Exploring the Relationship Between House Area, Cooling Systems, and Price: A Comparative Modeling Approach Using Linear, Polynomial, and kNN Regression

Isaias Garcia

```r
# Load necessary libraries for data manipulation, modeling, and visualization
library(caret)   # For machine learning and modeling
```

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

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

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

```
## Warning: package 'lattice' was built under R version 4.2.3
```

```r
library(ggplot2) # For data visualization
library(openintro) # For the duke_forest dataset
```

```
## Loading required package: airports
```

```
## Loading required package: cherryblossom
```

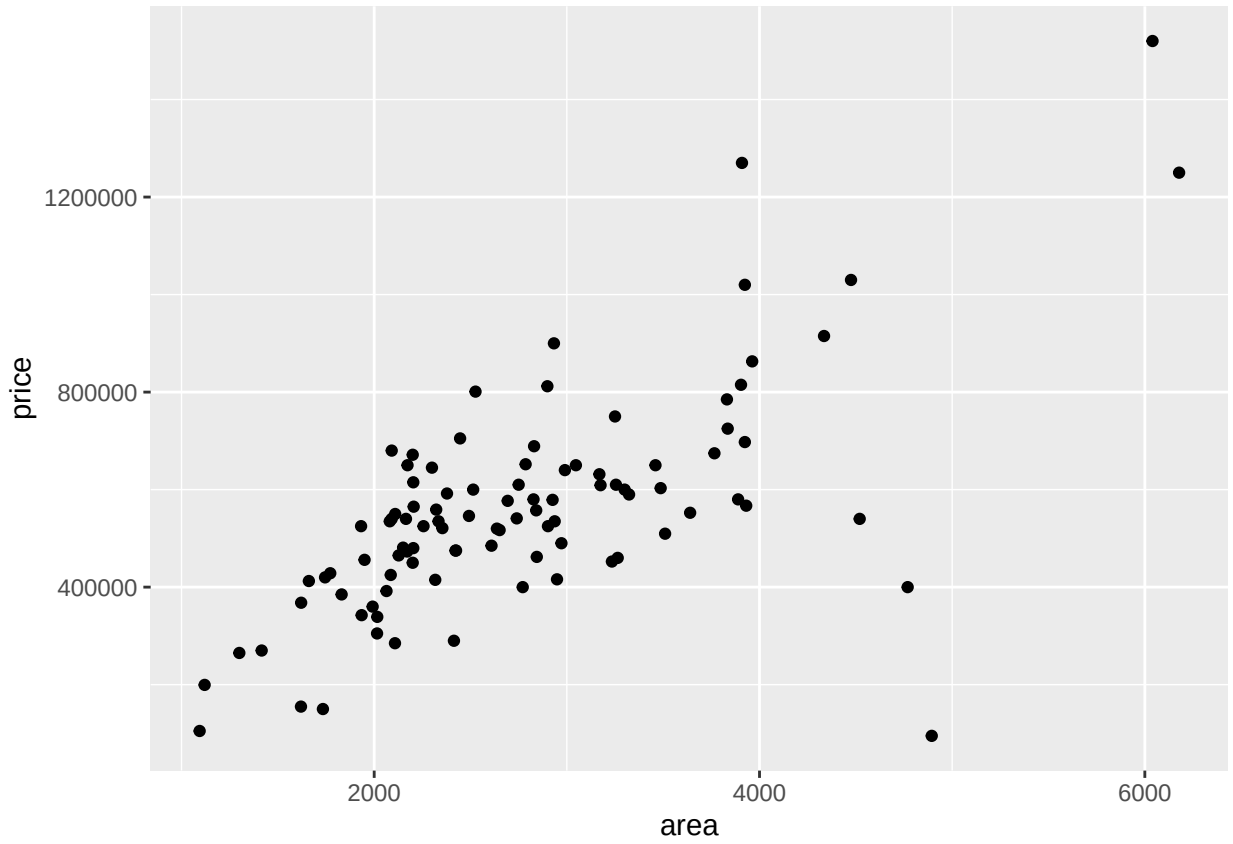
```
## Loading required package: usdata
```

```
##
## Attaching package: 'openintro'
```

```
## The following object is masked from 'package:caret':
##
##     dotPlot
```

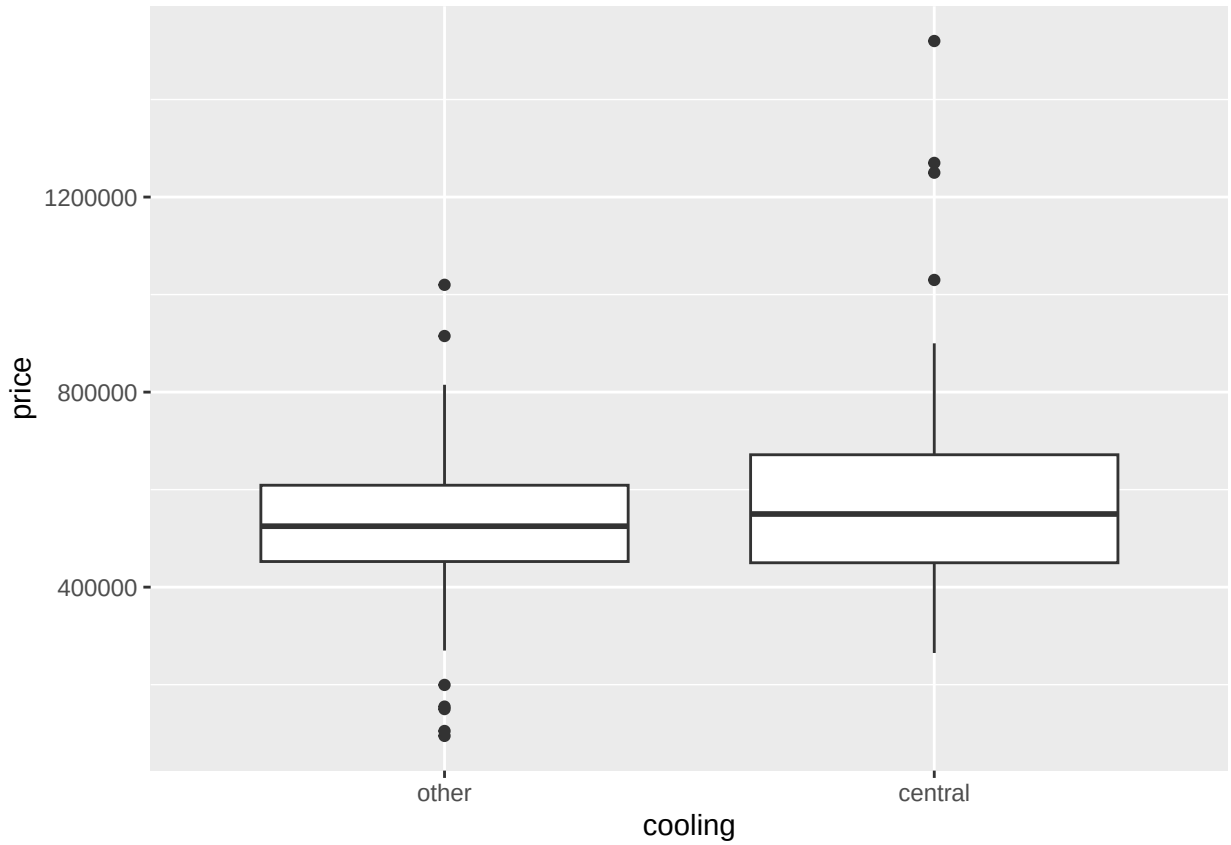
```
## The following objects are masked from 'package:lattice':
##
##     ethanol, lsegments
```

```r
