,
                                            property_type, room_type, accommodates,
                                            bathrooms_text, bedrooms, beds, amenities,
                                            number_of_reviews, instant_bookable, price,
                                            host_is_superhost,maximum_nights,minimum_nights,
                                            host_has_profile_pic, host_acceptance_rate,
                                            calculated_host_listings_count))
data_tree <- XI2
data_tree[data_tree == ''] <- NA
data_tree <- drop_na(data_tree)
colSums(is.na(data_tree))
```

```
                  review_scores_rating               host_identity_verified 
                                     0                                    0 
                 neighbourhood_cleansed                              latitude 
                                     0                                    0 
                             longitude                         property_type 
                                     0                                    0 
                             room_type                          accommodates 
                                     0                                    0 
                        bathrooms_text                              bedrooms 
                                     0                                    0 
                                  beds                             amenities 
                                     0                                    0 
                     number_of_reviews                      instant_bookable 
                                     0                                    0 
                                 price                     host_is_superhost 
                                     0                                    0 
                      maximum_nights                       minimum_nights 
                                     0                                    0 
              host_has_profile_pic                host_acceptance_rate 
                                     0                                    0 
calculated_host_listings_count 
                                     0 
```

Part III Classification Tree

fivenum(data_tree$review_scores_rating)				
minimum	lower quartile	average	upper quartile	maximum
0	4.565	4.8	5	5

Classify data into three grades:

0~4.565	Under Average Rating
4.565~4.8	Average Rating
4.8~5	Above Average Rating

Part III Classification Tree

★★★★★

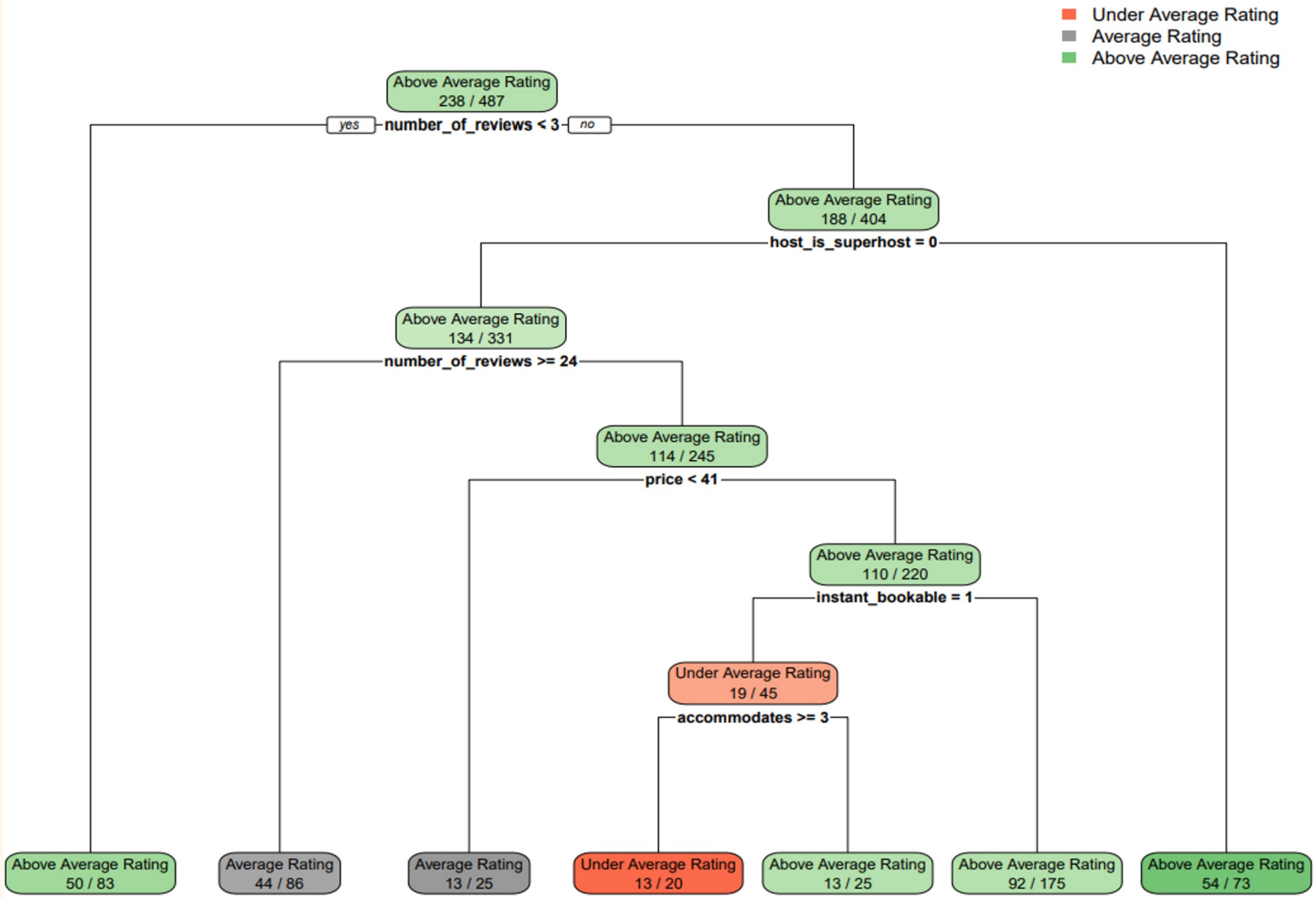
Factors that we think reflect people's choices about rooms & the reason

- number of reviews → popularity / advantages and disadvantages
- host is superhost → recognition of landlords / cannot be faked / with reference value
- accommodates → people directly care about
- beds → people directly care about
- bedrooms → people directly care about
- bookable → everyone must consider
- price → everyone must consider

Part III Classification Tree

