# Part 1: Basic Regression Analysis of Perth Housing Prices

Sunni Magan

Source: https://www.kaggle.com/datasets/syuzai/perth-house-prices

```
df <- read.csv("perth.csv")
```

## Data Pre-Processing

```
str(df)
```

```
## 'data.frame':    33656 obs. of  19 variables:
##  $ ADDRESS         : chr  "1 Acorn Place" "1 Addis Way" "1 Ainsley Court" "1 Albert Street" ...
##  $ SUBURB          : chr  "South Lake" "Wandi" "Camillo" "Bellevue" ...
##  $ PRICE           : int  565000 365000 287000 255000 325000 409000 400000 370000 565000 685000 ...
##  $ BEDROOMS        : int  4 3 3 2 4 4 3 4 4 3 ...
##  $ BATHROOMS       : int  2 2 1 1 1 2 2 2 2 2 ...
##  $ GARAGE          : chr  "2" "2" "1" "2" ...
##  $ LAND_AREA       : int  600 351 719 651 466 759 386 468 875 552 ...
##  $ FLOOR_AREA      : int  160 139 86 59 131 118 132 158 168 126 ...
##  $ BUILD_YEAR      : chr  "2003" "2013" "1979" "1953" ...
##  $ CBD_DIST        : int  18300 26900 22600 17900 11200 27300 28200 41700 12100 5900 ...
##  $ NEAREST_STN     : chr  "Cockburn Central Station" "Kwinana Station" "Challis Station" "Midland Sta
##  $ NEAREST_STN_DIST: int  1800 4900 1900 3600 2000 1000 3700 1100 2500 508 ...
##  $ DATE_SOLD       : chr  "09-2018\n" "02-2019\n" "06-2015\n" "07-2018\n" ...
##  $ POSTCODE        : int  6164 6167 6111 6056 6054 6112 6112 6169 6022 6053 ...
##  $ LATITUDE        : num  -32.1 -32.2 -32.1 -31.9 -31.9 ...
##  $ LONGITUDE       : num  116 116 116 116 116 ...
##  $ NEAREST_SCH     : chr  "LAKELAND SENIOR HIGH SCHOOL" "ATWELL COLLEGE" "KELMSCOTT SENIOR HIGH SCHO
##  $ NEAREST_SCH_DIST: num  0.828 5.524 1.649 1.571 1.515 ...
##  $ NEAREST_SCH_RANK: int  NA 129 113 NA NA NA NA NA NA 29 ...
```

Selecting only wanted features

```
df <- subset(df, select=c(PRICE, BEDROOMS, BATHROOMS, GARAGE, LAND_AREA, FLOOR_AREA, BUILD_YEAR, NEAREST
```

GARAGE and BUILD_YEAR is a string but would be better suited as an integer. However, after doing this the NA values must then be removed.

```
df$GARAGE <- as.integer(df$GARAGE)
df <- df[!is.na(df$GARAGE),]

df$BUILD_YEAR <-as.integer(df$BUILD_YEAR)
df <- df[!is.na(df$BUILD_YEAR),]
```

