BDA Assignment 2

Lalith Venkat Perugu

11/01/2021

Loading the Melbourne data set.

```
housingdata <- read.csv("C:/Program Files/R/melbourne_housing_data.csv")
```

Lets see the structure of the dataset to know the no of objects and variables present

```
str(housingdata)
```

```
## 'data.frame':    48433 obs. of  14 variables:
##  $ X            : int  1 2 3 4 5 6 7 8 10 11 ...
##  $ Suburb       : chr  "Abbotsford" "Abbotsford" "Abbotsford" "Aberfeldie" ...
##  $ Address      : chr  "49 Lithgow St" "59A Turner St" "119B Yarra St" "68 Vida St" ...
##  $ Rooms        : int  3 3 3 3 2 2 2 3 3 3 ...
##  $ Type         : chr  "h" "h" "h" "h" ...
##  $ Price        : int  1490000 1220000 1420000 1515000 670000 530000 540000 715000 1925000 515000 ..
##  $ Method       : chr  "S" "S" "S" "S" ...
##  $ SellerG      : chr  "Jellis" "Marshall" "Nelson" "Barry" ...
##  $ Date         : chr  "1/04/2017" "1/04/2017" "1/04/2017" "1/04/2017" ...
##  $ Postcode     : int  3067 3067 3067 3040 3042 3042 3042 3042 3206 3020 ...
##  $ Regionname   : chr  "Northern Metropolitan" "Northern Metropolitan" "Northern Metropolitan" "West
##  $ Propertycount: int  4019 4019 4019 1543 3464 3464 3464 3464 3280 2185 ...
##  $ Distance     : num  3 3 3 7.5 10.4 10.4 10.4 10.4 3 10.5 ...
##  $ CouncilArea  : chr  "Yarra City Council" "Yarra City Council" "Yarra City Council" "Moonee Valley
```

```
summary(housingdata)
```

```
##        X            Suburb            Address              Rooms
##  Min.   :    1   Length:48433       Length:48433       Min.   : 1.000
##  1st Qu.:15797   Class :character   Class :character   1st Qu.: 2.000
##  Median :31587   Mode  :character   Mode  :character   Median : 3.000
##  Mean   :31562                                         Mean   : 3.072
##  3rd Qu.:47365                                         3rd Qu.: 4.000
##  Max.   :63021                                         Max.   :31.000
##      Type               Price             Method            SellerG
##  Length:48433       Min.   :   85000   Length:48433       Length:48433
##  Class :character   1st Qu.:  620000   Class :character   Class :character
```

```
##  Mode  :character   Median :  830000   Mode  :character   Mode  :character
##                     Mean   :  997898
##                     3rd Qu.: 1220000
##                     Max.   :11200000
##      Date              Postcode     Regionname        Propertycount
##  Length:48433       Min.   :3000   Length:48433       Min.   :   39
##  Class :character   1st Qu.:3051   Class :character   1st Qu.: 4280
##  Mode  :character   Median :3103   Mode  :character   Median : 6567
##                     Mean   :3123                      Mean   : 7566
##                     3rd Qu.:3163                      3rd Qu.:10412
##                     Max.   :3980                      Max.   :21650
##     Distance    CouncilArea
##  Min.   : 0.0   Length:48433
##  1st Qu.: 7.0   Class :character
##  Median :11.7   Mode  :character
##  Mean   :12.7
##  3rd Qu.:16.7
##  Max.   :55.8
```

Task A

Hypothesis 1

one sample Test

From the housing data set we take the price variable and lets define the hypothesis on house prices with respective to the house types h and u. Lets go with the Z test as we have the mean and standard deviation for the house prices.

1 Defining the hypothesis

Null hypothesis Ho: The Average price of the houses with rescpective to the house type h is equal to the Average price of the houses with house type u

i.e Ho = mu1-mu2=0

Alternate Hypothesis H1: The Average price of the houses with rescpective to the house type h is not equal to the Average price of the houses with house type u

i.e H1: mu1-mu1 not equal to zero

2 State Alpha:

Lets take the significance level as 0.05

3 Confidence level = 95

4 Decision Rule:

If the z value is less than -1.96 or greater than 1.96,Reject the null hypothesis.

Test statestic

We use the z test method as we have the sample size greater than 30 , and also the standard deviation is known.

Here we are using two samples of different house type prices.

```
typeh<-subset(housingdata$Price, housingdata$Type=="h")
length(typeh)
```

```
## [1] 34161
```

```
mean(typeh)
```

```
## [1] 1110587
```

```
sd(typeh)
```

```
## [1] 637894.1
```

```
typeu<-subset(housingdata$Price,housingdata$Type=="u")
mean(typeu)
```

```
## [1] 630105.3
```

```r
sd(typeu)
```

```
## [1] 286087.8
```

```r
length(typeu)
```

```
## [1] 9292
```

```r
set.seed(100)
sampleoftypeh<-sample(typeh,50,replace = FALSE)
sampleoftypeu<-sample(typeh,50,replace = FALSE)
```

```r
mu1=mean(sampleoftypeh)
mu2=mean(sampleoftypeu)
mu=mu1-mu2
mu
```

```
## [1] 137530
```

```r
z.test(sampleoftypeh,sampleoftypeu,alternative = "two.sided",mu=0,sigma.x = sd(sampleoftypeh),sigma.y =
```

```
##
##  Two-sample z-Test
##
## data:  sampleoftypeh and sampleoftypeu
## z = 1.2486, p-value = 0.2118
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -78352.55 353412.55
## sample estimates:
## mean of x mean of y
##   1152300   1014770
```

Conclusion: From the above two sample z test using z.test function we can notice that z value is lies between -1.96 and +1.96, Which means we should not reject the null hypothesis according to the Decision Rule.

