Home_Sale_Prediction_using_RF_KNN_LM_RPART.R

santhoshthirumalai

2019-06-16

```
#==========================================================================#
#  Author: Santhosh Thirumalai
#  Date Written: 2019/06/16
#==========================================================================#
#                  !!!!!!!!!!!!!!WARNING!!!!!!!!
#==========================================================================#
#  1. The R version used to create this script is 3.5.3
#==========================================================================#

#==========================================================================#
#   Algorithm to predict the prices of the homes in Sacramento CA Area
#==========================================================================#
#   The following methods are used to predict the data
#==========================================================================#
#   1. Classification and Regression Trees - rpart()
#   2. K - Nearest Neigbors - knn()
#   3. Random Forest - randomforest()
#   4. Linear model - lm()
#==========================================================================#

#==========================================================================#
#  Add the libraries required to run the algorithm
#==========================================================================#
library(dslabs)
library(tidyverse)

## ── Attaching packages ────────────────────────────────────────────────────
tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.1     ✔ purrr   0.3.2
## ✔ tibble  2.1.2     ✔ dplyr   0.8.1
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0

## ── Conflicts ─────────────────────────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

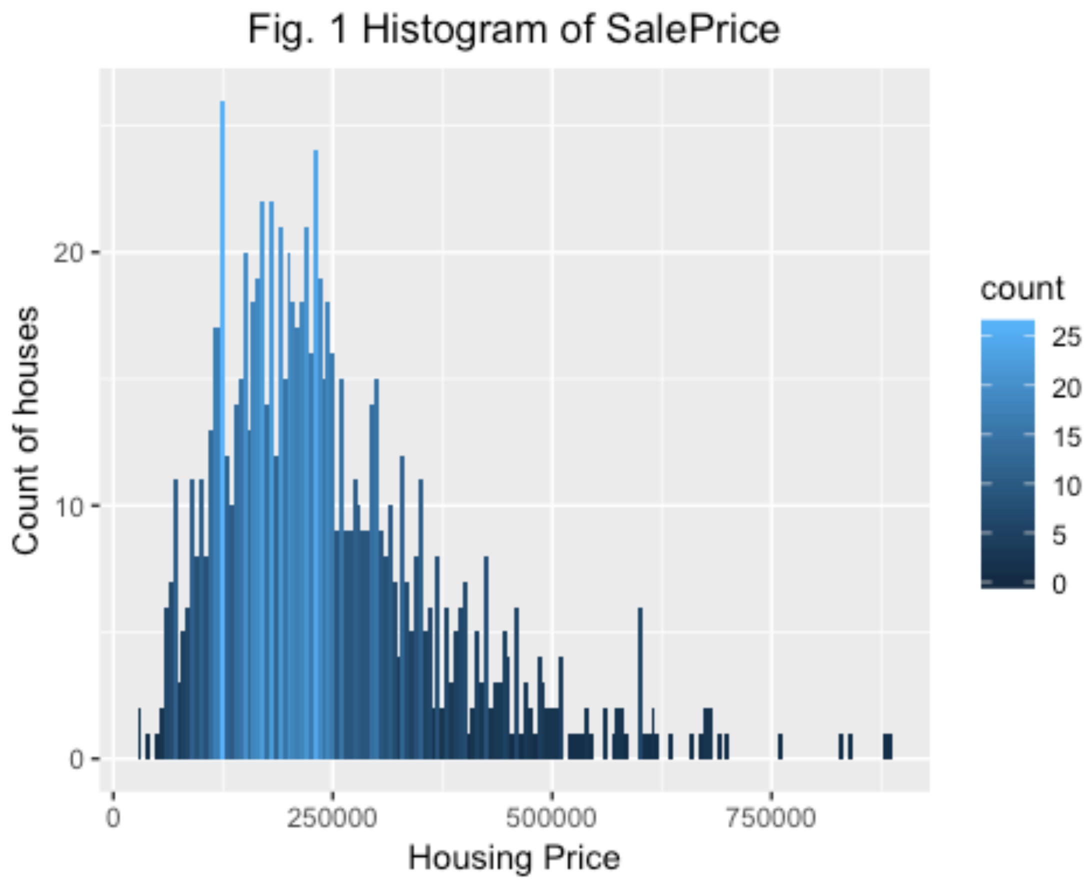
```r
library(ggplot2)
library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

library(reshape2)

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

library(forecast)
library(rpart)
library(rpart.plot)
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

#=========================================================================#
# set the seed to 1 to get the same result every time
#=========================================================================#
set.seed(1)

options(warn=-1)

#=========================================================================#
# Load the dataframe "Sacramento" from the dslabs package
#=========================================================================#
```

```r
data("Sacramento")

#=========================================================================#
# This area will research on the data present in the Sacramento dataset
#=========================================================================#

#=========================================================================#
# Print the dimensions on the dataset to find the predictors and outcomes
#=========================================================================#

dim(Sacramento)

## [1] 932   9

#=========================================================================#
#  Let's print the summary of the Sacramento dataset
#=========================================================================#
summary(Sacramento)

##              city            zip            beds           baths
##  SACRAMENTO    :438   z95823 : 61   Min.   :1.000   Min.   :1.000
##  ELK_GROVE     :114   z95828 : 45   1st Qu.:3.000   1st Qu.:2.000
##  ROSEVILLE     : 48   z95758 : 44   Median :3.000   Median :2.000
##  CITRUS_HEIGHTS: 35   z95835 : 37   Mean   :3.276   Mean   :2.053
##  ANTELOPE      : 33   z95838 : 37   3rd Qu.:4.000   3rd Qu.:2.000
##  RANCHO_CORDOVA: 28   z95757 : 36   Max.   :8.000   Max.   :5.000
##  (Other)       :236   (Other):672
##       sqft                type          price            latitude
##  Min.   : 484   Condo       : 53   Min.   : 30000   Min.   :38.24
##  1st Qu.:1167   Multi_Family: 13   1st Qu.:156000   1st Qu.:38.48
##  Median :1470   Residential :866   Median :220000   Median :38.62
##  Mean   :1680                      Mean   :246662   Mean   :38.59
##  3rd Qu.:1954                      3rd Qu.:305000   3rd Qu.:38.69
##  Max.   :4878                      Max.   :884790   Max.   :39.02
##
##    longitude
##  Min.   :-121.6
##  1st Qu.:-121.4
##  Median :-121.4
##  Mean   :-121.4
##  3rd Qu.:-121.3
##  Max.   :-120.6
##

#=========================================================================#
# The Goal is to predict the sale price of the homes in Sacramento Area
# At the END of the module the RMSE values of each prediction will be
# displayed as a conclusion
#=========================================================================#
#  The outcome field: Sacramento$price - sale price
```

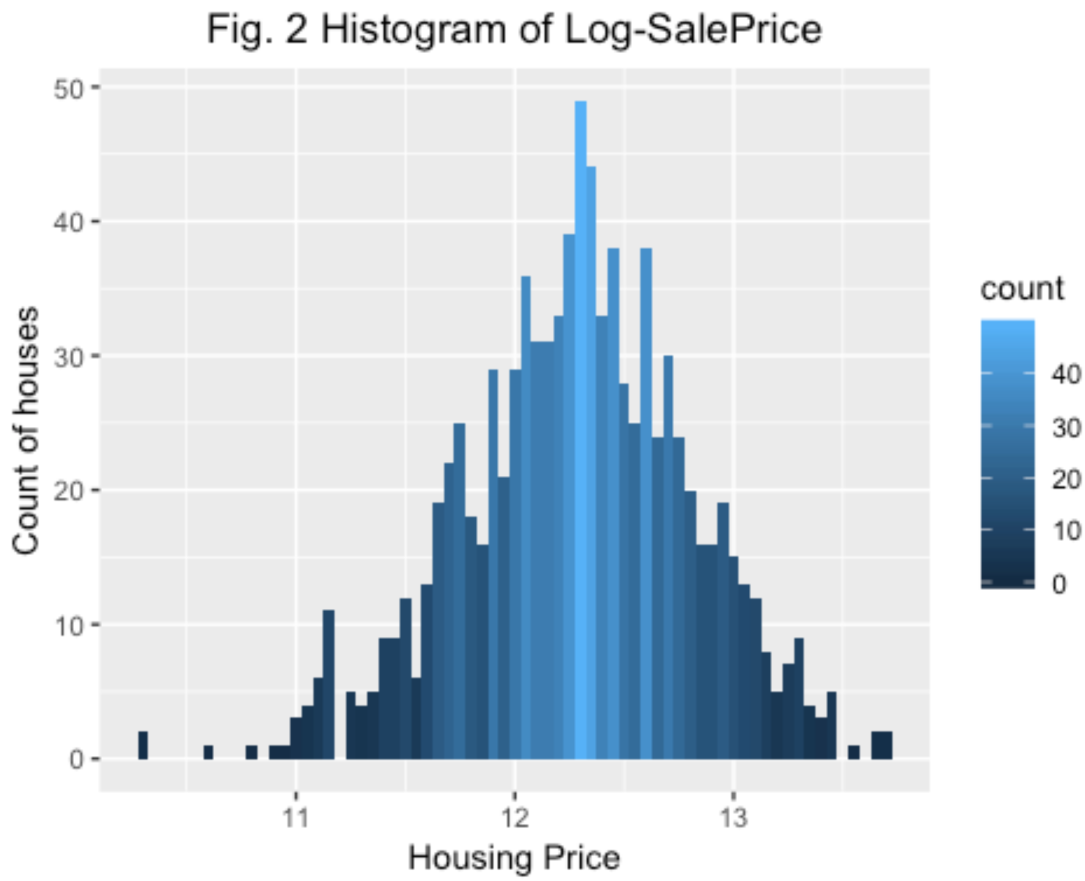
```
#=========================================================================#
#  The following fields may be used as predictors
#  1. city        - factor
#  2. zip         - factor
#  3. beds        - Integer
#  4. baths       - numeric
#  5. sqft        - integer
#  6. type        - factor
#  7. latitude    - numeric
#  8. longitude   - numeric
#=========================================================================#
names(Sacramento)
```

