```
install.packages(c("dplyr"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
install.packages(c("corrplot"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
install.packages(c("car"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘cowplot’, ‘doBy’, ‘carData’,
‘pbkrtest’, ‘quantreg’, ‘lme4’
```

```
install.packages(c("lmtest"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘zoo’
```

```
install.packages(c("wordcloud"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
install.packages(c("plotly"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘lazyeval’, ‘crosstalk’
```

```
install.packages(c("reshape2"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘plyr’
```

```
install.packages(c("caret"))
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
also installing the dependencies ‘future’, ‘future.apply’, ‘diagram’,
‘lava’, ‘prodlim’, ‘clock’, ‘hardhat’, ‘ipred’, ‘timeDate’, ‘e1071’,
‘foreach’, ‘ModelMetrics’, ‘pROC’, ‘recipes’
```

```
# Load the required libraries
library(ggplot2)
library(dplyr)
library(corrplot)
library(car)
library(lmtest)
library(wordcloud)
library(RColorBrewer)
library(plotly)
library(reshape2)
library(caret)
```

```
Loading required package: lattice
```

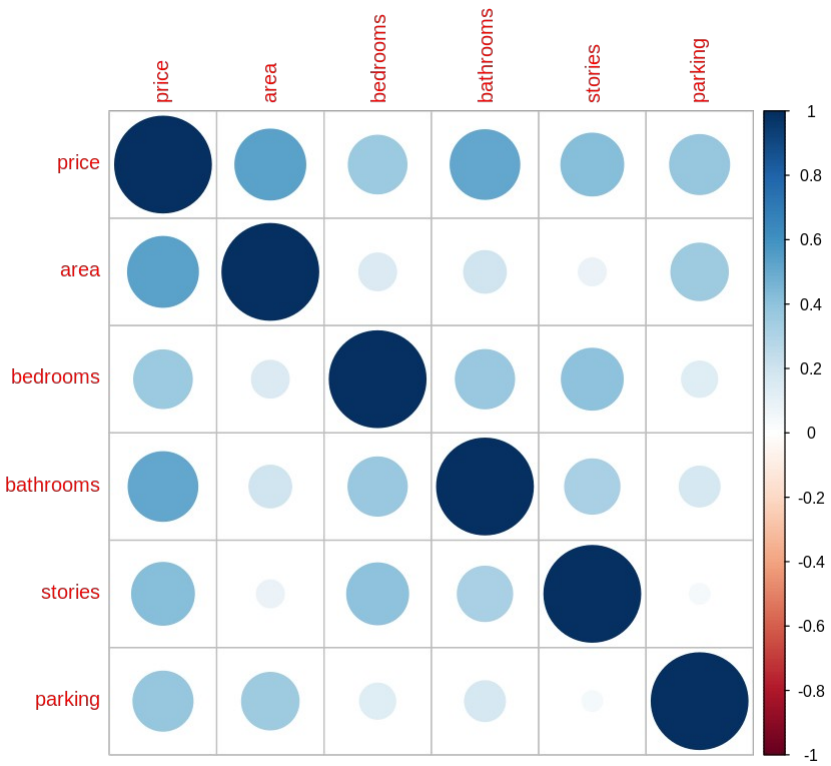
```
# Load the dataset
Housing <- read.csv("Housing.csv")
```

```
# 1. Data Exploration
# Summary statistics
summary(Housing)
```

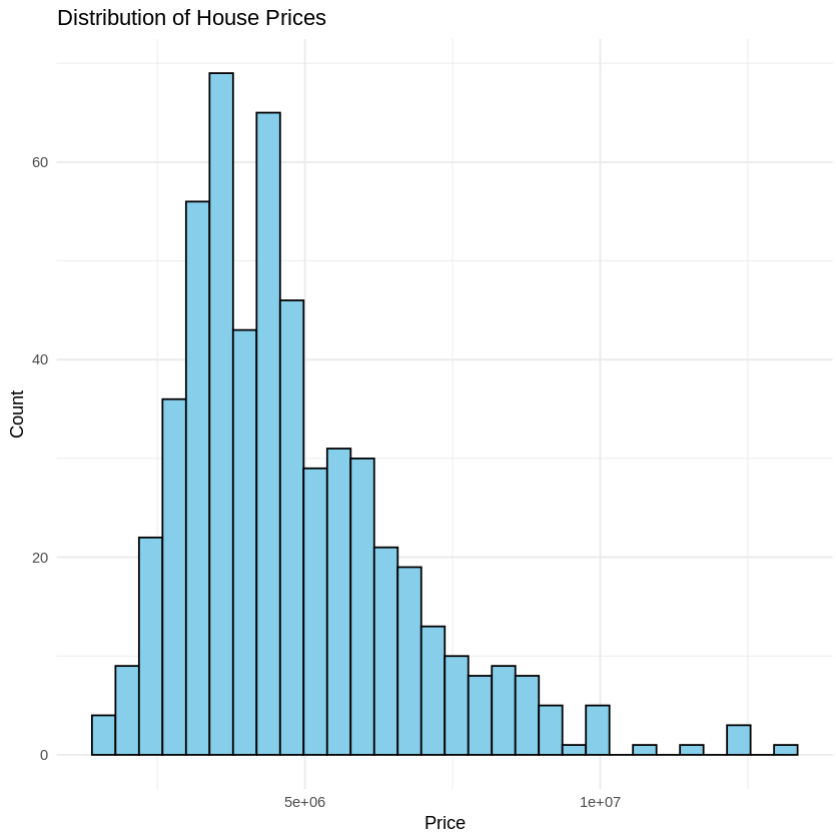
```
     price               area          bedrooms       bathrooms
 Min.   : 1750000   Min.   : 1650   Min.   :1.000   Min.   :1.000
 1st Qu.: 3430000   1st Qu.: 3600   1st Qu.:2.000   1st Qu.:1.000
 Median : 4340000   Median : 4600   Median :3.000   Median :1.000
 Mean   : 4766729   Mean   : 5151   Mean   :2.965   Mean   :1.286
 3rd Qu.: 5740000   3rd Qu.: 6360   3rd Qu.:3.000   3rd Qu.:2.000
 Max.   :13300000   Max.   :16200   Max.   :6.000   Max.   :4.000
    stories        mainroad          guestroom           basement

 Min.   :1.000   Length:545         Length:545         Length:545

 1st Qu.:1.000   Class :character   Class :character
Class :character
 Median :2.000   Mode  :character   Mode  :character
Mode  :character
 Mean   :1.806
```

```
 3rd Qu.:2.000

 Max.   :4.000

 hotwaterheating    airconditioning       parking          prefarea

 Length:545         Length:545         Min.   :0.0000   Length:545

 Class :character   Class :character   1st Qu.:0.0000
Class :character
 Mode  :character   Mode  :character   Median :0.0000
Mode  :character
                                       Mean   :0.6936

                                       3rd Qu.:1.0000

                                       Max.   :3.0000

 furnishingstatus
 Length:545
 Class :character
 Mode  :character
```

```r
# Correlation matrix
correlation_matrix <- cor(Housing[sapply(Housing, is.numeric)])
corrplot(correlation_matrix, method = "circle")
```