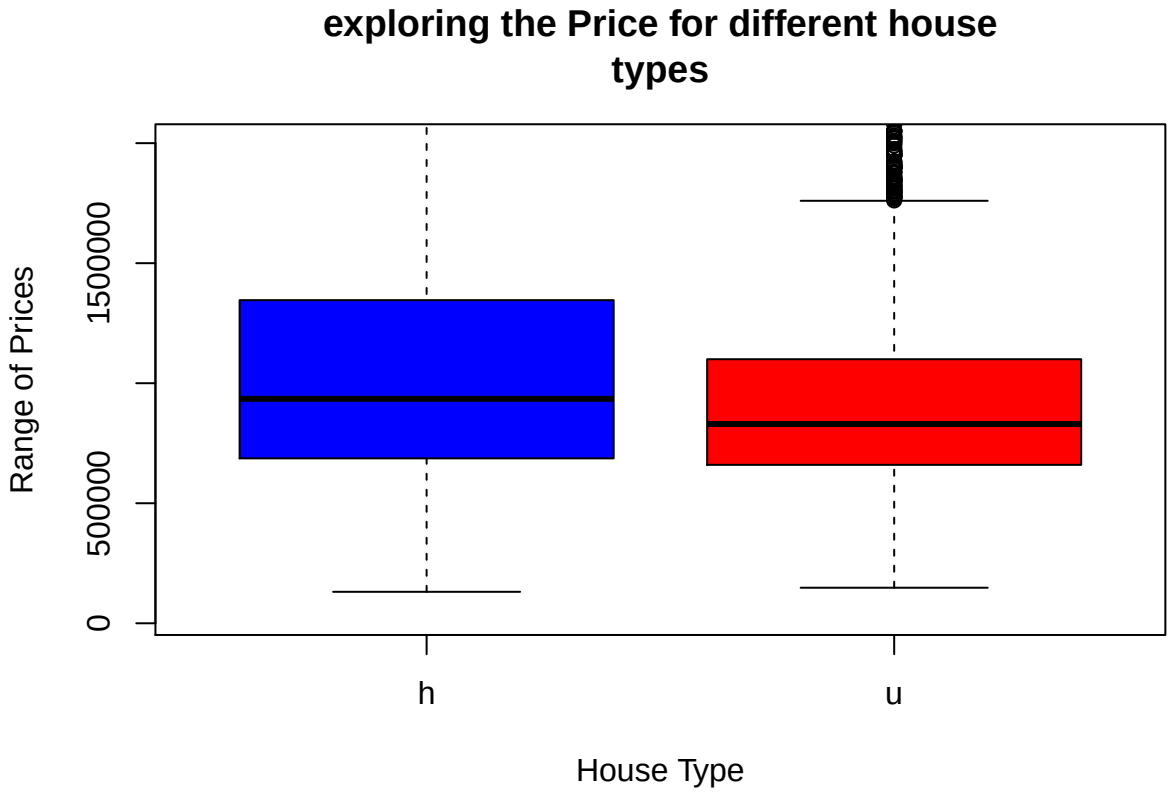
Also we can also find the p value is greater than the siginificane value (Alpha value), Which confirms that we should not reject the Null hypothesis

From the output we can also notice that the true difference in means of two samples is not equal to 0.

The output also provides that sample mean estimates of two samples.

Here we have BSDA package to use the z test function.

```
housetype<-c("h","u")
boxplot(housingdata$Price[housingdata$Type=="h"],
housingdata$Price[housingdata$Type=="t"],main= "exploring the Price for different house
types",ylim=c(30000,2000000),col = c("blue","red","orange"),xlab="House Type"
,ylab="Range of Prices",names = housetype)
```



From the graph we can notice that varience of house price for house type h is double than the variance of house price for type u.

```
mean(housingdata$Price)
```

```
## [1] 997898.2
```

```
sd(housingdata$Price)
```

```
## [1] 593498.9
```

```
housingdata1<-housingdata
sample1<-housingdata$Price
sample2<-housingdata$Type
sample2<-as.factor(sample2)
levels(sample2)
```

```
## [1] "h" "t" "u"
```

```r
class(sample1)
```

```
## [1] "integer"
```

Hypothesis 2:

two sample z test

From the housing data set we take the price variable and lets define the hypothesis on house prices with respective to the no of rooms that house has. Here we are considering the houses that contain only 2 and 3 rooms. We take two samples and make the conclusion for the following hypothesis. ## Defining Hypothesis

Null Hypothesis: The Average price of houses with only 2 rooms is less than or equal to the average price of the houses with only 3 rooms.

i.e Ho: u<= u1

Alternate Hypothesis: The Average price of houses with only 2 rooms is greater than the average price of the houses with only 3 rooms.