# Plot a scatter plot showing the relationship between area and price
ggplot(duke_forest) +
  geom_point(aes(x = area, y = price))
```



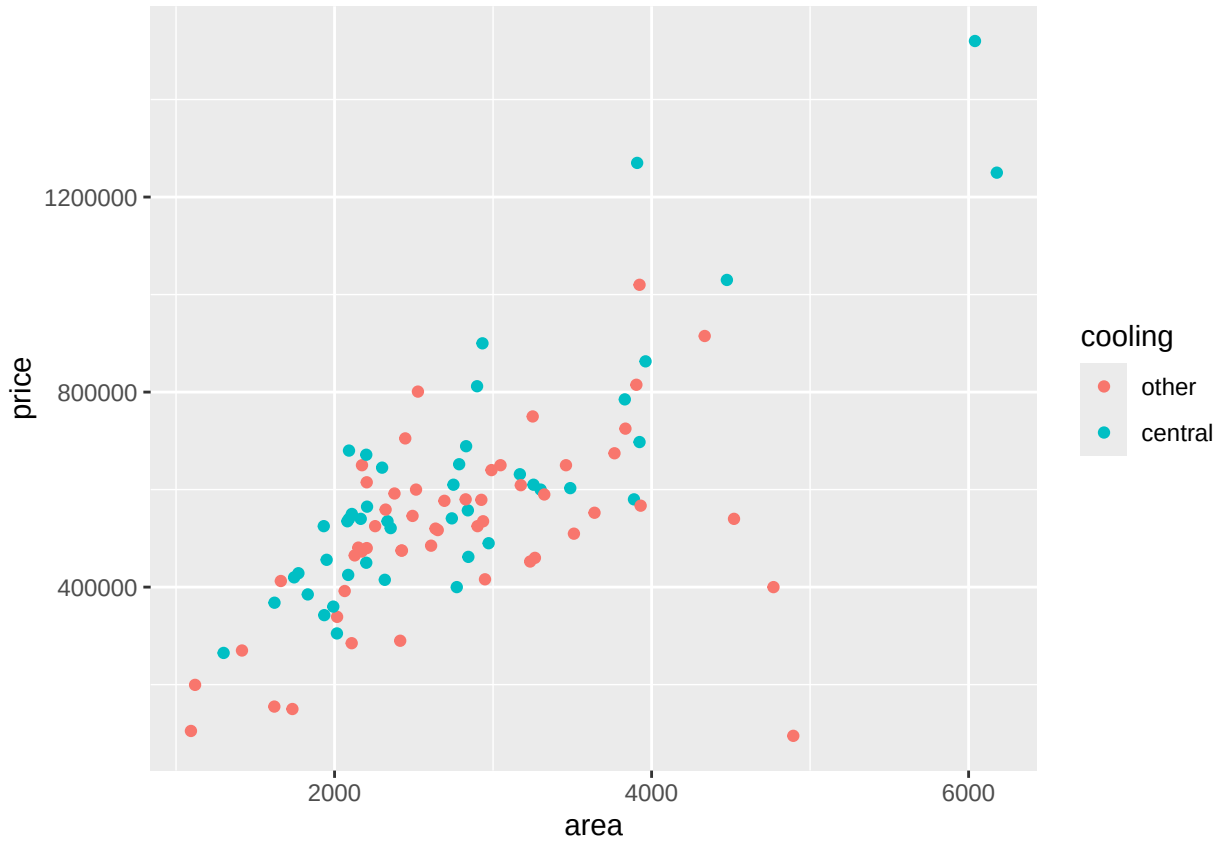

```
# Scatter plot with area on x-axis and price on y-axis
#  This plot visualizes how the price of houses changes with the area. It helps identify
# any correlation.
```

```
# Plot a boxplot to show the relationship between the type of cooling system and price
ggplot(duke_forest) +
  geom_boxplot(aes(x = cooling, y = price))
```




```
# Boxplot with cooling types on x-axis and price on y-axis
# This boxplot compares the house prices based on different types of cooling systems.
```

```
# Plot a scatter plot with colors indicating the type of cooling system
ggplot(duke_forest) +
  geom_point(aes(x = area, y = price, color = cooling))
```




```
# As the area increases, the house price tends to increase. Central cooling systems tend
# to be associated with higher prices.
```

```
# Fit a linear model predicting price based on area
m1 <- lm(price ~ area, data = duke_forest)
summary(m1)
```

```
##
## Call:
## lm(formula = price ~ area, data = duke_forest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -802163  -70824   -3786   85449  529928
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 116652.33   53302.46   2.188   0.0311 *
## area           159.48      18.17   8.777 6.29e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 168800 on 96 degrees of freedom
## Multiple R-squared:  0.4452, Adjusted R-squared:  0.4394
## F-statistic: 77.03 on 1 and 96 DF,  p-value: 6.292e-14
```

```
# Fit a generalized linear model (GLM) with the same formula
m2 <- glm(price ~ area, data = duke_forest)
```

```r
summary(m2)
```

```
##
## Call:
## glm(formula = price ~ area, data = duke_forest)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -802163   -70824    -3786    85449   529928
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 116652.33   53302.46   2.188   0.0311 *
## area           159.48      18.17   8.777 6.29e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 28492742139)
##
##     Null deviance: 4.9302e+12  on 97  degrees of freedom
## Residual deviance: 2.7353e+12  on 96  degrees of freedom
## AIC: 2641.2
##
## Number of Fisher Scoring iterations: 2
```