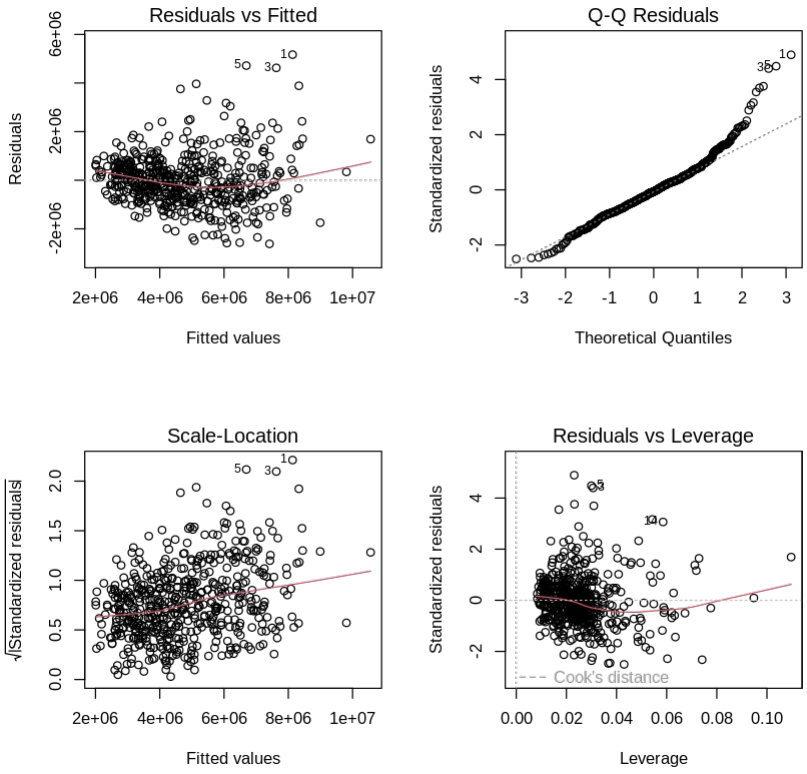

```
# Histogram of price
ggplot(Housing, aes(x = price)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  labs(title = "Distribution of House Prices", x = "Price", y =
"Count") +
  theme_minimal()
```




```
# 2. Linear Regression
model <- lm(price ~ area + bedrooms + bathrooms + stories + mainroad +
guestroom + basement + hotwaterheating + airconditioning + parking +
prefarea + furnishingstatus, data = Housing)
summary(model)


Call:
lm(formula = price ~ area + bedrooms + bathrooms + stories +
    mainroad + guestroom + basement + hotwaterheating +
airconditioning +
    parking + prefarea + furnishingstatus, data = Housing)

Residuals:
     Min       1Q   Median       3Q      Max
-2619718  -657322   -68409   507176  5166695

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)

(Intercept)                       42771.69  264313.31   0.162 0.871508

area                                244.14      24.29  10.052  < 2e-16
***
bedrooms                         114787.56   72598.66   1.581 0.114445
```

```
bathrooms                          987668.11  103361.98   9.555  < 2e-16
***
stories                            450848.00   64168.93   7.026 6.55e-12
***
mainroadyes                        421272.59  142224.13   2.962 0.003193
**
guestroomyes                       300525.86  131710.22   2.282 0.022901
*
basementyes                        350106.90  110284.06   3.175 0.001587
**
hotwaterheatingyes                 855447.15  223152.69   3.833 0.000141
***
airconditioningyes                 864958.31  108354.51   7.983 8.91e-15
***
parking                            277107.10   58525.89   4.735 2.82e-06
***
prefareayes                        651543.80  115682.34   5.632 2.89e-08
***
furnishingstatussemi-furnished     -46344.62  116574.09  -0.398 0.691118

furnishingstatusunfurnished       -411234.39  126210.56  -3.258 0.001192
**
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1068000 on 531 degrees of freedom
Multiple R-squared:  0.6818,    Adjusted R-squared:  0.674
F-statistic: 87.52 on 13 and 531 DF,  p-value: < 2.2e-16
```

```r
# Diagnostic plots
par(mfrow = c(2,2))
plot(model)
par(mfrow = c(1,1))
```




```
# VIF for multicollinearity
vif(model)
```

```
                   GVIF     Df GVIF^(1/(2*Df))
area               1.325250 1  1.151195
bedrooms           1.369477 1  1.170246
bathrooms          1.286621 1  1.134293
stories            1.478055 1  1.215753
mainroad           1.172728 1  1.082926
guestroom          1.212838 1  1.101289
basement           1.323050 1  1.150239
hotwaterheating    1.041506 1  1.020542
airconditioning    1.211840 1  1.100836
parking            1.212837 1  1.101289
prefarea           1.149196 1  1.072006
furnishingstatus   1.109639 2  1.026350
```

```
# Breusch-Pagan test for heteroscedasticity
bptest(model)
```

```
      studentized Breusch-Pagan test

data:  model
BP = 68.416, df = 13, p-value = 1.569e-09
```

```r
# Scatter plot with regression line
ggplot(Housing, aes(x = area, y = price)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Price vs Area with Regression Line", x = "Area", y =
"Price") +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```