i.e H1: u>u1

State Alpha

The significance level(alpha) is defined as 0.05

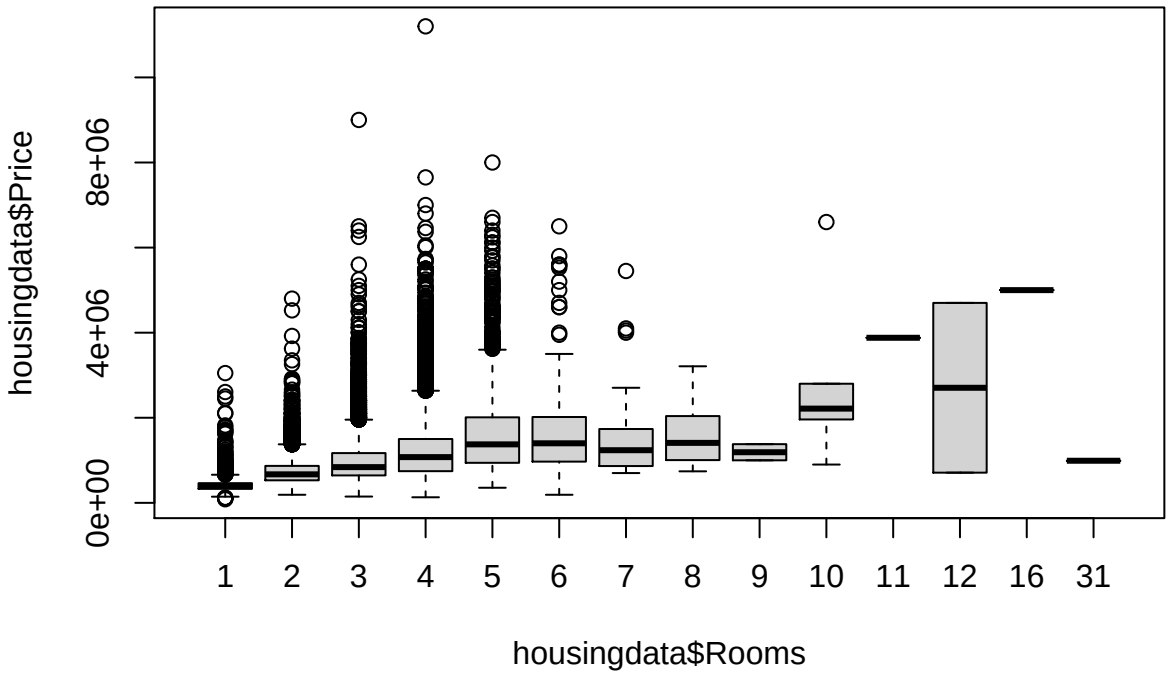
Confidence level is defined as 95

Decision Rule:If the z value is less than -1.96 or greater than 1.96,Reject the null hypothesis.

Test statistic:We use the z test method as we have the sample size greater than 30 , and also the standard deviation is known.

Here we are using two samples of with respect to no of rooms

```r
boxplot(housingdata$Price~housingdata$Rooms)
```




```
library("car")
rooms2<-subset(housingdata$Price, housingdata$Rooms==2)
length(rooms2)
```

```
## [1] 10674
```

```
mean(rooms2)
```

```
## [1] 746092.6
```

```
rooms3<-subset(housingdata$Price,housingdata$Rooms==3)
length(rooms3)
```

```
## [1] 21812
```

```
mean(rooms3)
```

```
## [1] 958528
```

```
library(BSDA)
set.seed(120)
rooms2sample<-sample(rooms2,50,replace = FALSE)
rooms3sample<-sample(rooms3,50,replace= FALSE)
mu=mean(rooms3)-mean(rooms2)
z.test(rooms2sample,rooms3sample,alternative="less",mu=212435.4,sigma.x = sd(rooms2sample),sigma.y = sd
```

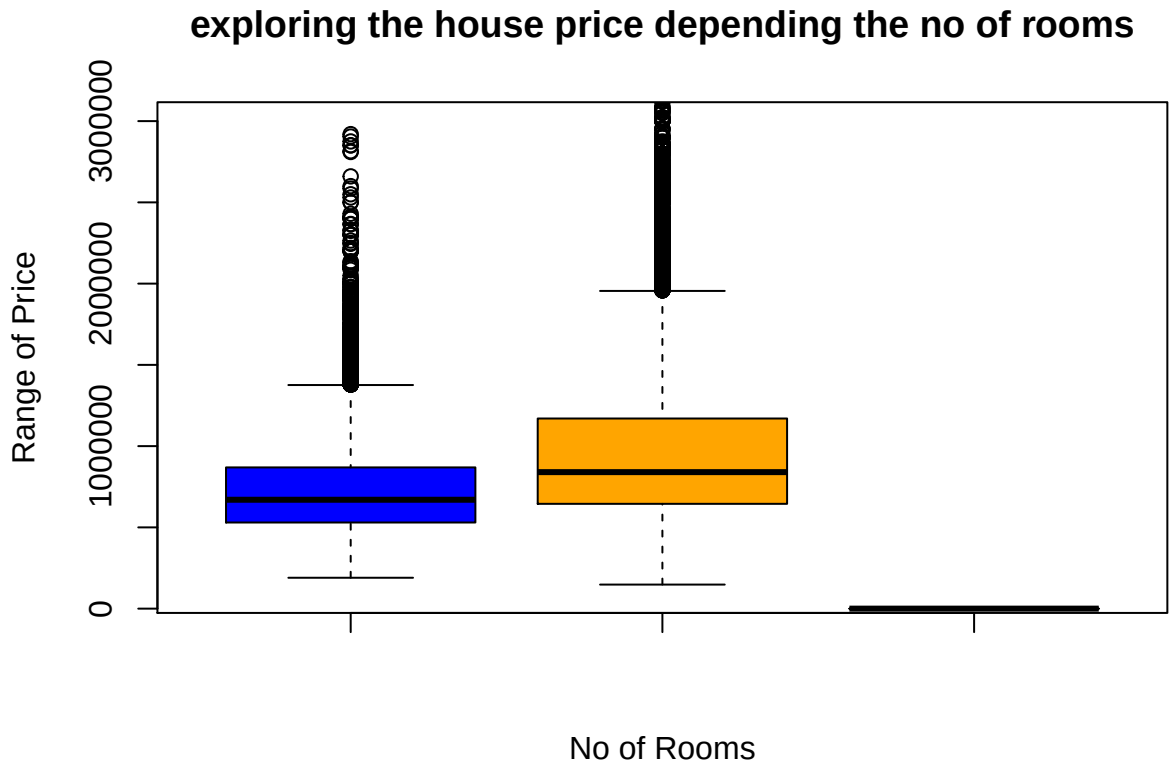
```
##
##  Two-sample z-Test
##
## data:  rooms2sample and rooms3sample
## z = -6.5124, p-value = 3.699e-11
## alternative hypothesis: true difference in means is less than 212435.4
## 95 percent confidence interval:
##         NA -59562.13
## sample estimates:
## mean of x mean of y
##    697985    849462
```

From the output we can notice that the Z value is -6.5124.

