Assignment 2

Shubhkumar Bharatkumar patel(3077432)

30/11/2021

```
knitr::opts_chunk$set(echo = TRUE)
housing.dataset <- read.csv("C:/Users/shubh/Downloads/melbourne_housing_data.csv")
str(housing.dataset)
```

```
## '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
```

```
options(scipen = 999)
library(car)
```

```
## Warning: package 'car' was built under R version 4.1.2
```

```
## Loading required package: carData
```

```
library(nortest)
library(HH)
```

```
## Warning: package 'HH' was built under R version 4.1.2
```

```
## Loading required package: lattice
```

```
## Loading required package: grid
```

```
## Loading required package: latticeExtra
```

```
## Warning: package 'latticeExtra' was built under R version 4.1.2
```

```
## Loading required package: multcomp

## Warning: package 'multcomp' was built under R version 4.1.2

## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Warning: package 'TH.data' was built under R version 4.1.2

## Loading required package: MASS

##
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
##
##     geyser

## Loading required package: gridExtra

## Warning: package 'gridExtra' was built under R version 4.1.2

##
## Attaching package: 'HH'

## The following objects are masked from 'package:car':
##
##     logit, vif
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:HH':
##
##     logit

## The following object is masked from 'package:car':
##
##     logit
```

```
library(corrgram)
```

```
## Warning: package 'corrgram' was built under R version 4.1.2
```

```
##
## Attaching package: 'corrgram'

## The following object is masked from 'package:latticeExtra':
##
##     panel.ellipse

## The following object is masked from 'package:lattice':
##
##     panel.fill
```

```
library(PerformanceAnalytics)
```

```
## Loading required package: xts

## Warning: package 'xts' was built under R version 4.1.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.2

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

## The following object is masked from 'package:latticeExtra':
##
##     layer
```

```
library(gridExtra)
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.1.2
```

```
library(carat)
```

```
## Warning: package 'carat' was built under R version 4.1.2
```

```
##
## Attaching package: 'carat'
```

```
## The following object is masked from 'package:psych':
##
##     corr.test
```

```
library(modeldata)
```

```
## Warning: package 'modeldata' was built under R version 4.1.2
```

```
library(C50)
```

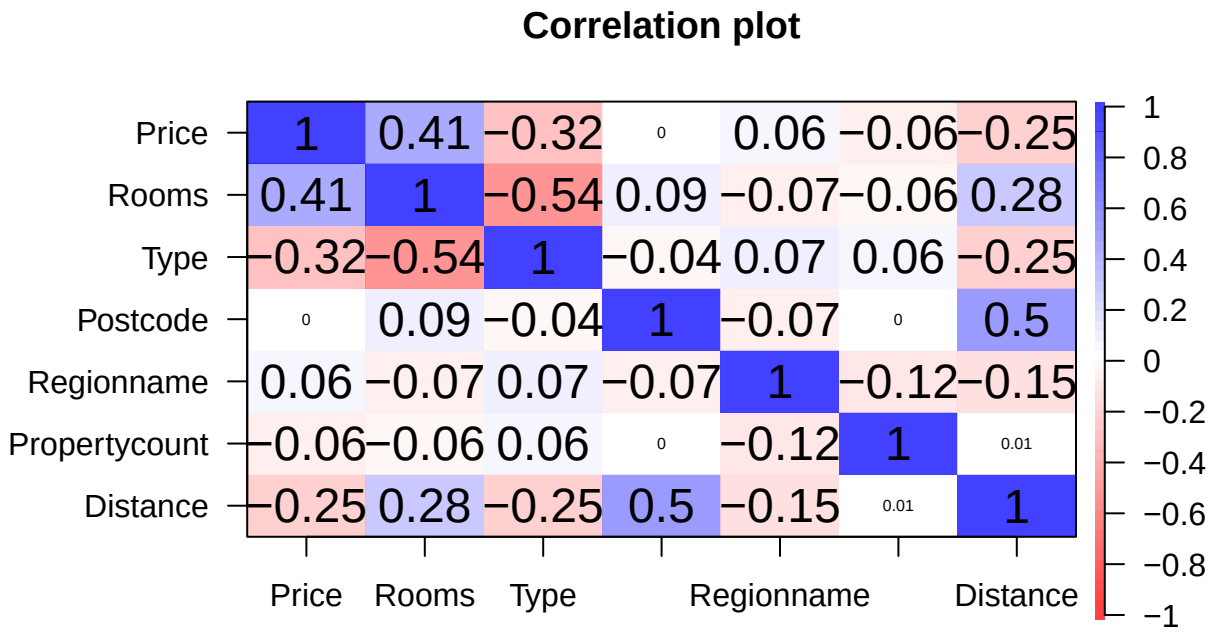
```
## Warning: package 'C50' was built under R version 4.1.2
```

```
library(neuralnet)
```

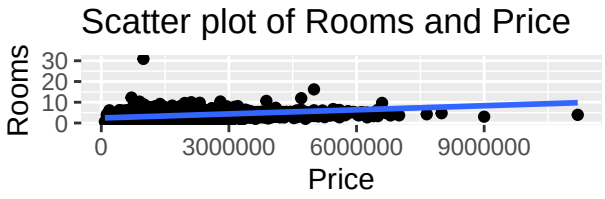
```
## Warning: package 'neuralnet' was built under R version 4.1.2
```