```r
# 3. Logistic Regression
median_price <- median(Housing$price)
Housing$price_category <- ifelse(Housing$price > median_price, 1, 0)

logistic_model <- glm(price_category ~ area + bedrooms + bathrooms +
stories + mainroad + guestroom + basement + hotwaterheating +
airconditioning + parking + prefarea + furnishingstatus,
                      data = Housing, family = binomial)
summary(logistic_model)
```

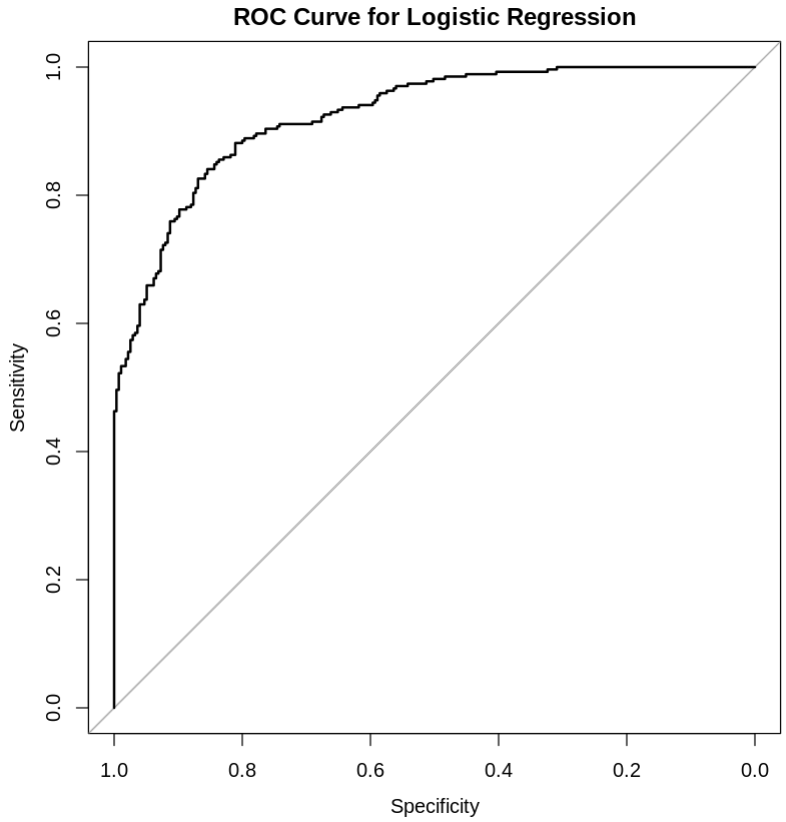
```
Call:
glm(formula = price_category ~ area + bedrooms + bathrooms +
    stories + mainroad + guestroom + basement + hotwaterheating +
```

```
    airconditioning + parking + prefarea + furnishingstatus,
    family = binomial, data = Housing)

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)

(Intercept)                     -9.674e+00  1.062e+00  -9.110  < 2e-16
***
area                             5.076e-04  7.845e-05   6.470 9.78e-11
***
bedrooms                         3.761e-01  2.077e-01   1.811
0.070159 .
bathrooms                        1.071e+00  3.145e-01   3.405 0.000663
***
stories                          9.681e-01  2.198e-01   4.403 1.07e-05
***
mainroadyes                      1.741e+00  4.885e-01   3.564 0.000365
***
guestroomyes                     1.306e+00  4.150e-01   3.148 0.001642
**
basementyes                      8.536e-01  2.980e-01   2.865 0.004175
**
hotwaterheatingyes               4.028e-01  5.788e-01   0.696 0.486458

airconditioningyes               1.453e+00  3.015e-01   4.819 1.44e-06
***
parking                          1.439e-01  1.595e-01   0.902 0.366877

prefareayes                      1.273e+00  3.328e-01   3.826 0.000130
***
furnishingstatussemi-furnished   6.035e-01  3.278e-01   1.841
0.065626 .
furnishingstatusunfurnished     -7.272e-01  3.725e-01  -1.952
0.050937 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 755.48  on 544  degrees of freedom
Residual deviance: 378.38  on 531  degrees of freedom
AIC: 406.38

Number of Fisher Scoring iterations: 6

# Install the pROC package
install.packages("pROC")

# Load the pROC package
library(pROC)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Type 'citation("pROC")' for a citation.