### Data Exploration

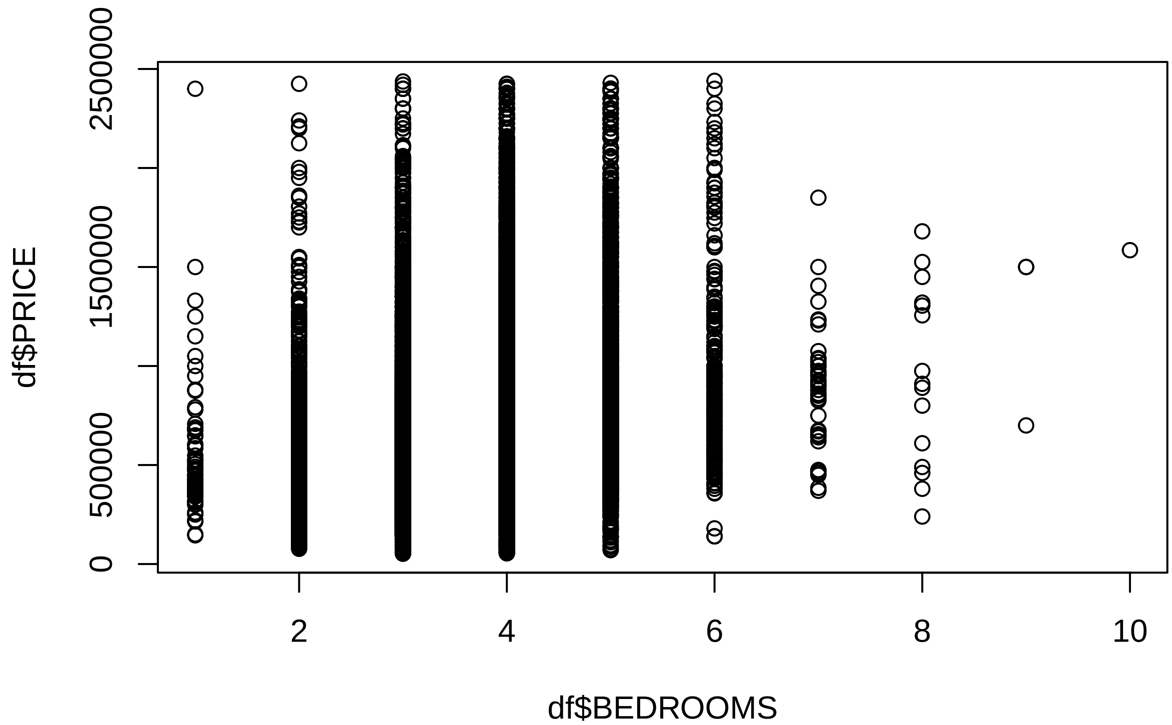
```
summary(df)
```

```
##      PRICE           BEDROOMS        BATHROOMS         GARAGE     
##  Min.   :  52000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
##  1st Qu.: 412000   1st Qu.: 3.000   1st Qu.: 2.000   1st Qu.: 2.000  
##  Median : 540000   Median : 4.000   Median : 2.000   Median : 2.000  
##  Mean   : 643243   Mean   : 3.674   Mean   : 1.841   Mean   : 2.196  
##  3rd Qu.: 770000   3rd Qu.: 4.000   3rd Qu.: 2.000   3rd Qu.: 2.000  
##  Max.   :2440000   Max.   :10.000   Max.   :16.000   Max.   :99.000  
##    LAND_AREA        FLOOR_AREA      BUILD_YEAR   NEAREST_STN_DIST
##  Min.   :    61   Min.   :  1.0   Min.   :1868   Min.   :   46   
##  1st Qu.:   504   1st Qu.:130.0   1st Qu.:1979   1st Qu.: 1700   
##  Median :   681   Median :172.0   Median :1995   Median : 3200   
##  Mean   :  2492   Mean   :183.3   Mean   :1990   Mean   : 4414   
##  3rd Qu.:   822   3rd Qu.:222.0   3rd Qu.:2005   3rd Qu.: 5200   
##  Max.   :999999   Max.   :849.0   Max.   :2017   Max.   :35500   
##  NEAREST_SCH_DIST
##  Min.   : 0.07091
##  1st Qu.: 0.87322
##  Median : 1.32923
##  Mean   : 1.76819
##  3rd Qu.: 2.05559
##  Max.   :20.72091
```