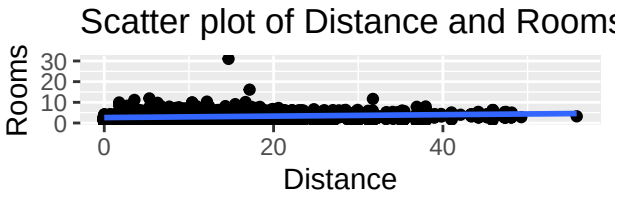
```
library(class)
```

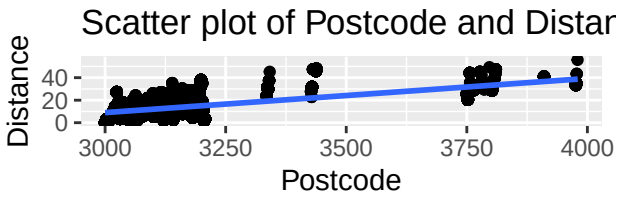
```
#Converting Type Column from Chr to int.

housing.dataset$Type[housing.dataset$Type == 'h'] <- 1
housing.dataset$Type[housing.dataset$Type == 't'] <- 2
housing.dataset$Type[housing.dataset$Type == 'u'] <- 3
housing.dataset$Type <- as.integer(housing.dataset$Type)

housing.dataset$Regionname[housing.dataset$Regionname == 'Eastern Metropolitan'] <- 1
housing.dataset$Regionname[housing.dataset$Regionname == 'Eastern Victoria'] <- 2
housing.dataset$Regionname[housing.dataset$Regionname == 'Northern Metropolitan'] <- 3
housing.dataset$Regionname[housing.dataset$Regionname == 'Northern Victoria'] <- 4
housing.dataset$Regionname[housing.dataset$Regionname == 'South-Eastern Metropolitan'] <- 5
housing.dataset$Regionname[housing.dataset$Regionname == 'Southern Metropolitan'] <- 6
housing.dataset$Regionname[housing.dataset$Regionname == 'Western Metropolitan'] <- 7
housing.dataset$Regionname[housing.dataset$Regionname == 'Western Victoria'] <- 8
housing.dataset$Regionname <- as.integer(housing.dataset$Regionname)

##There's a positive correlation with Regionname and Rooms with Price,Regionname and Propertycount with
## while it is showing negative correlation with Type and Rooms
corPlot(housing.dataset[c(6,4,5,10,11,12,13)], cex = 1)
```

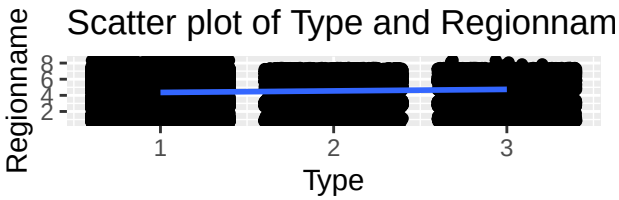


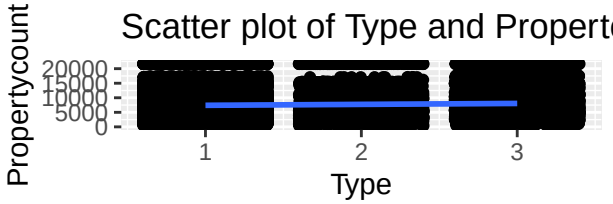

```
#plotting posiive Correlative pairs
p1=ggplot(housing.dataset, aes(x = Price, y = Rooms)) +
  geom_jitter() +  geom_smooth(formula=y~x,method = "lm", se = FALSE)+labs(title="Scatter plot of Rooms

p2=ggplot(housing.dataset, aes(x =Distance , y =  Rooms)) +
  geom_jitter() +  geom_smooth(formula=y~x,method = "lm", se = FALSE)+labs(title="Scatter plot of Distar

p3=ggplot(housing.dataset, aes(x =Postcode , y =  Distance)) +
  geom_jitter() +  geom_smooth(formula=y~x,method = "lm", se = FALSE)+labs(title="Scatter plot of Postc

p4=ggplot(housing.dataset, aes(x =Type , y =  Regionname)) +
  geom_jitter() +  geom_smooth(formula=y~x,method = "lm", se = FALSE)+labs(title="Scatter plot of Type a

p5=ggplot(housing.dataset, aes(x =Type , y =  Propertycount)) +
  geom_jitter() +  geom_smooth(formula=y~x,method = "lm", se = FALSE)+labs(title="Scatter plot of Type a

p6=ggplot(housing.dataset, aes(x =Postcode , y =  Rooms)) +
  geom_jitter() +  geom_smooth(formula=y~x,method = "lm", se = FALSE)+labs(title="Scatter plot of Postc

p7=ggplot(housing.dataset, aes(x =Price , y =Regionname )) +
  geom_jitter() +  geom_smooth(formula=y~x,method = "lm", se = FALSE)+labs(title="Scatter plot of Regio

grid.arrange(p1,p2,p3,p4,p5,p6,p7,nrow=4)
```

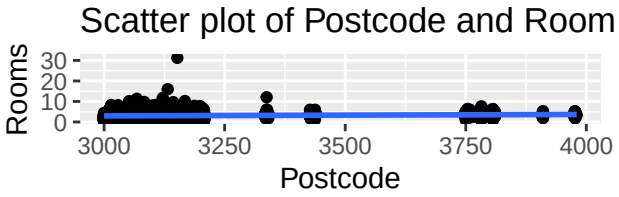


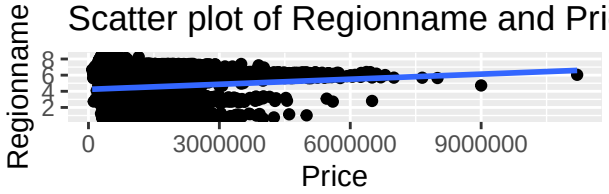













```
#Normalizing Data

norm1 <- function(x) {(x - min(x)) / (max(x) - min(x))}

hdd_norm <- as.data.frame(lapply(housing.dataset[,c(4,6,11)], norm1))
head(hdd_norm)
```

```
##        Rooms      Price Regionname
## 1 0.06666667 0.12640576  0.2857143
## 2 0.06666667 0.10211426  0.2857143
## 3 0.06666667 0.12010796  0.2857143
## 4 0.06666667 0.12865497  0.8571429
## 5 0.03333333 0.05263158  0.8571429
## 6 0.03333333 0.04003599  0.8571429
```

```
##HYPHOTHESIS 1

RPC <- aov(Rooms~Price, data = hdd_norm)
RPC
```

