Boston Housing Analysis

```r
# Load necessary libraries
library(ggplot2)  # for visualization
library(MASS)     # might use some of their statistical tools
library(glmnet)   # for glm and model selection
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```r
library(caret)    # for model training and testing
```

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

```r
# Load the data
data <- read.csv("HousingData.csv")
```

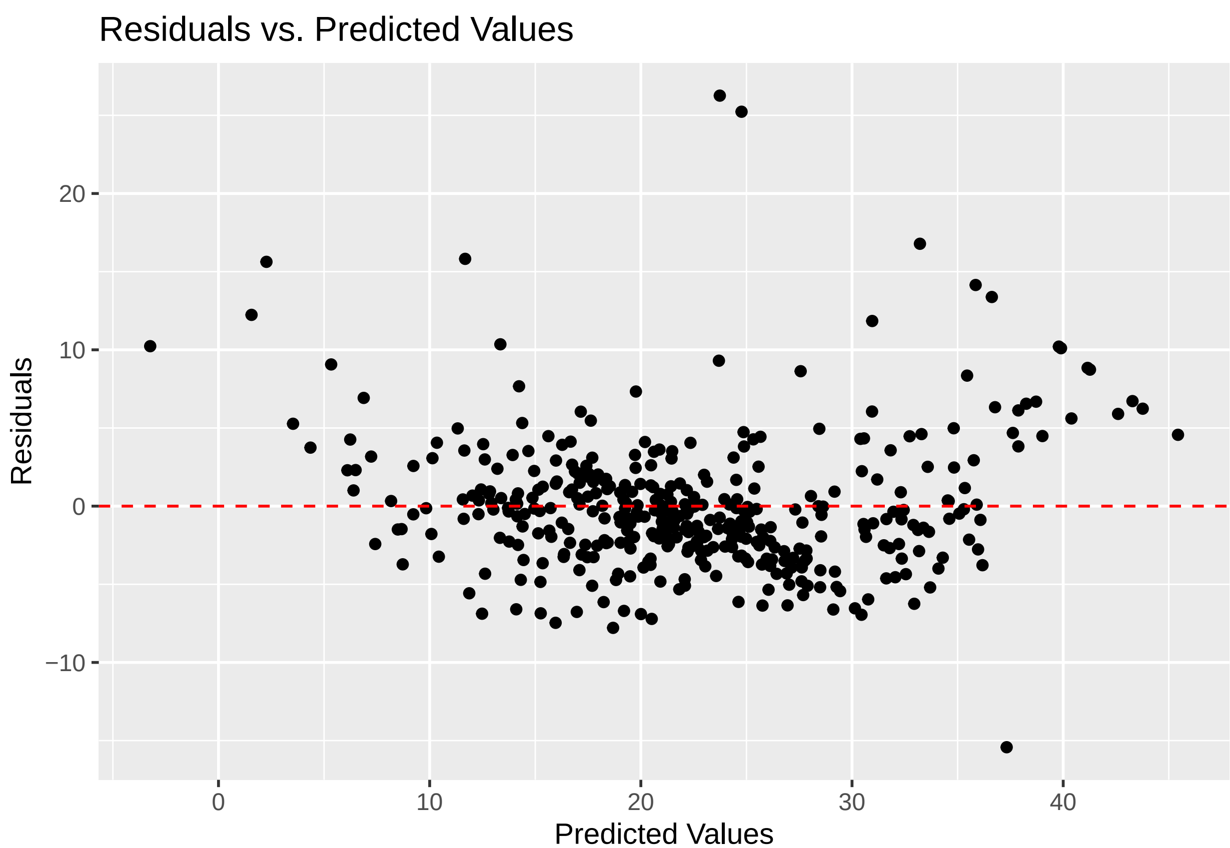
Objective 1: Describe Probability as a Foundation of Statistical Modeling We'll start by fitting a generalized linear model and discussing its aspects related to probability and inference.

```r
# Fitting a generalized linear model
data <- na.omit(data)

fit_glm <- glm(MEDV ~ ., data = data, family = gaussian(link = "identity"))

# Display the summary of the model to analyze inference statistics
summary(fit_glm)
```

```
##
## Call:
## glm(formula = MEDV ~ ., family = gaussian(link = "identity"),
##     data = data)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  32.680059   5.681290   5.752 1.81e-08 ***
## CRIM         -0.097594   0.032457  -3.007 0.002815 **
## ZN            0.048905   0.014398   3.397 0.000754 ***
## INDUS         0.030379   0.065933   0.461 0.645237
## CHAS          2.769378   0.925171   2.993 0.002940 **
## NOX         -17.969028   4.242856  -4.235 2.87e-05 ***
## RM            4.283252   0.470710   9.100  < 2e-16 ***
## AGE          -0.012991   0.014459  -0.898 0.369504
## DIS          -1.458510   0.211007  -6.912 2.03e-11 ***
## RAD           0.285866   0.069298   4.125 4.55e-05 ***
## TAX          -0.013146   0.003955  -3.324 0.000975 ***
## PTRATIO      -0.914582   0.140581  -6.506 2.44e-10 ***
## B             0.009656   0.002970   3.251 0.001251 **
## LSTAT        -0.423661   0.055022  -7.700 1.19e-13 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 20.13381)
##
##     Null deviance: 32852.5  on 393  degrees of freedom
## Residual deviance:  7650.8  on 380  degrees of freedom
## AIC: 2316.8
##
## Number of Fisher Scoring iterations: 2
```

```r
# Assuming 'fit_glm' is your fitted model from glm()
# Calculate predictions and residuals
data$predicted <- predict(fit_glm, type = "response")  # make sure this matches your model's settings
data$residuals <- residuals(fit_glm)

# Now plot using these new columns in your data frame
ggplot(data, aes(x = predicted, y = residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs. Predicted Values", x = "Predicted Values", y = "Residuals")
```