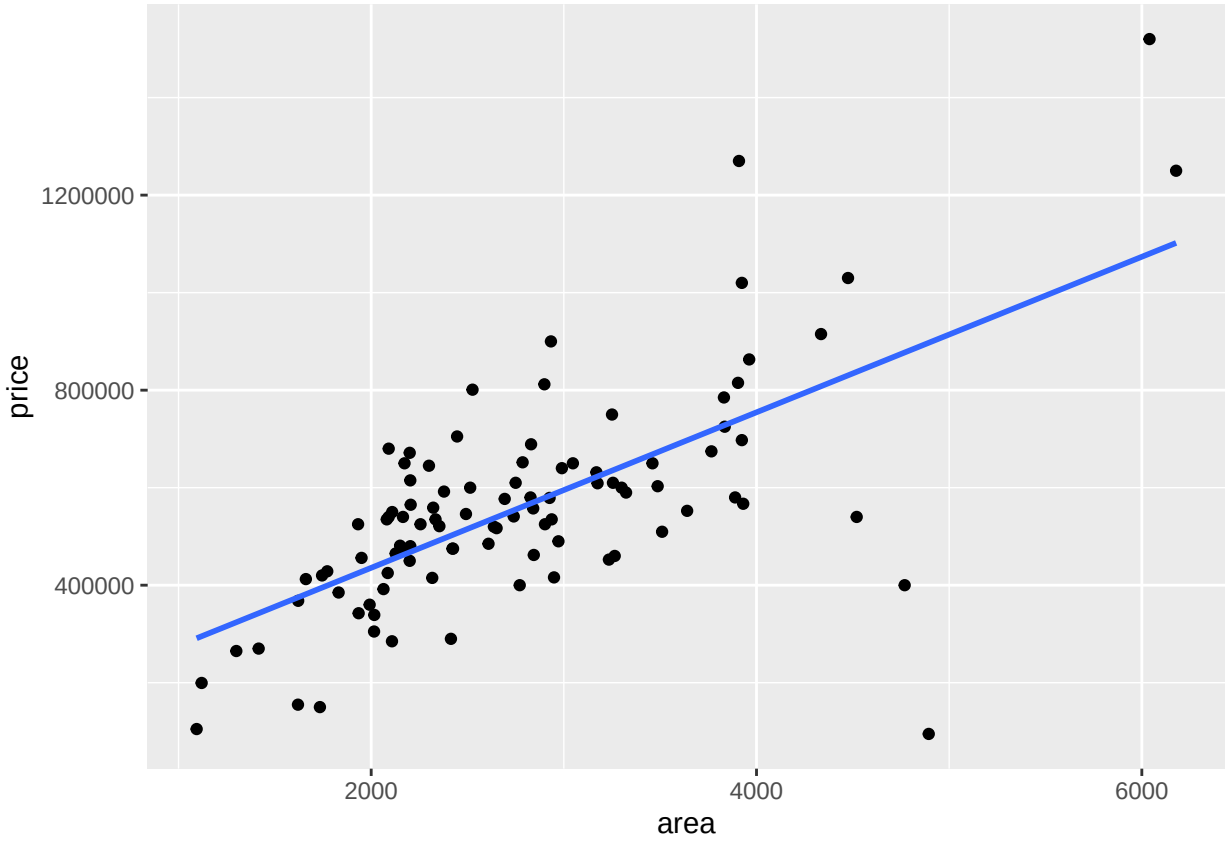
```r
# A GLM is fitted for comparison, but the coefficients remain the same because the
# default family is Gaussian.
```

```r
# Fit a linear model using the caret package, providing additional modeling options
m3 <- train(price ~ area, data = duke_forest, method = "lm")
summary(m3)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -802163  -70824   -3786   85449  529928
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 116652.33   53302.46   2.188   0.0311 *
## area           159.48      18.17   8.777 6.29e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 168800 on 96 degrees of freedom
## Multiple R-squared:  0.4452, Adjusted R-squared:  0.4394
## F-statistic: 77.03 on 1 and 96 DF,  p-value: 6.292e-14
```