Attaching package: ‘pROC’


The following objects are masked from ‘package:stats’:

    cov, smooth, var
```

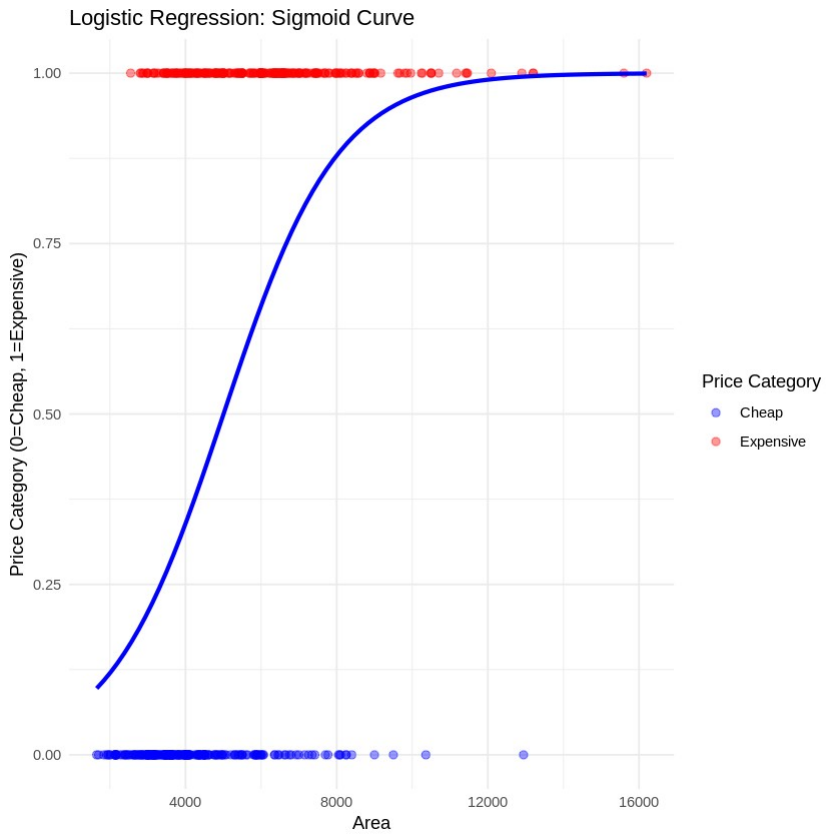
```r
# ROC curve
predictions <- predict(logistic_model, type = "response")
roc_obj <- roc(Housing$price_category, predictions)
plot(roc_obj, main = "ROC Curve for Logistic Regression")
auc(roc_obj)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases


Area under the curve: 0.9246
```




```
# Visualization of logistic regression
ggplot(Housing, aes(x = area, y = price_category, color =
factor(price_category))) +
  geom_point(alpha = 0.4, size = 2) +
  stat_smooth(method = "glm", method.args = list(family = "binomial"),
se = FALSE, color = "blue", lwd = 1.2) +
  labs(title = "Logistic Regression: Sigmoid Curve", x = "Area", y =
"Price Category (0=Cheap, 1=Expensive)") +
  scale_color_manual(values = c("blue", "red"), name = "Price
Category", labels = c("Cheap", "Expensive")) +
  theme_minimal()
```

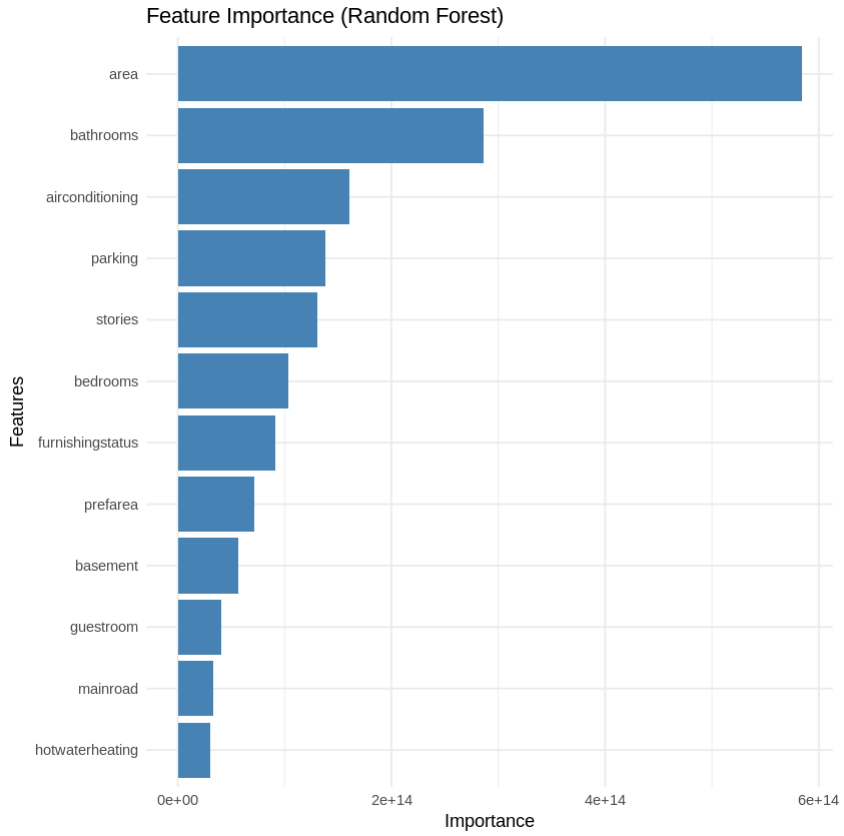
```
`geom_smooth()` using formula = 'y ~ x'
```