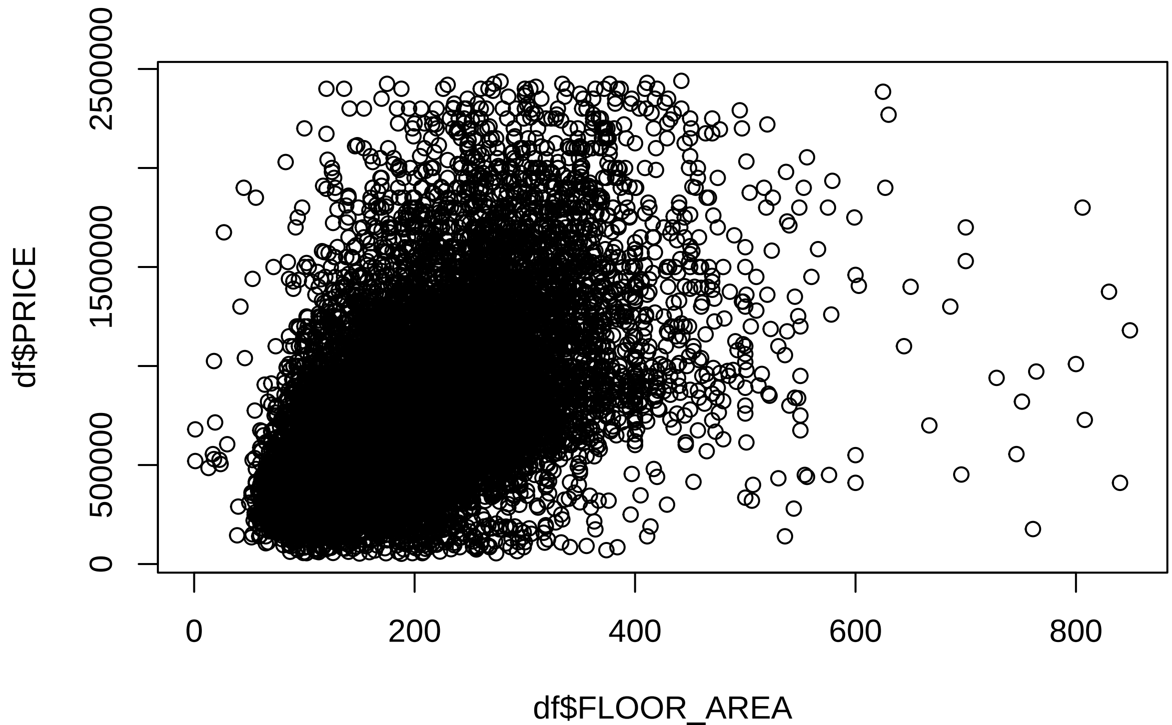
```
cor(df)
```

```
##                        PRICE   BEDROOMS  BATHROOMS     GARAGE   LAND_AREA
## PRICE             1.00000000 0.26925496 0.39290393 0.12947644 0.055033219
## BEDROOMS          0.26925496 1.00000000 0.56530152 0.19063445 0.050623320
## BATHROOMS         0.39290393 0.56530152 1.00000000 0.18186469 0.031912497
## GARAGE            0.12947644 0.19063445 0.18186469 1.00000000 0.053779668
## LAND_AREA         0.05503322 0.05062332 0.03191250 0.05377967 1.000000000
## FLOOR_AREA        0.56630505 0.55139974 0.57998214 0.19663942 0.065111231
## BUILD_YEAR       -0.16087941 0.22196494 0.34345839 0.04037070 0.004999639
## NEAREST_STN_DIST -0.08904211 0.11019277 0.04870937 0.10813823 0.211553978
## NEAREST_SCH_DIST -0.01211007 0.09361769 0.07170308 0.09392427 0.252991830
##                  FLOOR_AREA   BUILD_YEAR NEAREST_STN_DIST NEAREST_SCH_DIST
## PRICE            0.56630505 -0.160879412      -0.08904211      -0.01211007
## BEDROOMS         0.55139974  0.221964943       0.11019277       0.09361769
## BATHROOMS        0.57998214  0.343458394       0.04870937       0.07170308
## GARAGE           0.19663942  0.040370696       0.10813823       0.09392427
## LAND_AREA        0.06511123  0.004999639       0.21155398       0.25299183
## FLOOR_AREA       1.00000000  0.222725375       0.10499182       0.11683520
## BUILD_YEAR       0.22272538  1.000000000       0.10961194       0.11220423
## NEAREST_STN_DIST 0.10499182  0.109611940       1.00000000       0.61731952
## NEAREST_SCH_DIST 0.11683520  0.112204234       0.61731952       1.00000000
```

```
plot(df$BEDROOMS, df$PRICE)
```




```
plot(df$FLOOR_AREA, df$PRICE)
```




### Train-Test Split

```
set.seed(1234)
i <- sample(1:nrow(df), 0.75*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

## Linear Regression

```
#linear_model <- lm(PRICE~., data=train)
linear_model <- lm(PRICE~., data=train)
summary(linear_model)
```

```
##
## Call:
## lm(formula = PRICE ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2235006  -141375   -36393    94920  1751735
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.146e+07  1.844e+05  62.160  < 2e-16 ***
## BEDROOMS          -4.402e+04  3.165e+03 -13.909  < 2e-16 ***
```