```
## [1] "city"      "zip"       "beds"      "baths"     "sqft"      "type"
## [7] "price"     "latitude"  "longitude"
```

```
#=========================================================================#
#  The summary shows that there are 3 types of homes with more sales
#  in Sacramento city, the sqft ranges from 484 to 4878. The beds in
#  the homes ranges from 1 to 8 and baths are between 1 and 5.
#  The price ranges are from 30000$ to a max of 885000$.
#=========================================================================#
#  Let's find out the predictors for the outcomes using vizualization
#=========================================================================#

#=========================================================================#
# Compute the mean of the sale price in Sacramento area
#=========================================================================#
avg_sale_price <- mean(Sacramento$price)

#=========================================================================#
# Let's see if the sale price and number of sales are normally
# Distributed
#=========================================================================#

options(scipen=10000)
ggplot(Sacramento, aes(x = price, fill = ..count..)) +
  geom_histogram(binwidth = 5000) +
  ggtitle("Fig. 1 Histogram of SalePrice") +
  ylab("Count of houses") +
  xlab("Housing Price") +
  theme(plot.title = element_text(hjust = 0.5))
```

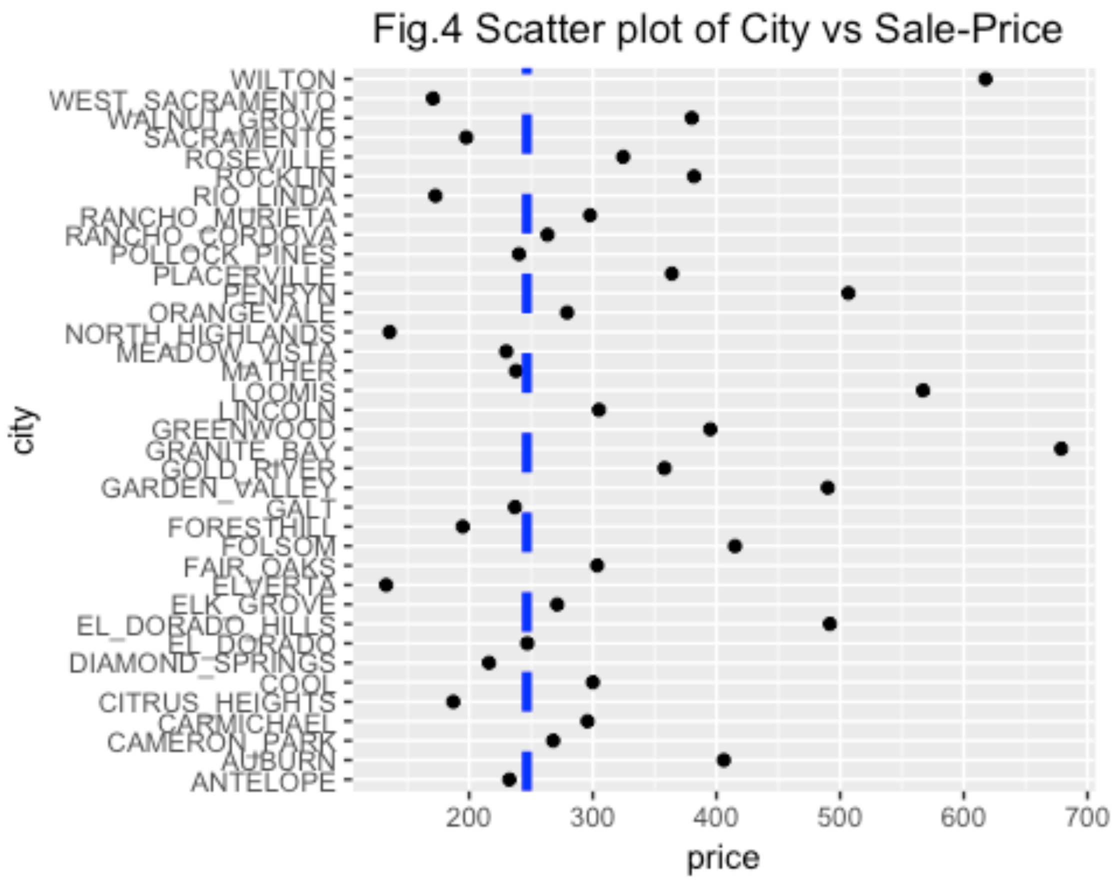



```r
#=======================================================================#
# The graph shows the data is normally distributed but with right
# skewness due to the price range and count. This can be fixed by
# transforming the price in logarithmic format and can be used throughout
# this exercise
#=======================================================================#
Sacramento$log_price <-log(Sacramento$price)

ggplot(Sacramento, aes(x = log_price, fill = ..count..)) +
  geom_histogram(binwidth = 0.05) +
  ggtitle("Fig. 2 Histogram of Log-SalePrice") +
  ylab("Count of houses") +
  xlab("Housing Price") +
  theme(plot.title = element_text(hjust = 0.5))
```

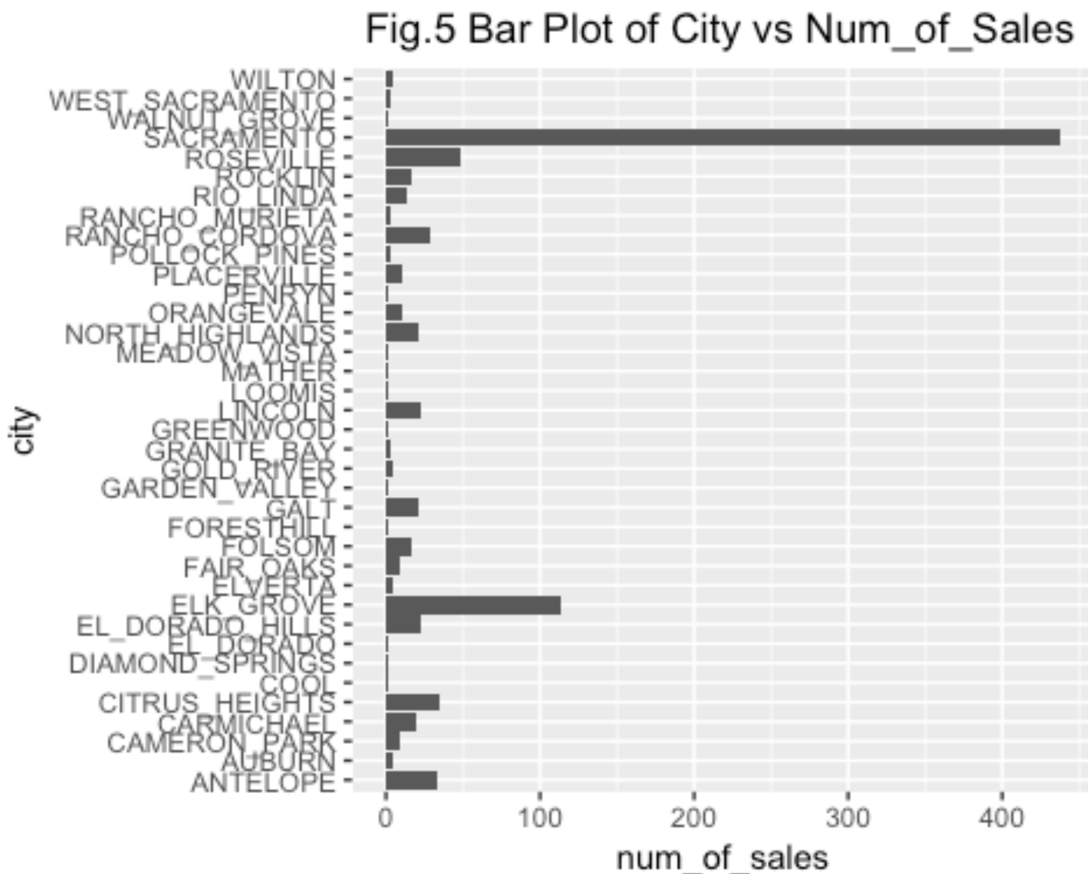



```
#======================================================================#
# After the log transforms the data looks normally distributed
#======================================================================#

#======================================================================#
#  Let's plot the price vs Sqft relationship to see if the sqft can be
#  used as a predictor. Let's use the scatter plot to determine this
#  note that the sale price is transformed to 100s and the mean for the
#  price for sqfts are plotted.
#======================================================================#

Sacramento %>%
  group_by(sqft) %>%
  summarize(price=mean(price/1000)) %>%
  select(sqft,price) %>%
  ggplot(aes(x=sqft,y=price)) +
  geom_point(shape=1) +
  geom_smooth(method=lm , color="blue", se=FALSE)+
  ggtitle("Fig.3 Scatter plot of Sqft vs Sale-Price") +
  theme(plot.title = element_text(hjust = 0.4))
```


```
#=======================================================================#
#  As the scatter plot (Fig.3) shows there is a linear relationship and the
#  sqft can be used as a predictor which we can confirm using heat map
#  for correlations
#=======================================================================#

#=======================================================================#
#  Let's plot the city vs Sale Price relationship to see if the city can
#  be used as a predictor. Let's use the scatter plot to determine this
#  note that the sale price is transformed to 100s and the mean for the
#  price for cities are plotted.
#=======================================================================#