```r
# Install the randomForest package
install.packages("randomForest")

# Load the randomForest package
library(randomForest)
```

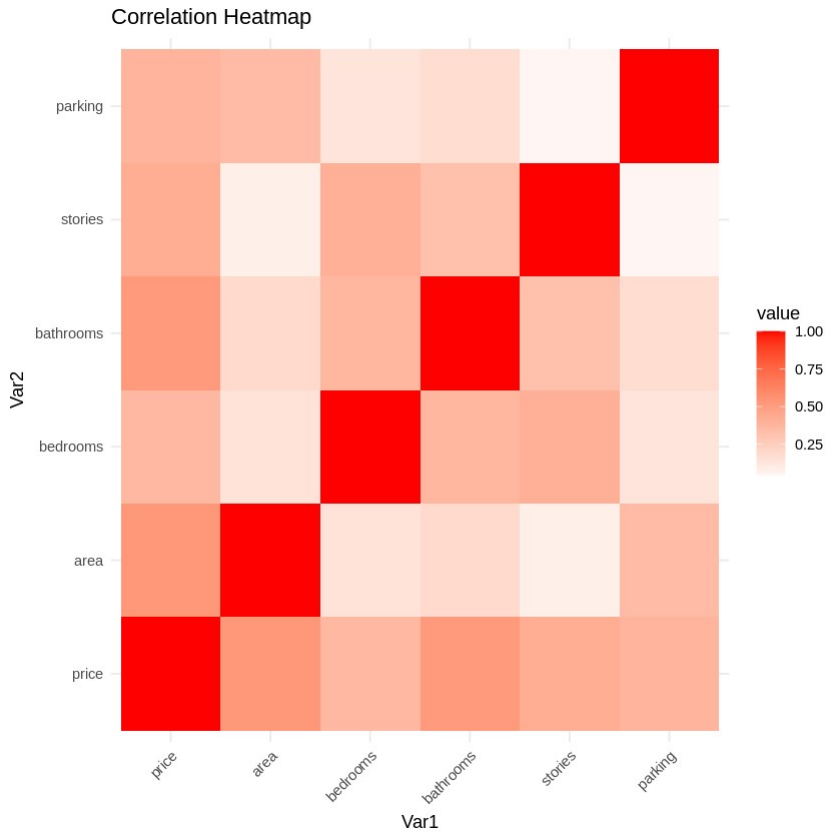
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

randomForest 4.7-1.2

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
```

```r
# 4. Feature Importance
# Random Forest for feature importance
rf_model <- randomForest(price ~ ., data = Housing[, -
which(names(Housing) == "price_category")], importance = TRUE)
importance_df <- as.data.frame(importance(rf_model))
importance_df$feature <- rownames(importance_df)

ggplot(importance_df, aes(x = reorder(feature, IncNodePurity), y =
IncNodePurity)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Feature Importance (Random Forest)", x = "Features", y
= "Importance") +
  theme_minimal()
```

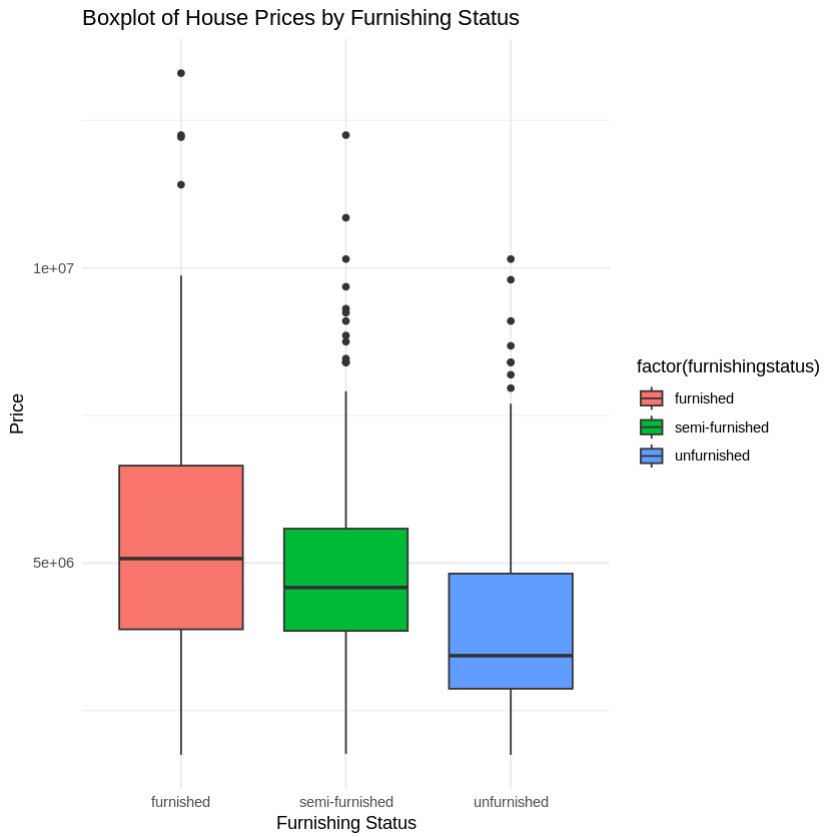



```r
# 5. Advanced Visualizations
# Heatmap of correlations
ggplot(melt(correlation_matrix), aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
midpoint = 0) +
```