```
## BATHROOMS         1.401e+05  4.178e+03  33.530  < 2e-16 ***
## GARAGE            6.215e+03  1.489e+03   4.173 3.02e-05 ***
## LAND_AREA         6.457e-01  1.050e-01   6.148 8.01e-10 ***
## FLOOR_AREA        2.845e+03  3.317e+01  85.773  < 2e-16 ***
## BUILD_YEAR       -5.733e+03  9.376e+01 -61.149  < 2e-16 ***
## NEAREST_STN_DIST -1.173e+01  5.306e-01 -22.107  < 2e-16 ***
## NEAREST_SCH_DIST  6.032e+03  1.397e+03   4.317 1.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 262100 on 21183 degrees of freedom
## Multiple R-squared:  0.4599, Adjusted R-squared:  0.4597
## F-statistic:  2255 on 8 and 21183 DF,  p-value: < 2.2e-16
```

```
pred <- predict(linear_model, newdata=test)
cor <- cor(pred, test$PRICE)
mse <- mean((pred - test$PRICE)^2)
print(paste("cor=", cor))
```

```
## [1] "cor= 0.664633754694884"
```

```
print(paste("mse=", mse))
```

```
## [1] "mse= 70482614044.9292"
```

## kNN Regression

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
kNN_model <- knnreg(train[,2:9],train[,1], k=5)
```

```
pred <- predict(kNN_model, newdata=test[,2:9])
cor <- cor(pred, test$PRICE)
mse <- mean((pred - test$PRICE)^2)
print(paste("cor=", cor))
```

```
## [1] "cor= 0.626252538902696"
```

```
print(paste("mse=", mse))
```

```
## [1] "mse= 77783050699.1665"
```

### Decision Tree Regression

```
library(tree)
```

```
dtree <- tree(PRICE~., data=train)
summary(dtree)
```

```
##
## Regression tree:
## tree(formula = PRICE ~ ., data = train)
## Variables actually used in tree construction:
## [1] "FLOOR_AREA"       "BUILD_YEAR"       "BATHROOMS"        "NEAREST_STN_DIST"
```

```
## Number of terminal nodes:  7
## Residual mean deviance:  7.65e+10 = 1.621e+15 / 21180
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1169000  -148900   -43890        0    99240  1851000
```

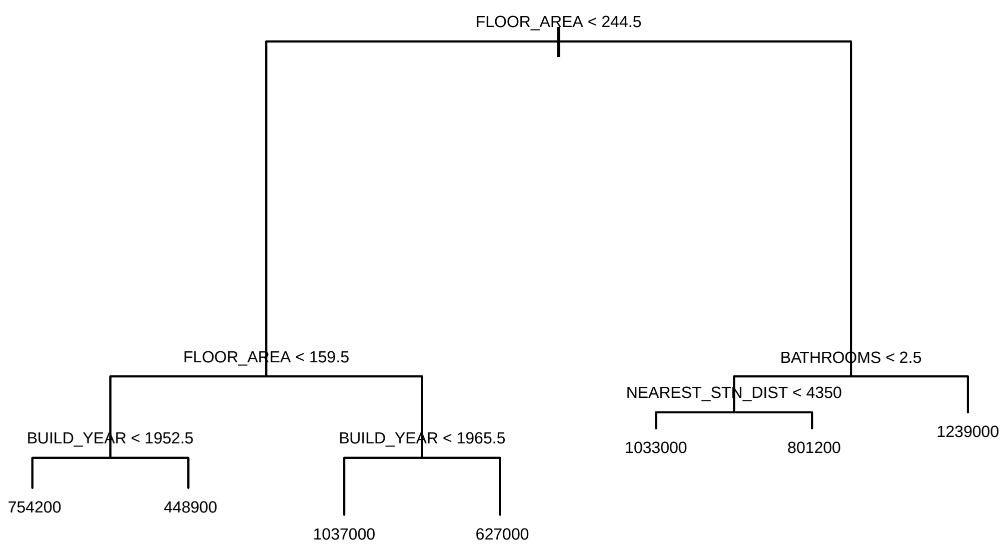
```
pred <- predict(dtree, newdata=test)
cor <- cor(pred, test$PRICE)
mse <- mean((pred - test$PRICE)^2)
print(paste("cor=", cor))
```

```
## [1] "cor= 0.622155502615899"
```

```
print(paste("mse=", mse))
```

```
## [1] "mse= 77279578826.384"
```

```
plot(dtree)
text(dtree, cex=0.5, pretty=1)
```




### Results

Interestingly, linear regression performed slightly better than both kNN Regression and decision trees. This could mean that there is a linearity to the data. Decision trees have the highest MSE which could be due to fact that the house prices are put into discrete bins. kNN had trouble as well, possibly due to the amount of features used.