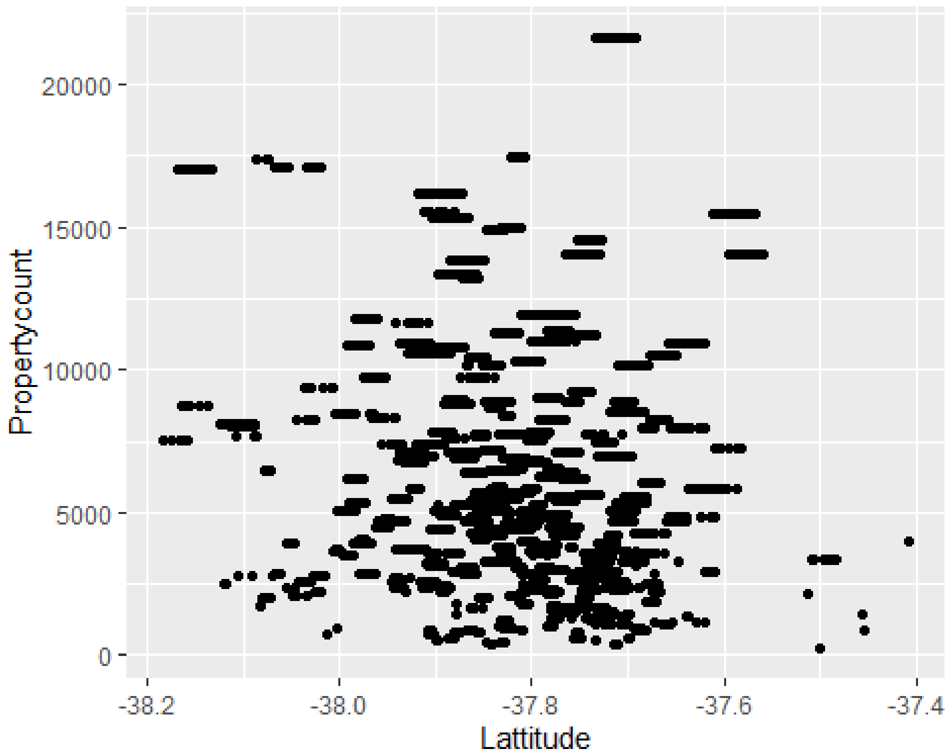
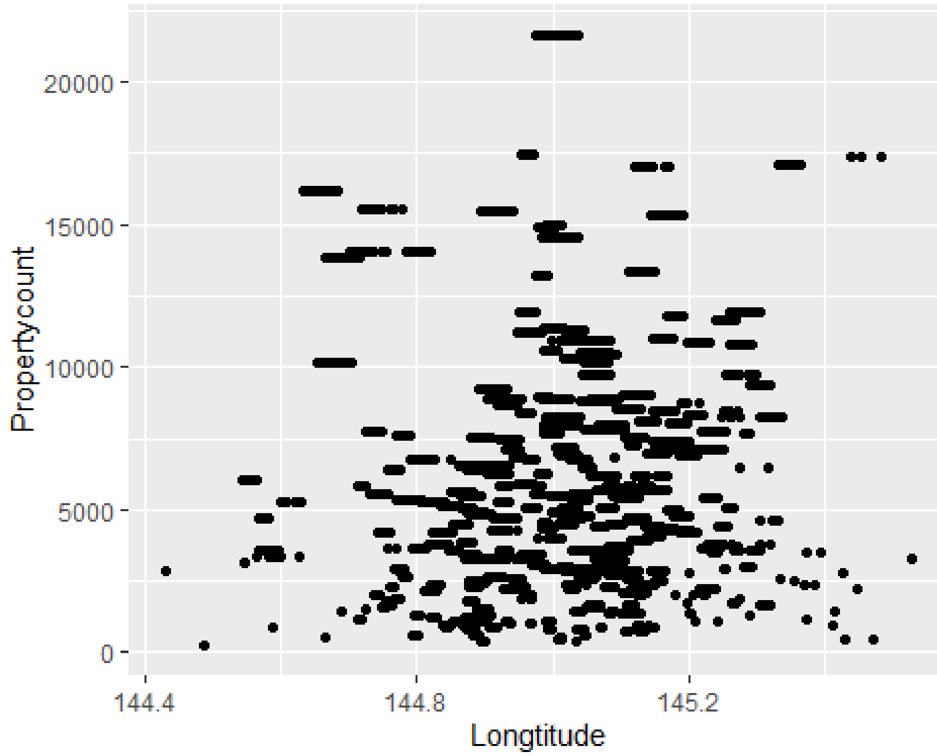
Individual Assignment4- Appendix

Appendix - 1

For latitude between -37.6 and -38.0 there are many houses,which would mean that it’s an ideal residential location.