```
Classification tree:
rpart(formula = review_scores_rating ~ accommodates + bedrooms +
    beds + number_of_reviews + instant_bookable + price + host_is_superhost,
    data = train, method = "class", xval = 5, cp = 0)

Variables actually used in tree construction:
[1] accommodates      bedrooms          beds              host_is_superhost instant_bookable
[6] number_of_reviews price

Root node error: 249/487 = 0.51129

n= 487

          CP nsplit rel error  xerror     xstd
1 0.0321285      0   1.00000 1.00000 0.044302
2 0.0160643      4   0.86747 0.91566 0.044224
3 0.0110442      6   0.83534 0.90361 0.044185
4 0.0080321     10   0.79116 0.93574 0.044272
5 0.0060241     13   0.76707 0.96787 0.044311
6 0.0040161     19   0.73092 0.97992 0.044313
7 0.0000000     30   0.67068 0.97992 0.044313
```

Pruning the tree


Under Average Rating
Average Rating
Above Average Rating
Above Average Rating
238 / 487
yes
number_of_reviews < 3
no
Above Average Rating
188 / 404
host_is_superhost = 0
Above Average Rating
134 / 331
number_of_reviews >= 24
Above Average Rating
114 / 245
price < 41
Above Average Rating
110 / 220
instant_bookable = 1
Under Average Rating
19 / 45
accommodates >= 3
Above Average Rating
50 / 83
Average Rating
44 / 86
Average Rating
13 / 25
Under Average Rating
13 / 20
Above Average Rating
13 / 25
Above Average Rating
92 / 175
Above Average Rating
54 / 73

Part III Classification Tree

```r
library(lattice)
library(caret)
pred.train<-predict(model_tree2,train[,c(8,10,11,13,14,15,16)],type="class")
confusionMatrix(pred.train, train$review_scores_rating)
```

```r
pred.valid<-predict(model_tree2,valid[,c(8,10,11,13,14,15,16)],type="class")
confusionMatrix(pred.valid, valid$review_scores_rating)
```

★★★★★

Test accuracy

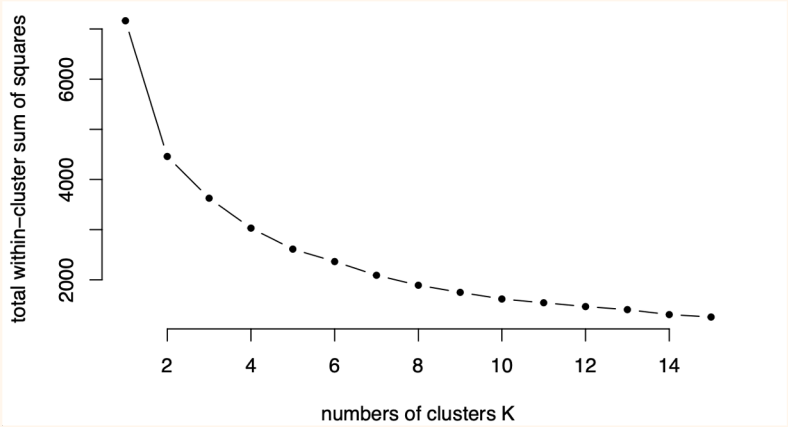
```
Overall Statistics

               Accuracy : 0.5729
                 95% CI : (0.5276, 0.6173)
    No Information Rate : 0.4887
    P-Value [Acc > NIR] : 0.0001189

                  Kappa : 0.2521

 Mcnemar's Test P-Value : < 0.00000000000000022
```

```
Overall Statistics

               Accuracy : 0.5077
                 95% CI : (0.4519, 0.5633)
    No Information Rate : 0.5015
    P-Value [Acc > NIR] : 0.434

                  Kappa : 0.1346

 Mcnemar's Test P-Value : 0.00000000000002327
```

Although the accuracy is not high
the closeness of these two numbers means that overfitting does not occur

Clustering

5



★★★★★

K-means Clustering Process

```
clusters$centers
```

```
##   accommodates  bedrooms       beds number_of_reviews      price  bathrooms
## 1    1.9108889  2.601751  2.2193440       0.038733806  1.6018164  1.5373391
## 2   -0.2133085 -0.290428 -0.2477407      -0.004323774 -0.1788074 -0.1716099
```