Conclusion

The Z values doesn't lie between the -1.96 and +1.96 as stated in decision rule, Hence we should reject the null hypothesis.

```
noofrooms<-c(2,3)
boxplot(housingdata$Price[housingdata$Rooms==2],housingdata$Price[housingdata$Rooms==3],main="exploring
```



From the graph we can observe that the house with only two rooms has less mean price than the houses with 3 rooms.

Hypothesis 3

One sample t test

To determine whether the sample mean of prices and mean of prices are equal. ## 1 Define Hypothesis ## Null hypothesis HO: To check whether the sample mean of house prices are equal to the mean of the house prices. ## Ho= 997898.2

Alternate Hpyothesis: To check whether the sample mean of house are not equal to the mean of house prices

H1 not equal to 997898.2

Alpha = 0.05

confidence level =95

Decision Rule: Reject null hypothesis if t > 2.26 and t<2.26

Test statistic: even though we assume to know the mean og the house price, we use t-test , as we assume that our sample is less than 30.

```
sampledata<-sample(housingdata$Price ,30,replace = FALSE)
sd(sampledata)
```

```
## [1] 440262.1
```

```
t.test(sampledata,alternative="two.sided",mu=997898.2,conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  sampledata
## t = -1.1441, df = 29, p-value = 0.2619
## alternative hypothesis: true mean is not equal to 997898.2
## 95 percent confidence interval:
##   741536.8 1070329.9
## sample estimates:
## mean of x
##  905933.3
```

conclusion: From the above output we can notice that t value is -0.3442 and p value =0.733

As t value lies betwwn the -2.26 and +2.26 as stated in the decision rule, We can confirm that we don't reject the null hypothesis.

It also provides values which are from 765713.3 and 1163153.4 for 95 confidence interval

Hypothesis 4

Linear Regression between the price and distance

1 Define hypothesis:

Null Hypothesis Ho= There is linear relation between price and distance

Alternative Hypothesis H1 = There is no linear relation between price and distance.

```
lmod<-lm(Price ~ Distance, data=housingdata)
summary(lmod)
```

```
##
## Call:
## lm(formula = Price ~ Distance, data = housingdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1104275  -346822  -129411   210196 10158178
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1251197.7     5105.7  245.06   <2e-16 ***
## Distance     -19940.5      345.5  -57.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 574100 on 48431 degrees of freedom
## Multiple R-squared:  0.06435,    Adjusted R-squared:  0.06433
## F-statistic:  3331 on 1 and 48431 DF,  p-value: < 2.2e-16
```

From the above output we see R-squared value of 0.06433

Conclusion: From the above finding's with respect to R-squared value which is 0.06433 tells that there is very less variance for price and distance. As the p value 2.2e-16<0.05 suggests that this overall a good model. But there are also many other variables which have a significant relantionship with price.

```
#Task 2

The task is to split the data into training data and test data with a split percentage of 75/25.
First lets load the caret library, which contains many functions for modeling training process in regres

## Performing linear regression with multiple variables to predict the house price

'''r
library(caret)
set.seed(100)
## Setting a seed number ensures that we get the same result when we run this process with the same seed

##splitting the data set into training data and test data
trainingindex <- createDataPartition(housingdata$Price, p= 0.75, list = F)
trainingdata<-housingdata[trainingindex,]
testdata<-housingdata[-trainingindex,]

linear_model<-lm(Price~Rooms+Type+Distance+Postcode+Regionname+Propertycount,data=housingdata)
linear_model


##
## Call:
## lm(formula = Price ~ Rooms + Type + Distance + Postcode + Regionname +
##     Propertycount, data = housingdata)
##
## Coefficients:
##                            (Intercept)                                   Rooms
##                              286420.43                               244700.31
##                                  Typet                                   Typeu
##                             -220878.43                              -436273.77
##                               Distance                                Postcode
##                              -40153.15                                  202.02
##           RegionnameEastern Victoria       RegionnameNorthern Metropolitan
##                             181388.64                              -281728.91
##          RegionnameNorthern Victoria  RegionnameSouth-Eastern Metropolitan
##                             118854.90                               169658.22
##      RegionnameSouthern Metropolitan        RegionnameWestern Metropolitan
##                             256549.72                              -352243.90
##           RegionnameWestern Victoria                           Propertycount
##                             -95325.13                                   1.61
```

```r
summary(linear_model)
```

```
##
## Call:
## lm(formula = Price ~ Rooms + Type + Distance + Postcode + Regionname +
##     Propertycount, data = housingdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6940043  -233380   -53838   152568  9439174
##
## Coefficients:
##                                     Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                         2.864e+05  6.062e+04    4.725 2.31e-06
## Rooms                               2.447e+05  2.412e+03  101.460  < 2e-16
## Typet                              -2.209e+05  6.341e+03  -34.834  < 2e-16
## Typeu                              -4.363e+05  5.842e+03  -74.680  < 2e-16
## Distance                           -4.015e+04  3.818e+02 -105.167  < 2e-16
## Postcode                            2.020e+02  1.979e+01   10.207  < 2e-16
## RegionnameEastern Victoria          1.814e+05  2.312e+04    7.845 4.40e-15
## RegionnameNorthern Metropolitan    -2.817e+05  6.272e+03  -44.921  < 2e-16
## RegionnameNorthern Victoria         1.189e+05  2.109e+04    5.637 1.74e-08
## RegionnameSouth-Eastern Metropolitan  1.697e+05  8.652e+03   19.609  < 2e-16
## RegionnameSouthern Metropolitan     2.565e+05  6.570e+03   39.047  < 2e-16
## RegionnameWestern Metropolitan     -3.522e+05  6.529e+03  -53.954  < 2e-16
## RegionnameWestern Victoria         -9.533e+04  3.130e+04   -3.046 0.002324
## Propertycount                       1.610e+00  4.394e-01    3.664 0.000249
##
## (Intercept)                        ***
## Rooms                              ***
## Typet                              ***
## Typeu                              ***
## Distance                           ***
## Postcode                           ***
## RegionnameEastern Victoria         ***
## RegionnameNorthern Metropolitan    ***
## RegionnameNorthern Victoria        ***
## RegionnameSouth-Eastern Metropolitan ***
## RegionnameSouthern Metropolitan    ***
## RegionnameWestern Metropolitan     ***
## RegionnameWestern Victoria         **
## Propertycount                      ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 409400 on 48419 degrees of freedom
## Multiple R-squared:  0.5242, Adjusted R-squared:  0.5241
## F-statistic:  4103 on 13 and 48419 DF,  p-value: < 2.2e-16
```

```r
coef(linear_model)
```

```
##                        (Intercept)                              Rooms
```

```
##                          2.864204e+05                          2.447003e+05
##                                 Typet                                 Typeu
##                         -2.208784e+05                         -4.362738e+05
##                              Distance                              Postcode
##                         -4.015315e+04                          2.020193e+02
##            RegionnameEastern Victoria      RegionnameNorthern Metropolitan
##                          1.813886e+05                         -2.817289e+05
##           RegionnameNorthern Victoria RegionnameSouth-Eastern Metropolitan
##                          1.188549e+05                          1.696582e+05
##       RegionnameSouthern Metropolitan       RegionnameWestern Metropolitan
##                          2.565497e+05                         -3.522439e+05
##            RegionnameWestern Victoria                         Propertycount
##                         -9.532513e+04                          1.609704e+00
```

```
prediction1 <-predict(linear_model, newdata = housingdata)
traindata <- lm(Price~Rooms+Type+Distance+Postcode+Regionname+Propertycount,data = trainingdata)

coef(traindata)
```

```
##                           (Intercept)                                 Rooms
##                          3.002739e+05                          2.418391e+05
##                                 Typet                                 Typeu
##                         -2.181669e+05                         -4.360268e+05
##                              Distance                              Postcode
##                         -3.998920e+04                          2.007199e+02
##            RegionnameEastern Victoria      RegionnameNorthern Metropolitan
##                          1.693504e+05                         -2.799954e+05
##           RegionnameNorthern Victoria RegionnameSouth-Eastern Metropolitan
##                          1.223915e+05                          1.708618e+05
##       RegionnameSouthern Metropolitan       RegionnameWestern Metropolitan
##                          2.488023e+05                         -3.541139e+05
##            RegionnameWestern Victoria                         Propertycount
##                         -9.044951e+04                          1.433912e+00
```

```
prediction1 <- predict(traindata, newdata = testdata)
summary(prediction1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -565371  723057  982235 1001295 1280127 2905093
```

```
trainingdata1<-trainingdata
```