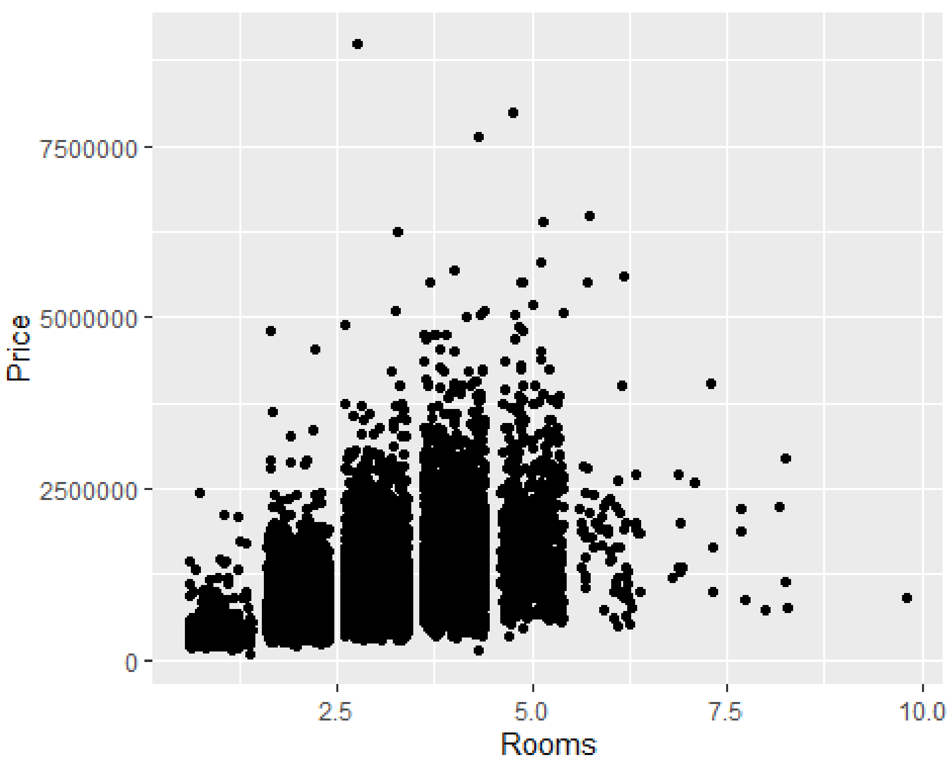


But when we talk about longitude we can see that concentration is high between 144.8 to 145.1 (This is

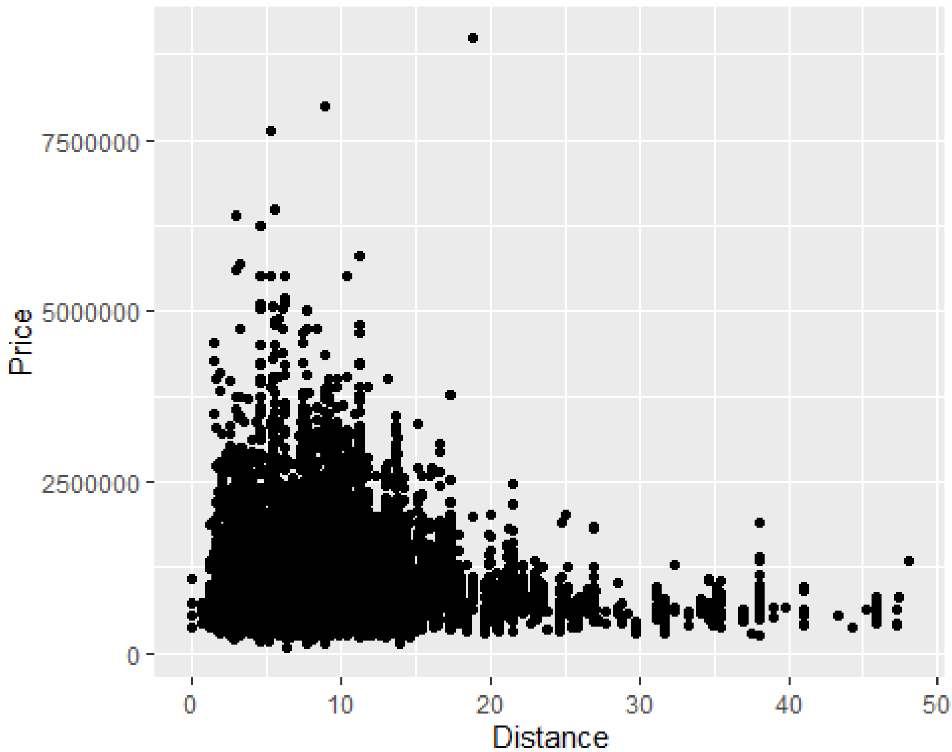


close to the location of CBD)

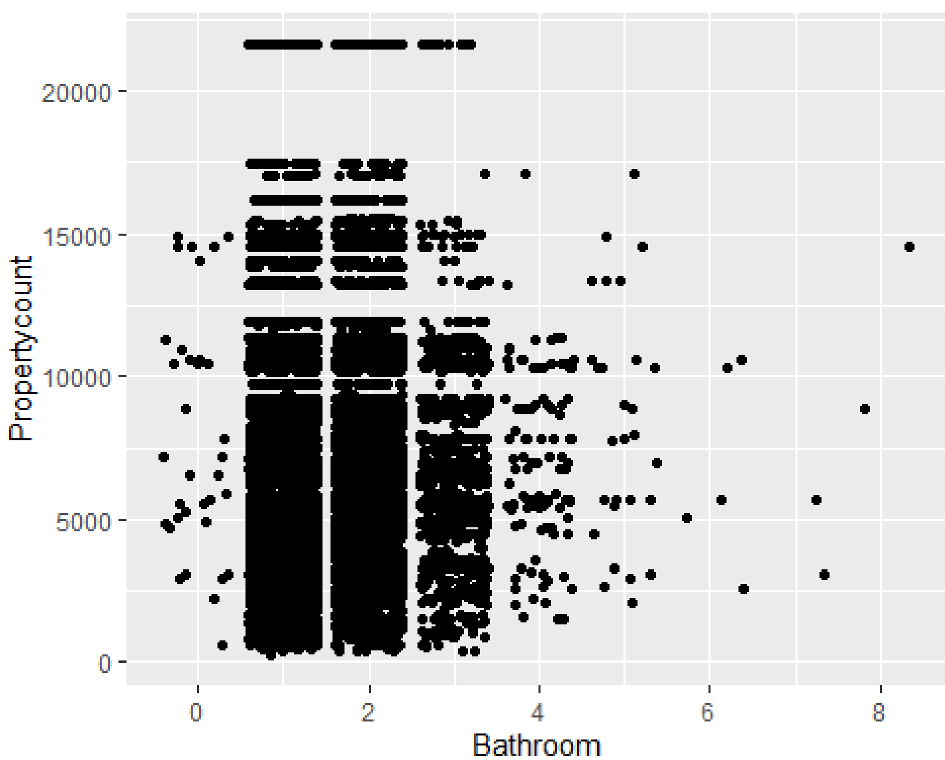
Given number of rooms between 3-5, there seems to be houses in all price ranges. As the number of rooms increases we can find higher price range houses. For 1-2 room houses there are lower price range houses