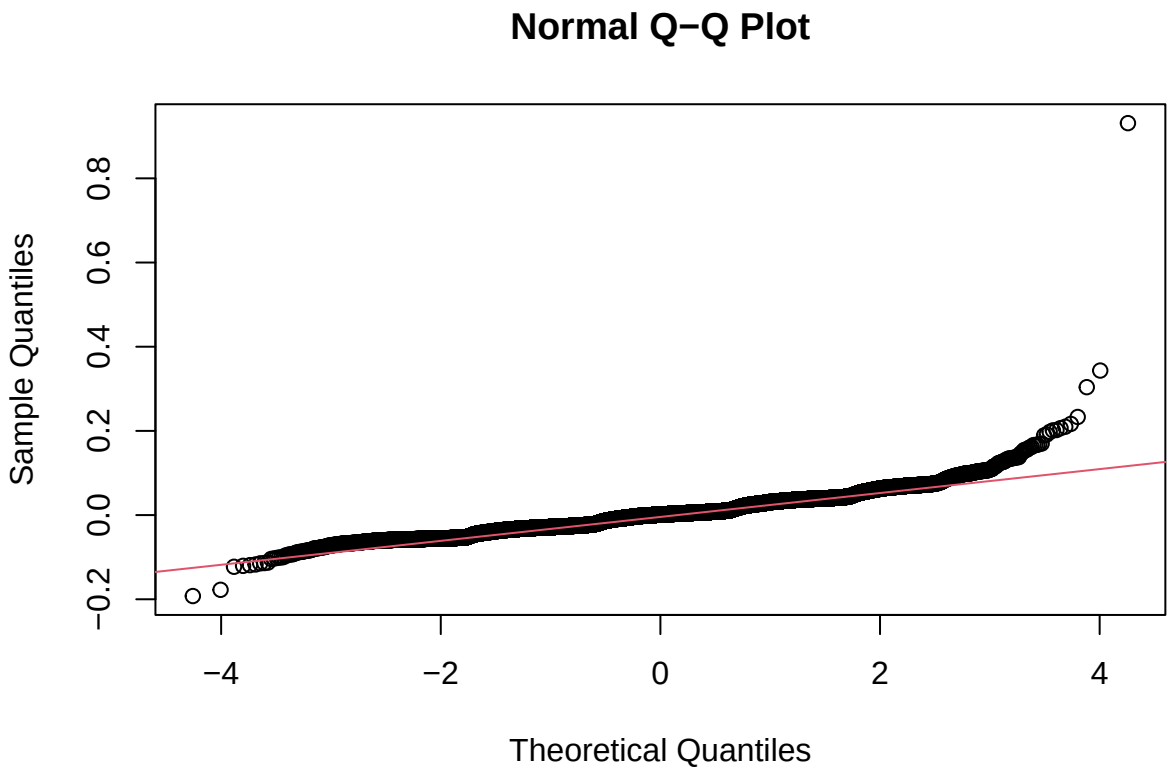
```
## Call:
##    aov(formula = Rooms ~ Price, data = hdd_norm)
##
## Terms:
##                     Price Residuals
```

```
## Sum of Squares   8.16962  39.85732
## Deg. of Freedom        1     48431
##
## Residual standard error: 0.02868748
## Estimated effects may be unbalanced
```

```
summary(RPC)
```

```
##                Df Sum Sq Mean Sq F value              Pr(>F)
## Price           1   8.17   8.170    9927 <0.0000000000000002 ***
## Residuals   48431  39.86   0.001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Normality Plot
qqnorm(residuals(RPC));qqline(residuals(RPC), col = 2)
```

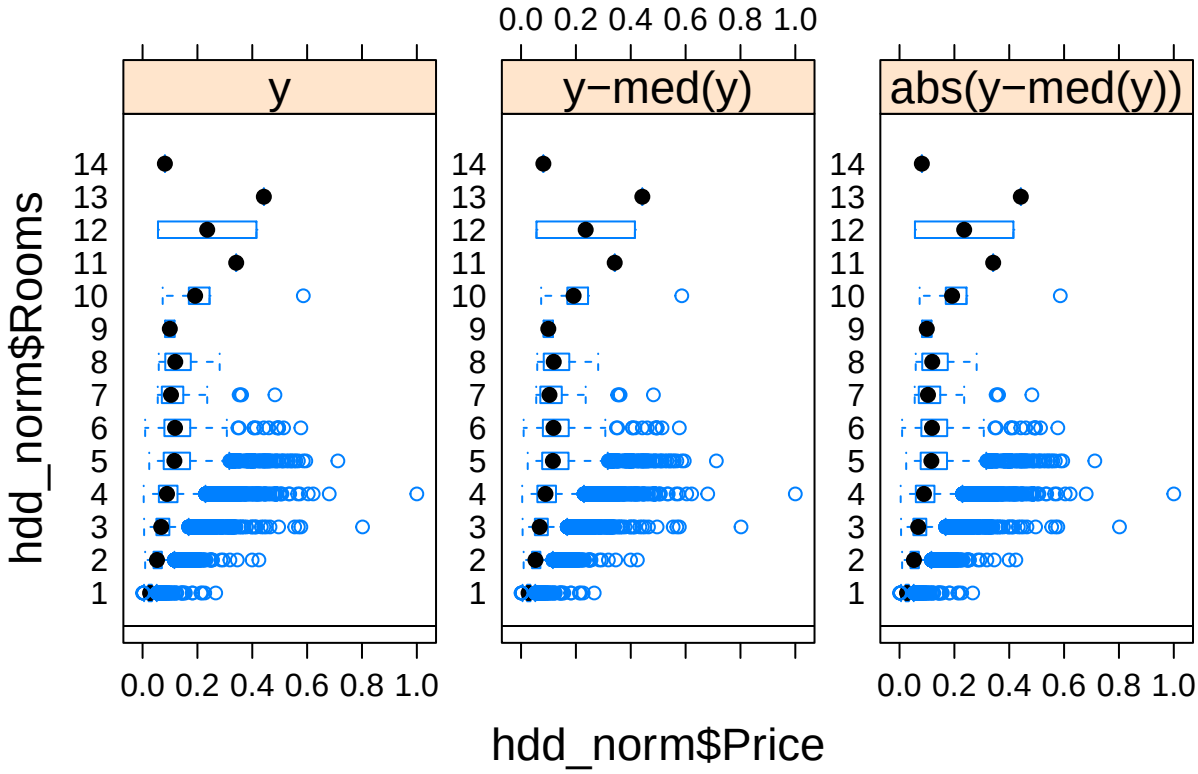



```
#Homogeneity of Variance Test
hov1<-hov(hdd_norm$Rooms ~ hdd_norm$Price)
hov1
```

```
##
##  hov: Brown-Forsyth
##
```

```
## data:  hdd_norm$Rooms
## F = 9927, df:hdd_norm$Price = 1, df:Residuals = 48431, p-value <
## 0.00000000000000022
## alternative hypothesis: variances are not identical
```

```
hovplot1<-hovPlot(hdd_norm$Rooms ~ hdd_norm$Price)
hovplot1
```