```r
# Plot the regression line from the linear model on top of the scatter plot
ggplot(duke_forest) +
  geom_point(aes(x = area, y = price)) +
  geom_smooth(method = "lm", aes(x = area, y = price), se = FALSE) # Adds regression line
```

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

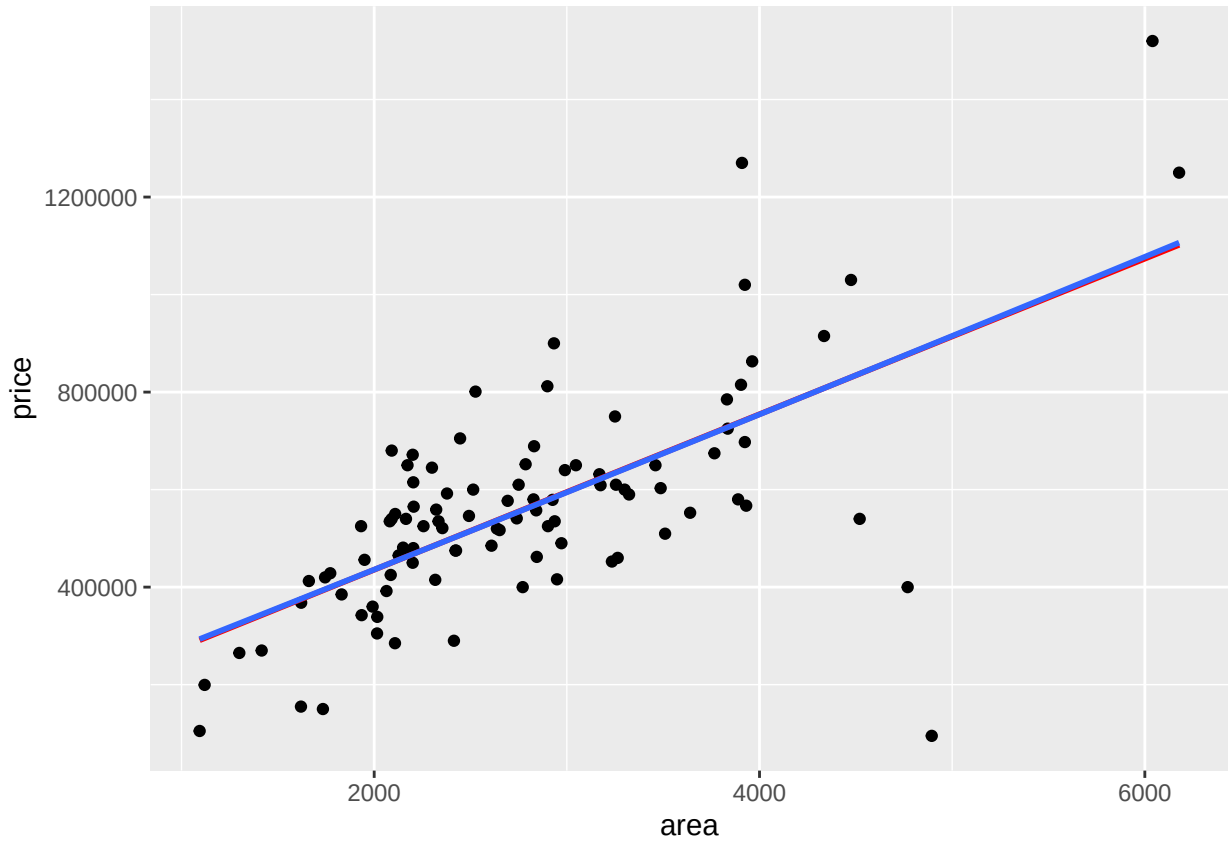



```r
# Fit a linear model with an interaction term between area and cooling system
lm(price ~ area * cooling, data = duke_forest)
```

```
##
## Call:
## lm(formula = price ~ area * cooling, data = duke_forest)
##
## Coefficients:
##         (Intercept)                 area       coolingcentral
##           239255.82                99.41           -211987.70
## area:coolingcentral
##              111.61
```

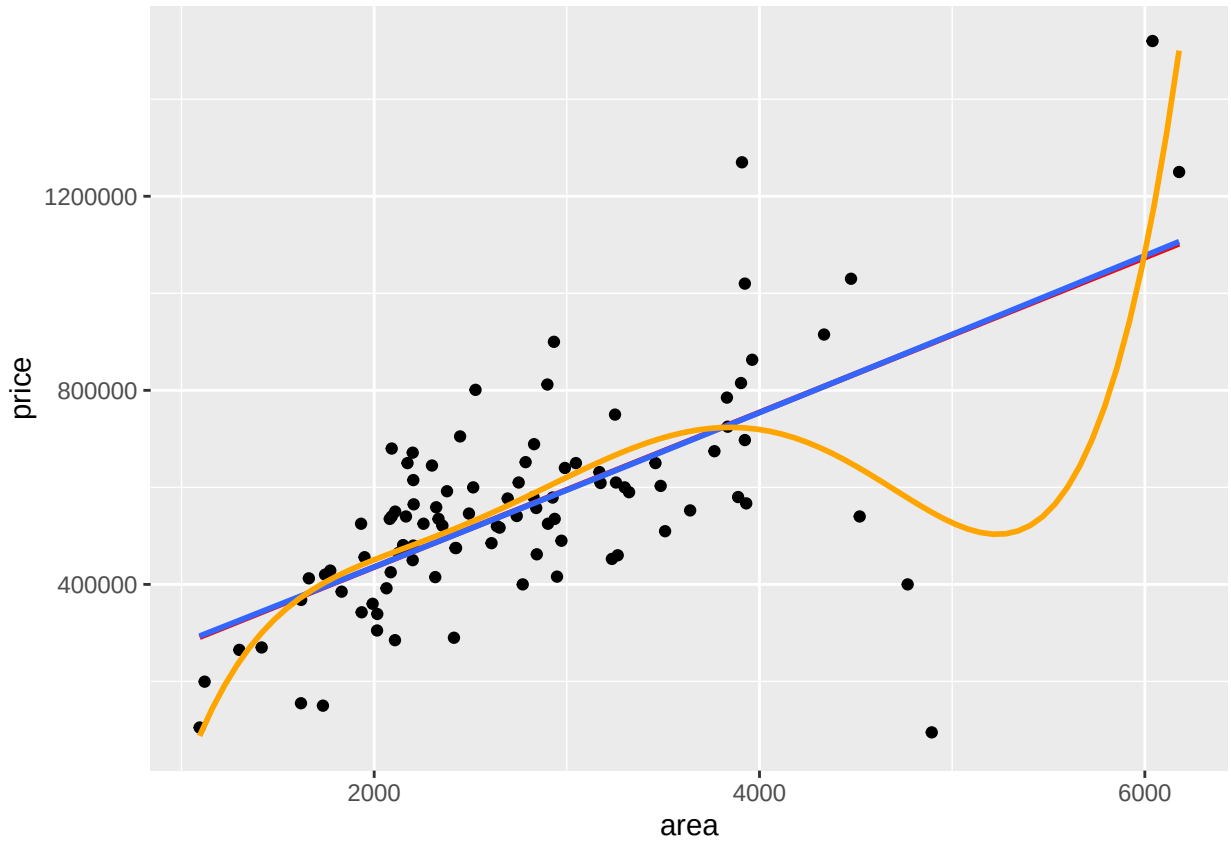
```r
# This model includes an interaction term to assess how the cooling system modifies the
# relationship between area and price.
```

```r
# Fit a polynomial regression model including an area-squared term
m_poly2 <- lm(price ~ area + I(area^2), data = duke_forest)

# Plot scatter plot with linear and quadratic regression lines
ggplot(duke_forest, aes(x = area, y = price)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, col = "red") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE)
