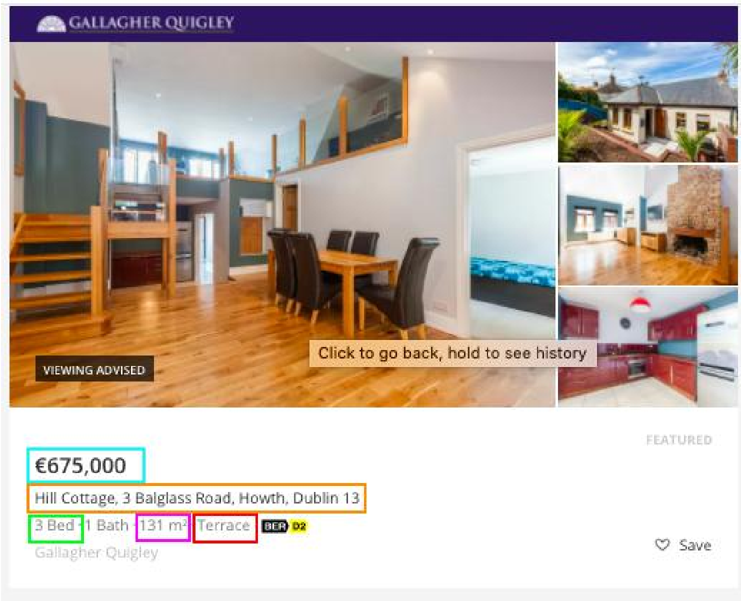
Dublin Property Analysis

Choosing a property is choosing a lifestyle.

1.Dataset

The dataset is from Daft.ie and focus on Dublin city's apartments and houses.The original data includes location (orange), area (purple), the number of bedroom (green), price (blue), and property's type (red).



1-1 Daft.ie property in Dublin

The original data is very rough and has above 2900 data. Using the EXCEL to deal with the data at first. Remove duplicate and missing data. Extracting Dublin's districts from location. For example, from the 1-1 graph, Dublin 13 is the district in location information. Considering the address's information is not very direct. If we need to draw map graph, it needs latitude and longitude. Using tools to gain latitude and longitude. Finally, the dataset includes 8 attributes.

1. Location: The detailed sale apartment or house address.
2. District: Includes 22 district in Dublin city.
3. Area: Square meter.
4. Bed: Number of bedrooms in this apartment or house.
5. Latitude
6. Longitude

7. Price: Unit is euro.
8. Type: Includes 8 types.

```
'data.frame':   2389 obs. of  8 variables:
 $ location : Factor w/ 2372 levels "1 An Tearmann, Palmerstown, Dublin 20",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ district : Factor w/ 22 levels "1","10","11",..: 12 20 6 10 6 14 18 7 11 14 ...
 $ area     : Factor w/ 275 levels "100 m²","101 m²",..: 84 28 20 275 257 5 242 52 246 245 ...
 $ bed      : Factor w/ 10 levels "1 Bed","12 Bed",..: 5 6 5 3 4 4 3 5 4 4 ...
 $ latitude : num  53.4 53.4 53.3 53.2 53.3 ...
 $ longitude: num  -6.37 -6.3 -6.28 -6.12 -6.27 ...
 $ price    : Factor w/ 307 levels "€ 1,000,000",..: 253 202 247 147 217 185 242 213 217 115 ...
 $ type     : Factor w/ 8 levels "Apartment","Bungalow",..: 3 5 6 1 6 3 5 6 8 5 ...
```

2.Relevant information and data

2.1 Residential holding type

1. Freehold

It means the unconditional ownership of a house and land. This ownership includes not only the building itself, but also the land it occupies. Terrace, Townhouse, Detached, Semi-D, etc.are belong to freehold.

2. Leasehold

It means the buyer has the right to use the property. The maximum term is 999 years and needs to pay a land rental fee every year. In general, the rental fee with management fee are paid to the real estate company. Almost all apartments are sold in this way.

3. Share of freehold

It is a combination of freehold and tenancy, which is used for some small apartment buildings. If two-thirds of the owners vote to buy the entire property, they can form a corporation that officially owns the building, and each owner has ownership of the property in proportion to the amount of space owner occupies in the building. This type is very rare.

2.2 Building type

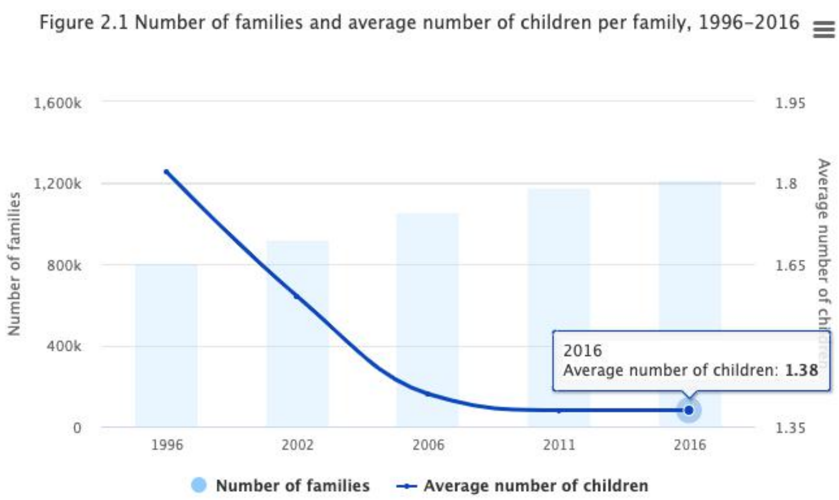
Terrace: Has a small back garden and share two walls with neighbors. It belongs to freehold.
End of Terrace: It is similar with Terrace, the only difference is public one wall with neighbor.
Townhouse: compare with other types, it has more floors, general 3-5 storeies. Space is larger. Usually each townhouse has its own garden and park.
Semi-D: A single villa is divided into about two sets of independent housing, the middle of a public wall. It has front and back garden.
Detached: independent villa, front and back garden. With a high level of privacy, the

house is surrounded by spacious courtyards.
Bungalow: It is a small house or cottage that is either single-storey or has a second storey built into a sloping roof (usually with dormer windows),and may be surrounded by wide verandas.
Apartment: Usually belongs to leasehold, annual ground rent, small size and managed by a property company.
Duplex: Duplex flat is similar with apartment, which are larger in size than regular apartments.

2.3 Relevant data

https://www.cso.ie/en/releasesandpublications/ep/p-cp4hf/cp4hf/fmls/

According CSO Irish family statistic, an average family has 1.38 child. It can be seen that as fertility rates have fallen, the housing needs of Irish households have shifted from large families to small ones in two decades.



2-1 Number of families and average number of children per family, 1996-2016

http://census.cso.ie/sapmap2016/Results.aspx?Geog_Type=CTY31&Geog_Code=2AE19629143313A3E055000000000001#SAPMAP_T4_400

According the Dublin family statistic 2-2. In Dublin, 1/3 families they without children. Following, 1 child and 2 children families are close to 1/2 in total. Minority families had above 5 children in Dublin. Majority family's requirements focus on 0, 1, and 2 children. Normally, 4 or less bedrooms can satisfy most of families.

PDF Excel Print

Family units with children, by size and age of children

Number of children	All children under 15	All children 15 or over	Children both under and over 15	Total
No children	0	0	0	44,257
1 child	17,111	20,137	0	37,248
2 children	13,562	9,851	4,332	27,745
3 children	4,602	2,972	3,978	11,552
4 children	1,005	647	1,650	3,302
5 or more children	242	148	706	1,096
Total	36,522	33,755	10,666	125,200

2-2 Number of children in Dublin families in 2016

Graph 2-3, it refers one people household is majority in Dublin. Following is married couple and children. In Dublin, Childless households are 1.24 times more likely to have children than households with children. However, among them, the number of couple household who without children is less than couple who with children. When people became couple, more than half of them will choose have children.