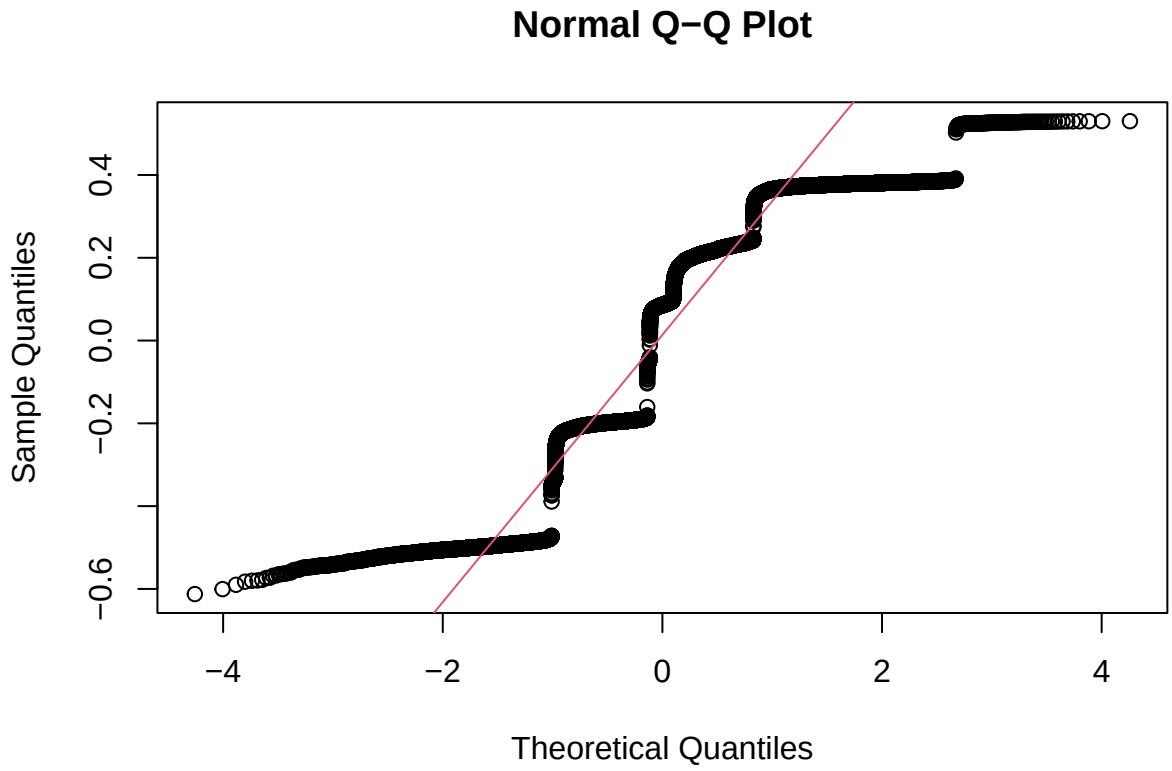

```
##HYPOTHESIS 2


DRD <- aov(Regionname~Price, data = hdd_norm)
DRD
```

```
## Call:
##    aov(formula = Regionname ~ Price, data = hdd_norm)
##
## Terms:
##                     Price Residuals
## Sum of Squares     15.246  4429.798
## Deg. of Freedom         1     48431
##
## Residual standard error: 0.3024337
## Estimated effects may be unbalanced
```

```
summary(DRD)
```

```
##                Df Sum Sq Mean Sq F value              Pr(>F)
## Price           1     15  15.246   166.7 <0.0000000000000002 ***
## Residuals   48431   4430   0.091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

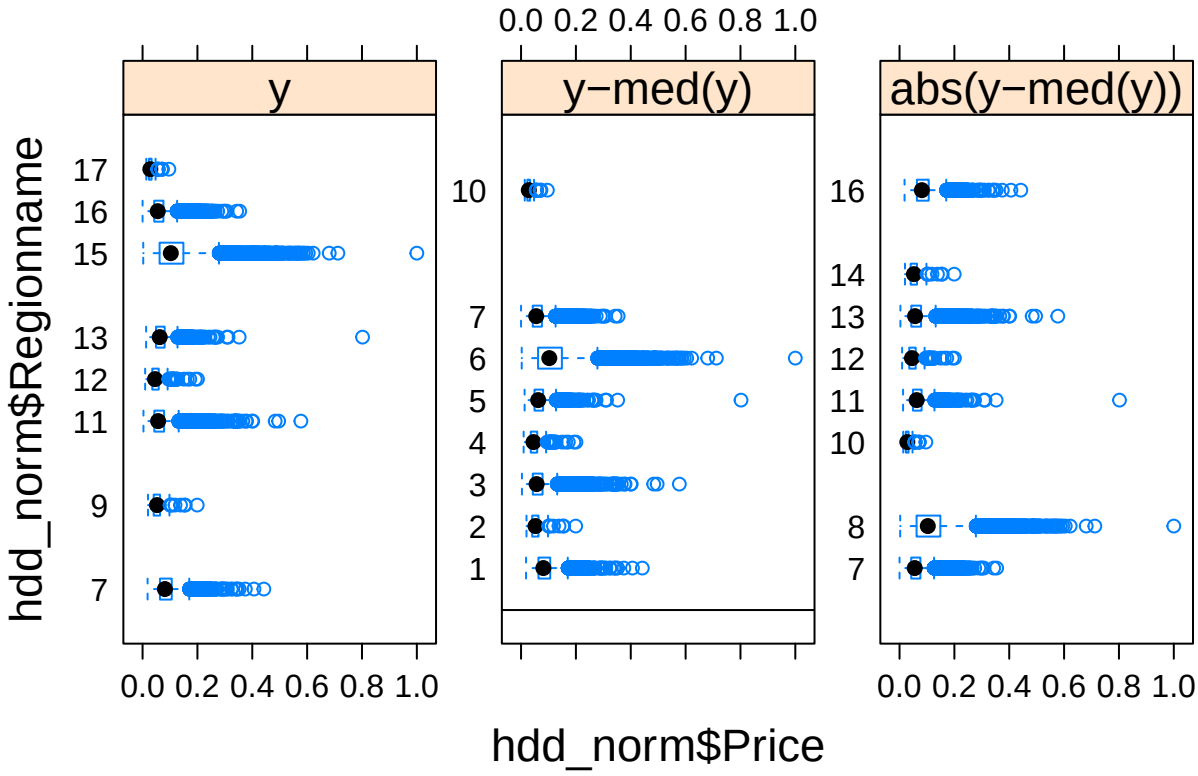
```
#Normality Plot
qqnorm(residuals(DRD));qqline(residuals(DRD), col = 2)
```