```



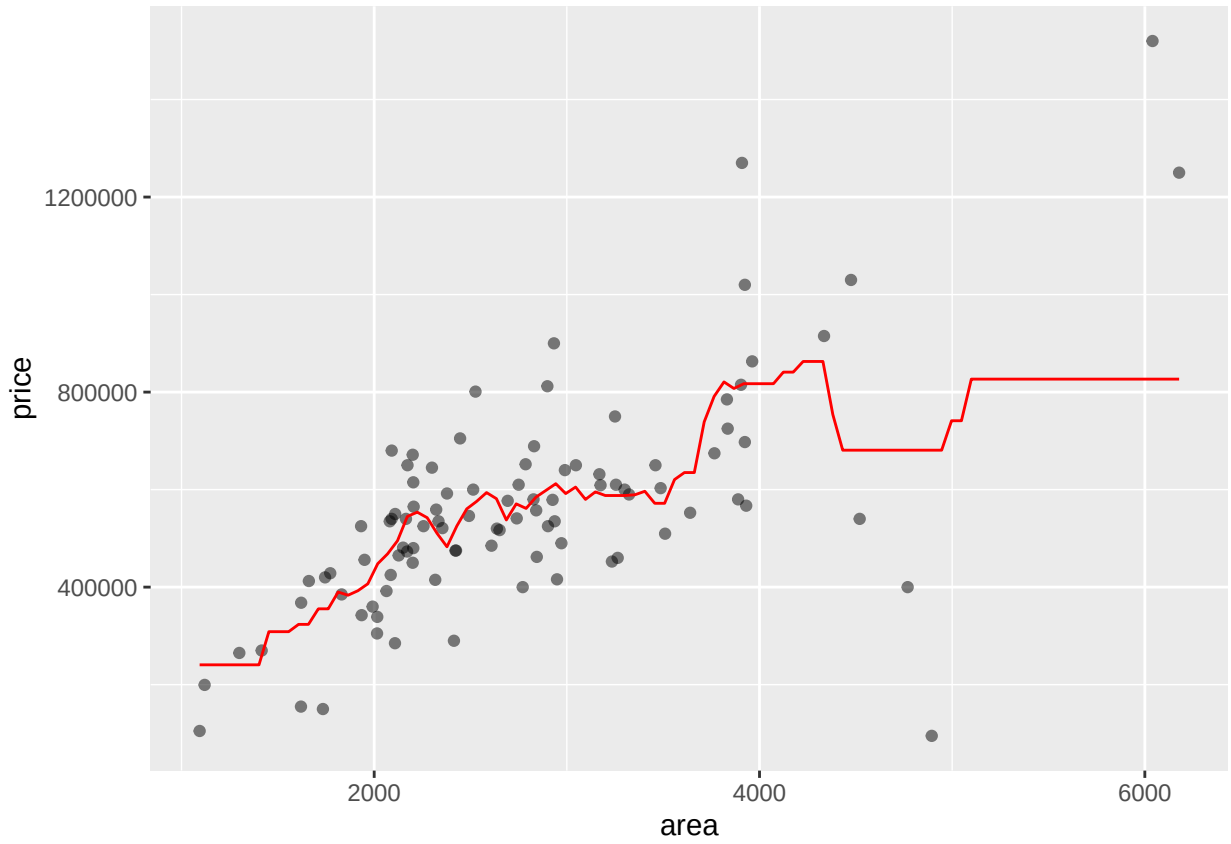

```
# Adds quadratic regression line
# Fits a quadratic model and visualizes it against a simple linear fit to assess added
# flexibility.
```

```
# Fit a 5th-degree polynomial regression model
m_poly5 <- lm(price ~ poly(area, 5), data = duke_forest)

# Plot scatter plot with linear, quadratic, and 5th-degree polynomial regression lines
ggplot(duke_forest, aes(x = area, y = price)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, col = "red") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 5), se = FALSE, col = "orange") # Adds 5th-degree re
```



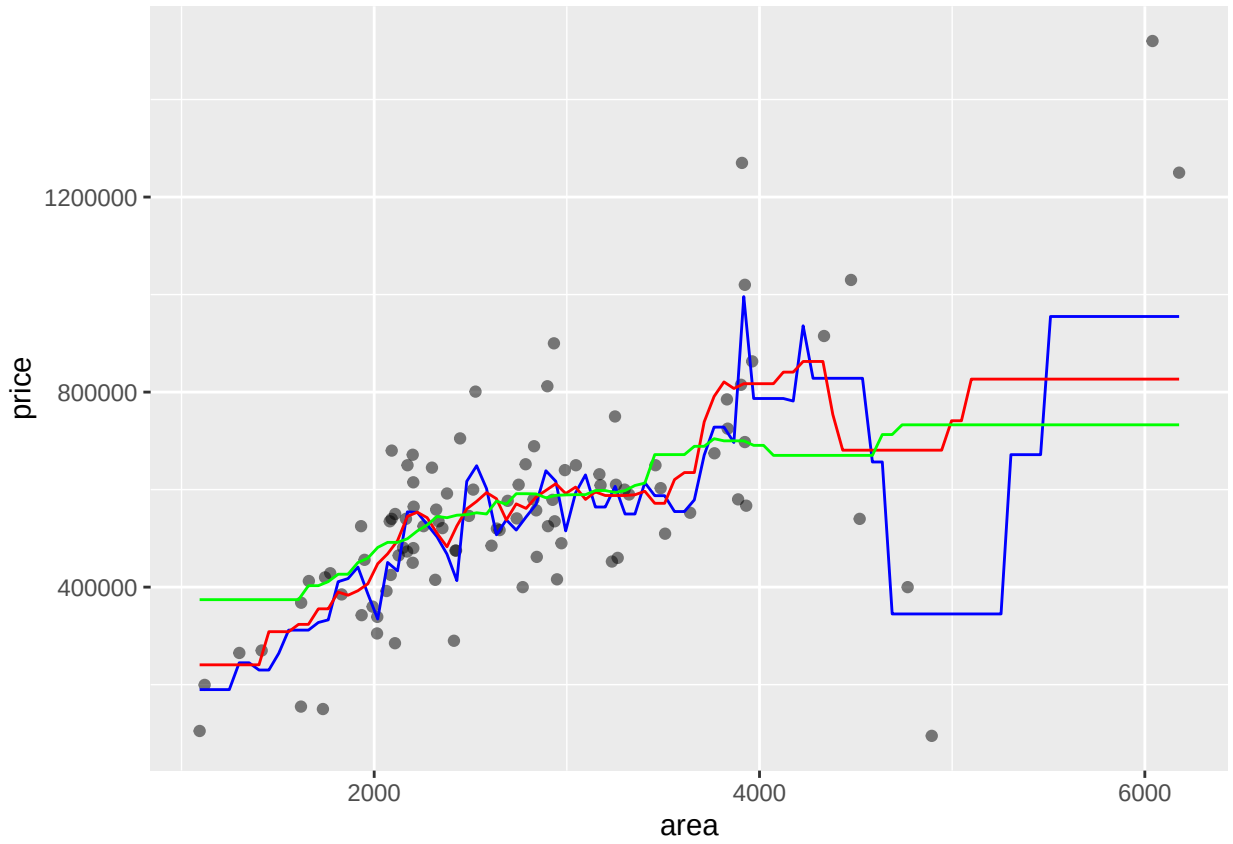