Private households by type

Type of Household	Households	Persons
One person	60,001	60,001
Married couple	24,308	48,616
Cohabiting couple	14,306	28,612
Married couple and children	40,058	158,610
Cohabiting couple and children	6,290	23,868
Father and children	2,613	6,294
Mother and children	19,207	52,302
Couple and others	4,081	14,320
Couple, children and others	3,756	19,072
Father, children and others	493	1,785
Mother, children and others	2,954	11,090
Two or more family units	3,514	19,020
Non-family households and relations	7,874	20,068
Two or more non-related persons	22,292	61,571
Total	211,747	525,229

2-3 Number of type of household in Dublin in 2016

From the 2-4, it shows the house type is most popular in Dublin. The number of apartment is close to a half of house.

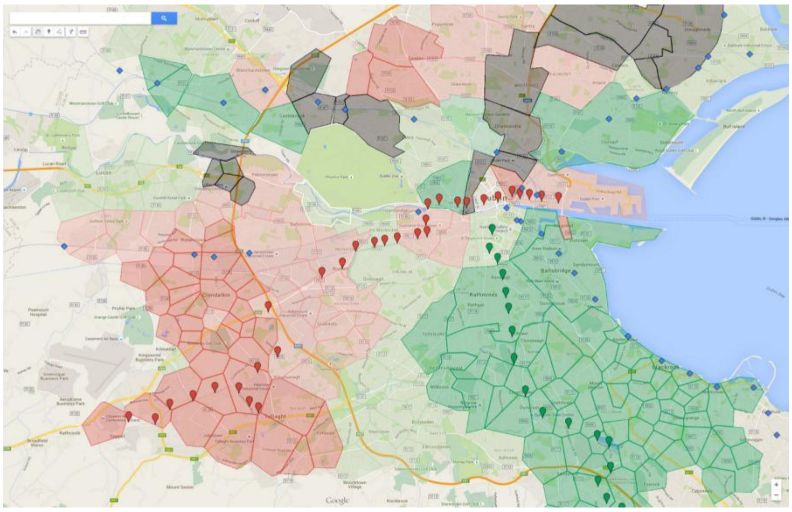
Private households by type of accommodation

Type of accommodation	Households	Persons
House/Bungalow	133,709	363,282
Flat/Apartment	72,526	149,540
Bed-sit	2,011	2,767
Caravan/Mobile Home	156	542
Not stated	3,345	9,098
Total	211,747	525,229

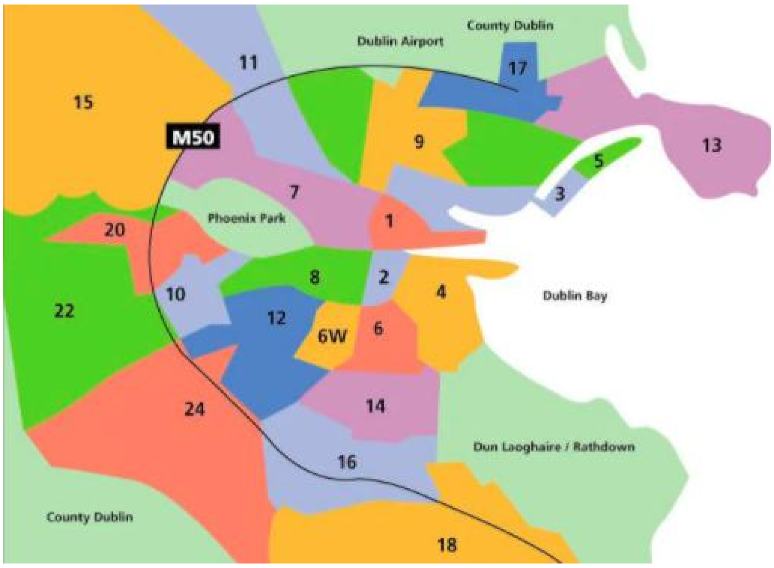
2-4 Accommodation's type in Dublin in 2016

https://dublinhousehunting.com/searching-for-the-perfect-house/2015/4/12/whatare-the-best-areas-and-those-that-you-should-avoid

The 2-5 graph shows the security evaluation is from Irish people. The red is dangerous, the green is safe. The 2-6 graph shows the distribution of Dublin district. Combine the two graphs, the 3, 4, 6, 14, 15, 6W, 16, 18, 13, etc.are safer than other districts.



2-5 Dublin safety level map



2-6 Dublin district map

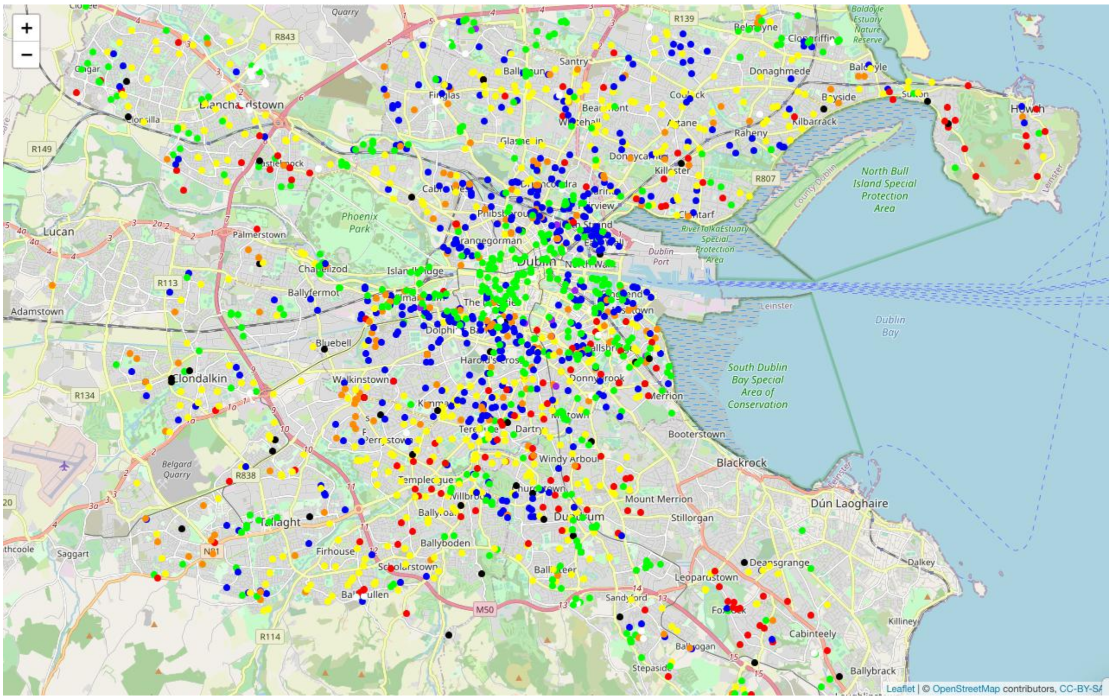
https://www.dailyedge.ie/alternative-dublin-maps-1374128-Mar2014/

The 2-7 shows the distribution of the middle class in Dublin, the darker the purple the more middle class in the area. The middle class map and regional safety show a strong correlation.

2-7 distribution of Middle classes

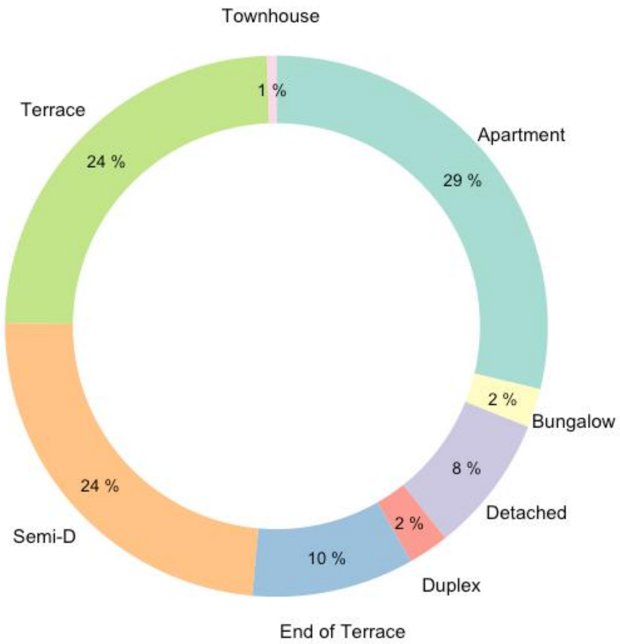
3.Data visualization

In total, there are 8 types that Townhouse(purple), Apartment(green), Bungalow(black), Detached(red), Duplex(white), End of Terrace(orange), Semi-D(yellow), Terrace(blue). Apartment, Duplex belong to apartment type. Others belong to house type. The 3-1 refers a large number of Apartments in the city center and a large number of Terraces and End of Terraces close to the city center. Outside the Terraces and and End of Terraces circle there is a concentration of Semi-D.