```
#Homogeneity of Variance Test
hov2<-hov(hdd_norm$Regionname ~ hdd_norm$Price)
hov2
```

```
##
##  hov: Brown-Forsyth
##
## data:  hdd_norm$Regionname
## F = 188.9, df:hdd_norm$Price = 1, df:Residuals = 48431, p-value <
## 0.00000000000000022
## alternative hypothesis: variances are not identical
```

```
hovplot2<-hovPlot(hdd_norm$Regionname ~ hdd_norm$Price)
hovplot2
```

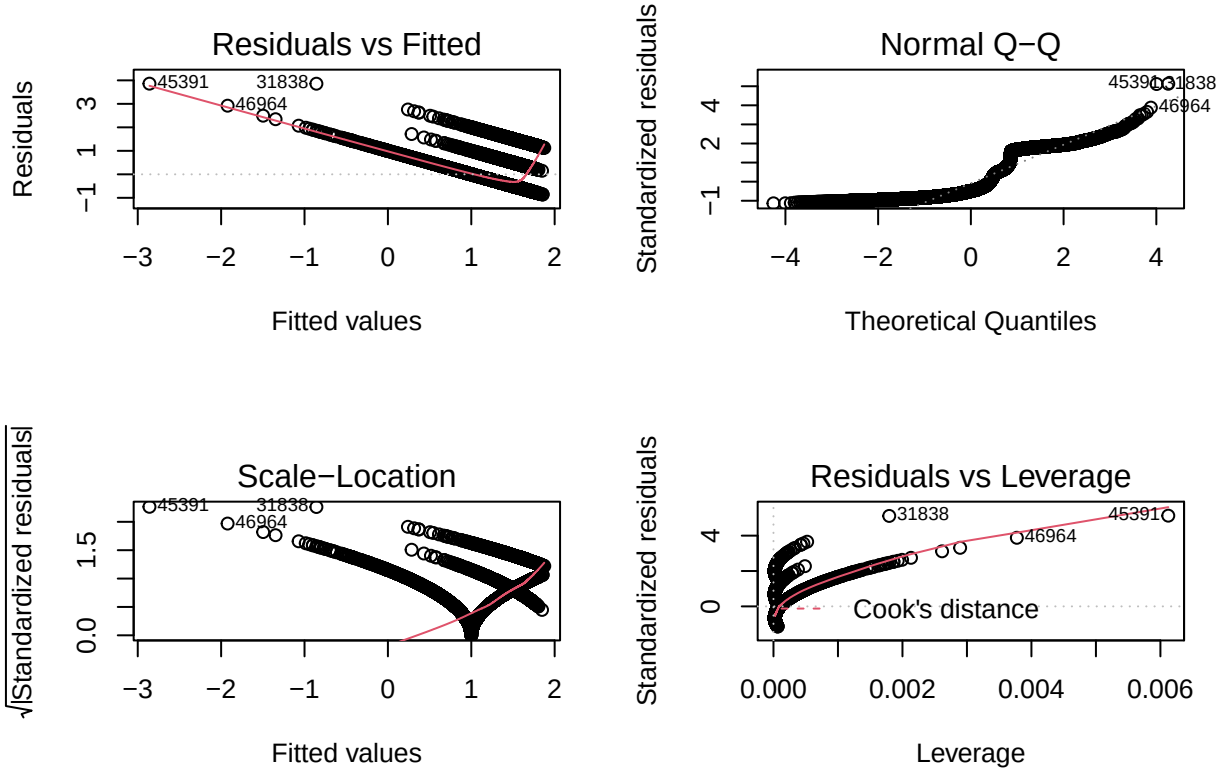



```
#split data into train and test data 75/25.

housing.dataset0<- housing.dataset <- read.csv("C:/Users/shubh/Downloads/melbourne_housing_data.csv")

housing.dataset0$Type[housing.dataset0$Type == 'h'] <- 1
housing.dataset0$Type[housing.dataset0$Type == 't'] <- 2
housing.dataset0$Type[housing.dataset0$Type == 'u'] <- 3
housing.dataset0$Type <- as.integer(housing.dataset0$Type)

housing.dataset0$Regionname[housing.dataset0$Regionname == 'Eastern Metropolitan'] <- 1
housing.dataset0$Regionname[housing.dataset0$Regionname == 'Eastern Victoria'] <- 2
housing.dataset0$Regionname[housing.dataset0$Regionname == 'Northern Metropolitan'] <- 3
housing.dataset0$Regionname[housing.dataset0$Regionname == 'Northern Victoria'] <- 4
housing.dataset0$Regionname[housing.dataset0$Regionname == 'South-Eastern Metropolitan'] <- 5
housing.dataset0$Regionname[housing.dataset0$Regionname == 'Southern Metropolitan'] <- 6
housing.dataset0$Regionname[housing.dataset0$Regionname == 'Western Metropolitan'] <- 7
housing.dataset0$Regionname[housing.dataset0$Regionname == 'Western Victoria'] <- 8
housing.dataset0$Regionname <- as.integer(housing.dataset0$Regionname)


set.seed(101)
sample = sample.split(housing.dataset0, SplitRatio = .75)
```

```r
train.data = subset(housing.dataset0, sample == TRUE)
test.data  = subset(housing.dataset0, sample == FALSE)

#Linear regression using multiple Variable to predict house price
Price.lm<-lm(Type ~ Price, data = housing.dataset0)
summary(Price.lm)
```

```
##
## Call:
## lm(formula = Type ~ Price, data = housing.dataset0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8558 -0.5772 -0.3727  0.4515  3.8593
##
## Coefficients:
##                    Estimate      Std. Error t value            Pr(>|t|)
## (Intercept)  1.911611026835  0.006709057376  284.93 <0.0000000000000002 ***
## Price       -0.000000425979  0.000000005778  -73.72 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7547 on 48431 degrees of freedom
## Multiple R-squared:  0.1009, Adjusted R-squared:  0.1009
## F-statistic:  5434 on 1 and 48431 DF,  p-value: < 0.00000000000000022
```