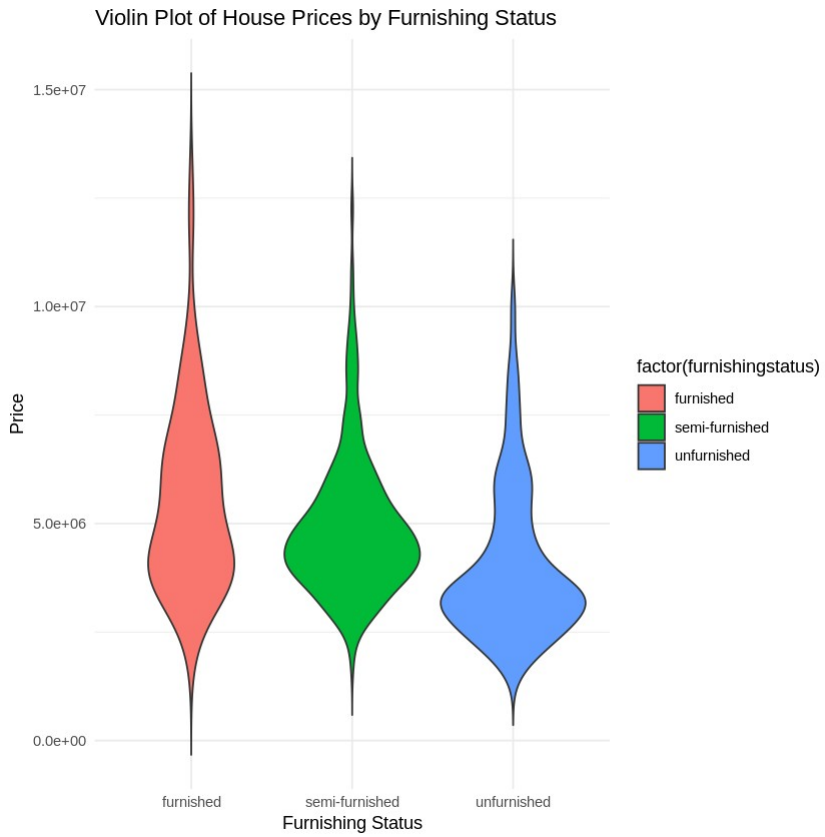
```
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation Heatmap")
```




```
# Boxplot of price by furnishing status
ggplot(Housing, aes(x = factor(furnishingstatus), y = price, fill =
factor(furnishingstatus))) +
  geom_boxplot() +
  labs(title = "Boxplot of House Prices by Furnishing Status", x =
"Furnishing Status", y = "Price") +
  theme_minimal()
```




```
# Violin plot of price by furnishing status
ggplot(Housing, aes(x = factor(furnishingstatus), y = price, fill =
factor(furnishingstatus))) +
  geom_violin(trim = FALSE) +
  labs(title = "Violin Plot of House Prices by Furnishing Status", x =
"Furnishing Status", y = "Price") +
  theme_minimal()
```

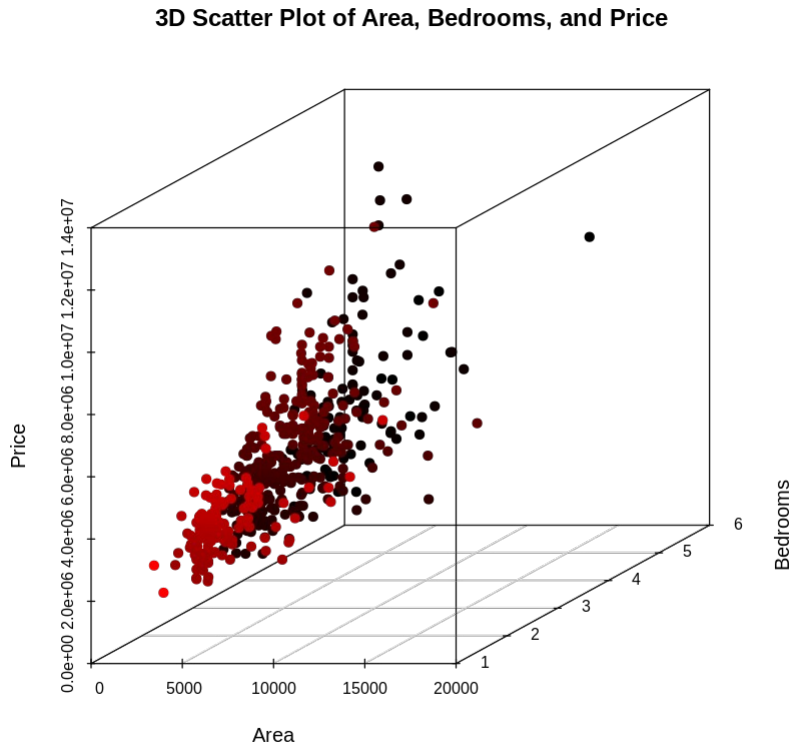



```
# Install and load the scatterplot3d package
install.packages("scatterplot3d")
library(scatterplot3d)
```