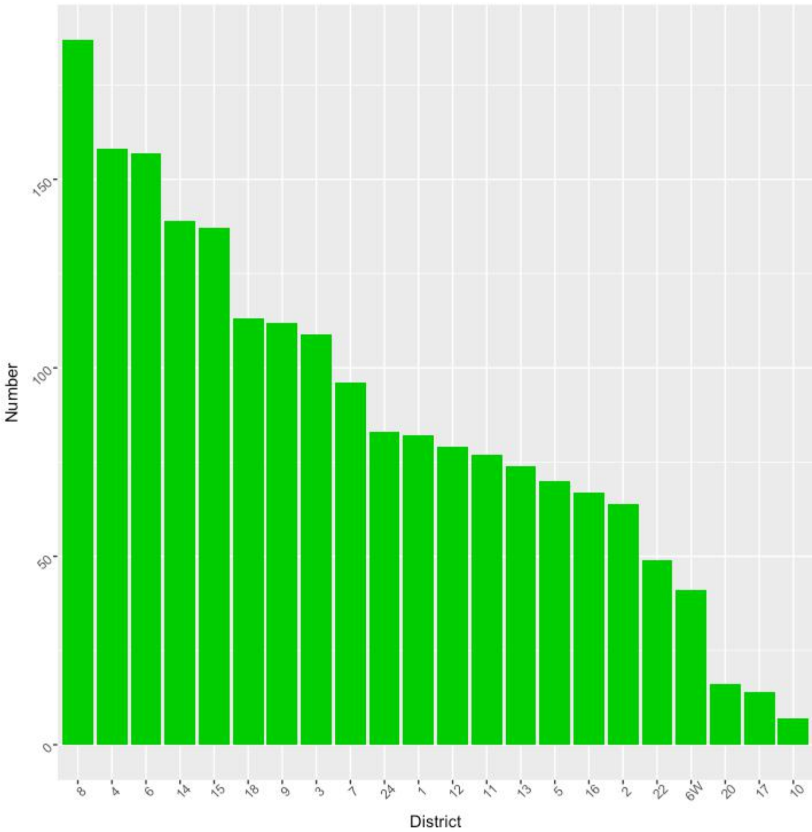
3-1 Distribution of the different types on the map

From 3-2, Apartment type has the highest percentage of accommodations for sale, 29%. The next is Semi-D and Terrace, both at 24%. These three types of accommodations are the most widespread in Dublin. In conclusion, house type accounts for the largest proportion in Dublin.



3-2 Circle-type graph

From 3-3, in all districts, most accommodations for sale in Zone 8. The graph 2-6 shows Zone 8 is city center. Sequence, Zone 4, 6, 14, 15 have a lot of accommodations for sale. At the same time, the number of accommodations is influenced by district's area and building's density.

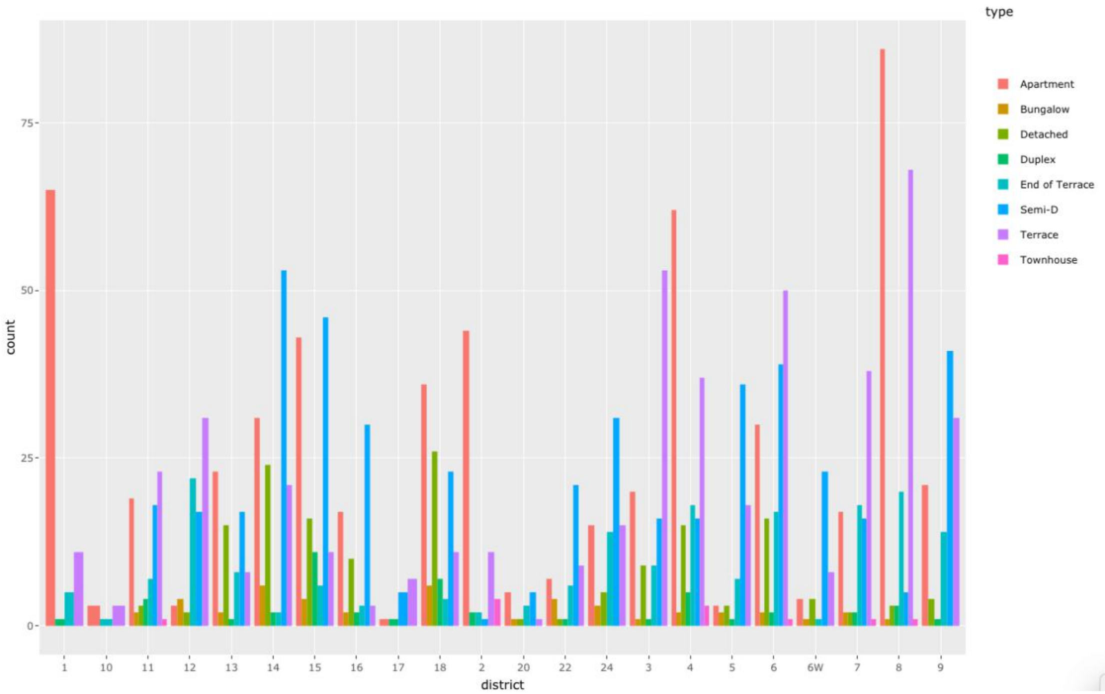


3-3 Hist-the number of accommodation in every district

Next, to know the distribution of the various types of accommodations in each district. According 3-4, Apartment is the most widespread in Zone 1, 18, 2, 4, and 8. Among them, Zone 1, 2, and 8 are city center.

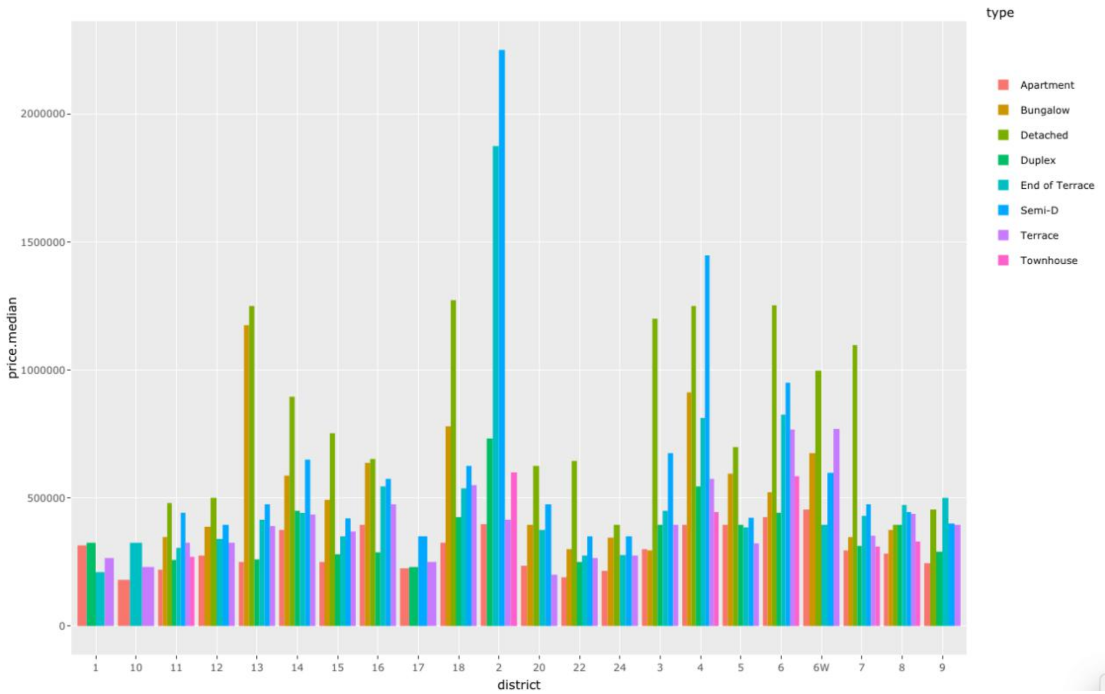
Why are these more apartments in the city center? There are some analysis and speculations.