Sacramento %>%
  group_by(city) %>%
  summarize(price=mean(price/1000)) %>%
  select(city,price) %>%
  ggplot( aes(x=price, y=city)) +
  geom_point()+geom_vline(xintercept = avg_sale_price/1000,
                          linetype="dashed", color = "blue", size=1.5) +
  ggtitle("Fig.4 Scatter plot of City vs Sale-Price") +
  theme(plot.title = element_text(hjust = 0.4))
```

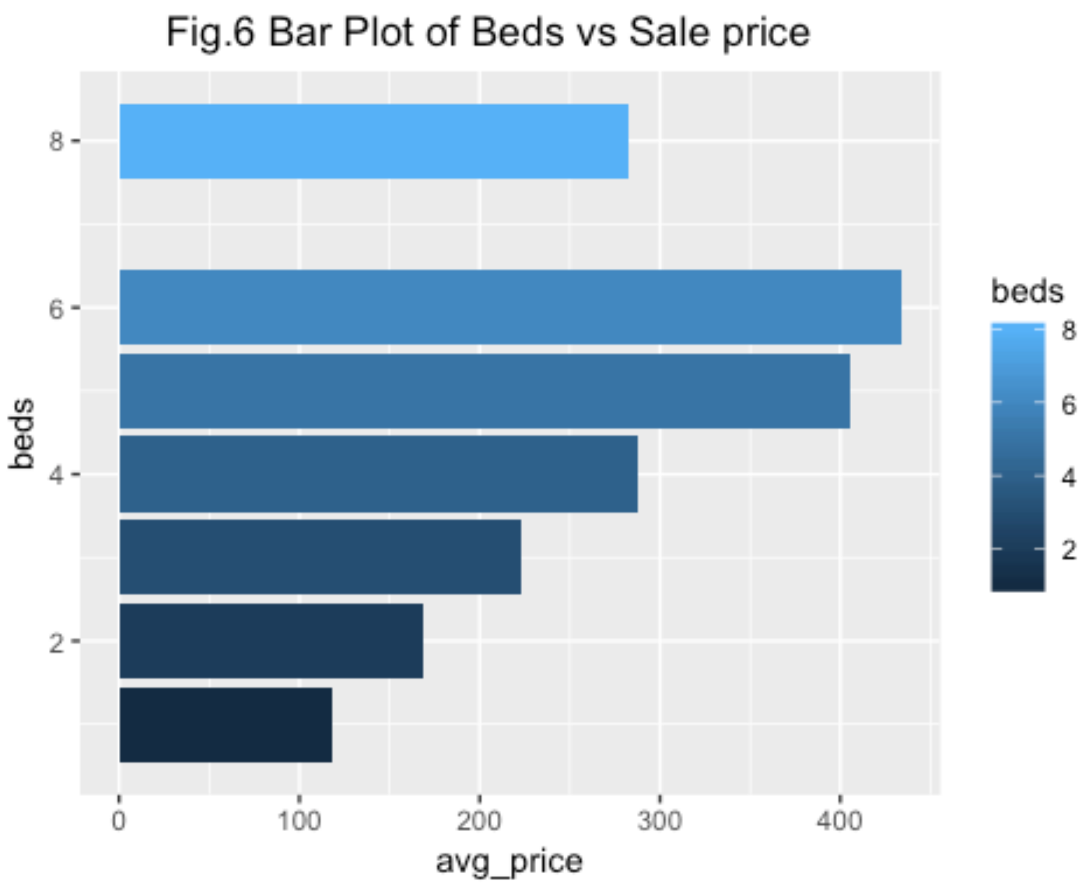



```
#======================================================================#
#  Note that the Fig.4 illustrates the data points are not on a linear
#  fashion and some cities has less sales which can be proved below
#======================================================================#

Sacramento %>%
  group_by(city) %>%
  summarize(num_of_sales=n()) %>%
  select(city,num_of_sales) %>%
  ggplot(aes(x=city,y=num_of_sales)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.5 Bar Plot of City vs Num_of_Sales") +
  theme(plot.title = element_text(hjust = 0.4))
```

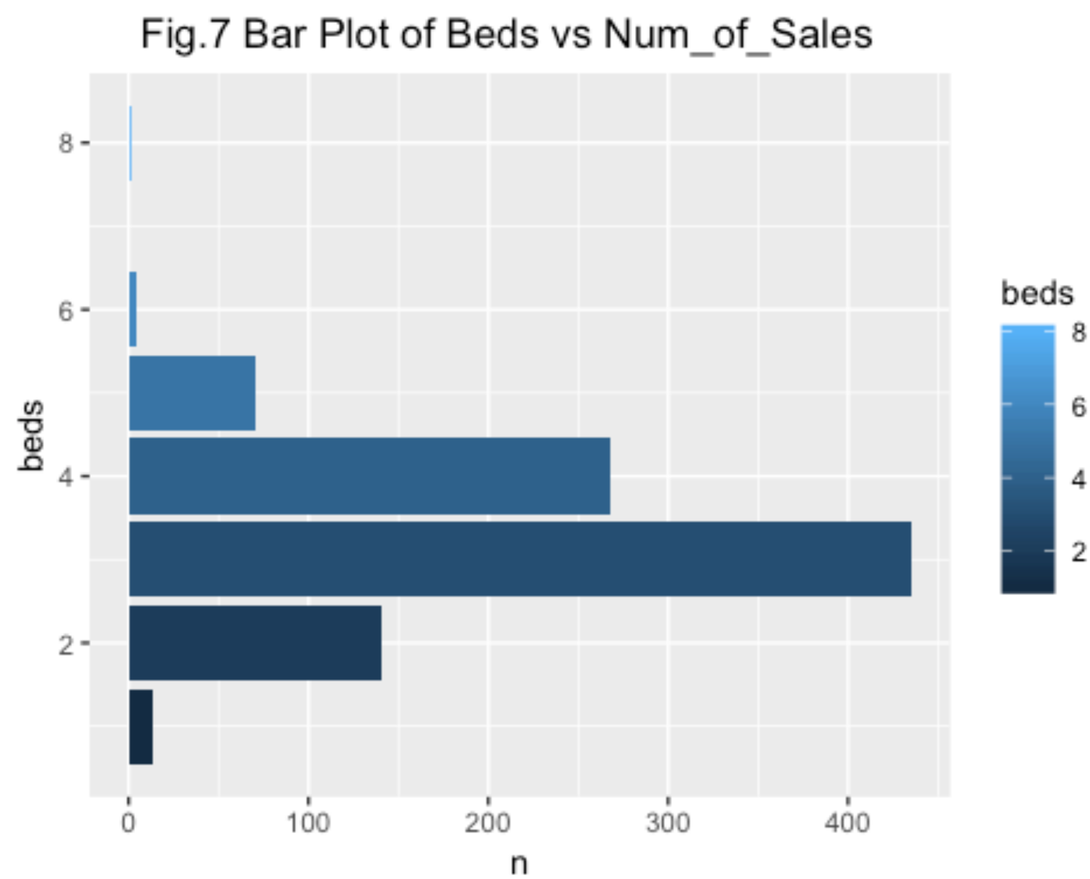



```
Sacramento %>% group_by(city) %>%  summarize(num_of_sale = n())

## # A tibble: 37 x 2
##    city            num_of_sale
##    <fct>                 <int>
##  1 ANTELOPE                 33
##  2 AUBURN                    5
##  3 CAMERON_PARK              9
##  4 CARMICHAEL               20
##  5 CITRUS_HEIGHTS           35
##  6 COOL                      1
##  7 DIAMOND_SPRINGS           1
##  8 EL_DORADO                 2
##  9 EL_DORADO_HILLS          23
## 10 ELK_GROVE               114
## # … with 27 more rows

#========================================================================#
#  This confirms that the city cannot be used as a predictor
#========================================================================#


#========================================================================#
#  Now, Let's plot the Beds vs Sale price relationship to see if the Beds
```

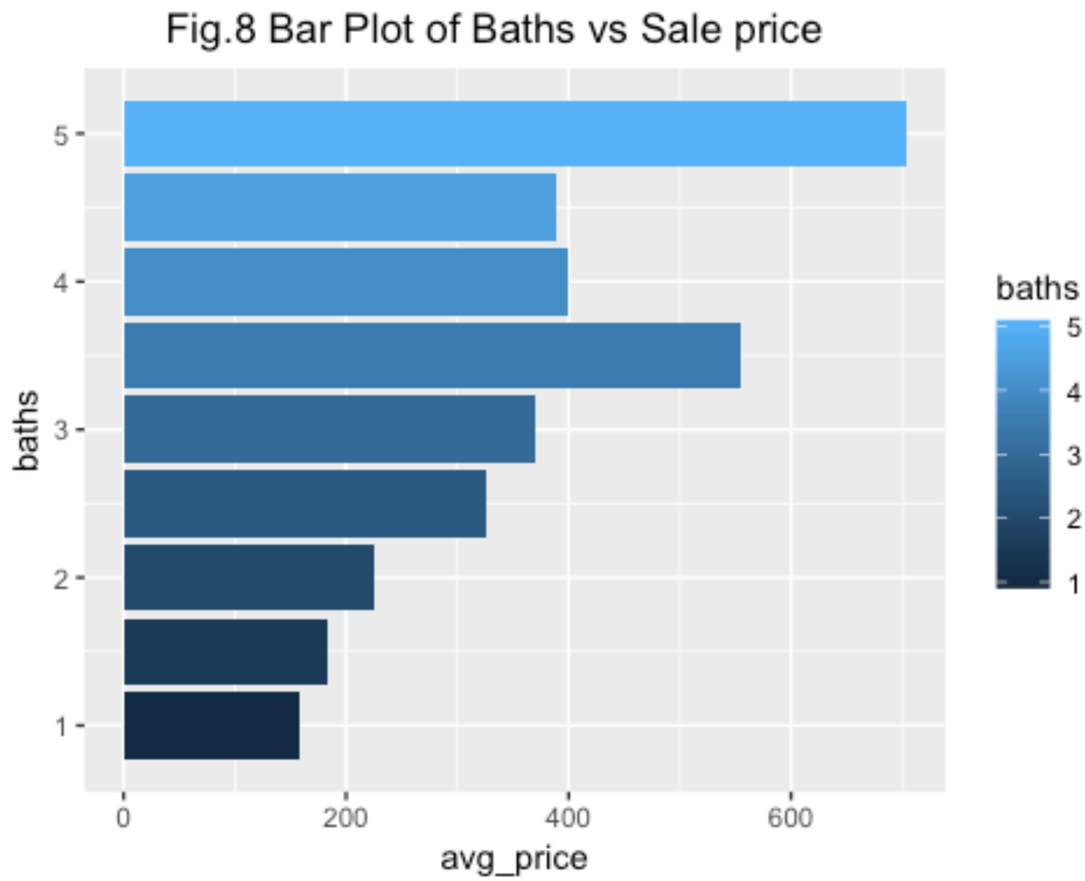
```r
#  can be used as a predictor. Let's use the scatter plot to determine
#  this note that the sale price is transformed to 100s and the mean for
#  the price for beds are plotted.
#========================================================================#