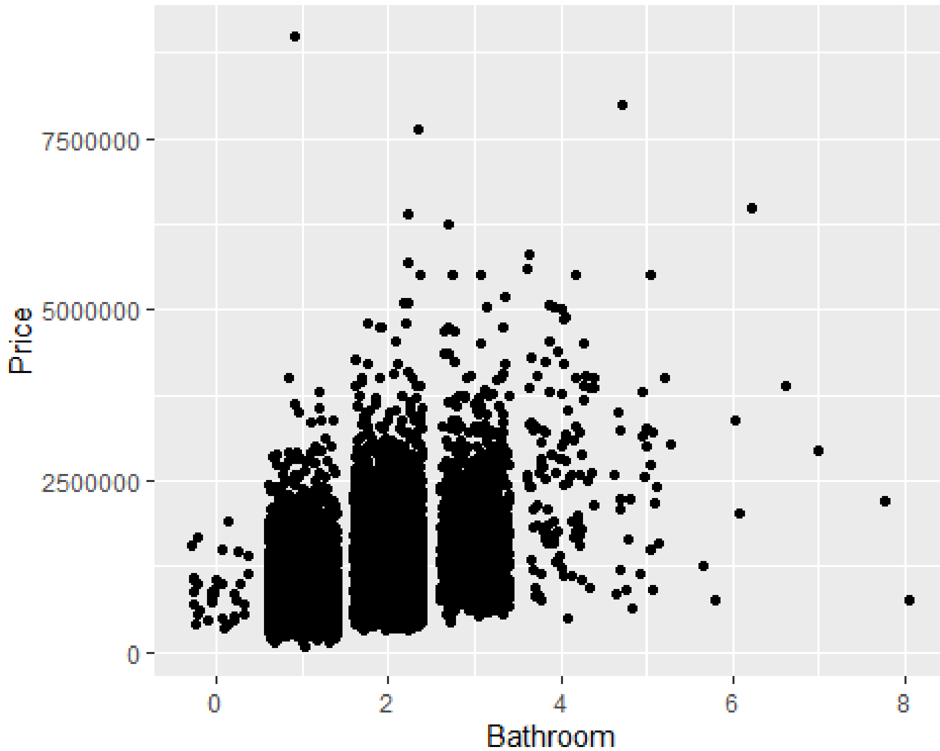
Closer to CBD, there are a number of houses with varying price range Farther away, the number reduces drastically and the price too



As the number of bathrooms becomes more than 4, the number of houses decreases drastically.

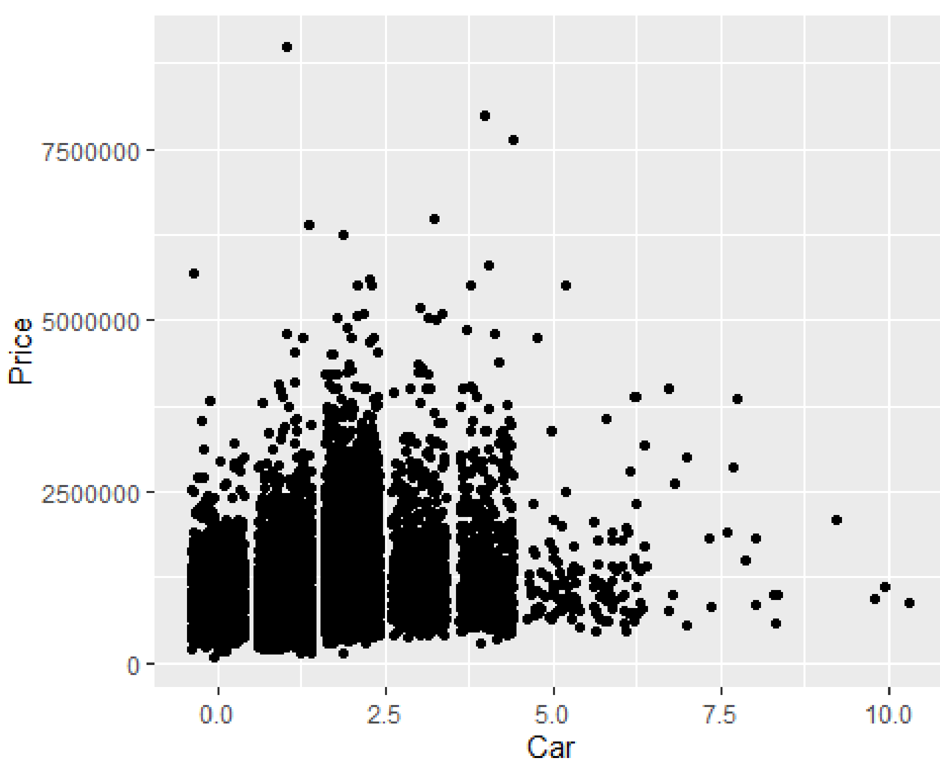


Initially, the price range increases as the number of bathrooms increases.

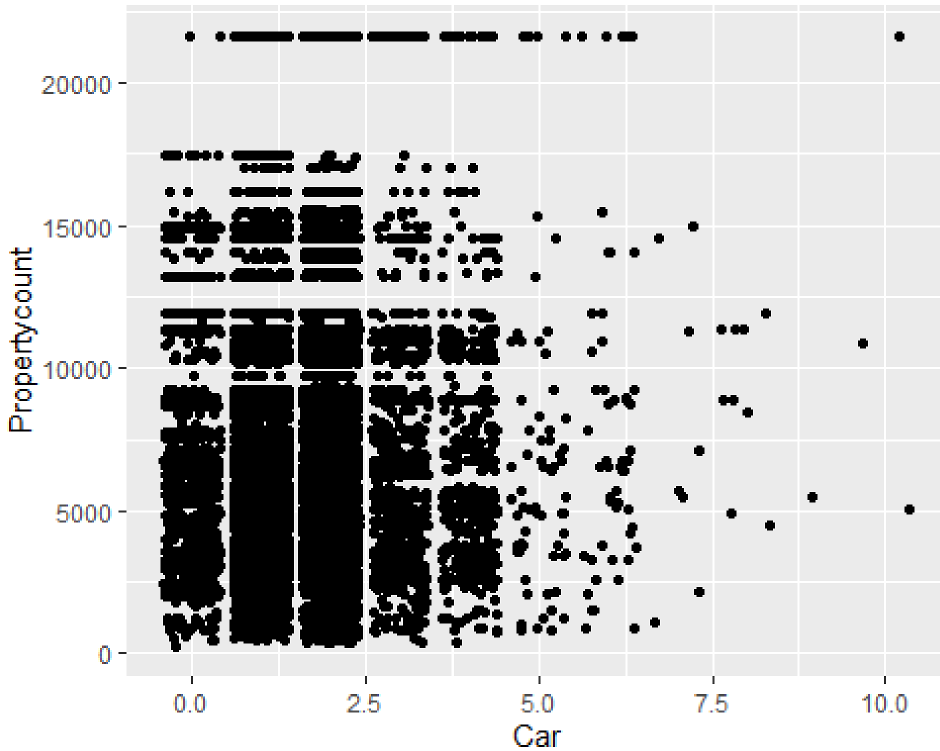


People with 2 or 4 cars have been seen to be buying the costliest of houses As people have more number of houses the price decreases