Here we have replicated training data as trainingdata1 as a precautionary to not disturb the original training data while trying to know the possible variables for predicting the house price using corrplot

```
str(trainingdata1)
```

```
## 'data.frame':    36326 obs. of  14 variables:
##  $ X            : int  1 2 6 7 8 12 13 14 17 19 ...
##  $ Suburb       : chr  "Abbotsford" "Abbotsford" "Airport West" "Airport West" ...
##  $ Address      : chr  "49 Lithgow St" "59A Turner St" "4/32 Earl St" "3/74 Hawker St" ...
```

```
##  $ Rooms        : int  3 3 2 2 3 4 2 4 3 2 ...
##  $ Type         : chr  "h" "h" "t" "u" ...
##  $ Price        : int  1490000 1220000 530000 540000 715000 717000 1675000 2008000 720000 2110000 ..
##  $ Method       : chr  "S" "S" "S" "S" ...
##  $ SellerG      : chr  "Jellis" "Marshall" "Jellis" "Barry" ...
##  $ Date         : chr  "1/04/2017" "1/04/2017" "1/04/2017" "1/04/2017" ...
##  $ Postcode     : int  3067 3067 3042 3042 3042 3020 3078 3078 3025 3143 ...
##  $ Regionname   : chr  "Northern Metropolitan" "Northern Metropolitan" "Western Metropolitan" "Weste
##  $ Propertycount: int  4019 4019 3464 3464 3464 2185 2211 2211 5132 4836 ...
##  $ Distance     : num  3 3 10.4 10.4 10.4 10.5 5.7 5.7 9.4 6.3 ...
##  $ CouncilArea  : chr  "Yarra City Council" "Yarra City Council" "Moonee Valley City Council" "Moone
```