Sacramento %>%
  group_by(beds) %>%
  summarize(avg_price=mean(price/1000)) %>%
  select(beds,avg_price) %>%
  ggplot(aes(x=beds,y=avg_price,fill=beds)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.6 Bar Plot of Beds vs Sale price") +
  theme(plot.title = element_text(hjust = 0.4))
```




```r
Sacramento%>%group_by(beds)%>%summarize(n=n()) %>%
  select(beds,n) %>%
  ggplot(aes(x=beds,y=n,fill=beds)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.7 Bar Plot of Beds vs Num_of_Sales") +
  theme(plot.title = element_text(hjust = 0.4))
```

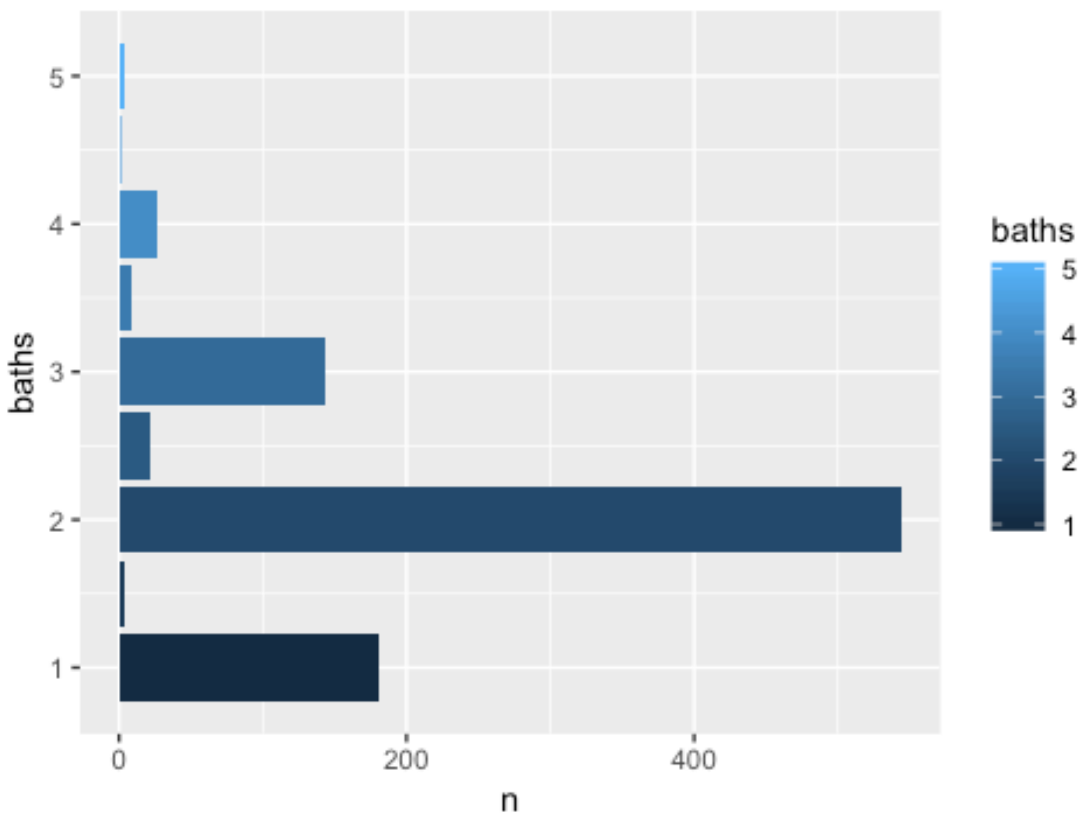



```
#==========================================================================#
#  The Bar plot Fig.6 clearly shows that when Beds increases the price
#  increases and Fig.7 shows there is an outlier with bedrooms 8 beds
#  which has only one sample which can be removed from the data
#==========================================================================#

#==========================================================================#
#  Now, Let's plot the Baths vs Sale price relationship to see if the
#  Baths can be used as a predictor. Let's use the scatter plot to
#  determine this note that the sale price is transformed to 100s and
#  the mean for the price for baths are plotted.
#==========================================================================#

Sacramento %>%
  group_by(baths) %>%
  summarize(avg_price=mean(price/1000)) %>%
  select(baths,avg_price) %>%
  ggplot(aes(x=baths,y=avg_price,fill=baths)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.8 Bar Plot of Baths vs Sale price") +
  theme(plot.title = element_text(hjust = 0.4))
```



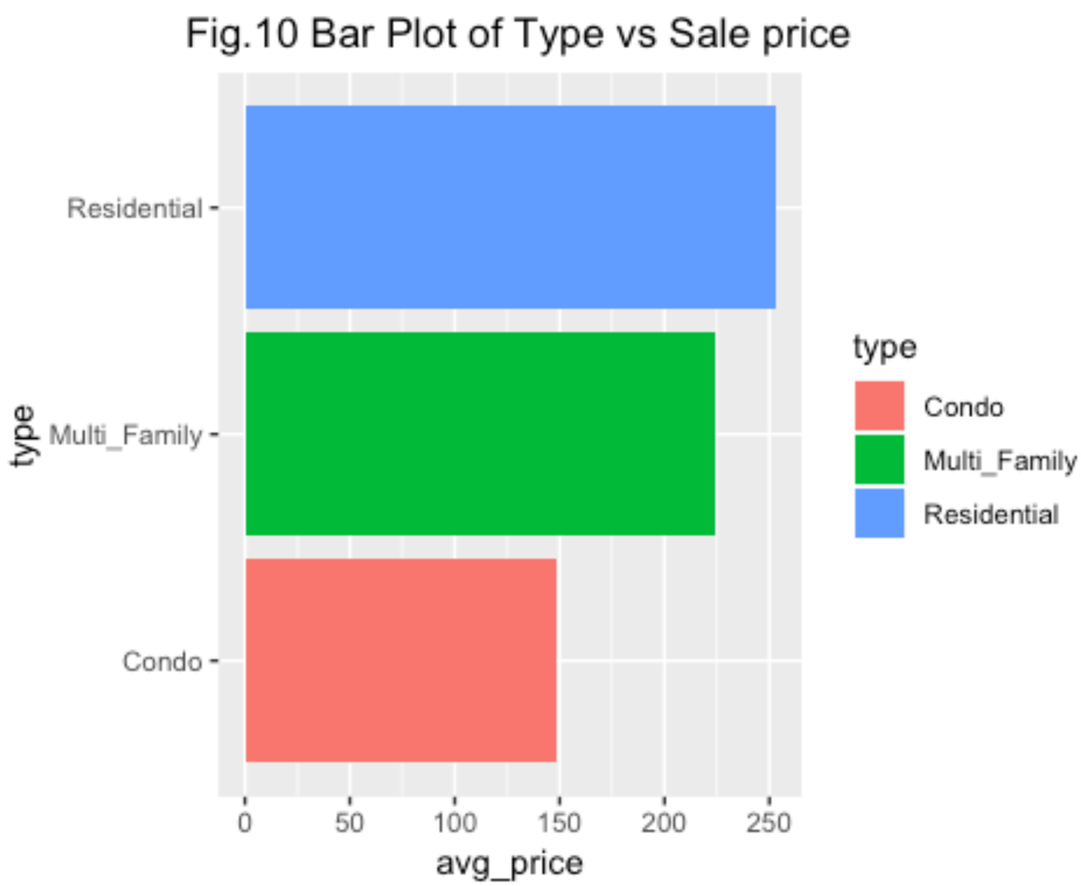

```
Sacramento%>%group_by(baths)%>%summarize(n=n()) %>%
  select(baths,n) %>%
  ggplot(aes(x=baths,y=n,fill=baths)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.9 Bar Plot of Baths vs Num_of_Sales") +
  theme(plot.title = element_text(hjust = 0.4))
```

Fig.9 Bar Plot of Baths vs Num_of_Sales




```
Sacramento%>%group_by(baths)%>%summarize(n=n()) %>%
  select(baths,n)

## # A tibble: 9 x 2
##   baths     n
##   <dbl> <int>
## 1   1     180
## 2   1.5     4
## 3   2     544
## 4   2.5    22
## 5   3     143
## 6   3.5     8
## 7   4      27
## 8   4.5     1
## 9   5       3