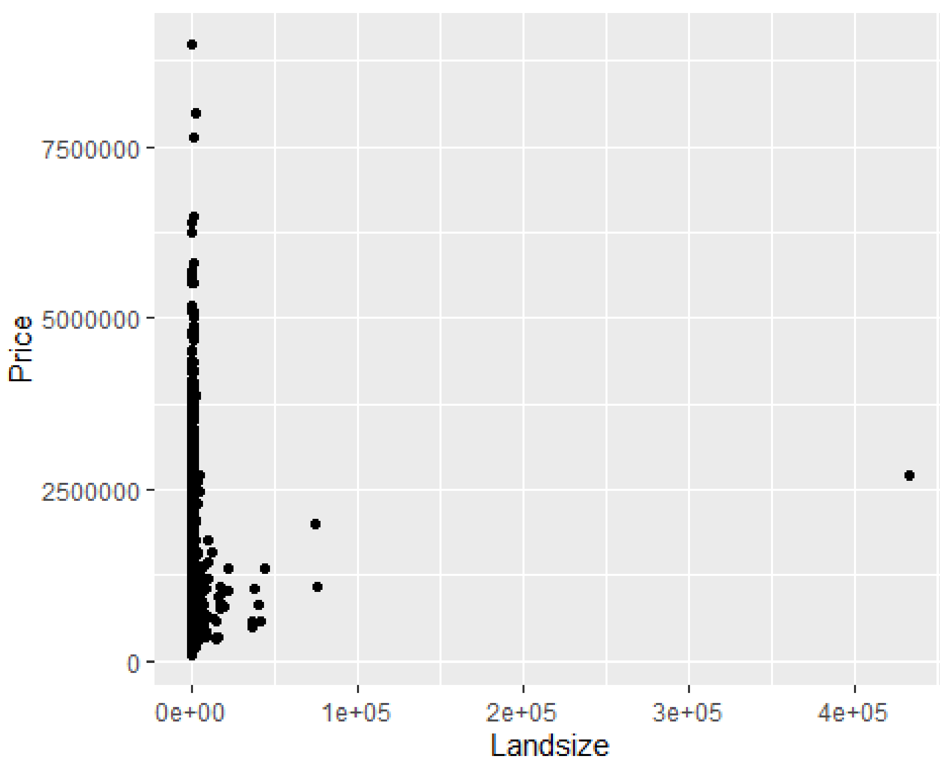
```
## Warning: Removed 62 rows containing missing values (geom_point).
```




```
## Warning: Removed 62 rows containing missing values (geom_point).
```