1. Apartments have a higher land use than houses, more accommodations can be built on the same area of land to meet residential needs, and the number of apartments shown in 3-3 graph is logical in a city center where land is at a premium.
2. The apartment type is built on land bought by a real estate company and then sold to customers. Apartment are a long term gain for the real estate company and the income from property fees is more stable than the profit gained from building houses. As a result, real estate companies prefer to build apartments in locations where there is a high demand for living and they are easy to sell.
3. One speculation is apartment’s price is lower than other types in same district.



3-4 Distribution of the types

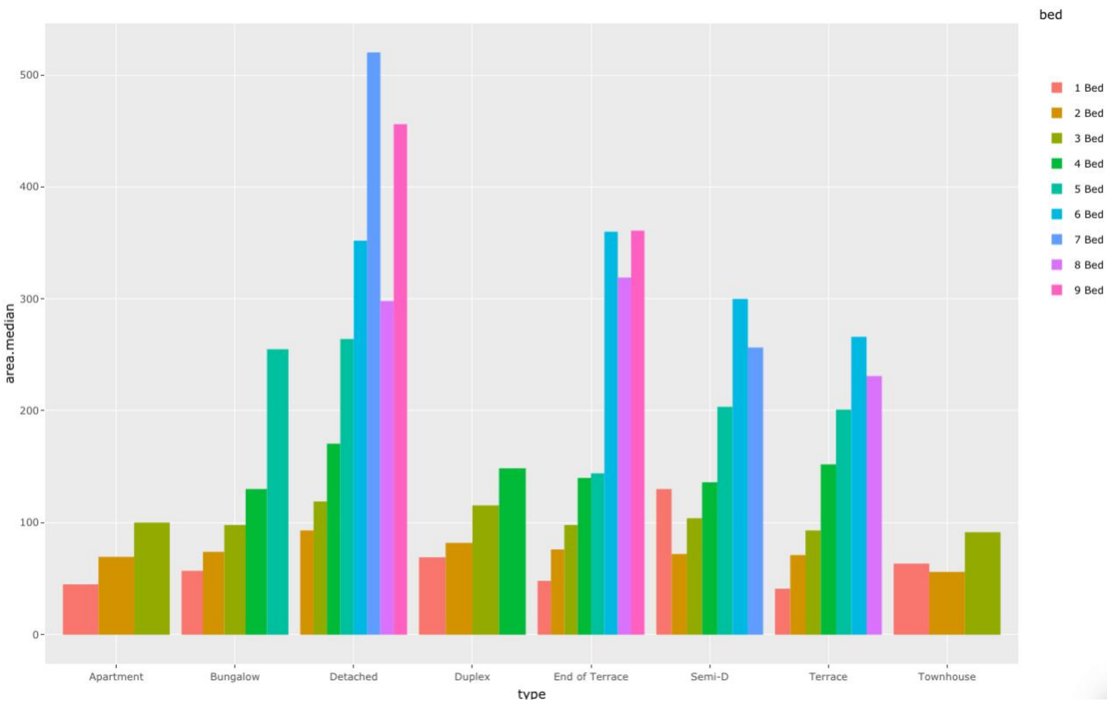
Following, to know the distribution of the various types median price of accommodations in each district. From 3-5, the median price of apartment is the lowest in every districts. Except apartment properties are different from houses. Apartment price is lower than other types in same district, because apartment price is influenced by area and number of bedroom. One speculation is apartment has less area and number of bedroom than other types.



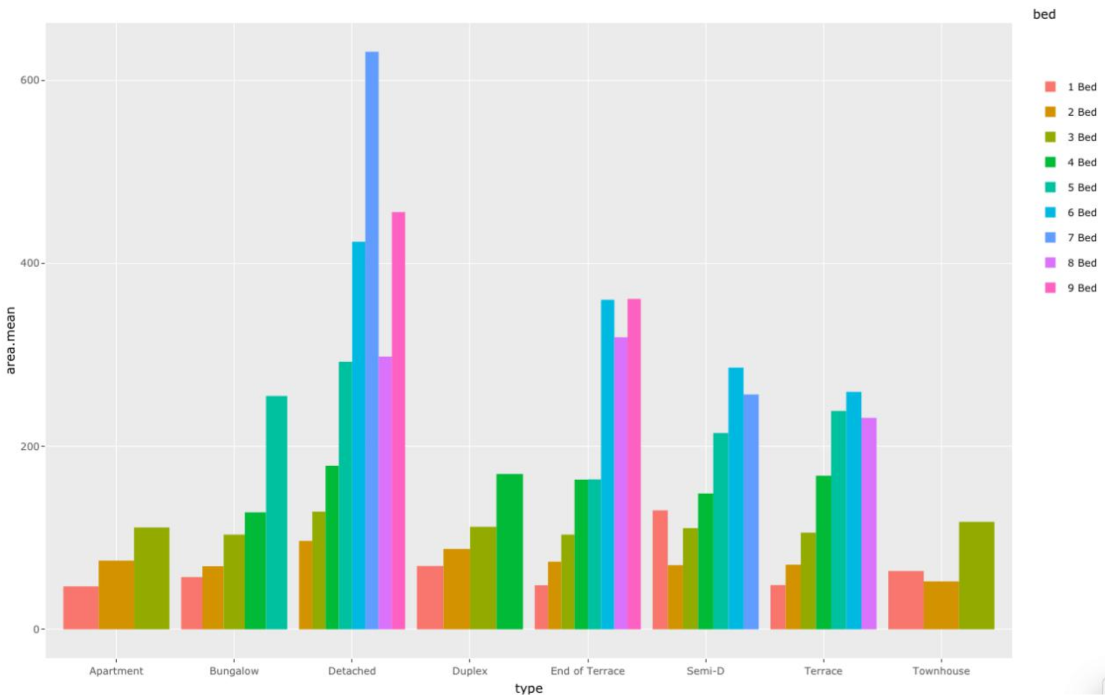
3-5 Distribution of the various types median price

Next, to know the every accommodation type's area and number of bedroom. From

3-6, it refers the number and area distribution of bedrooms in apartments and houses. Apartments have a maximum of 3 bedrooms, so if customers want an apartment with more than 3 bedrooms, they can only choose a duplex. Duplexes can offer up to 4 bedrooms. According to references, the distribution of family members in Dublin is concentrated at less than 4 people and couples can share a bedroom, so 3 bedrooms will suit most families living in Dublin. One bedroom terrace's median area is the smallest, only 41 square meters(sqm). The sequence is apartment, median area is 45 sqm. From 3-7, one bedroom apartment's average area(46.8 sqm) is less than others.