#=========================================================================#
#  The Bar plot Fig.8 clearly shows that when baths increases the price
#  increases and Fig.9 shows there is an outlier with 4.5 baths which
#  has only one sample which can be removed from the data
#=========================================================================#
```

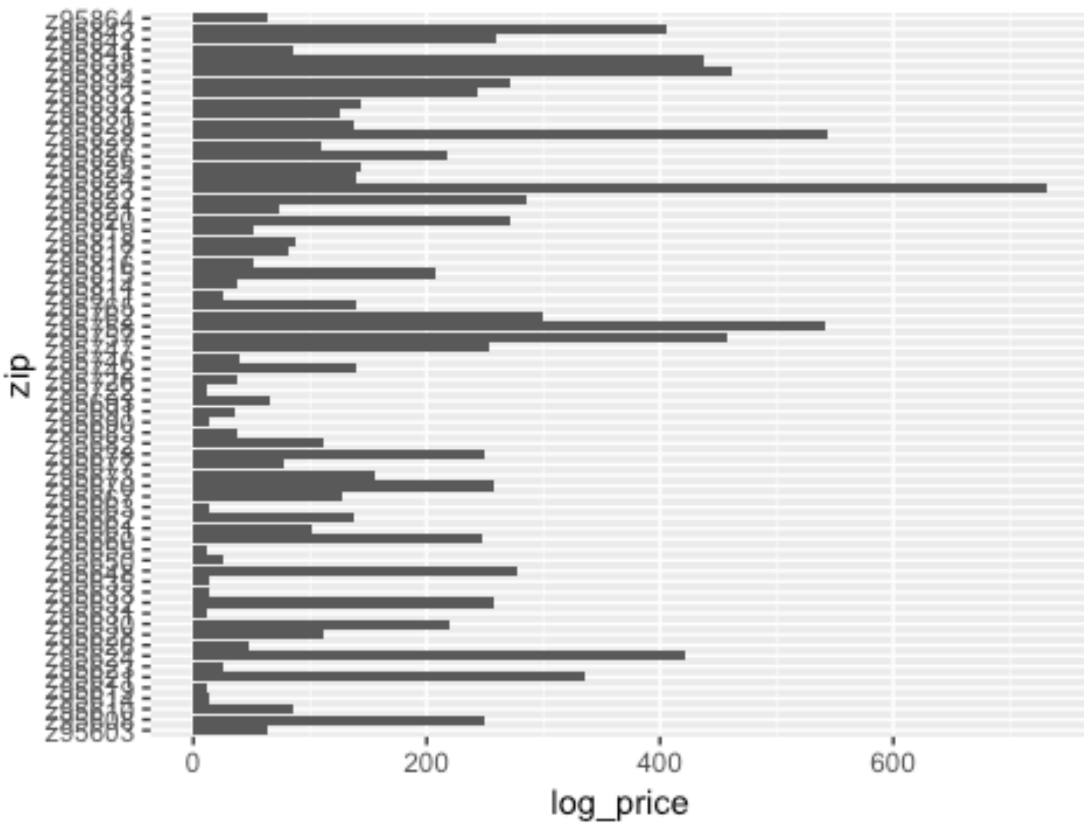
```
#=========================================================================#
#  Now, Let's plot the Tye vs Sale price relationship to see if the
#  Baths can be used as a predictor. Let's use the scatter plot to
#  determine this note that the sale price is transformed to 100s and
#  the mean for the price for types are plotted.
#=========================================================================#

Sacramento %>%
  group_by(type) %>%
  summarize(avg_price=mean(price/1000)) %>%
  select(type,avg_price) %>%
  ggplot(aes(x=type,y=avg_price,fill=type)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.10 Bar Plot of Type vs Sale price") +
  theme(plot.title = element_text(hjust = 0.4))
```



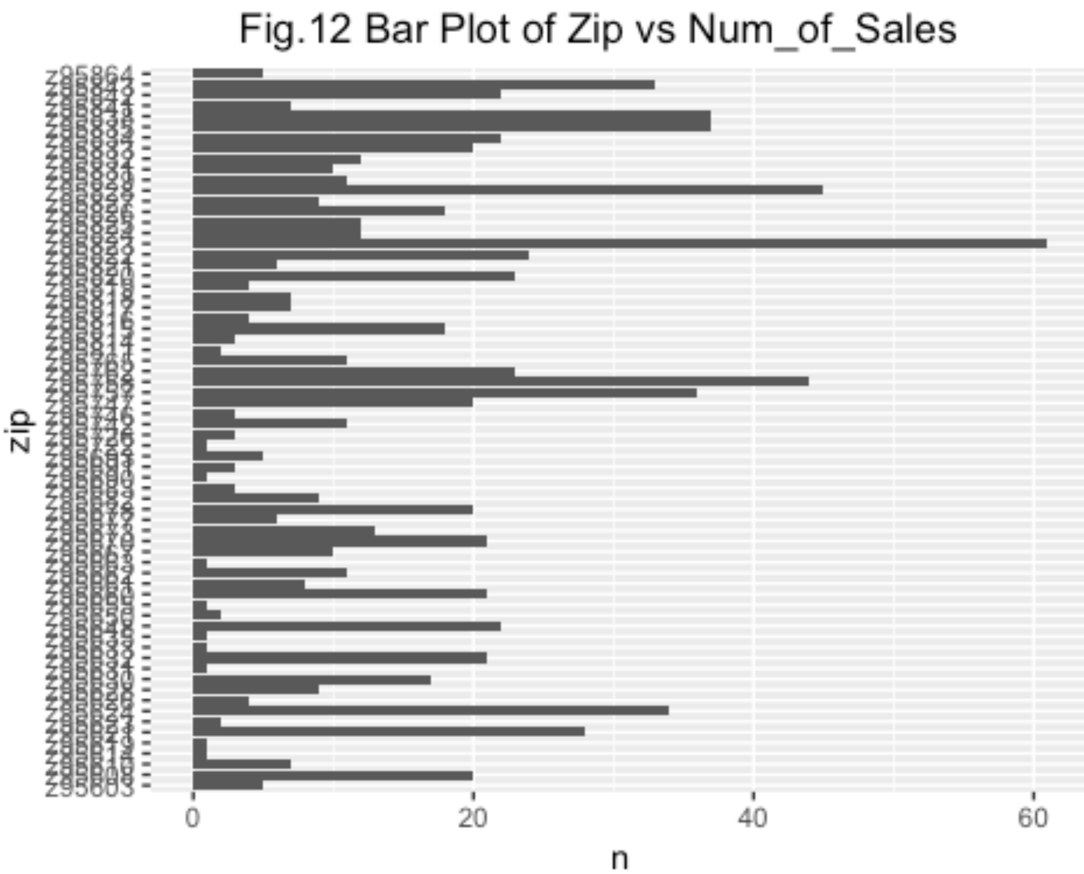

```
#=========================================================================#
#  The Bar plot Fig.10 shows that the condos and multi family units
#  are cheaper compared to Single family units and has a correlation
#  between sale price and hence can be picked up as a predictor
#=========================================================================#
```

```
#===========================================================================#
# Let's plot the zip field to see if it can be used as a
# Predictor
#===========================================================================#

Sacramento %>%
  group_by(zip) %>%
  select(zip,log_price) %>%
  ggplot(aes(x=zip,y=log_price)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.11 Bar Plot of Zip vs Sale price") +
  theme(plot.title = element_text(hjust = 0.4))
```

Fig.11 Bar Plot of Zip vs Sale price

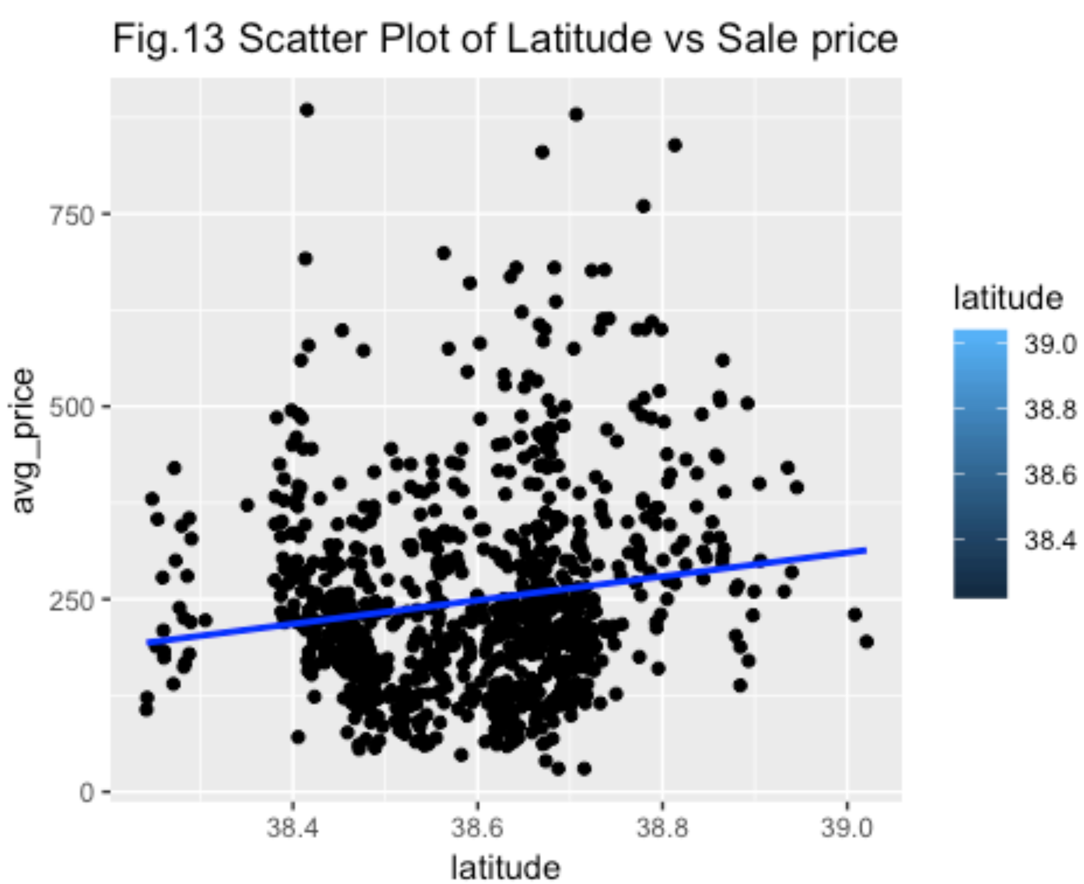