```
XI.clu %>% group_by(cluster) %>% summarise_all("mean")
```

```
## # A tibble: 2 x 7
##   cluster accommodates bedrooms  beds number_of_reviews price bathrooms
##     <int>        <dbl>    <dbl> <dbl>             <dbl> <dbl>     <dbl>
## 1       1         5.45     2.54  3.53              22.1 178.       1.55
## 2       2         2.57     1.01  1.36              20.2  75.2      1.04
```

Clusters: Better Services to Target Clients

1

For Leisure Guests

- With a **higher price**, these rentals offer more accommodates, bedrooms and beds.
- The rentals can have some little touches to make the **stay special**.
- They could also provide luxuriously soft bedding, toys for the little ones or a comprehensive guide to the city.

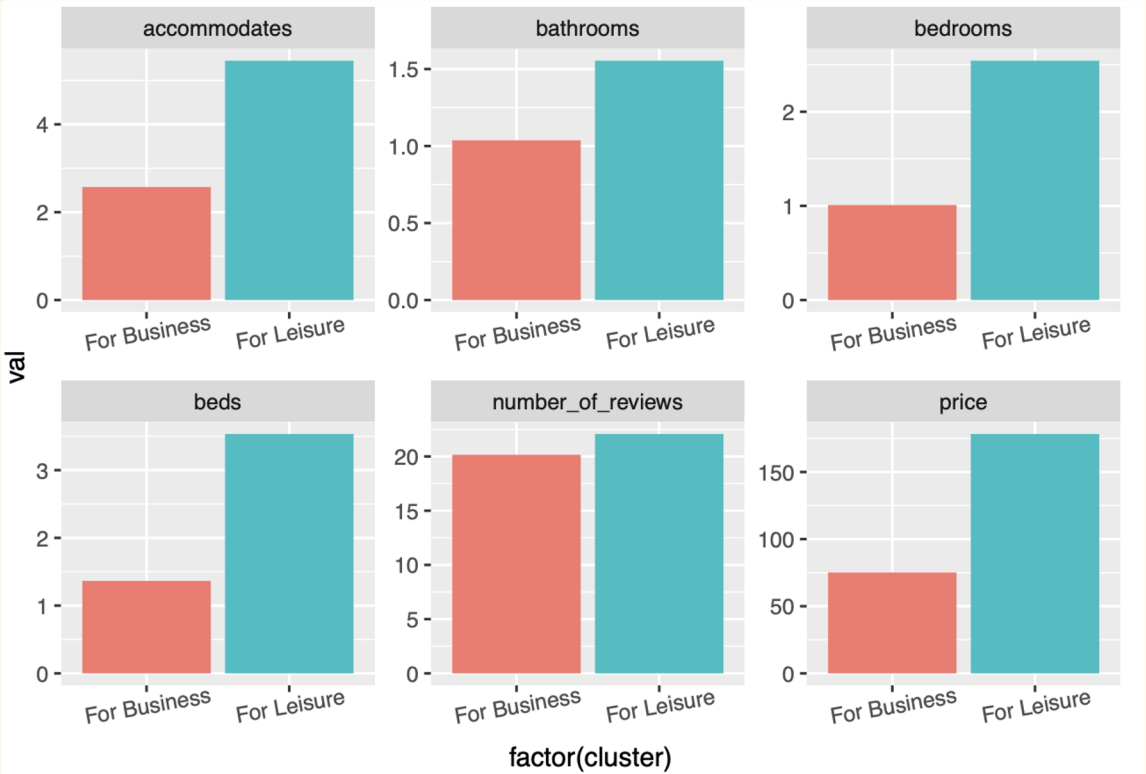
2

For Business Travelers

- With a **lower price**, the guests want a quiet work space and one bed is enough for them.
- A laptop-friendly **workspace**, WiFi and extra chargers are **vital**.
- These rentals could also provide some information on transport services and taxi numbers, or free parking.

Faceted Bar Plot

The faceted bar plot shows the **per-cluster comparison** across each feature, which demonstrates those key differences among each of the two clusters.



Conclusion

6

Airbnb Analysis Project Conclusions

- Based on our prediction model, we will suggest people who want to be a host in this area to invest a big house which can contain more visitor near the center of Paris. It's better for them to rent the entire house. Furthermore, try to become a super host by improving the customers' satisfaction. These will help the hosts to increase their price and be most profitable.
- By running the k-nn model, if a particular Airbnb was rented out, we can provide more similar options to renters who are looking for specific amenity.
- Using this Naïve Bayes model built on 6 predictors, we can predict whether a particular rental will be instantly bookable.
- For the classification tree, we can see the content related to the house rating and what kind of relationship there is between them, which is a strong reference for hosts who are running Airbnb or people who want to join Airbnb in the future.
- By clustering, the hosts can know how to differentiate themselves and provide better services to their target clients.

Thank You!

Any Questions?