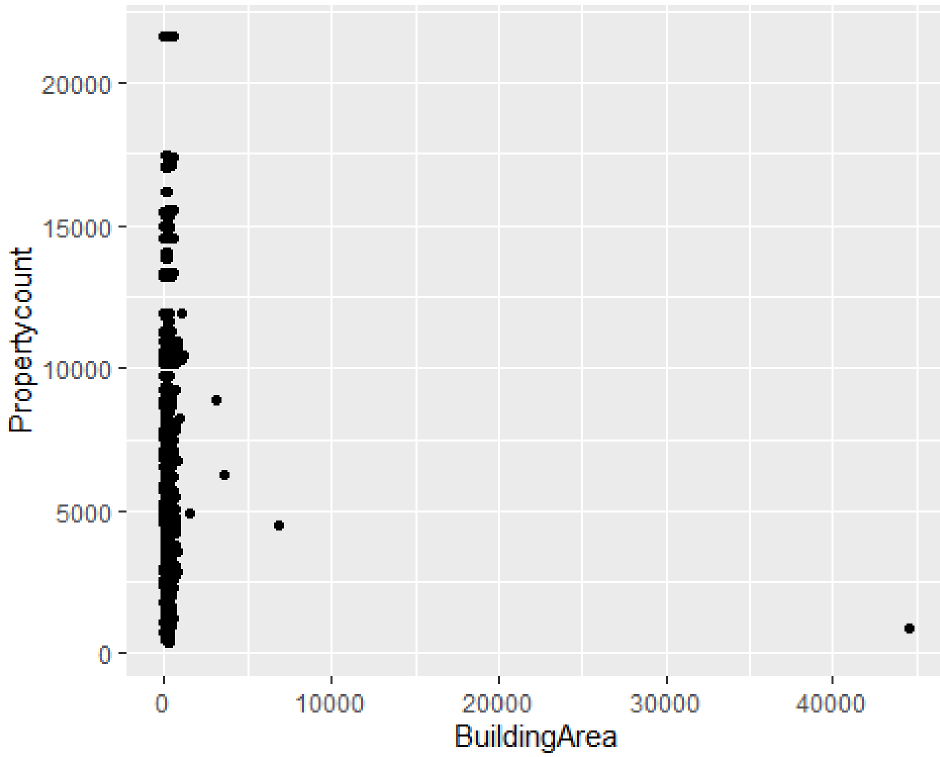


Landsize doesn’t distinctly affect the price of a house

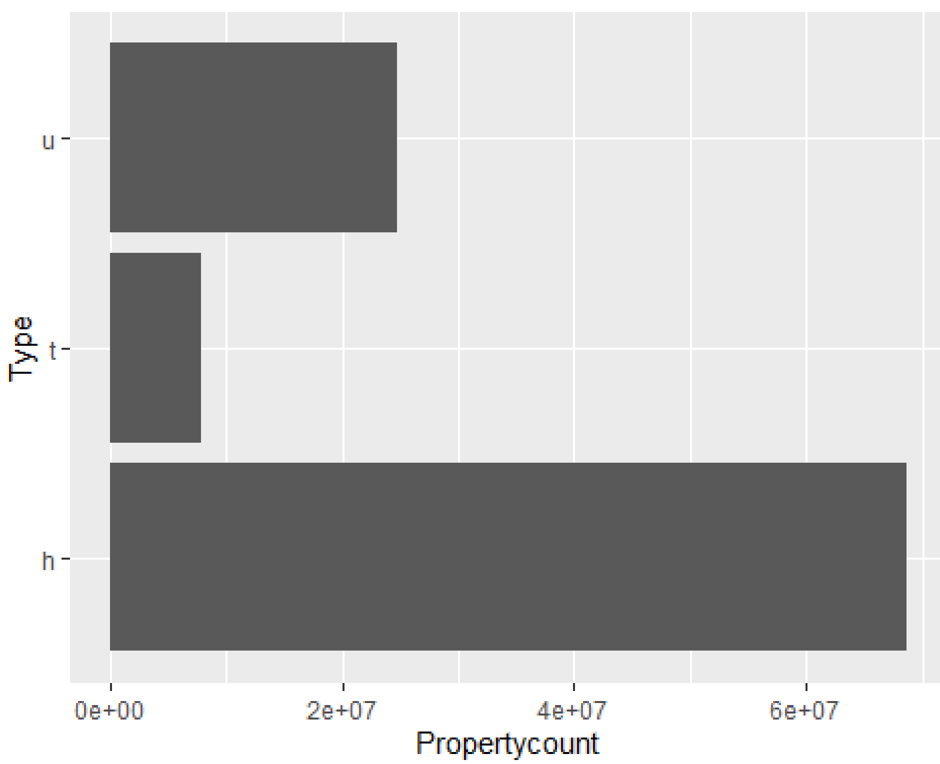


As building area increases we can see a slight shift of the range of the houses available

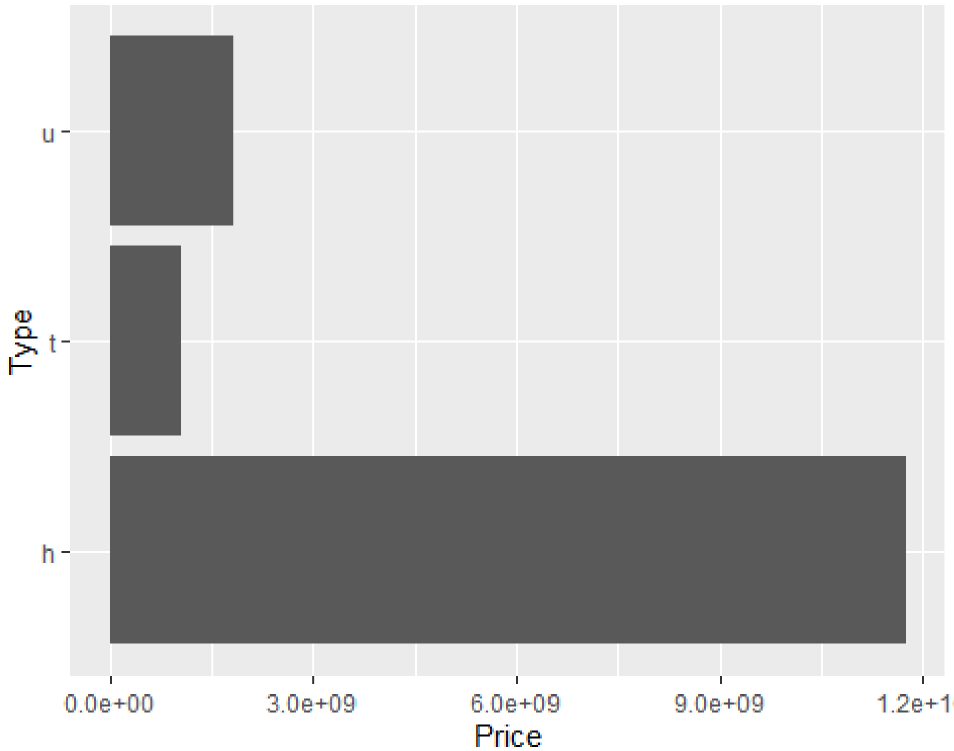
```
## Warning: Removed 6450 rows containing missing values (geom_point).
```