```
Sacramento%>%group_by(zip)%>%summarize(n=n()) %>%arrange(desc(n)) %>%
  select(zip,n) %>%
  ggplot(aes(x=zip,y=n)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Fig.12 Bar Plot of Zip vs Num_of_Sales") +
  theme(plot.title = element_text(hjust = 0.4))
```



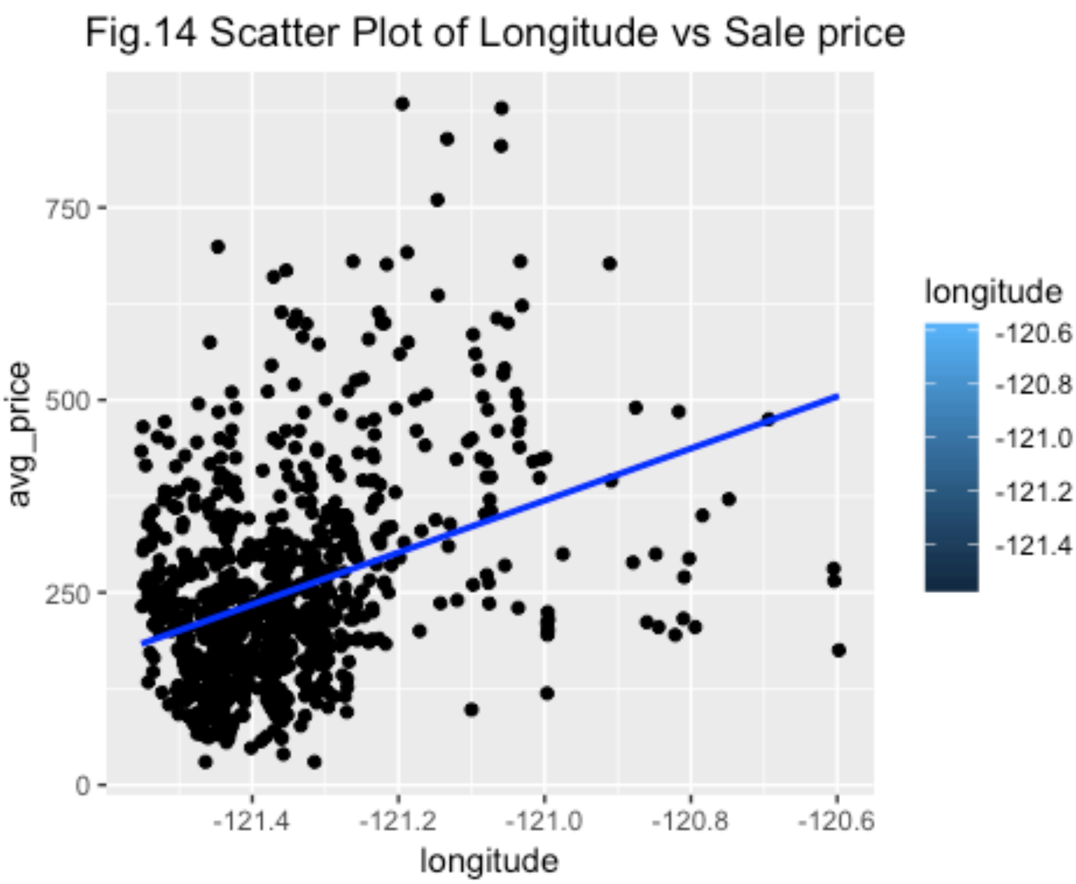

```r
Sacramento %>% select(zip,beds) %>% group_by(zip,beds) %>%
summarize(n_bed=n())

## # A tibble: 196 x 3
## # Groups:   zip [68]
##    zip     beds n_bed
##    <fct>  <int> <int>
##  1 z95603     2     2
##  2 z95603     3     1
##  3 z95603     4     2
##  4 z95608     2     4
##  5 z95608     3    11
##  6 z95608     4     5
##  7 z95610     3     5
##  8 z95610     4     1
##  9 z95610     5     1
## 10 z95614     3     1
## # … with 186 more rows

#=====================================================================#
# Even though the Fig 11 shows some relationship between zip and sale
# price, there is no concrete evidence to support the reason for the
# dependency. For instance, the number of sales based on beds were
```

```
# summarized in an assumption that the bedroom counts may play a role
# in the raise in saleprice on the particular zipcode but the summary
# defies that, hence this can be eliminated to be a predictor
#=========================================================================#


#=========================================================================#
# Let's plot the latitude and longitude field to see if it can be used
# as Predictors
#=========================================================================#

Sacramento %>%
  group_by(latitude) %>%
  summarize(avg_price=mean(price/1000)) %>%
  select(latitude,avg_price) %>%
  ggplot(aes(x=latitude,y=avg_price,fill=latitude)) +
  geom_point() +
  geom_smooth(method=lm , color="blue", se=FALSE)+
  ggtitle("Fig.13 Scatter Plot of Latitude vs Sale price") +
  theme(plot.title = element_text(hjust = 0.4))
```



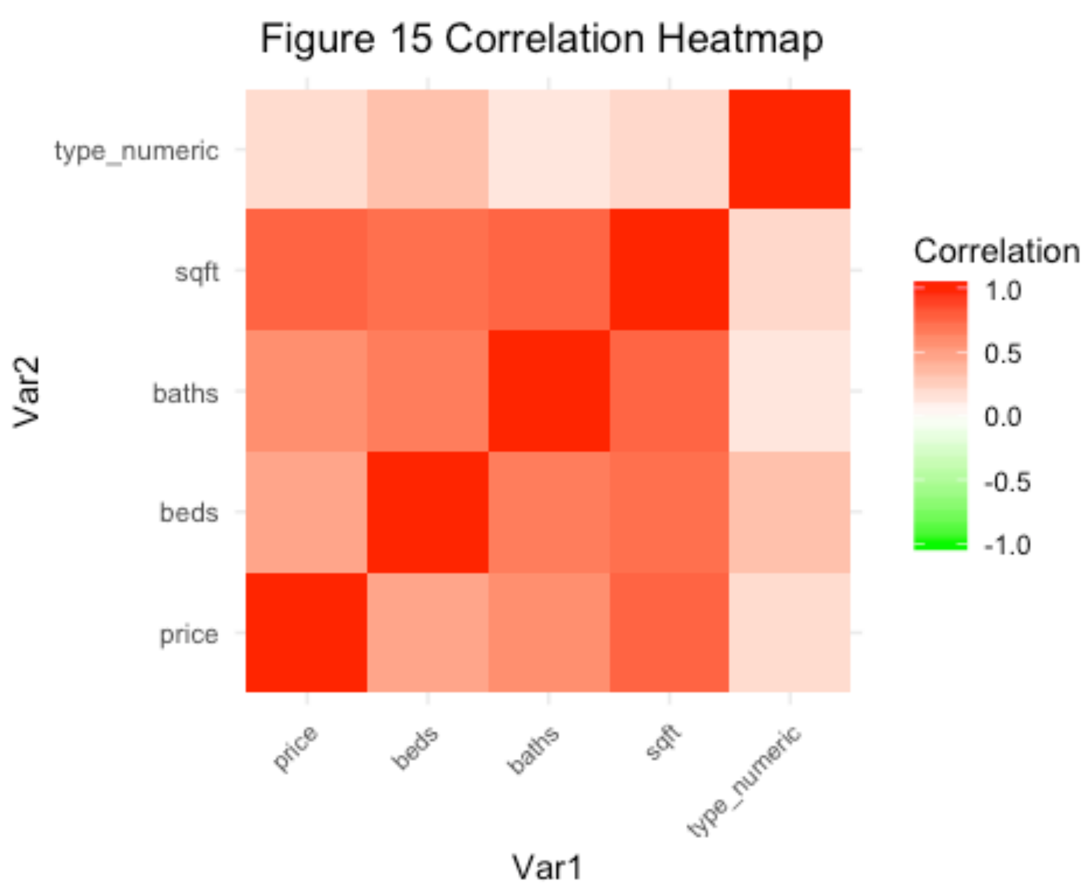

```
Sacramento %>%
  group_by(longitude) %>%
```

```
  summarize(avg_price=mean(price/1000)) %>%
  select(longitude,avg_price) %>%
  ggplot(aes(x=longitude,y=avg_price,fill=longitude)) +
  geom_point() +
  geom_smooth(method=lm , color="blue", se=FALSE)+
  ggtitle("Fig.14 Scatter Plot of Longitude vs Sale price") +
  theme(plot.title = element_text(hjust = 0.4))
```




```
#=====================================================================#
# The scatter plots of latitude and longitudes shows there is no linear
# relationship with prices and hence can be eliminated as predictors
#=====================================================================#

#=====================================================================#
# Let's plot the heat map to confirm the predictors we have chosen
# are good for out modeling
# Note that the red tiles in the diagonal shows the strong corelation
# between price and it's predictors
# Before we plot the Heat Map, lets convert the type to a numeric from
# factor
#=====================================================================#

Sacramento$type_numeric <-
```

```
as.numeric(factor(Sacramento$type,levels=c("Condo","Multi_Family","Residentia
l"),
                    labels=c(1,2,3), ordered = TRUE))

heat_var <- Sacramento %>% select(price,beds,baths,sqft,type_numeric)

qplot(x=Var1, y=Var2, data=melt(cor(heat_var, use="p")), fill=value,
geom="tile") +
  scale_fill_gradient2(low = "green", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 8, hjust =
1))+
  coord_fixed()+
  ggtitle("Figure 15 Correlation Heatmap") +
  theme(plot.title = element_text(hjust = 0.4))
```




```
#==================================================================#
# Okay, we vizualized the data and picked the predictors.
# Now lets remove the outliers as explained and split the data into
# train and test sets
```

```r
#=========================================================================#

#=========================================================================#
# Remove the house with beds = 8 and bath = 4.5
#=========================================================================#

Sacramento_data <- Sacramento %>% filter((beds != 8)) %>% filter((baths !=
4.5))

#=========================================================================#
# Confirm the outliers were removed
#=========================================================================#
unique(Sacramento_data$baths)

## [1] 1.0 2.0 3.0 4.0 5.0 1.5 2.5 3.5

unique(Sacramento_data$beds)

## [1] 2 3 1 4 5 6

#=========================================================================#
# Select the required data for modeling
#=========================================================================#

Sacramento_model_data <- Sacramento_data %>%
  select(log_price, beds, baths, sqft, type_numeric)

#=========================================================================#
# Partition the data into test and train datasets
#=========================================================================#

y_all <- Sacramento_model_data$log_price

index<-createDataPartition(y_all,times=1,p=0.8,list=FALSE)

train_data<- Sacramento_model_data[index,]
test_data<- Sacramento_model_data[-index,]