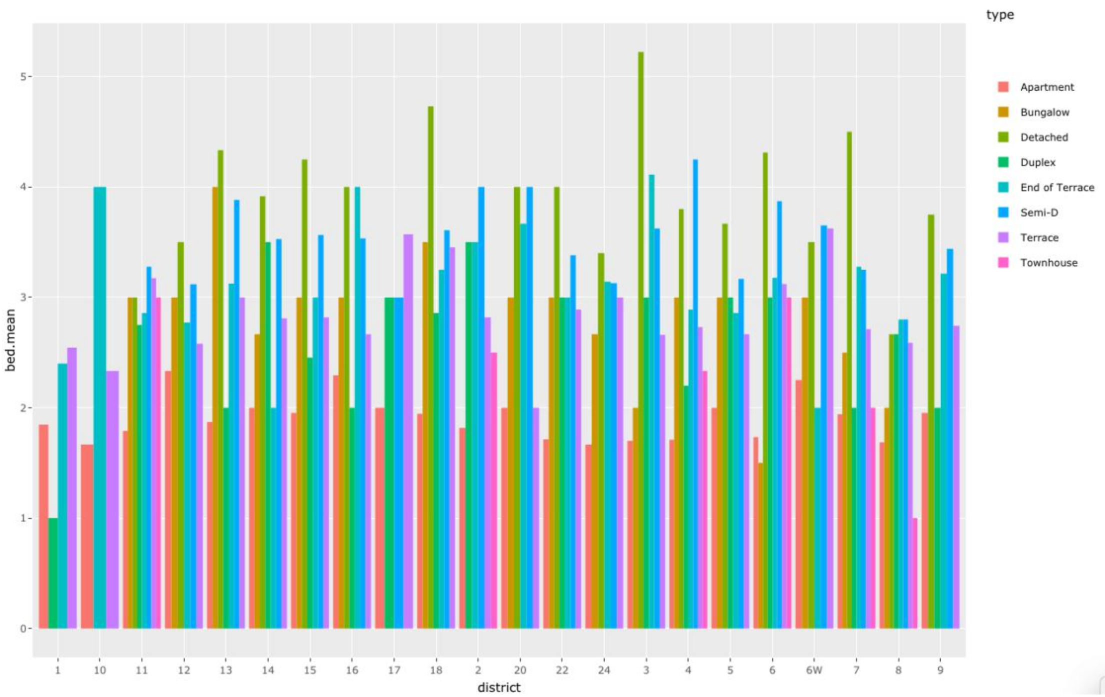


3-6 Type's median area and number of bedroom



3-7 Type's average area and number of bedroom

Compare to the average number of bedrooms in different types in each district (3-8), the average number of bedrooms in the apartments in each district is the lowest, followed by the Duplex. Detached has most bedrooms in all types.



3-8 Average number of bedrooms in different types in each district

An good apartment is based on good neighbors, a good house is based on a good community. For the new middle class or young families seeking a certain quality of life, the following here are two tips.

(As property data is not comprehensive and property is an alternative investment, there are large individual variations. The following recommendations are therefore based on the data in this article only)

1. Apartment group: As for single family or couples who without children. Apartment is a suitable choice. City center has many apartments and price is lower than other types. Youngsters can choose an apartment as their first accommodation as a transition. Apartments are managed by property company, which can save time in management. And the number of apartments in the city center is relatively large, it is easier to pick a satisfactory accommodation. At present, Ireland's policy is not limit the purchase of apartments, investors and young independent groups will be the first choice of apartments.

Recommend: Zone 4, 8, 18.
Zone 4's safety is better than other center districts and price is suitable. Zone 8 has more apartment and lower price than other center districts. Ignore the safety, Zone 8's location and cost-efficient are excellent. Different with other districts, Zone 18 has the largest distribution of apartment and Detached, which shows that this district is newer and also has a concentration of new rich and middle class people, and the age of apartments is younger compared to other non-central districts, while the apartments' price and safety of the Zone 18 district is good when compared horizontally to other non-central districts.

2. House group: The higher the price of Detached in a district, the better the district is for living in. The price of detached houses is a regional trendsetter, representing the choice of the middle class and above families. They are less sensitive to price and more concerned about safety and living environment. So it is good to choose the lower priced Terrace and Semi-D in districts with high Detached prices to enjoy the benefits of the district.

Recommend: Zone 13, 18, 7.
The median price of a Detached in these three districts is twice the price of a Terrace and Semi-D. If people are on a budget and want to consider the living environment and convenience of the district, people can choose Terrace and Semi-D in these three districts.

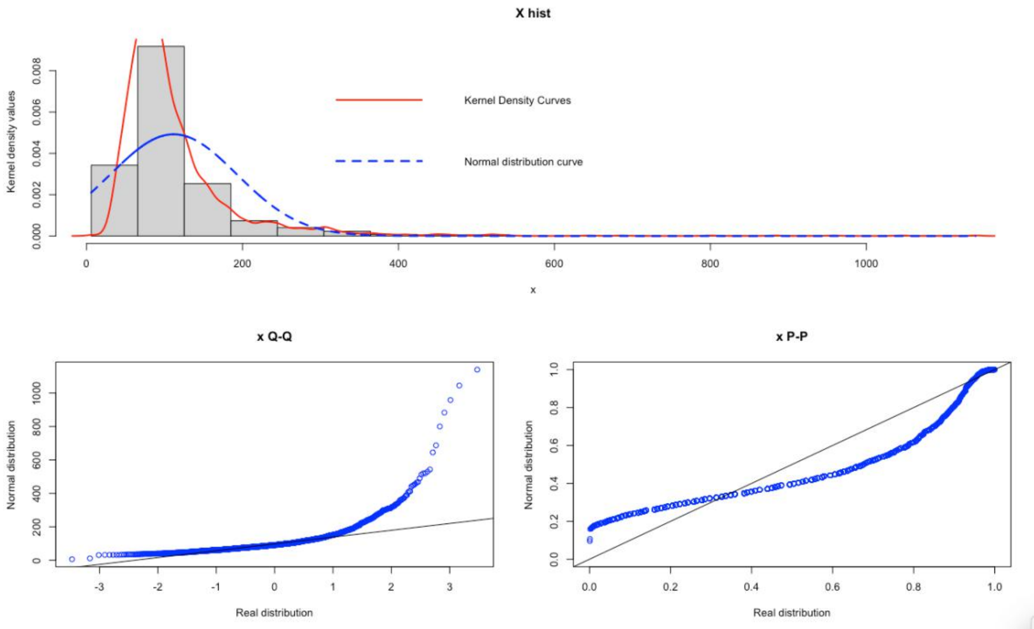
4.Model

4.1 K-Means-Cluster (via area and price to classify)

Step 1: Deal with data

```
dataDublin$area[which(dataDublin$area=="null")]=NA
dataDublin$bed[which(dataDublin$bed=="null")]=NA
data=na.omit(dataDublin)

#S2:convert to num
data$price=as.numeric(gsub('[,€]', '', data$price))
data$area=as.numeric(gsub('[ac,m²]','',data$area))
data$bed=as.numeric(gsub('[,Bed]','',data$bed))
```

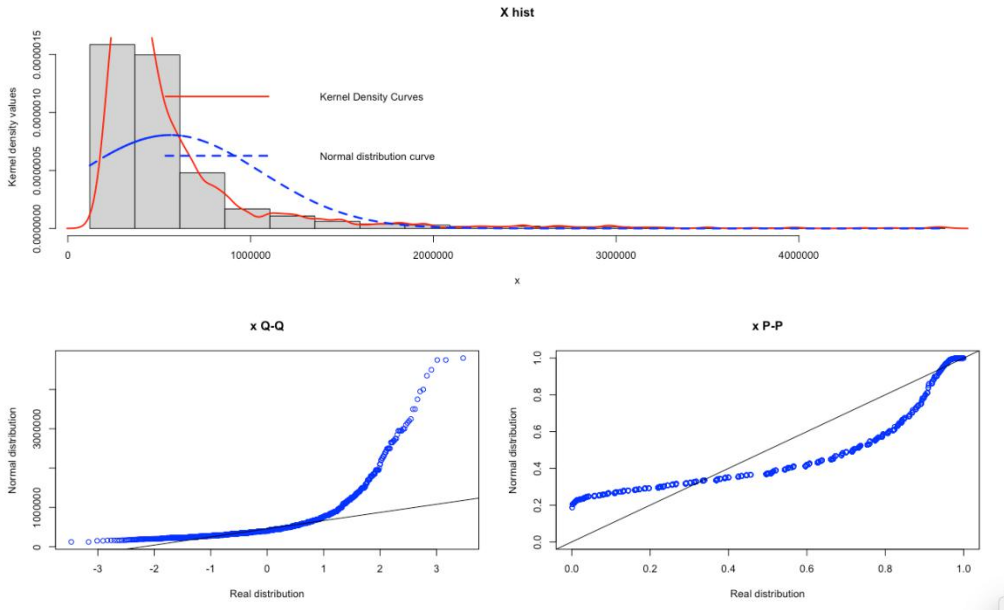
Step 2: Verify the normal distribution of area and price