Appendix - 2



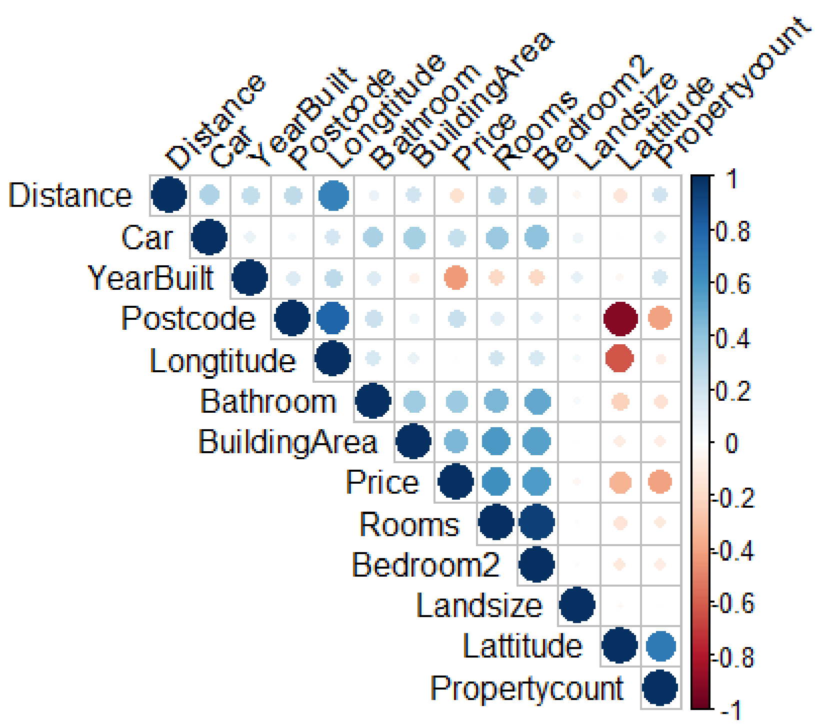
Appendix - 3



Appendix - 4

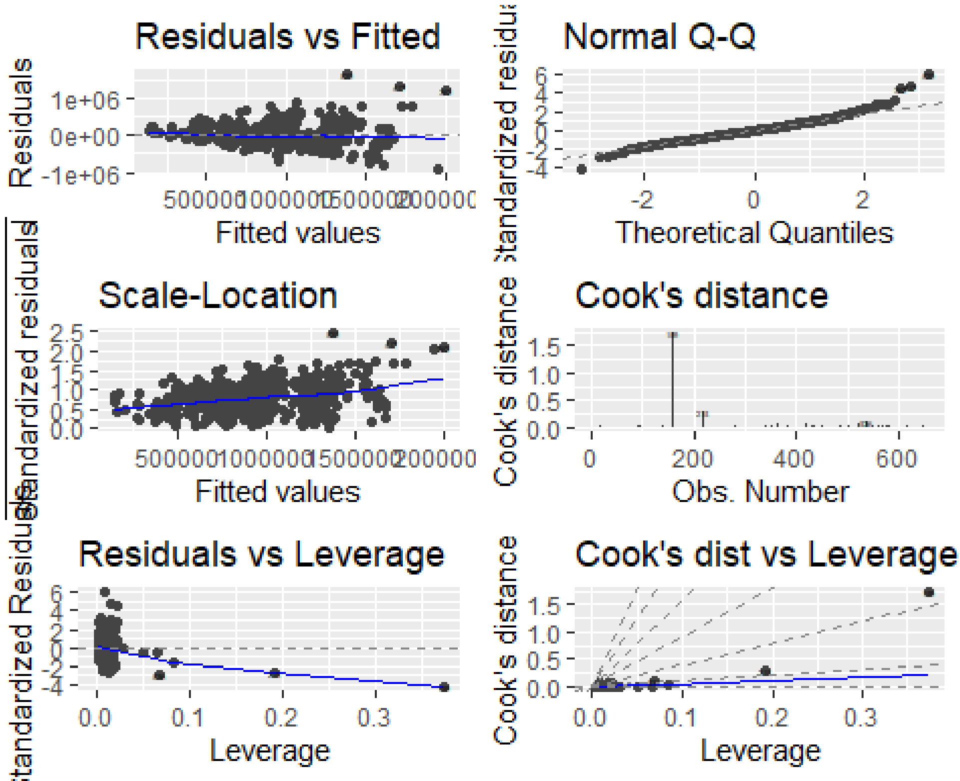
Appendix -4b

```
## Warning: package 'corrplot' was built under R version 4.0.4

## corrplot 0.84 loaded
```



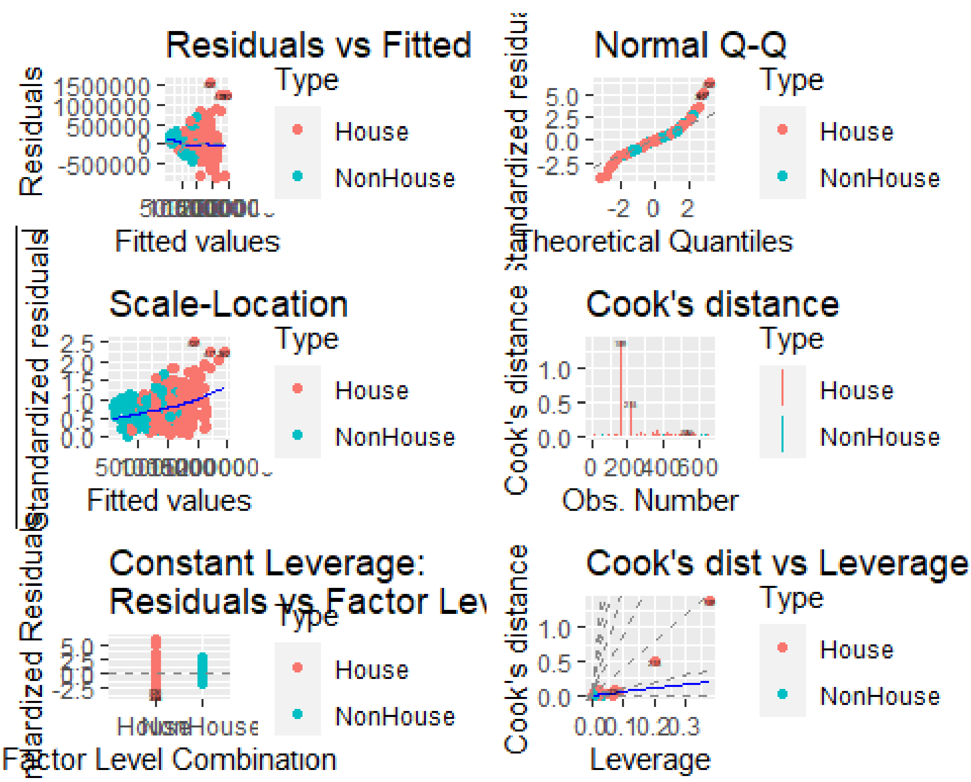
Appendix -5

```
##
## Call:
## lm(formula = Price ~ BuildingArea + Postcode + Rooms + Bathroom +
##     Distance, data = dataset, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -895579 -165361   -9515  149806 1618878
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7536414.8   883638.9  -8.529  < 2e-16 ***
## BuildingArea     1021.7      180.2   5.670 2.15e-08 ***
## Postcode         2553.4      288.3   8.856  < 2e-16 ***
## Rooms          277024.3    15792.7  17.541  < 2e-16 ***
## Bathroom        10256.4    19039.5   0.539     0.59
## Distance       -42889.9     2761.4 -15.532  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 271400 on 652 degrees of freedom
## Multiple R-squared:  0.576,  Adjusted R-squared:  0.5727
## F-statistic: 177.1 on 5 and 652 DF,  p-value: < 2.2e-16

## Warning: `arrange_()` was deprecated in dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
```




```
##
## Call:
## lm(formula = Price ~ BuildingArea + Postcode + Rooms + Bathroom +
```

```
##     Distance + Type, data = dataset, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -853455 -160250   -6818  135489 1544108
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8054081.4   822855.6  -9.788  < 2e-16 ***
## BuildingArea      879.3      168.1   5.232 2.26e-07 ***
## Postcode         2802.1      269.1  10.413  < 2e-16 ***
## Rooms          199133.0    16550.6  12.032  < 2e-16 ***
## Bathroom        47126.9    18062.3   2.609  0.00929 **
## Distance       -40944.8     2573.7 -15.909  < 2e-16 ***
## TypeNonHouse  -252053.4    24744.3 -10.186  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252300 on 651 degrees of freedom
## Multiple R-squared:  0.6343, Adjusted R-squared:  0.6309
## F-statistic: 188.2 on 6 and 651 DF,  p-value: < 2.2e-16

autoplot(model.2, which = 1:6, label.size = 1,  colour = 'Type')
```