```
# Compares a 5th-degree polynomial fit to the linear and quadratic fits, which may lead
# to overfitting.
```

```
# Fit a k-nearest neighbors (kNN) regression model with k = 8
knn_fit_8 <- knnreg(price ~ area, data = duke_forest, k = 8)

# Generate predictions for a sequence of area values
area_seq <- seq(min(duke_forest$area), max(duke_forest$area), length.out = 100)
pred_data <- data.frame(area = area_seq)
preds_8 <- predict(knn_fit_8, newdata = pred_data)
pred_data$price_8 <- preds_8

# Plot the kNN predictions for k = 8
ggplot(duke_forest, aes(x = area, y = price)) +
  geom_point(alpha = 0.5) +
  geom_line(data = pred_data, aes(x = area, y = price_8), color = "red")
```




```
# Adds kNN prediction line
# Fits a kNN model and plots predictions with k=8k=8, showing how well the model fits the data.
```

```
# Fit additional kNN models with different k values
knn_fit_3 <- knnreg(price ~ area, data = duke_forest, k = 3)
knn_fit_25 <- knnreg(price ~ area, data = duke_forest, k = 25)

# Generate predictions for k = 3 and k = 25
preds_3 <- predict(knn_fit_3, newdata = pred_data)
preds_25 <- predict(knn_fit_25, newdata = pred_data)

# Add predictions to the data frame for plotting
pred_data$price_3 <- preds_3
pred_data$price_25 <- preds_25

# Update the plot with kNN predictions for different k values
ggplot(duke_forest, aes(x = area, y = price)) +
  geom_point(alpha = 0.5) +
  geom_line(data = pred_data, aes(x = area, y = price_3), color = "blue") + # k = 3
  geom_line(data = pred_data, aes(x = area, y = price_8), color = "red") +  # k = 8
  geom_line(data = pred_data, aes(x = area, y = price_25), color = "green") # k = 25
```




```
#  The graph illustrates the results of applying k-Nearest Neighbors (kNN)
# regression to predict housing prices based on property area, using three
# different values of kk: 3, 8, and 25. The black dots represent the actual
# data points from the duke_forest dataset, showing the relationship between
# area and price. The colored lines indicate the predictions made by
# the kNN models: blue for k=3, red for k=8, and green for k=25.
```

```
# This project explores various modeling approaches to understand how housing
# prices in the duke_forest dataset relate to property area and other features,
# such as cooling system type. It begins with visualizations, including scatter
# plots and boxplots, to highlight relationships between variables. Several
# linear models are then fitted, including simple linear regression,
# polynomial regression, and interaction terms, to examine how well different
# formulations capture the price trends.
# In addition to traditional linear modeling, the project introduces more
# flexible methods such as k-Nearest Neighbors (kNN) regression. By fitting
# kNN models with different values of kk (3, 8, and 25), the analysis visually
# compares how model flexibility affects predictions. Smaller kk values closely
# follow the data and capture local variations, while larger kk values produce
# smoother, more generalized fits.
# Overall, the project demonstrates how both parametric (linear, polynomial)
# and non-parametric (kNN) methods can be used to model and visualize
# complex relationships in real-world housing data, providing insights into
# model behavior, trade-offs, and the importance of selecting appropriate
# methods for prediction tasks.
```