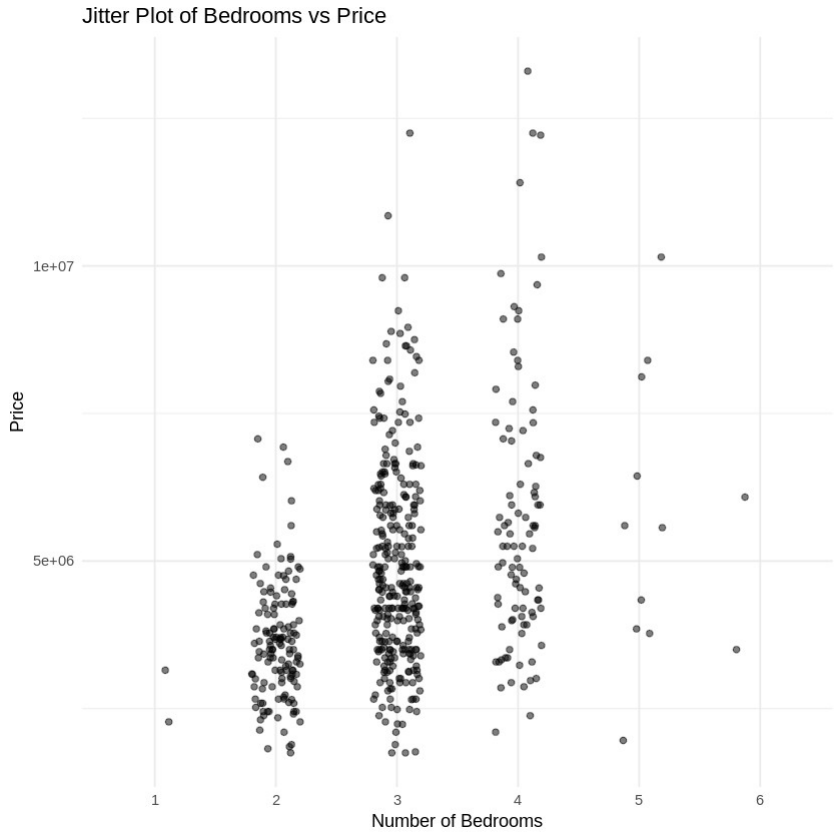
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
# 3D Scatter plot of area, bedrooms, and price
scatterplot3d(Housing$area, Housing$bedrooms, Housing$price,
              main="3D Scatter Plot of Area, Bedrooms, and Price",
              xlab="Area", ylab="Bedrooms", zlab="Price",
              color="blue", pch=19, highlight.3d=TRUE)
```

```
Warning message in scatterplot3d(Housing$area, Housing$bedrooms,
Housing$price, :
“color is ignored when highlight.3d = TRUE”
```

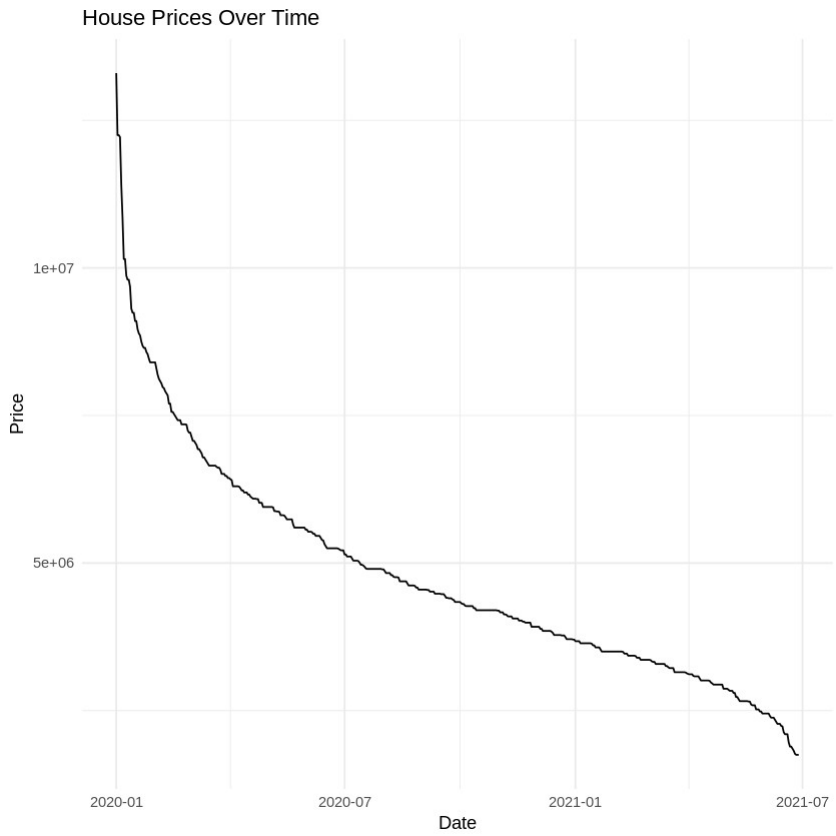