```r
#plotting
par(mfrow=c(2,2))
plot(Price.lm)
```




```
par(mfrow=c(1,1))


#Prediction of Price
p123<-predict(Price.lm,newdata = train.data)
lmp<-sqrt(mean((p123-train.data$Price)^2))
lmp
```

```
## [1] 1160021
```

```
#RMSE Correlation Predection of Test data
#Reporting R Squared
TRp<- lm(Rooms ~ Price, data = train.data)
summary(TRp)$adj.r.squared
```

```
## [1] 0.1705309
```

```
pr<- lm(Regionname ~ Price, data = train.data)
summary(pr)$adj.r.squared
```

```
## [1] 0.003170581
```

```r
#Reporting Prediction
Predict1<-predict(TRp, newdata = test.data)
RMSE1<-sqrt(mean((Predict1-test.data$Rooms)^2))
RMSE1
```

```
## [1] 0.8461915
```

```r
Predict2<-predict(pr, newdata = test.data)
RMSE2<-sqrt(mean((Predict2-test.data$Regionname)^2))
RMSE2
```

```
## [1] 2.117851
```

```r
#creating Table
tabrmse<-table(RMSE1,RMSE2)

#Calculating Accuracy
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
 accuracy(tabrmse)
```

```
## [1] 100
```

```r
#Normalizing Data

norm2 <- function(x) {(x - min(x)) / (max(x) - min(x))}

hdd_norm2 <- as.data.frame(lapply(housing.dataset0[,c(5,6)], norm2))
head(hdd_norm2)
```

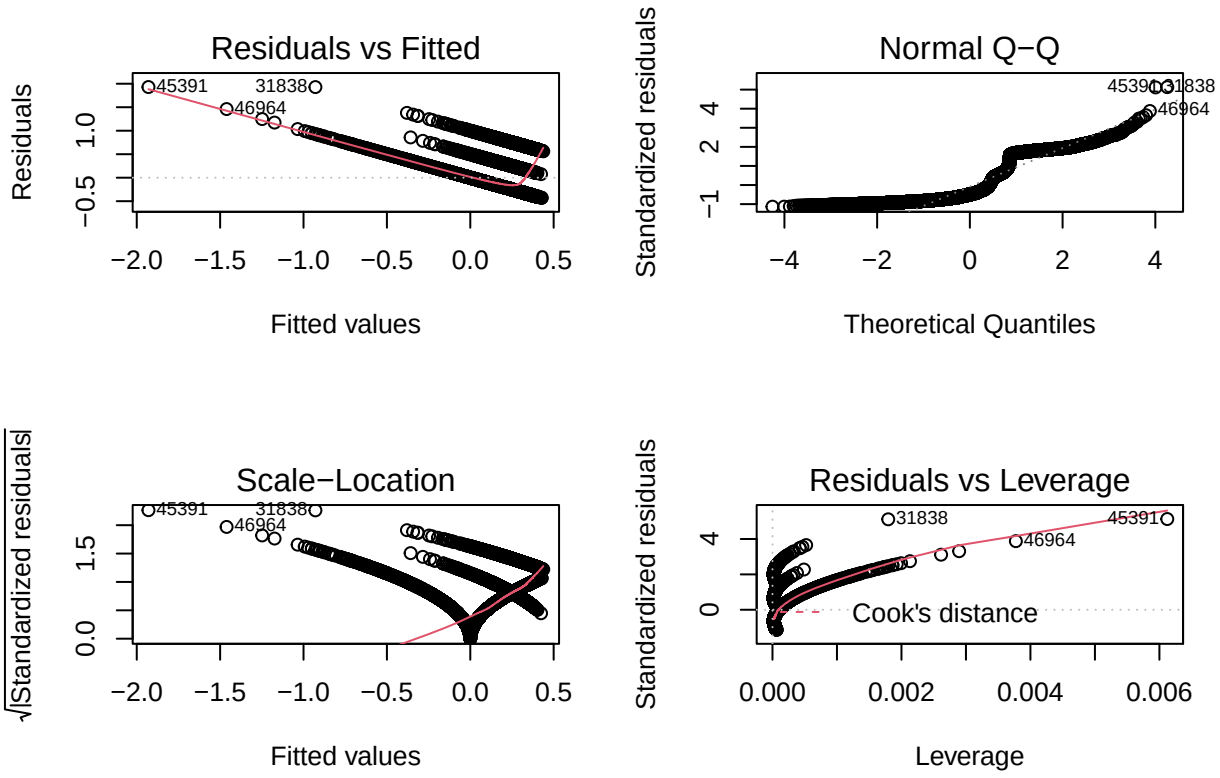
```
##   Type      Price
## 1  0.0 0.12640576
## 2  0.0 0.10211426
## 3  0.0 0.12010796
## 4  0.0 0.12865497
## 5  0.0 0.05263158
## 6  0.5 0.04003599
```

```r
#Creating a simple regression model

Price.lm2<-lm(Type ~ Price, data = hdd_norm2)
summary(Price.lm2)
```

```
##
## Call:
## lm(formula = Type ~ Price, data = hdd_norm2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4279 -0.2886 -0.1864  0.2258  1.9297
##
```

```
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept)  0.437701   0.003146  139.13 <0.0000000000000002 ***
## Price       -2.367376   0.032114  -73.72 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3774 on 48431 degrees of freedom
## Multiple R-squared:  0.1009, Adjusted R-squared:  0.1009
## F-statistic:  5434 on 1 and 48431 DF,  p-value: < 0.00000000000000022
```

```
#plotting
par(mfrow=c(2,2))
plot(Price.lm2)
```




```
par(mfrow=c(1,1))