#=========================================================================#
#  Display the details of Train and Test datasets
#=========================================================================#

dim(train_data)

## [1] 745   5

dim(test_data)

## [1] 185   5

summary(train_data)
```

```
##    log_price          beds           baths           sqft     
##  Min.   :10.31   Min.   :1.000   Min.   :1.000   Min.   : 484  
##  1st Qu.:11.96   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1172  
##  Median :12.30   Median :3.000   Median :2.000   Median :1477  
##  Mean   :12.28   Mean   :3.289   Mean   :2.058   Mean   :1682  
##  3rd Qu.:12.63   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:1940  
##  Max.   :13.69   Max.   :6.000   Max.   :5.000   Max.   :4400  
##   type_numeric  
##  Min.   :1.000  
##  1st Qu.:3.000  
##  Median :3.000  
##  Mean   :2.867  
##  3rd Qu.:3.000  
##  Max.   :3.000
```

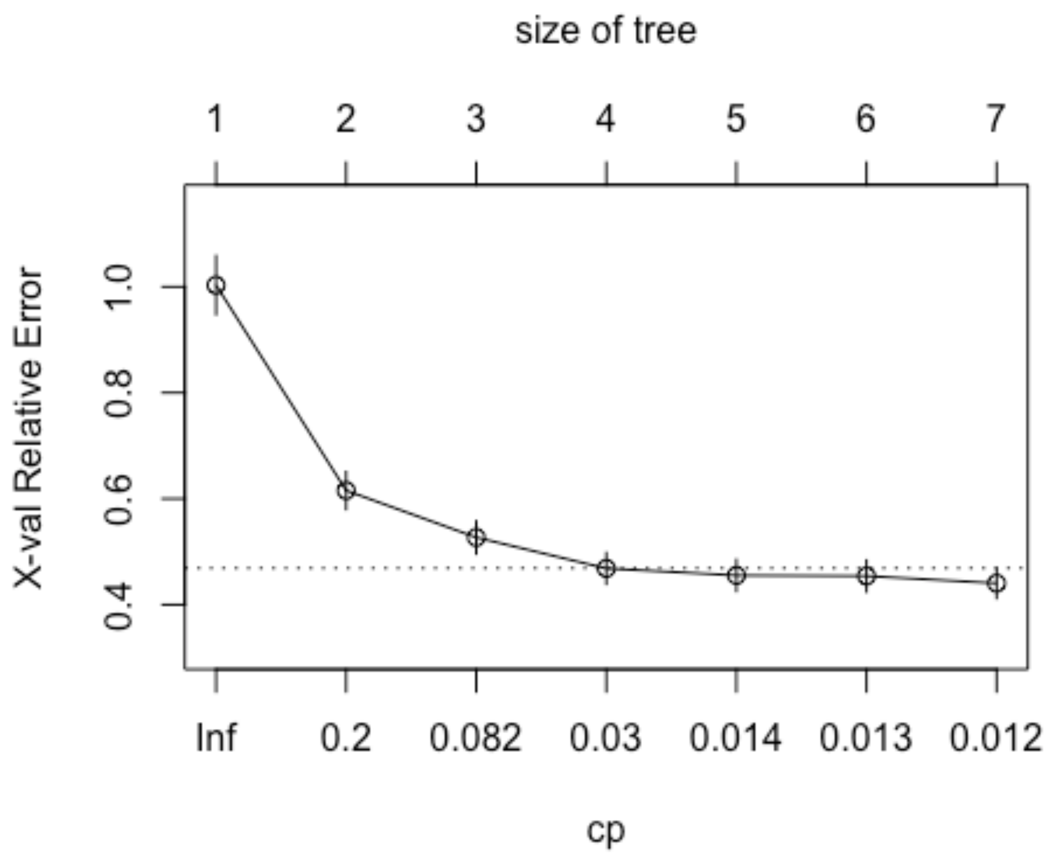
```
summary(test_data)
```

```
##    log_price          beds           baths           sqft     
##  Min.   :10.99   Min.   :2.000   Min.   :1.000   Min.   : 539  
##  1st Qu.:11.95   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1140  
##  Median :12.30   Median :3.000   Median :2.000   Median :1440  
##  Mean   :12.30   Mean   :3.189   Mean   :2.011   Mean   :1657  
##  3rd Qu.:12.61   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:1980  
##  Max.   :13.63   Max.   :6.000   Max.   :5.000   Max.   :4878  
##   type_numeric  
##  Min.   :1.000  
##  1st Qu.:3.000  
##  Median :3.000  
##  Mean   :2.897  
##  3rd Qu.:3.000  
##  Max.   :3.000
```

```
#======================================================================#
# Let us use the linear model to predict the RMSE to make sure the 
# squared errors and mimimal, so that we can use the data for other
# models to predict the fit
#======================================================================#

linreg <- lm(log_price~.,data = train_data)
summary(linreg)
```

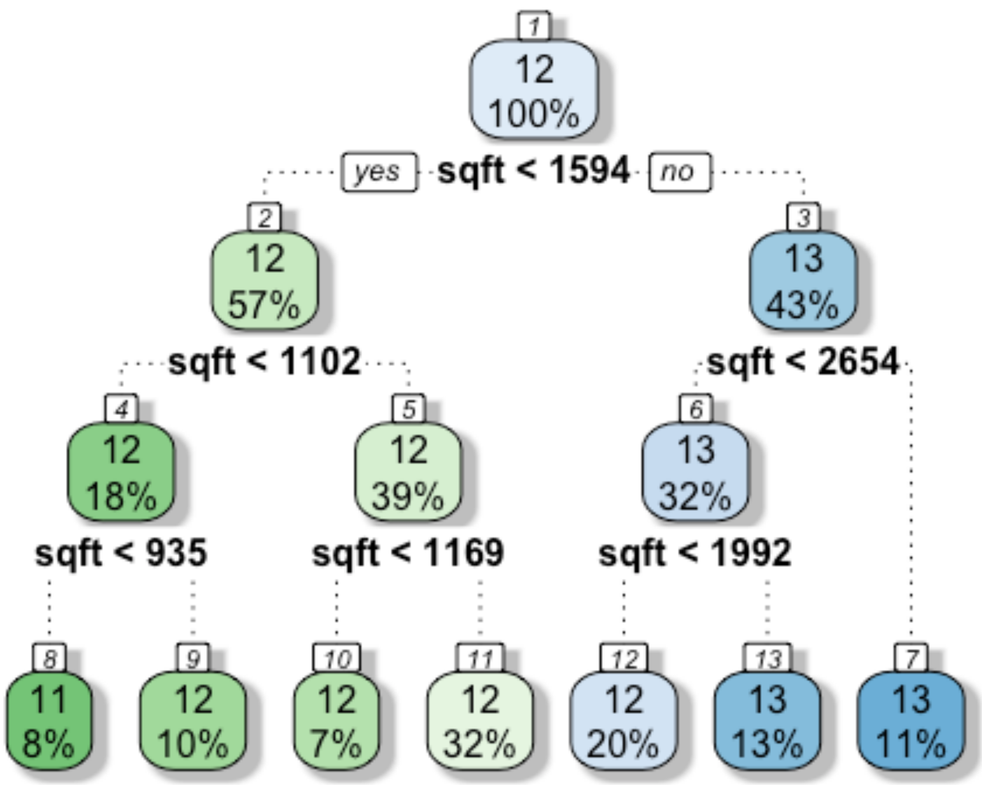
```
## 
## Call:
## lm(formula = log_price ~ ., data = train_data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.75587 -0.22238  0.00903  0.23030  1.22231 
## 
## Coefficients:
##                 Estimate  Std. Error t value             Pr(>|t|)
```

```
## (Intercept)  11.10717828  0.08399718 132.233 < 0.0000000000000002 ***
## beds         -0.05055495  0.02273543  -2.224            0.026475 *
## baths         0.09921736  0.02933540   3.382            0.000757 ***
## sqft          0.00049507  0.00003063  16.165 < 0.0000000000000002 ***
## type_numeric  0.10490029  0.02913622   3.600            0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3579 on 740 degrees of freedom
## Multiple R-squared:  0.5509, Adjusted R-squared:  0.5485
## F-statistic: 226.9 on 4 and 740 DF,  p-value: < 0.00000000000000022

pred_lm_lp <- predict(linreg,test_data,type="response")

#=======================================================================#
#  Combine the results into a dataframe
#=======================================================================#
residuals <- test_data$log_price - pred_lm_lp
linreg_pred <- data.frame("Method"="LM",
                          "Predicted" = pred_lm_lp,
                          "Actual" = test_data$log_price,
                          "Residual" = residuals)

acc_lm<-accuracy(pred_lm_lp, test_data$log_price)

accuracy_details <- data.frame("Method" = "LM", RMSE=acc_lm[2])

#=======================================================================#
# The RMSE is less (0.341) hence we are proceeding to the next model
#=======================================================================#

#=======================================================================#
# Classification and regression trees - Recursive partitioning
#=======================================================================#