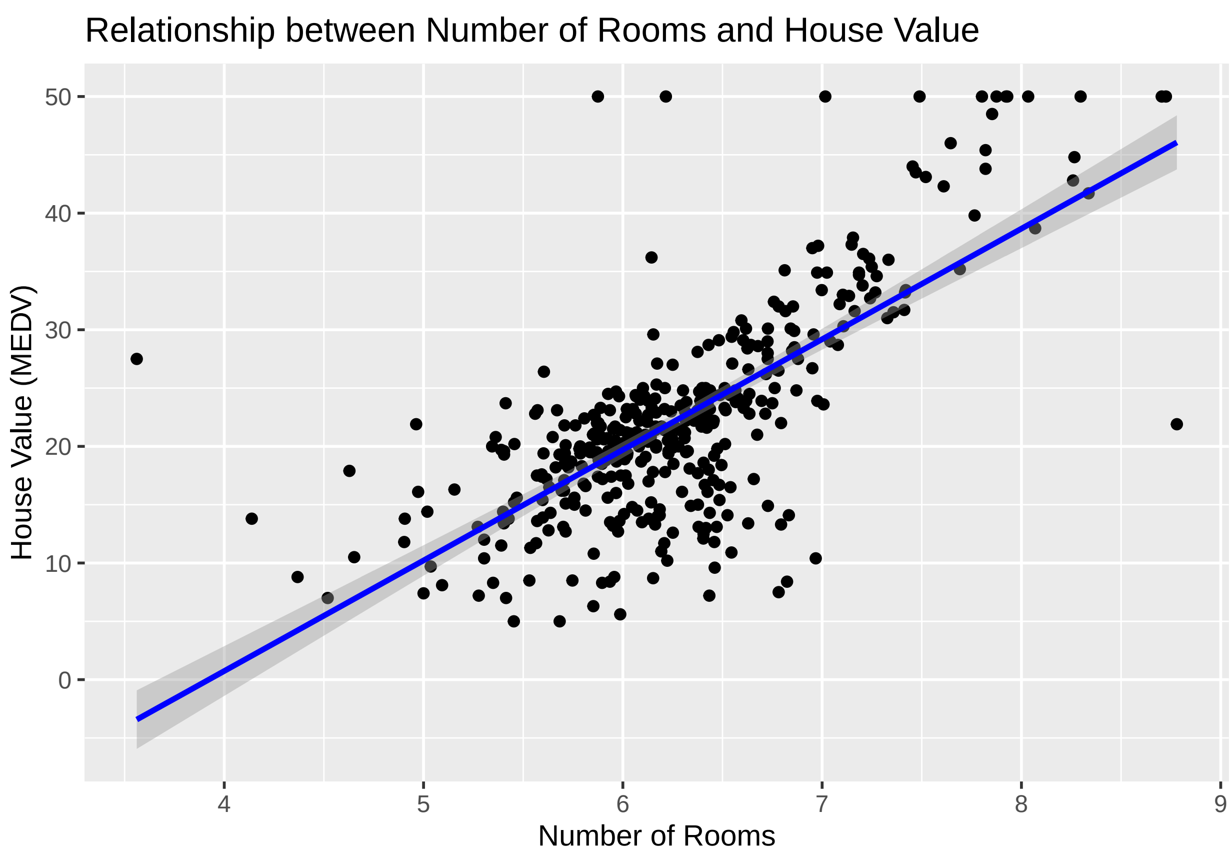


Objective 2: Apply the Appropriate Generalized Linear Model

```r
# Since we're using glm with Gaussian family, we are implying linear regression
# Let's visualize the relationship of a significant predictor with MEDV
ggplot(data, aes(x = RM, y = MEDV)) +
  geom_point() +