PND <- predict(Price.lm2, newdata = housing.dataset0)
summary(PND)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -26514608  -2888198  -1964921  -2362400  -1467773   -201227
```

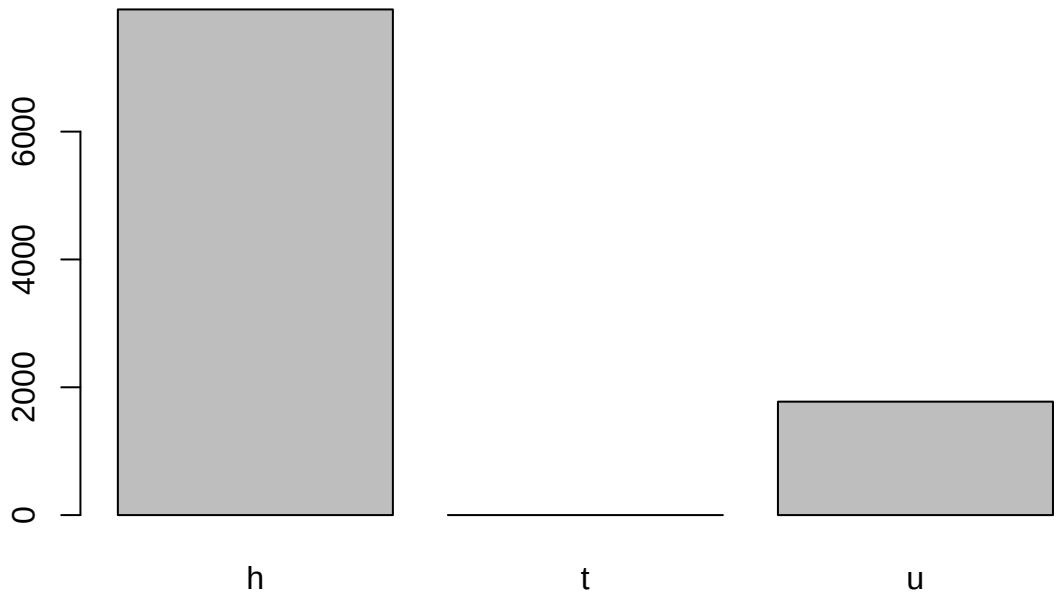
```r
HD1<-housing.dataset <- read.csv("C:/Users/shubh/Downloads/melbourne_housing_data.csv")




#spliting data into 80/20.
NHD <- sample(1:nrow(HD1), 0.8 * nrow(HD1))

#normalizing data
nor <-function(x) { (x -min(x))/(max(x)-min(x))   }

HD_norm <- as.data.frame(lapply(HD1[,c(4,6)], nor))
head(HD_norm)
```

```
##        Rooms      Price
## 1 0.06666667 0.12640576
## 2 0.06666667 0.10211426
## 3 0.06666667 0.12010796
## 4 0.06666667 0.12865497
## 5 0.03333333 0.05263158
## 6 0.03333333 0.04003599
```

```r
HD_train <- HD_norm[NHD,]

HD_test <- HD_norm[-NHD,]

HD_target <- as.factor(HD1[NHD,5])

HDtest_target <- as.factor(HD1[-NHD,5])

##run knn function
HDKNN <- knn(HD_train,HD_test,cl=HD_target,k=16)

#finding accuracy
tabHD<- table(HDKNN,HDtest_target)
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tabHD)
```

```
## [1] 81.34613
```

```r
plot(HDKNN)
```




```
# C 5.0 Classification Model

HD2<-housing.dataset <- read.csv("C:/Users/shubh/Downloads/melbourne_housing_data.csv")

 ctr <- c("Price","Rooms")
 str(HD2[, c(ctr, "Type")])
```

```
## 'data.frame':    48433 obs. of  3 variables:
##  $ Price: int  1490000 1220000 1420000 1515000 670000 530000 540000 715000 1925000 515000 ...
##  $ Rooms: int  3 3 3 3 2 2 2 3 3 3 ...
##  $ Type : chr  "h" "h" "h" "h" ...
```

```
set.seed(1001)
sample = sample.split(HD2, SplitRatio = .80)
train.HD2 = subset(HD2, sample == TRUE)
test.HD2  = subset(HD2, sample == FALSE)


train.HD2$Type<-as.factor(train.HD2$Type)
str(train.HD2$Type)
```

```
##  Factor w/ 3 levels "h","t","u": 1 1 1 1 3 1 1 3 1 1 ...
```

```
hdc5.0 <- C5.0(x = train.HD2[ctr], y = train.HD2$Type)
hdc5.0
```

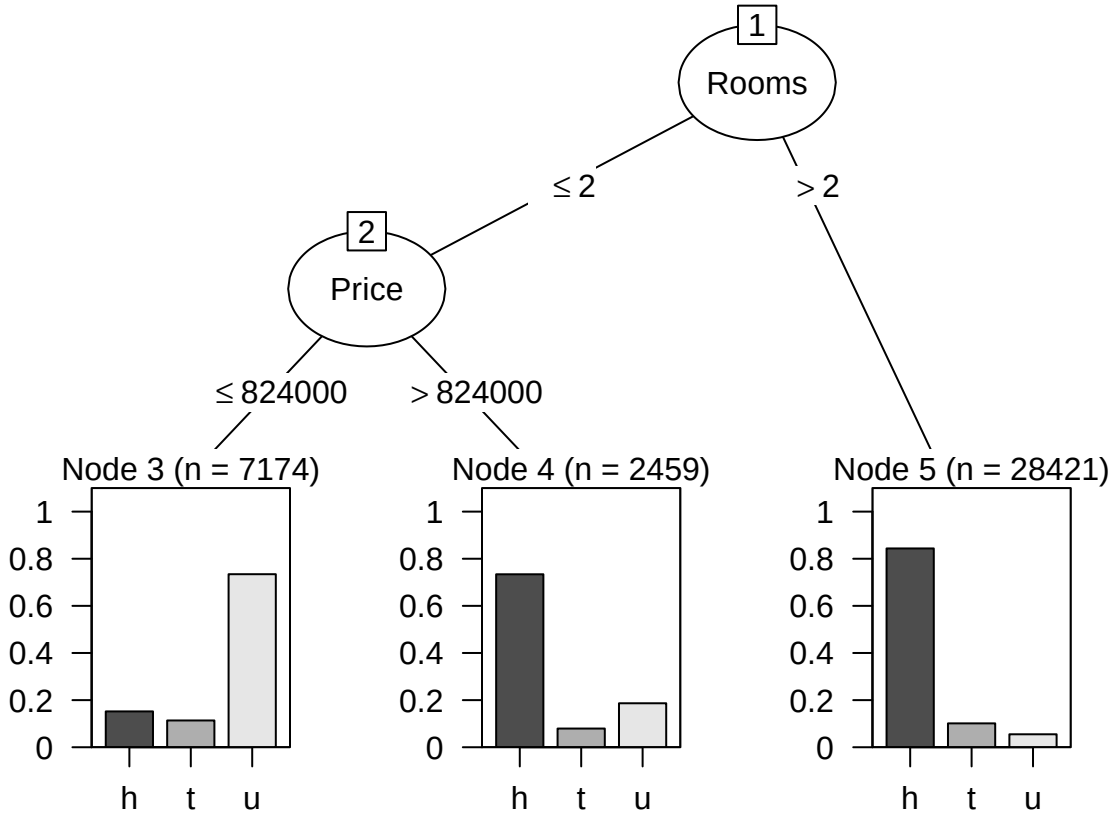
```
##
## Call:
## C5.0.default(x = train.HD2[ctr], y = train.HD2$Type)
##
## Classification Tree
## Number of samples: 38054
## Number of predictors: 2
##
## Tree size: 3
##
## Non-standard options: attempt to group attributes
```