#=======================================================================#
# Lets pick the confusion parameter for rpart function for cross
# validation
#=======================================================================#

rpart_cp = rpart.control(cp=0.01)

rpart_tree <- rpart(log_price ~ .,data=train_data,control = rpart_cp)

plotcp(rpart_tree)
```



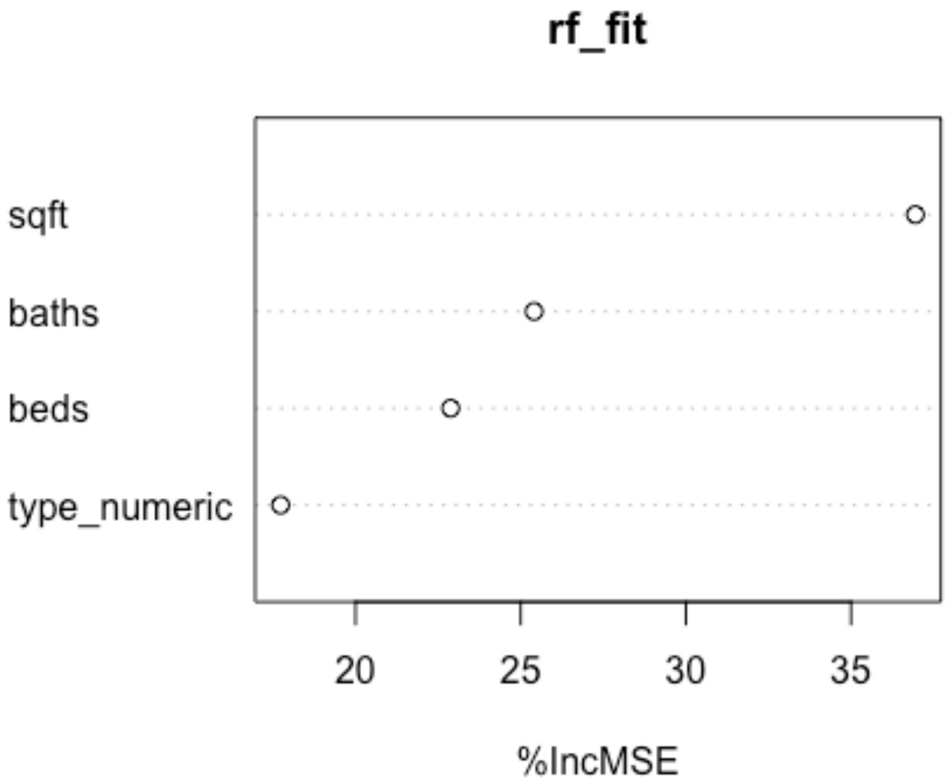

```
#=====================================================================#
# The plot shows the relative error is getting minimized when cp=0.011
#=====================================================================#
printcp(rpart_tree)

##
## Regression tree:
## rpart(formula = log_price ~ ., data = train_data, control = rpart_cp)
##
## Variables actually used in tree construction:
## [1] sqft
##
## Root node error: 211.02/745 = 0.28325
##
## n= 745
##
##          CP nsplit rel error  xerror     xstd
## 1 0.394402      0   1.00000 1.00304 0.055433
## 2 0.099840      1   0.60560 0.61544 0.035568
## 3 0.066656      2   0.50576 0.52689 0.030900
## 4 0.013766      3   0.43910 0.46840 0.029624
## 5 0.013514      4   0.42534 0.45522 0.029884
```

```
## 6 0.013442      5   0.41182 0.45403 0.029849
## 7 0.010000      6   0.39838 0.44044 0.028513

#======================================================================#
# Print the tree split which shows the split based on sqft
# note that the rpart() function ignored other predictors since those
# gave the same split as like sqft
#======================================================================#
rpart.plot(rpart_tree,
           box.palette="GnBu",
           branch.lty=3, shadow.col="gray", nn=TRUE)
```



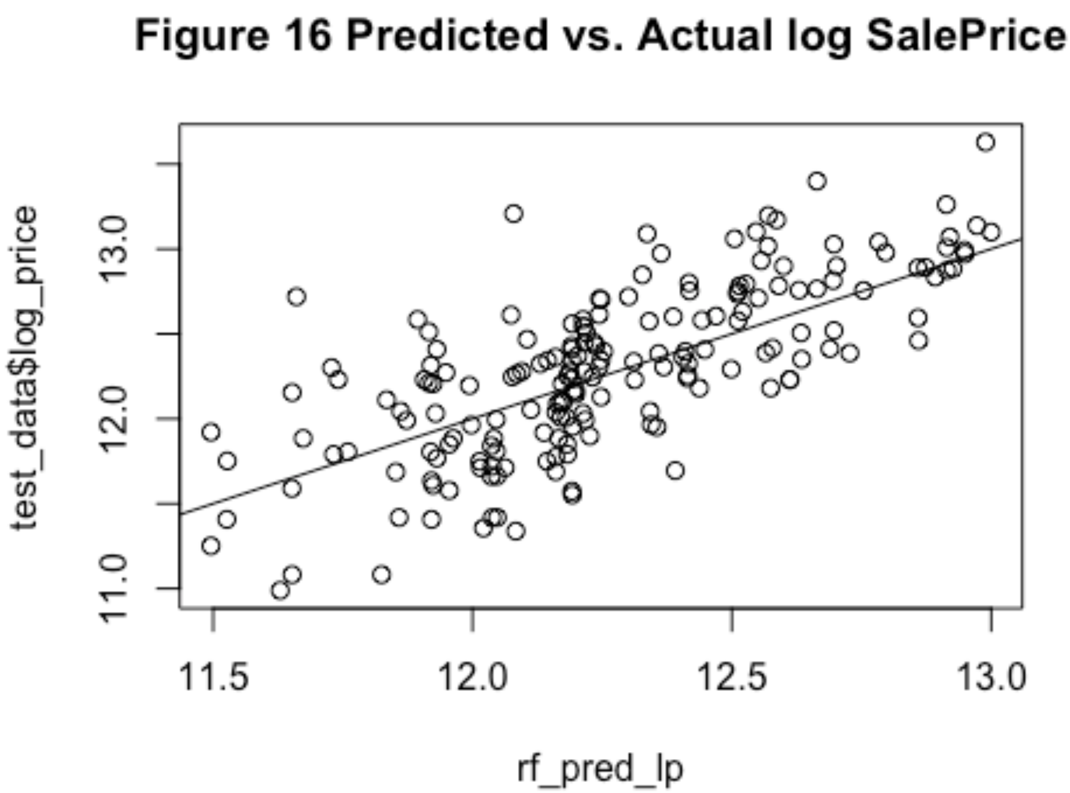

```
rpart_pred_lp <- predict(rpart_tree, newdata=test_data )

#======================================================================#
#  Combine the results into a dataframe
#======================================================================#

residuals <- test_data$log_price - rpart_pred_lp
rpart_pred <- data.frame("Method"="Rpart",
                         "Predicted" = rpart_pred_lp,
                         "Actual" = test_data$log_price,
                        "Residual" = residuals)
```

```
combined_predictions <- bind_rows(linreg_pred,rpart_pred)

acc_rpart<-accuracy(rpart_pred_lp, test_data$log_price)

accuracy_details <- bind_rows(accuracy_details,
                    data.frame("Method" = "RPART", RMSE=acc_rpart[2]))

#========================================================================#
# Note that the RMSE is decreased as 0.339 as compared to LM
#========================================================================#

#========================================================================#
# Random Forest model
#========================================================================#

rf_fit <- randomForest(log_price ~ .,data=train_data,
                       importance =TRUE,ntree=500,
                       nodesize=7, na.action=na.roughfix)

#========================================================================#
# Plot the variable importance graph
#========================================================================#
options(repr.plot.width=9, repr.plot.height=6)
varImpPlot(rf_fit, type=1)
```




```
#==========================================================================#
# Predict for the test data
#==========================================================================#

rf_pred_lp <- predict(rf_fit, newdata=test_data )
accuracy(rf_pred_lp, test_data$log_price)

##                   ME      RMSE       MAE       MPE     MAPE
## Test set 0.03285384 0.342537 0.274664 0.1843465 2.241556

#==========================================================================#
#  Combine the results into a dataframe
#==========================================================================#

residuals <- test_data$log_price - rf_pred_lp
rf_pred <- data.frame("Method"="RandomForest",
                      "Predicted" = rf_pred_lp,
                         "Actual" = test_data$log_price,
                         "Residual" = residuals)

combined_predictions <- bind_rows(combined_predictions,rf_pred)

acc_rf<-accuracy(rf_pred_lp, test_data$log_price)
```

```
accuracy_details <- bind_rows(accuracy_details,
                              data.frame("Method" = "RF", RMSE=acc_rf[2]))

#=====================================================================#
#  Plot the predicted and actual values for Random Forest prediction
#=====================================================================#

plot(rf_pred_lp, test_data$log_price,
     main = "Figure 16 Predicted vs. Actual log SalePrice")
abline(0,1)
```




```
#=====================================================================#
# KNN Algorithm
#=====================================================================#

#=====================================================================#
# create the control for Cross Validation Kfold = 10 with 90%
# of data for training
#=====================================================================#
control_knn<-trainControl(method = "cv", number = 10, p=0.9)

#=====================================================================#
```

```r
# Knn algorithm to pick the K value
#========================================================================#
train_knn <- train(log_price~.,data=train_data,method="knn",
                   tuneGrid = data.frame(k=seq(1,10,by=0.25)),
                   trControl = control_knn)

#========================================================================#
# Get the correct K value from Cross fold validation
#========================================================================#
k_value <- as.numeric(train_knn$bestTune)
print(k_value)
```

```
## [1] 9.75
```

```r
#========================================================================#
# get the fit for prediction
#========================================================================#
fit_knn<-train(log_price~.,data=train_data,method="knn",
               tuneGrid=data.frame(k=k_value))

#========================================================================#
# predict the fit using the test data
#========================================================================#
knn_pred_lp<-predict(fit_knn,newdata = test_data)

#========================================================================#
#  Combine the results into a dataframe
#========================================================================#

residuals <- test_data$log_price - knn_pred_lp
knn_pred <- data.frame("Method"="RandomForest",
                       "Predicted" = knn_pred_lp,
                      "Actual" = test_data$log_price,
                      "Residual" = residuals)

combined_predictions <- bind_rows(combined_predictions,knn_pred)

acc_knn<-accuracy(knn_pred_lp, test_data$log_price)

accuracy_details <- bind_rows(accuracy_details,
                        data.frame("Method" = "KNN", RMSE=acc_knn[2]))


#========================================================================#
# Display the RMSE values for each model used in this script
#========================================================================#

accuracy_details %>% knitr::kable()
```

Method	RMSE
LM	0.3418035
RPART	0.3390634
RF	0.3425370
KNN	0.3483518

```
#=========================================================================#
# Please note that the predictions for each model is combined into
# the dataframe "combined_prediction"
#=========================================================================#
```