Appendix - 6

```
summary(model.1)

##
## Call:
## lm(formula = Price ~ BuildingArea + Postcode + Rooms + Bathroom +
##     Distance, data = dataset, na.action = na.exclude)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.12522 -0.19421  0.00462  0.18949  0.96369 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -39.27983    7.16820  -5.480 6.09e-08 ***
## BuildingArea   0.19586    0.02490   7.865 1.54e-14 ***
## Postcode       6.46265    0.89243   7.242 1.25e-12 ***
## Rooms          0.78034    0.04084  19.107  < 2e-16 ***
## Bathroom       0.01547    0.03295   0.470    0.639    
## Distance      -0.33527    0.02148 -15.608  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2761 on 652 degrees of freedom
## Multiple R-squared:  0.6443, Adjusted R-squared:  0.6415 
## F-statistic: 236.2 on 5 and 652 DF,  p-value: < 2.2e-16
```

```
summary(model.2)
```

```
## 
## Call:
## lm(formula = Price ~ BuildingArea + Postcode + Rooms + Bathroom + 
##     Distance + Type, data = dataset, na.action = na.exclude)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.29296 -0.16896  0.00522  0.17394  0.93493 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -46.49184    6.42316  -7.238 1.29e-12 ***
## BuildingArea   0.14669    0.02255   6.504 1.56e-10 ***
## Postcode       7.42083    0.80009   9.275  < 2e-16 ***
## Rooms          0.55974    0.04025  13.906  < 2e-16 ***
## Bathroom       0.09956    0.03012   3.305    0.001 ** 
## Distance      -0.31608    0.01923 -16.435  < 2e-16 ***
## TypeNonHouse  -0.31771    0.02457 -12.930  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2465 on 651 degrees of freedom
## Multiple R-squared:  0.7169, Adjusted R-squared:  0.7143 
## F-statistic: 274.8 on 6 and 651 DF,  p-value: < 2.2e-16
```

```
lm.beta(model.1)
```

```
## BuildingArea     Postcode        Rooms     Bathroom     Distance 
##   0.23865221   0.17512561   0.60317432   0.01271967  -0.38180199
```

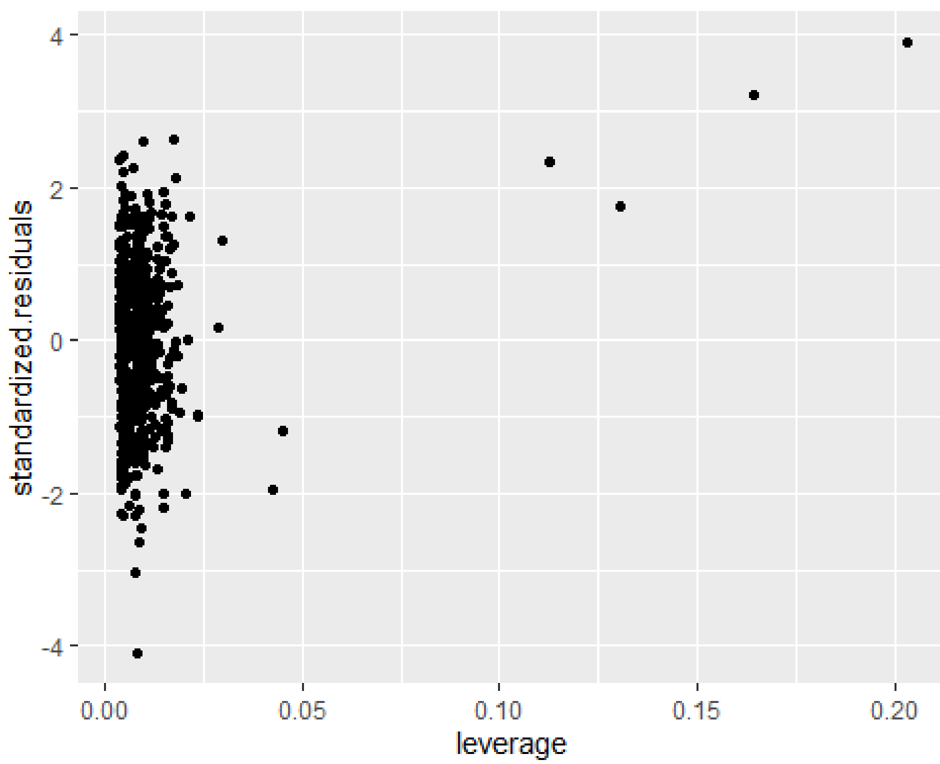
```
lm.beta(model.2)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
```

```
## BuildingArea     Postcode        Rooms     Bathroom     Distance TypeNonHouse
##   0.17874047   0.20109032   0.43266151   0.08183822  -0.35994886           NA
```