Both the P-P and Q-Q plots are graphs used to test whether the data conform to a normal distribution. When the data conform to a normal distribution, the points in the P-P plot are approximately in a straight line, while the points in the Q-Q plot are approximately in the vicinity of a straight line. The P-P and Q-Q plots show that the area does not fit a normal distribution.

```
> norm.test(data$area)
[1] "The quantitative results are:  x not follows a normal distribution,  P value = 0 <= 0.05"

        Shapiro-Wilk normality test

data:  x
W = 0.63266, p-value < 0.00000000000000022
```

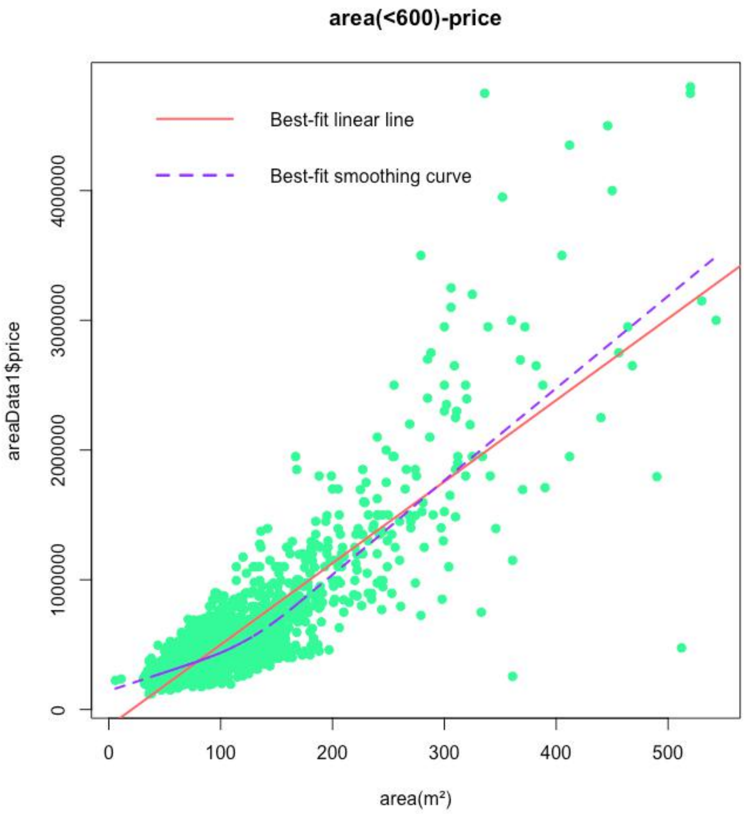
Using quantitative analysis, the W test(Shapiro-Wilk test) for small data samples and KS test(Kolmogorov-Smirnov test) for large data samples. The result is P value < 0.5, area does not fit a normal distribution.




```
> norm.test(data$price)
[1] "The quantitative results are:  x not follows a normal distribution,  P value = 0 <= 0.05"

        Shapiro-Wilk normality test

data:  x
W = 0.59509, p-value < 0.00000000000000022
```

The result is P value < 0.5, price does not fit a normal distribution.

S3: The correlation between price and area

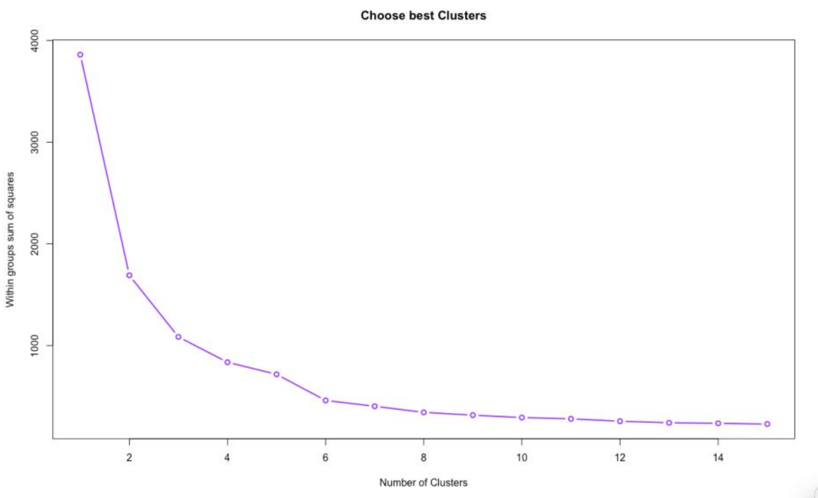


As can see from the graph, as the area increases, so does the price, with most of properties being under 200 sqm and price for under 1 million Euro. The right corner in the graph shows some properties above 4 millions in Dublin.

Step 4: Classification of properties

The gap within groups should be small and the gap between groups should be large. The two variables of size and rent are used to classify the listings.

```
#model- K-means
modeldata=data
k.plot = function(data, nc, seed=1234){

  k = (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){

    set.seed(seed)
    k[i] = kmeans(data, centers=i, iter.max = 100)$tot.withinss
  }
  plot(1:nc, k, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares",col = '#9933FF',
       lwd = 2, main = 'Choose best Clusters')
}

standrad =data.frame(scale(modeldata[,c('area','price')]))
Kplot =k.plot(standrad, nc = 15)
```



As can be seen from the graph, when K=6, it is clear that the curve decreases at a lower rate, which suggests that it would be more appropriate for us to divide the listings into 6 categories. Therefore it is determined that here K=6.

S5: K-Means clustering is performed

```
› set.seed(1234)
› clust = kmeans(x = standrad, centers = 6, iter.max = 100)
› table(clust$cluster)

  1   2   3   4   5   6
610 186   6  29  77 1023
```

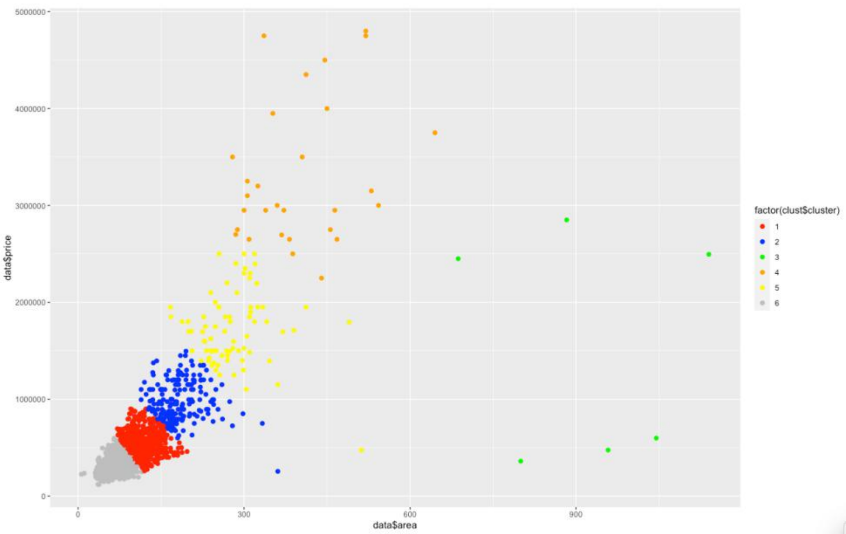
According to the clustering results, the regional distribution in each category is based on the property type situation.

```
table(data$district,clust$cluster)

      1   2   3   4   5   6
1     8   0   0   0   0  74
10    0   0   0   0   0   7
11   24   3   0   0   0  50
12   21   0   0   0   0  58
13   25  10   1   0   9  29
14   52  29   2   1   3  52
15   58   5   1   0   3  70
16   44   6   0   0   0  17
17    3   0   0   0   0  11
18   44  17   0   4  11  37
2    12   4   0   2   3  43
20    5   2   0   0   0   9
22   13   0   0   0   0  36
24   19   1   0   0   0  63
3    32  12   0   1   5  59
4    39  25   0   9  16  69
5    34   1   0   0   1  34
6    44  38   1  12  18  44
6W   20  11   0   0   3   7
7    30   4   0   0   2  60
8    43   7   1   0   2 134
9    40  11   0   0   1  60
```

```
table(data$type,clust$cluster)

                  1   2   3   4   5   6
Apartment        62   7   1   1   3 481
Bungalow         15   6   0   0   1  23
Detached         43  58   3  13  32  10
Duplex           20   2   0   0   0  25
End of Terrace   72  16   0   3   5  91
Semi-D          269  54   1   8  22 105
Terrace         127  42   1   4  13 281
Townhouse         2   1   0   0   1   7
```

The following compare to the average area(m²), average price(€) and average price(€/m²) in each category.

```
aggregate(y,list(clust$cluster),mean)
Group.1   price.€    area.m²     €/m²
      1  539679.3 116.76230 4785.652
      2  978214.8 183.18817 5548.038
      3 1537991.7 918.83333 1750.078
      4 3308448.3 399.79310 8572.173
      5 1717987.0 278.16883 6427.042
      6  325743.0  70.26882 4884.370
```



Apartments are the most distributed in Group 6, followed by Terrace, which is the only group with houses of less than 100 sqm. The price per square meter is slightly higher than in Group 1. Zone 6 with 6 types of clusters, and Zone 6 has the most Group 4,5. Group 4,5 belongs to a large house.