```
summary(hdc5.0)
```

```
##
## Call:
## C5.0.default(x = train.HD2[ctr], y = train.HD2$Type)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Dec 14 23:05:48 2021
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 38054 cases (3 attributes) from undefined.data
##
## Decision tree:
##
## Rooms > 2: h (28421/4444)
## Rooms <= 2:
## :...Price <= 824000: u (7174/1906)
##     Price > 824000: h (2459/654)
##
##
## Evaluation on training data (38054 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       3 7004(18.4%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##   25782        1091    (a): class h
##    3075         815    (b): class t
##    2023        5268    (c): class u
##
##
```

```
##  Attribute usage:
##
##  100.00% Rooms
##   25.31% Price
##
##
## Time: 0.0 secs
```

```
plot(hdc5.0)
```




```
#Showing Price Range With Rooms And Type Data
```

```
#Ann Classification

HD3<-housing.dataset <- read.csv("C:/Users/shubh/Downloads/melbourne_housing_data.csv")



set.seed(1001)
sample = sample.split(HD3, SplitRatio = .80)
train.HD3 = subset(HD3, sample == TRUE)
test.HD3  = subset(HD3, sample == FALSE)


train.HD3$Type<-as.factor(train.HD3$Type)
str(train.HD3$Type)
```

```
##  Factor w/ 3 levels "h","t","u": 1 1 1 1 3 1 1 3 1 1 ...
```

```
train.HD3$Rooms<-as.factor(train.HD3$Rooms)
str(train.HD2$Rooms)
```

```
##  int [1:38054] 3 3 3 2 2 3 3 3 4 2 ...
```

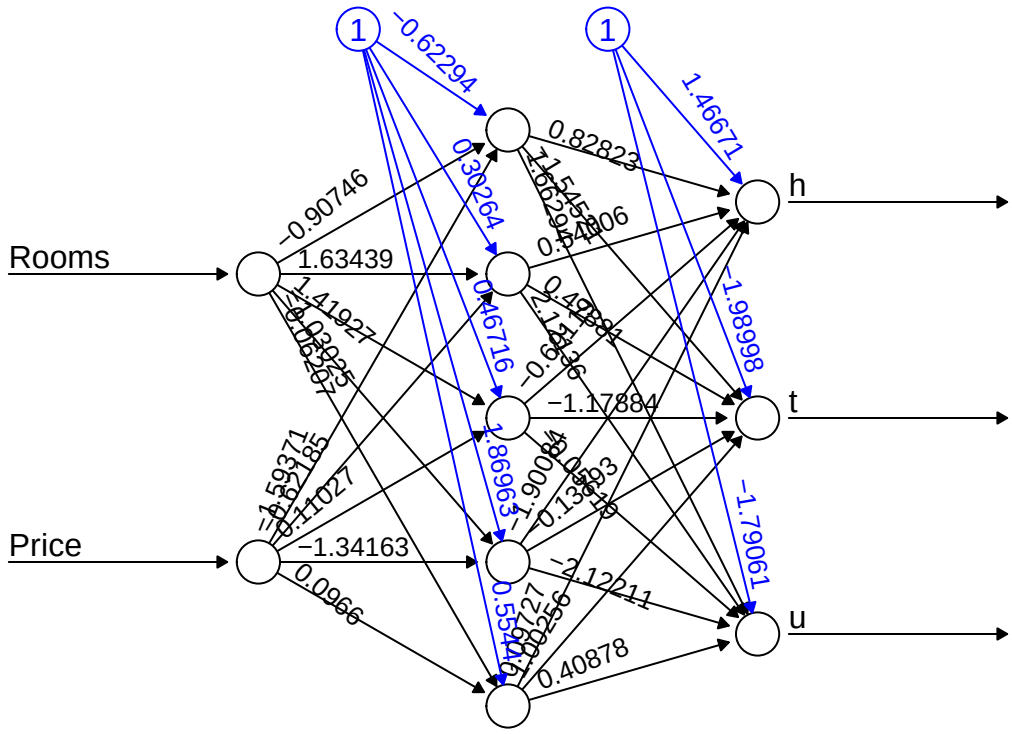
```
train.HD3$Price<-as.factor(train.HD3$Price)
str(train.HD3$Price)
```

```
##  Factor w/ 3131 levels "85000","112000",..: 1874 2126 2229 866 535 975 2566 474 982 2372 ...
```

```
#HD3N <- model.matrix(~Price+Rooms+Type+Regionname, data = HD3 )

Ann.HD3=neuralnet(Type~Rooms+Price,data=HD3, hidden=5,threshold = 0.01,
                linear.output = FALSE)
summary(Ann.HD3)
```

```
##                     Length Class      Mode
## call                     6 -none-     call
## response            145299 -none-     logical
## covariate            96866 -none-     numeric
## model.list               2 -none-     list
## err.fct                  1 -none-     function
## act.fct                  1 -none-     function
## linear.output            1 -none-     logical
## data                    14 data.frame list
## exclude                  0 -none-     NULL
## net.result               1 -none-     list
## weights                  1 -none-     list
## generalized.weights      1 -none-     list
## startweights             1 -none-     list
## result.matrix           36 -none-     numeric
```

```
plot(Ann.HD3 ,rep = 'best')
```



Compairing ANN Performance C 5.0 and KNN Showing accurate And Valid prediction on price so C 5.0 Performing Well.##### ## While Knn gave us approx 81% Accuracy.