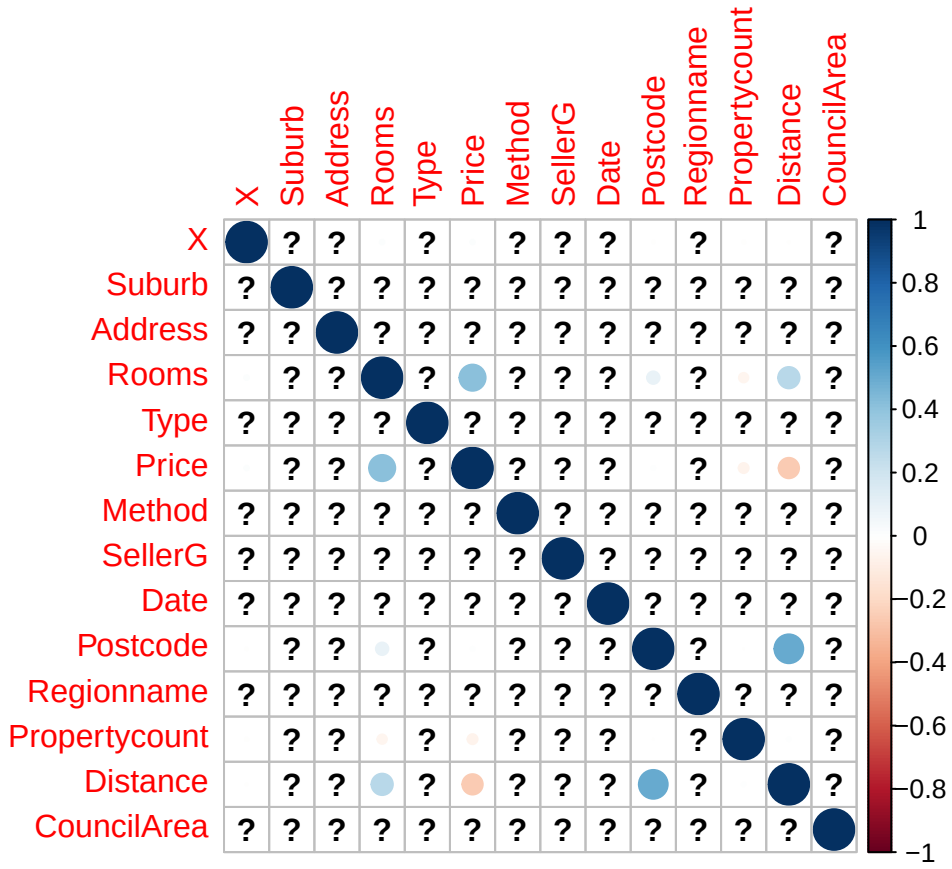
```
library(corrplot)
sapply(trainingdata1,class)
```

```
##             X        Suburb       Address         Rooms          Type
##     "integer"   "character"   "character"     "integer"   "character"
##         Price        Method       SellerG          Date      Postcode
##     "integer"   "character"   "character"   "character"     "integer"
##    Regionname Propertycount      Distance   CouncilArea
##   "character"     "integer"     "numeric"   "character"
```

```
i<-c(1,2,3,4,5,6,7,8,9,10,11,12,14)
trainingdata1[, i]<-apply(trainingdata1[, i],2,function(x) as.numeric(as.character(x)))
```

```
## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion
```

```
A<-cor(trainingdata1)
corrplot(A,method = "circle")
```



From the above corrplot we can confirm that the variables Rooms,Type and Distance are most associated with the price variable.

```
cor(prediction1,testdata$Price)
```

```
## [1] 0.7299545
```

```
cor(prediction1,testdata$Price)^2
```

```
## [1] 0.5328335
```

```
RMSE(testdata$Price, prediction1)
```

```
## [1] 411240.8
```

Here we can find that the Adjusted R squared value is 0.5212 and the multiple R-squared value is 0.5213.

The R-squared value gives the propotion of variance for dependent variable with respect to indepedent variables.

The prediction accuracy of the model on the test data with respect to RMSE and correlation is 411240.8 and 0.5328335

Normalization of data

Here we are using min-max scaling normalization to bring all variables in same range.

```
normal <- preProcess(housingdata[,c(4:6,10:13)], method=c("range"))
normdata <- predict(normal, housingdata[,c(4:6,10:13)])
summary(normdata)
```

```
##      Rooms             Type               Price            Postcode
##  Min.   :0.00000   Length:48433       Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.03333   Class :character   1st Qu.:0.04813   1st Qu.:0.05204
##  Median :0.06667   Mode  :character   Median :0.06703   Median :0.10510
##  Mean   :0.06906                      Mean   :0.08213   Mean   :0.12572
##  3rd Qu.:0.10000                      3rd Qu.:0.10211   3rd Qu.:0.16633
##  Max.   :1.00000                      Max.   :1.00000   Max.   :1.00000
##   Regionname        Propertycount       Distance
##  Length:48433       Min.   :0.0000   Min.   :0.0000
##  Class :character   1st Qu.:0.1962   1st Qu.:0.1254
##  Mode  :character   Median :0.3021   Median :0.2097
##                     Mean   :0.3483   Mean   :0.2276
##                     3rd Qu.:0.4800   3rd Qu.:0.2993
##                     Max.   :1.0000   Max.   :1.0000
```

```
library("caret")
set.seed(116)
trainingindex2 <- createDataPartition(normdata$Price, p= 0.75, list = F)
trainingdata2 <- normdata[trainingindex2,]
testdata2 <- normdata[-trainingindex2,]
```