```
# Jitter plot of bedrooms vs price
ggplot(Housing, aes(x = factor(bedrooms), y = price)) +
  geom_jitter(width = 0.2, height = 0, alpha = 0.5) +
  labs(title = "Jitter Plot of Bedrooms vs Price", x = "Number of
Bedrooms", y = "Price") +
  theme_minimal()
```



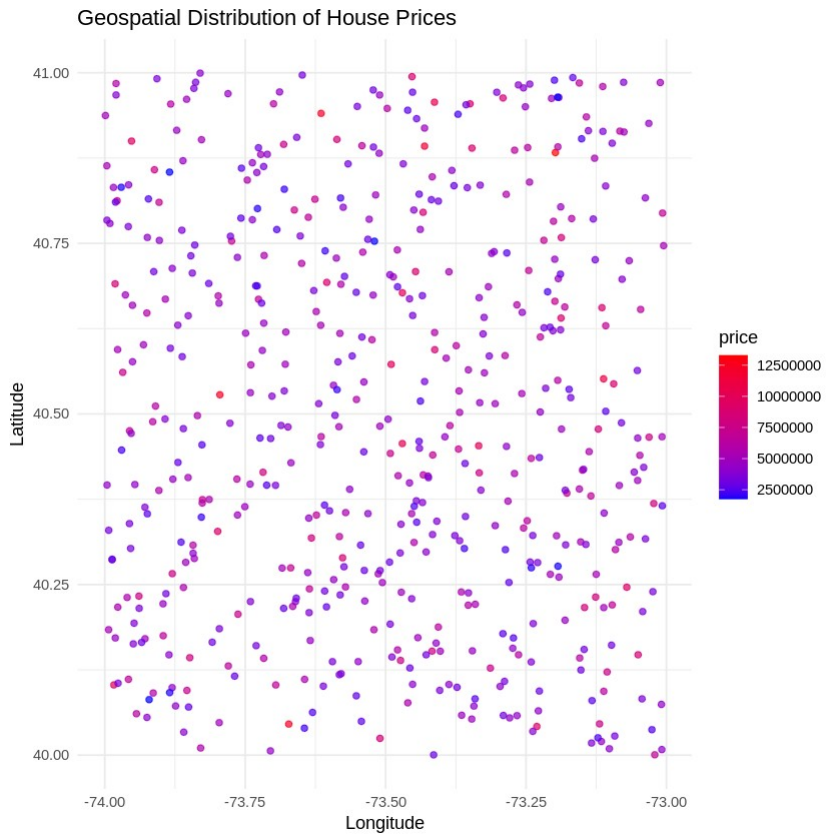

```
# WordCloud of furnishing status
furnishingstatus_text <- as.character(Housing$furnishingstatus)
furnishingstatus_table <- table(furnishingstatus_text)
wordcloud(names(furnishingstatus_table), furnishingstatus_table,
          colors = brewer.pal(8, "Dark2"), scale = c(3,0.5),
          random.order = FALSE)
```


```
# 6. Time Series Analysis (assuming we have a date column)
# If there's no date column, you can create a dummy one for
demonstration
Housing$date <- seq.Date(as.Date("2020-01-01"), by = "day", length.out
= nrow(Housing))

# Time series plot of prices
ggplot(Housing, aes(x = date, y = price)) +
  geom_line() +
  labs(title = "House Prices Over Time", x = "Date", y = "Price") +
  theme_minimal()
```




```
# 7. Clustering
# K-means clustering
set.seed(123)
kmeans_result <- kmeans(scale(Housing[, c("area", "price")]), centers
= 3)
Housing$cluster <- as.factor(kmeans_result$cluster)

ggplot(Housing, aes(x = area, y = price, color = cluster)) +
  geom_point(alpha = 0.7) +
  labs(title = "K-means Clustering of Houses", x = "Area", y =
"Price") +
  theme_minimal()
```


```r
# 8. Geospatial Analysis (if latitude and longitude are available)
# If not available, you can create dummy coordinates for demonstration
Housing$latitude <- runif(nrow(Housing), min = 40, max = 41)
Housing$longitude <- runif(nrow(Housing), min = -74, max = -73)

ggplot(Housing, aes(x = longitude, y = latitude, color = price)) +
  geom_point(alpha = 0.7) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Geospatial Distribution of House Prices", x =
"Longitude", y = "Latitude") +
  theme_minimal()
```




```r
# 9. Price Prediction
# Split the data
set.seed(123)
train_index <- createDataPartition(Housing$price, p = 0.8, list =
FALSE)
train_data <- Housing[train_index, ]
test_data <- Housing[-train_index, ]

# Train a random forest model
rf_model <- randomForest(price ~ area + bedrooms + bathrooms + stories
+ mainroad + guestroom + basement + hotwaterheating + airconditioning
+ parking + prefarea + furnishingstatus,
                         data = train_data)

# Make predictions
predictions <- predict(rf_model, newdata = test_data)

# Evaluate the model
mse <- mean((test_data$price - predictions)^2)
rmse <- sqrt(mse)
r_squared <- 1 - sum((test_data$price - predictions)^2) /
sum((test_data$price - mean(test_data$price))^2)

cat("Root Mean Squared Error:", rmse, "\n")
cat("R-squared:", r_squared, "\n")
```

```
Root Mean Squared Error: 1146735
R-squared: 0.6763417
```

```r
# Plot predicted vs actual prices
ggplot(data.frame(actual = test_data$price, predicted = predictions),
aes(x = actual, y = predicted)) +
  geom_point(alpha = 0.5) +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype =
"dashed") +
  labs(title = "Predicted vs Actual House Prices", x = "Actual Price",
y = "Predicted Price") +
  theme_minimal()
```