4.2 Logistic regression (predict property's type)

S1: Deal with data

Convert all factor and char types to numeric type. 6W changes to 66. Setting apartment, duplex as apartment type. Others belong to house type. Apartment type is 0, house type is 1.

```
data$price=as.numeric(gsub('[,€]', '', data$price))
data$area=as.numeric(gsub('[ac,m²]','',data$area))
data$bed=as.numeric(gsub('[Bed]','',data$bed))
#change district to numeric, needs to convert 6W to 66
data$district=as.numeric(gsub('[W]','6',data$district))

#Y should 0-1.So change apartment type is 0, house type is 1
data$type=factor(data$type,levels = c("Apartment","Bungalow","Detached","Duplex","End of Terrace",
                                      "Semi-D","Terrace","Townhouse"),
                 labels=c("0","1","1","0","1","1","1","1"))
```

S2: Set train and test data

According 8:2 rule, in total has 2389 data, 1911 data as training data, 478 data as testing data.

```
set.seed(1234)
data_rand=data[order(runif(2389)),]

data_train=data_rand[1:1911, ]
data_test=data_rand[1912:2389, ]
```

S3: Build model

```
Call:
glm(formula = type ~ district + area + bed + latitude + longitude +
    price, family = "binomial", data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9860  -0.3582   0.1550   0.4963   2.4450

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.773e+01  4.549e+01   0.610   0.5421
district     1.969e-03  9.428e-03   0.209   0.8346
area        -3.244e-03  2.011e-03  -1.613   0.1066
bed          2.417e+00  1.511e-01  15.996   <2e-16 ***
latitude    -6.684e-01  9.138e-01  -0.731   0.4645
longitude   -4.260e-01  5.416e-01  -0.787   0.4316
price        8.335e-07  4.425e-07   1.884   0.0596 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1920.1  on 1547  degrees of freedom
Residual deviance: 1174.3  on 1541  degrees of freedom
  (363 observations deleted due to missingness)
AIC: 1188.3

Number of Fisher Scoring iterations: 6
```

As can seen from the Pr of the regression coefficients, the P value for district, latitude, longitude are large (more than 0.05) and neither contributes significantly to the equation. Remove these variables to refit the model and test whether the new model is a good fit.

```
Call:
glm(formula = type ~ area + bed + price, family = "binomial",
    data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9821  -0.3571   0.1537   0.4970   2.4599

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.251e+00  3.164e-01 -16.596   <2e-16 ***
area        -3.166e-03  1.988e-03  -1.592   0.1113
bed          2.435e+00  1.468e-01  16.584   <2e-16 ***
price        7.643e-07  4.289e-07   1.782   0.0748 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1920.1  on 1547  degrees of freedom
Residual deviance: 1175.6  on 1544  degrees of freedom
  (363 observations deleted due to missingness)
AIC: 1183.6

Number of Fisher Scoring iterations: 6
```

The new model reduce some variables. Using Anova() function to compare, p=0.7184 > 0.5, indicates that the new model for the three predictor variables fits as well as the model as the six full predictor variables.

```
> anova(model_reduced,modeldata,test="Chisq")
Analysis of Deviance Table

Model 1: type ~ area + bed + price
Model 2: type ~ district + area + bed + latitude + longitude + price
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      1544     1175.6
2      1541     1174.3  3   1.3455   0.7184
```

S4: Create formulation

Viewing regression coefficients (three predictor variables) and conducting indexation. It can gain the formulation:

Y=0.996838902*area +11.414044577*bed +1.000000764*price

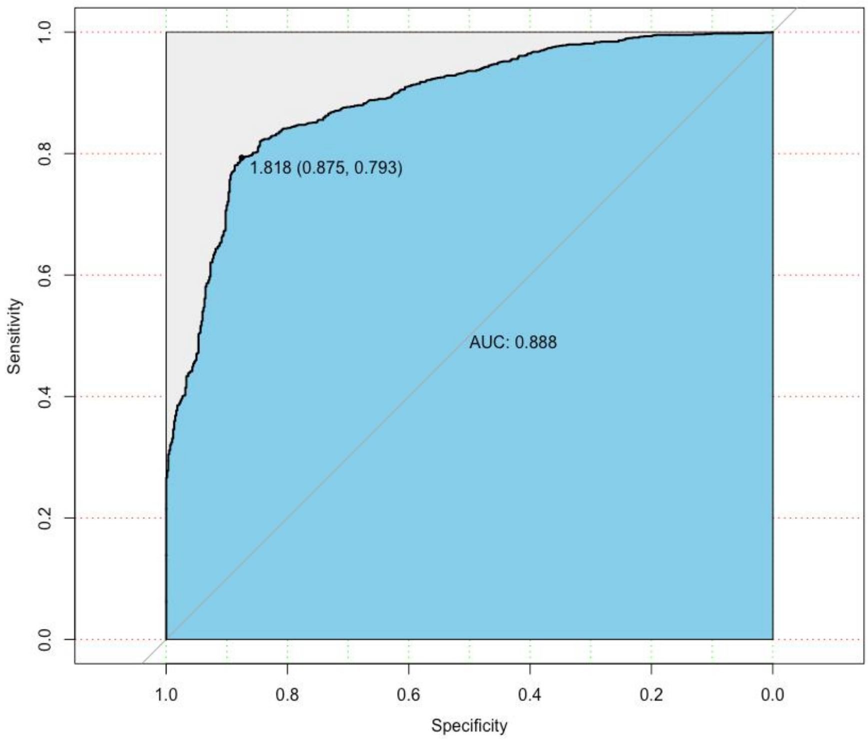
```
> coef(model_reduced)
  (Intercept)          area           bed         price
-5.251345e+00 -3.166105e-03  2.434845e+00  7.643071e-07
> exp(coef(model_reduced))
 (Intercept)        area          bed        price
 0.005240463 0.996838902 11.414044577  1.000000764
```

S5: Verify model

```
             | actual
   predicted |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
   apartment |       103 |        51 |       154 |
             |     0.265 |     0.131 |           |
-------------|-----------|-----------|-----------|
       house |        15 |       220 |       235 |
             |     0.039 |     0.566 |           |
-------------|-----------|-----------|-----------|
Column Total |       118 |       271 |       389 |
-------------|-----------|-----------|-----------|
```

Using testing data to verify the model, shows the accuracy of this model has 83%.

S6: Evaluate model



Using the pROC package, which facilitates the comparison of the two classifiers and also automatically labels the optimal critical point, in the figure below the optimal point FPR=1-TNR=0.875, TPR=0.793 and the AUC=0.888, indicating that the model is a good predictor.