Performing linear regression with multiple variables to predict the house price

```
linear_model2<-lm(Price~Rooms+Type+Distance+Postcode+Regionname+Propertycount,data=normdata)
linear_model2
```

```
##
## Call:
## lm(formula = Price ~ Rooms + Type + Distance + Postcode + Regionname +
##     Propertycount, data = normdata)
##
## Coefficients:
##                          (Intercept)                                 Rooms
##                             0.094669                              0.660460
```

```
##                               Typet                               Typeu
##                          -0.019872                           -0.039251
##                            Distance                            Postcode
##                          -0.201579                            0.017812
##          RegionnameEastern Victoria      RegionnameNorthern Metropolitan
##                           0.016319                           -0.025347
##          RegionnameNorthern Victoria RegionnameSouth-Eastern Metropolitan
##                           0.010693                            0.015264
##      RegionnameSouthern Metropolitan       RegionnameWestern Metropolitan
##                           0.023081                           -0.031691
##          RegionnameWestern Victoria                       Propertycount
##                          -0.008576                            0.003130
```

```
summary(linear_model2)
```

```
##
## Call:
## lm(formula = Price ~ Rooms + Type + Distance + Postcode + Regionname +
##     Propertycount, data = normdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62439 -0.02100 -0.00484  0.01373  0.84923
##
## Coefficients:
##                                    Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                       0.0946686  0.0008407  112.603  < 2e-16
## Rooms                             0.6604597  0.0065095  101.460  < 2e-16
## Typet                            -0.0198721  0.0005705  -34.834  < 2e-16
## Typeu                            -0.0392509  0.0005256  -74.680  < 2e-16
## Distance                         -0.2015786  0.0019167 -105.167  < 2e-16
## Postcode                          0.0178119  0.0017451   10.207  < 2e-16
## RegionnameEastern Victoria        0.0163193  0.0020801    7.845 4.40e-15
## RegionnameNorthern Metropolitan  -0.0253467  0.0005643  -44.921  < 2e-16
## RegionnameNorthern Victoria       0.0106932  0.0018970    5.637 1.74e-08
## RegionnameSouth-Eastern Metropolitan  0.0152639  0.0007784   19.609  < 2e-16
## RegionnameSouthern Metropolitan   0.0230814  0.0005911   39.047  < 2e-16
## RegionnameWestern Metropolitan   -0.0316909  0.0005874  -53.954  < 2e-16
## RegionnameWestern Victoria       -0.0085763  0.0028160   -3.046 0.002324
## Propertycount                     0.0031298  0.0008543    3.664 0.000249
##
## (Intercept)                       ***
## Rooms                             ***
## Typet                             ***
## Typeu                             ***
## Distance                          ***
## Postcode                          ***
## RegionnameEastern Victoria        ***
## RegionnameNorthern Metropolitan   ***
## RegionnameNorthern Victoria       ***
## RegionnameSouth-Eastern Metropolitan ***
## RegionnameSouthern Metropolitan   ***
## RegionnameWestern Metropolitan    ***
## RegionnameWestern Victoria        **
```

```
## Propertycount                         ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03684 on 48419 degrees of freedom
## Multiple R-squared:  0.5242, Adjusted R-squared:  0.5241
## F-statistic:  4103 on 13 and 48419 DF,  p-value: < 2.2e-16
```

```
coef(linear_model2)
```

```
##                           (Intercept)                                 Rooms
##                           0.094668584                           0.660459681
##                                 Typet                                 Typeu
##                          -0.019872103                          -0.039250901
##                              Distance                              Postcode
##                          -0.201578561                           0.017811865
##           RegionnameEastern Victoria      RegionnameNorthern Metropolitan
##                           0.016319266                          -0.025346730
##          RegionnameNorthern Victoria RegionnameSouth-Eastern Metropolitan
##                           0.010693198                           0.015263898
##      RegionnameSouthern Metropolitan       RegionnameWestern Metropolitan
##                           0.023081396                          -0.031690860
##           RegionnameWestern Victoria                         Propertycount
##                          -0.008576260                           0.003129763
```

```
#Testing the training data againist the test data
ins_predict2<-predict(linear_model2, newdata = testdata2)
summary(ins_predict2)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.05894  0.05702  0.08016  0.08215  0.10754  0.28141
```

```
cor(ins_predict2,testdata2$Price)
```

```
## [1] 0.7249854
```

```
cor(ins_predict2,testdata2$Price)^2
```

```
## [1] 0.5256038
```

```
RMSE(testdata2$Price, ins_predict2)
```

```
## [1] 0.03680975
```

From the normalized data we can observe that there is very slight differences in the values. The Adjusted and multiple R-squared value are 0.5236,0.5238 which is very similar the R-squared values 0.5212 and 0.5213 The p values remains same for the orginal data and normalized data. The main difference between these models are RMSE value. The Rmse value for the normalized data is 0.036820 which is very low with respect the rsme value of original data.