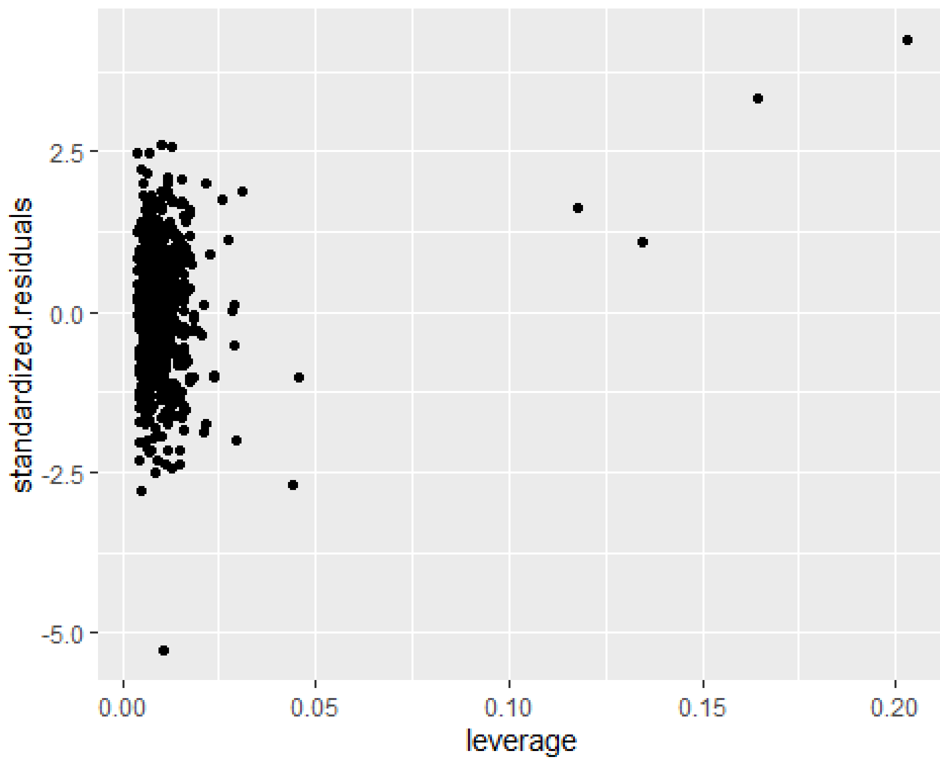
Appendix -7

```
dataset$residuals<-resid(model.1)  #page 323
dataset$standardized.residuals<- rstandard(model.1) #page 324
dataset$studentized.residuals<-rstudent(model.1)
dataset$cooks.distance<-cooks.distance(model.1)
dataset$dfbeta<-dfbeta(model.1)
dataset$dffit<-dffits(model.1)
dataset$leverage<-hatvalues(model.1)
dataset$covariance.ratios<-covratio(model.1)


e <- ggplot(dataset, aes(y=standardized.residuals, x=leverage))
e+geom_point()
```




```
summary(dataset$cooks)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0001239 0.0005914 0.0031541 0.0016212 0.6486616

dataset$residuals<-resid(model.2)  #page 323
dataset$standardized.residuals<- rstandard(model.2) #page 324
dataset$studentized.residuals<-rstudent(model.2)
dataset$cooks.distance<-cooks.distance(model.2)
dataset$dfbeta<-dfbeta(model.2)
dataset$dffit<-dffits(model.2)
dataset$leverage<-hatvalues(model.2)
dataset$covariance.ratios<-covratio(model.2)
```

```
e <- ggplot(dataset, aes(y=standardized.residuals, x=leverage))
e+geom_point()
```




```
summary(dataset$cooks)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0001281 0.0005717 0.0030707 0.0016027 0.6570990
```

Appendix-8

```
vif(model.1)

## BuildingArea     Postcode        Rooms     Bathroom     Distance
##     1.687631     1.071840     1.826461     1.344159     1.096637

1/vif(model.1)

## BuildingArea     Postcode        Rooms     Bathroom     Distance
##    0.5925467    0.9329750    0.5475068    0.7439596    0.9118790

mean(vif(model.1))

## [1] 1.405346

vif(model.2)

## BuildingArea     Postcode        Rooms     Bathroom     Distance         Type
##     1.737006     1.081114     2.226409     1.409876     1.103206     1.483621

1/vif(model.2)

## BuildingArea     Postcode        Rooms     Bathroom     Distance         Type
##    0.5757032    0.9249720    0.4491538    0.7092822    0.9064491    0.6740268

mean(vif(model.2))
```

```
## [1] 1.506872
```

Appendix -9

```
confint(model.2)

##                      2.5 %      97.5 %
## (Intercept)  -59.10445776 -33.8792250
## BuildingArea   0.10240145   0.1909742
## Postcode       5.84975658   8.9918942
## Rooms          0.48070195   0.6387792
## Bathroom       0.04041596   0.1587085
## Distance      -0.35384635  -0.2783171
## TypeNonHouse  -0.36595389  -0.2694600

exp(confint(model.2))

##                     2.5 %       97.5 %
## (Intercept)  2.144175e-26 1.933925e-15
## BuildingArea 1.107828e+00 1.210428e+00
## Postcode     3.471499e+02 8.037667e+03
## Rooms        1.617209e+00 1.894167e+00
## Bathroom     1.041244e+00 1.171996e+00
## Distance     7.019828e-01 7.570567e-01
## TypeNonHouse 6.935348e-01 7.637918e-01
```

Appendix -10

```
anova(model.1,model.2)

## Analysis of Variance Table
##
## Model 1: Price ~ BuildingArea + Postcode + Rooms + Bathroom + Distance
## Model 2: Price ~ BuildingArea + Postcode + Rooms + Bathroom + Distance +
##     Type
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1    652 49.708
## 2    651 39.550  1    10.158 167.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Appendix -11

Price prediction

```
##        1
## 13.51889

exp(log_price)

##        1
## 743328.8

log_newdata

##   BuildingArea Postcode    Rooms  Bathroom Distance
## 1      4.60517 8.024862 1.098612 0.6931472 2.484907

log_newdata$Type='House'
```

```r
log_price=predict(model.2, log_newdata)
exp(log_price)

##        1
## 833423.7

log_newdata$Type='NonHouse'

log_price=predict(model.2, log_newdata)
exp(log_price)

##        1
## 606579.1
```