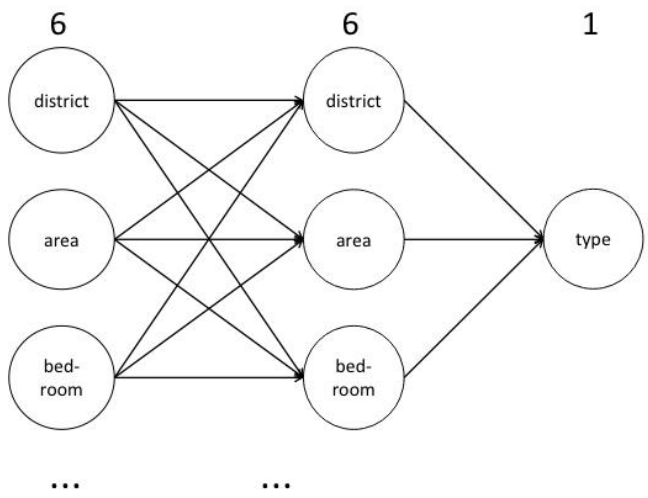
```
> testdata=data.frame(area=c(33,98),bed=c(1,3),price=c(195000,380000))
> testdata$prob=predict(model_reduced,newdata=testdata,type="response")
> testdata
  area bed  price       prob
1   33   1 195000 0.05928889
2   98   3 380000 0.88200488
```

Input two sets of data outside the dataset to see the likelihood of them being house type. The first group shows a high probability of not being a house, the second group shows an 88% probability of being a house type.

4.3 Numpy - CNN (predict property's type)

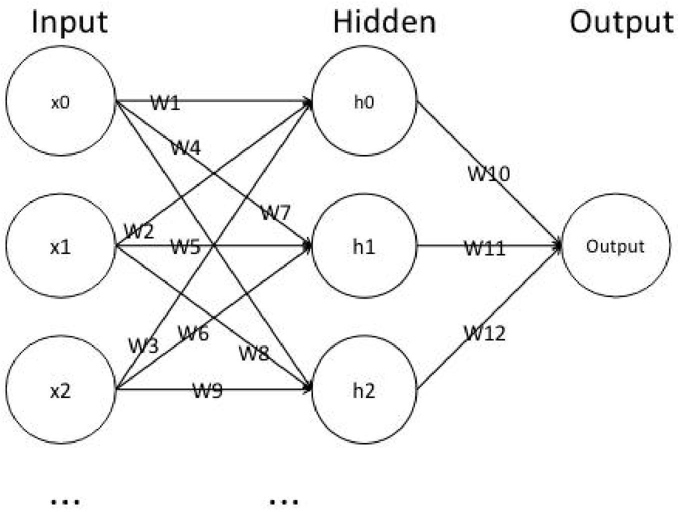
Using 6 variables to build three-layer neural network structure. Because using numpy, convert all factor to numeric, 6W to 66, and delete location variable. Latitude and longitude can replace location. The type converts factor to numeric.
Apartment: 1;
Bungalow: 2;
Detached: 3;
Duplex: 4;
End of Terrace: 5;
Semi-D: 6;
Terrace: 7;
Townhouse:8;
6 variables include district, area, the number of bedroom, latitude, longitude and price. The predicting variable is housing type.

S1: Three-layer neural network structure



The first layer : input layer, has 6 neurons and the required parameters are 6*6+6=42.
The second layer: hidden layer, has 6 neurons and the required parameters are 6+1=7.
The third layer: output layer, has 1 neuron and this is output result.

S2: Gradient descent method



Forward propagation:
h0=x0w1+x1w2+x2w3
H1=x0w4+x1w5+x2w6
H2=x0w7+x1w8+x2w9
Output=h0w10+h1w11+h2w12

Hidden layer's backward propagation:
Loss=1/2(output-y)^2
Output=h0w10+h1w11+h2w12
Setting a is learning rate:
w10=w10-a*∂Loss/∂w10
∂Loss/∂w10=(output-y)*h0

Output layer's backward propagation:
Loss=1/2(output-y)^2
Output=h0w10+h1w11+h2w12
Setting a is learning rate:
w1=w1-a*∂Loss/∂w1
w1=w1-a*(output-y)*w10-x0
∂Loss/∂w1=∂Loss/∂h0*∂h0/∂w1
∂Loss/∂h0=(output-y)*w10
∂h0/∂w1=x0
...
∂h0/∂w2=x1
∂h0/∂w3=x2

Residuals= the weights of the connected lines * the value of the neuron itself * (predicted value - exact value)

S3: Normalization of the 6 features of the data

Each feature is normalized so that each feature is scaled to a value between 0 and 1.

The advantage of this area is model training is more efficient.
Formulation:(x-min)/(max-min)

```
for i in range(6):
    Max =  np.max(housingdata[:,i])
    Min = np.min(housingdata[:,i])
    housingdata[:,i]=(housingdata[:,i]-Min)/(Max-Min)
```

S4: Set training and testing data

Training data is 80%, and testing data is 20%

```
Splitdata = round(len(housingdata)*0.8)
Train = housingdata[:Splitdata]
Test = housingdata[Splitdata:]
return Train,Test
```

S5: design model and update gradient

Based on Numpy mechanism, the gradient calculation can be implemented more quickly. Via (z-y)*hidden1 can gain a 6 dimensional vector, with each component representing the gradient in that dimension. Measuring the goodness of the model through the loss function.

```
class Model_Config(object):
    def __init__(self,firstnetnum,secondnetnum):
        np.random.seed(1)
        self.w0 = np.random.randn(firstnetnum*secondnetnum,1).reshape(firstnetnum,secondnetnum)
        self.w1 = np.random.randn(secondnetnum,1)
        self.b0 = np.random.randn(firstnetnum,1).reshape(1,firstnetnum)
        self.b1 = np.random.randn(1,1)

    def forward(self,x):
        hidden1 = np.dot(x,self.w0)+self.b0
        y = np.dot(hidden1,self.w1)+self.b1
        return hidden1,y

    def loss(self,z,y):
        error = z-y
        cost = error*error
        avg_cost = np.mean(cost)
        return avg_cost

    def back(self,x,y):
        hidden1,z = self.forward(x)
        gradient_w1 = (z-y)*hidden1
        gradient_w1 = np.mean(gradient_w1,axis=0)
        gradient_w1 = gradient_w1[:,np.newaxis]
        gradient_b1 = (z-y)
        gradient_b1 = np.mean(gradient_b1)
        gradient_w0 = np.zeros(shape=(6,6))
        for i in range(len(x)):
            data = x[i,:]
            data = data[:,np.newaxis]
            print("data.shape",data.shape)
            w1 = self.w1.reshape(1,6)
```

Updating the gradient, firstly, moving a small step in the opposite direction of the gradient to find the next point and observe the change in the loss function. To make sure the loss function whether is decreasing gradually.

```
def update(self,gradient_w1,gradient_b1,gradient_w0,gradient_b0,learning_rate):
    self.w1 = self.w1-learning_rate*gradient_w1
    self.b1 = self.b1-learning_rate*gradient_b1
    self.w0 = self.w0-learning_rate*gradient_w0
    self.b0 = self.b0-learning_rate*gradient_b0
```

S6: train data

```python
def train(self,epoch_num,x,y,learning_rate):

    losses=[]
    for i in range(epoch_num):
        _,z = self.forward(x)
        avg_loss = self.loss(z,y)
        gradient_w1,gradient_b1,gradient_w0,gradient_b0 = self.back(x,y)
        self.update(gradient_w1,gradient_b1,gradient_w0,gradient_b0,learning_rate)
        losses.append(avg_loss)
        if(i%20==0):
            print("iter:{},loss:{}".format(i,avg_loss))
```

S7: the result of training

Setting learning rate is 0.001. The training result is when epoch_num is between 0 and 100, the loss value fall rapidly. 200 later, the loss is stable, around 4.8. The model predicts the housing data is good.

```
iter:0,loss:87.94726762792936
iter:20,loss:39.00290638352118
iter:40,loss:20.068833742000372
iter:60,loss:11.566488421318232
iter:80,loss:7.7395363838971125
iter:100,loss:6.085797079544049
iter:120,loss:5.400408375430163
iter:140,loss:5.123514746785168
iter:160,loss:5.011329056320356
iter:180,loss:4.963483817950339
iter:200,loss:4.940290292680953
iter:220,loss:4.926469747370751
iter:240,loss:4.916240218044652
iter:260,loss:4.907436288330109
iter:280,loss:4.899251382228967
iter:300,loss:4.891385207323129
iter:320,loss:4.883725128040201
iter:340,loss:4.876227854256725
```