Task 3

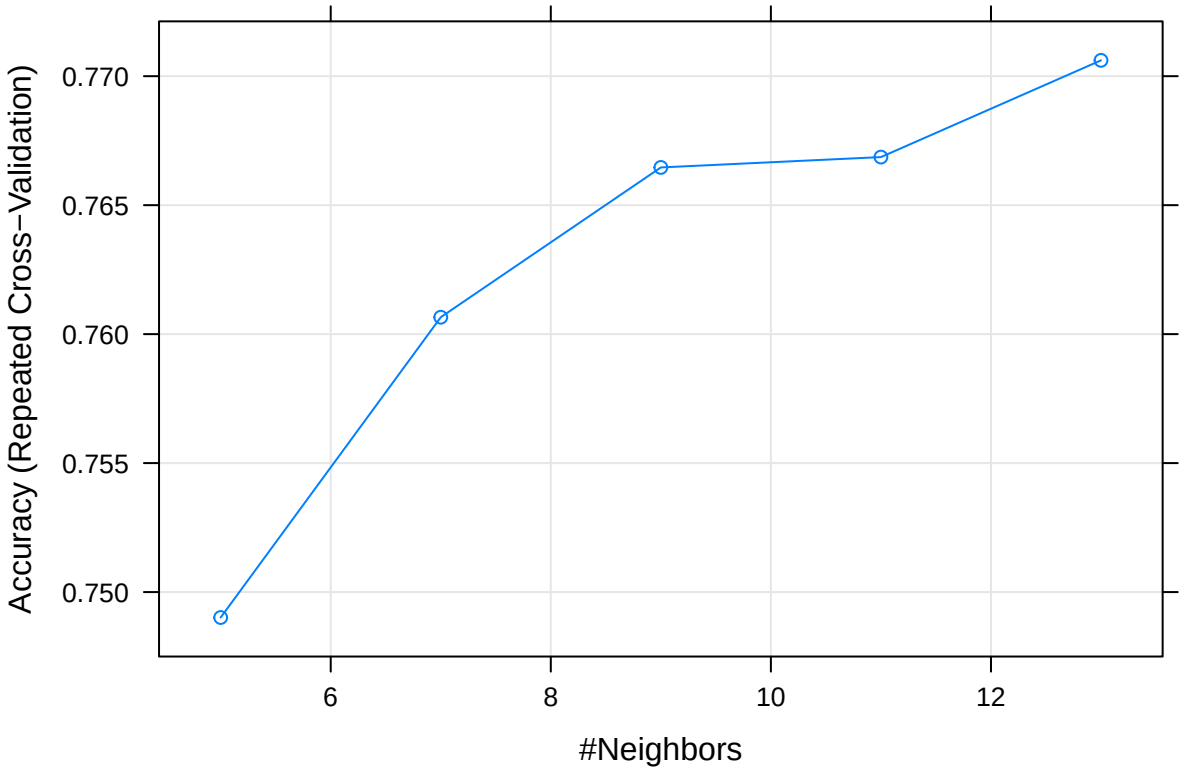
Dividing the dataset

Here we are splitting the data set into training data set and test data set with ratio of 80/20. we also peform normalization on the data set for better results.

```
housingdata.sub1<-housingdata[,c(4,5,6,7,12,13)]
samplesize <- sample(nrow(housingdata.sub1), size=1000, replace = FALSE, prob = NULL)
housingdata.sub2 <- housingdata.sub1[samplesize, ]
indxdata <- createDataPartition(y = housingdata.sub2$Type,p = 0.8,list = FALSE)
training5 <- housingdata.sub2[indxdata,]
testing5 <- housingdata.sub2[-indxdata,]

# Run k-NN:
set.seed(200)
ctrl <- trainControl(method="repeatedcv",repeats = 3)
knnFit <- train(Type ~ ., data = training5, method = "knn", trControl = ctrl, preProcess = c("center","
knnFit
```

```
## k-Nearest Neighbors
##
## 802 samples
##   5 predictor
##   3 classes: 'h', 't', 'u'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 722, 721, 721, 722, 722, 723, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.7490132  0.3879688
##    7  0.7606549  0.4055661
##    9  0.7664627  0.4136017
##   11  0.7668636  0.4076926
##   13  0.7706140  0.4157520
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 13.
```

```
#Plotting different k values against accuracy (based on repeated cross validation)
plot(knnFit)
```




```
confusionMatrix(knnFit)
```

```
## Cross-Validated (10 fold, repeated 3 times) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    h    t    u
##          h 65.5  9.8  7.7
##          t  0.1  0.0  0.1
##          u  3.5  1.7 11.6
##
##  Accuracy (average) : 0.7706
```

From the above output we can observe that the knn with 801 samples from a dataset has 5 predictors and 3 classes which are categorized by house types.

We can observe the respective accuracy for knn with respect to change in the value of k

Thus the Accuracy for knn is 79

2) C5.0

using c5.0 algorithm to classify house types into appropriate types based on their features.

```
library(caret)
library(C50)
library(lattice)
library(ggplot2)
housingdata.subset1<-housingdata[,c(4,5,6,7,12,13)]
samplesize <- sample(nrow(housingdata.subset1), size=500, replace = FALSE, prob = NULL)
housingdata.subset <- housingdata.subset1[samplesize, ]
housingdata.subset$Type<- as.factor(housingdata.subset$Type)

trainingindex <- createDataPartition(housingdata.subset$Type, p= 0.8, list = F)
c50_training <- housingdata.subset[trainingindex,]
c50_test <- housingdata.subset[-trainingindex,]
C5_fit <- train(Type~., data = c50_training, method = "C5.0")
summary(C5_fit)
```

```
##
## Call:
## (function (x, y, trials = 1, rules = FALSE, weights = NULL, control
##  2, fuzzyThreshold = FALSE, sample = 0, earlyStopping = TRUE, label
##  = "outcome", seed = 3017L))
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Jan 18 23:41:49 2021
## -------------------------------
##
## Class specified by attribute ‘outcome’
##
## Read 401 cases (9 attributes) from undefined.data
##
## -----  Trial 0:  -----
##
## Decision tree:
##
## Rooms > 2: h (309/46)
## Rooms <= 2:
## :...Price <= 755000: u (65/11)
##     Price > 755000: h (27/7)
##
## -----  Trial 1:  -----
##
```

```
## Decision tree:
##
## Rooms <= 2: u (96.2/44.9)
## Rooms > 2: h (304.8/95.1)
##
## -----  Trial 2:  -----
##
## Decision tree:
##  h (401/177.4)
##
## *** boosting reduced to 2 trials since last classifier is very inaccurate
##
## *** boosting abandoned (too few classifiers)
##
##
## Evaluation on training data (401 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       3   64(16.0%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##     283           6    (a): class h
##      35           5    (b): class t
##      18          54    (c): class u
##
##
##  Attribute usage:
##
##  100.00% Rooms
##   22.94% Price
##
##
## Time: 0.0 secs
```

we can observe that there are 401 samples with 5 predictor and 3 classes. It shows the classification of model types winnow and trails. the final value used for the model were trials = 20, model =tree and winnow =False

```
C5_predict <- predict(C5_fit, newdata = c50_test )
confusionMatrix(C5_predict, c50_test$Type )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  h  t  u
##          h 70  7  3
##          t  0  0  0
##          u  2  2 15
##
```

```
## Overall Statistics
##
##                Accuracy : 0.8586
##                  95% CI : (0.7741, 0.9205)
##     No Information Rate : 0.7273
##     P-Value [Acc > NIR] : 0.001427
##
##                   Kappa : 0.6253
##
##  Mcnemar's Test P-Value : 0.026747
##
## Statistics by Class:
##
##                      Class: h Class: t Class: u
## Sensitivity            0.9722  0.00000   0.8333
## Specificity            0.6296  1.00000   0.9506
## Pos Pred Value         0.8750      NaN   0.7895
## Neg Pred Value         0.8947  0.90909   0.9625
## Prevalence             0.7273  0.09091   0.1818
## Detection Rate         0.7071  0.00000   0.1515
## Detection Prevalence   0.8081  0.00000   0.1919
## Balanced Accuracy      0.8009  0.50000   0.8920
```

we can observe that from the above output c50 algorithm has an accuracy of 85 for the following data. conclusion: By observing the two classifications we found that decision-tree i.e c5.0 algorithm has more accuracy with large data sets.

“