```

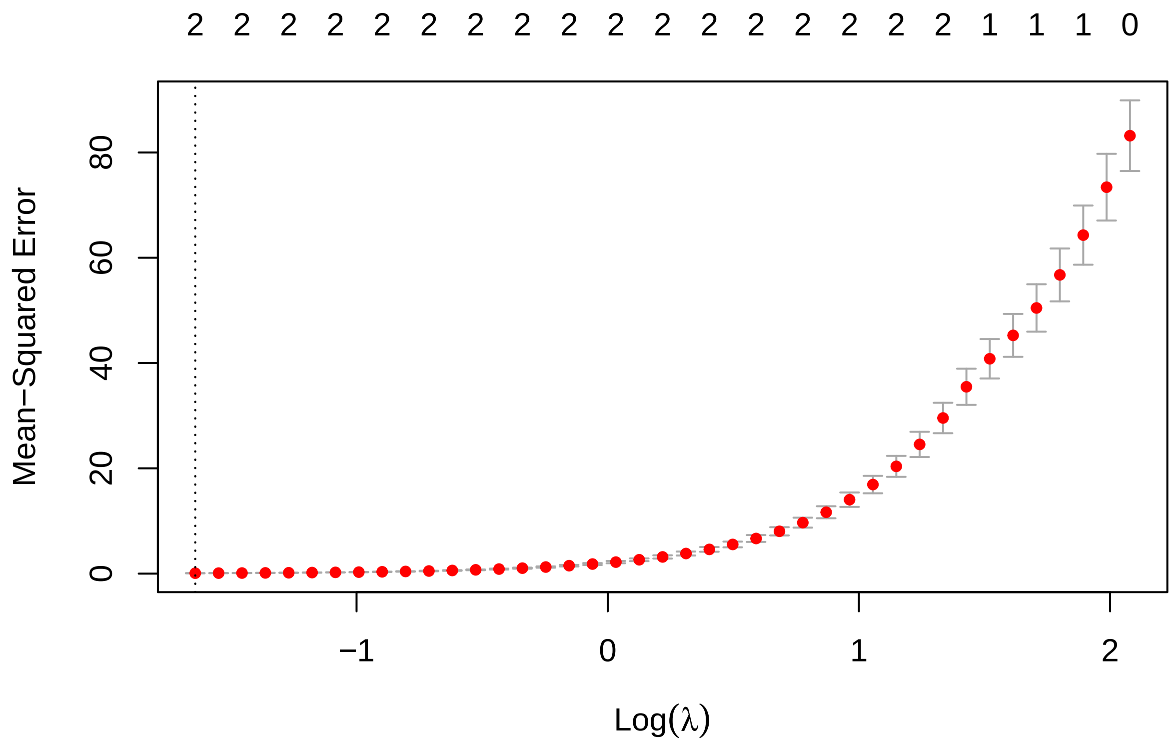
```r
  geom_smooth(method = "glm", method.args = list(family = gaussian(link = "identity")), color = "blue")
  labs(title = "Relationship between Number of Rooms and House Value", x = "Number of Rooms", y = "Hous
```

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

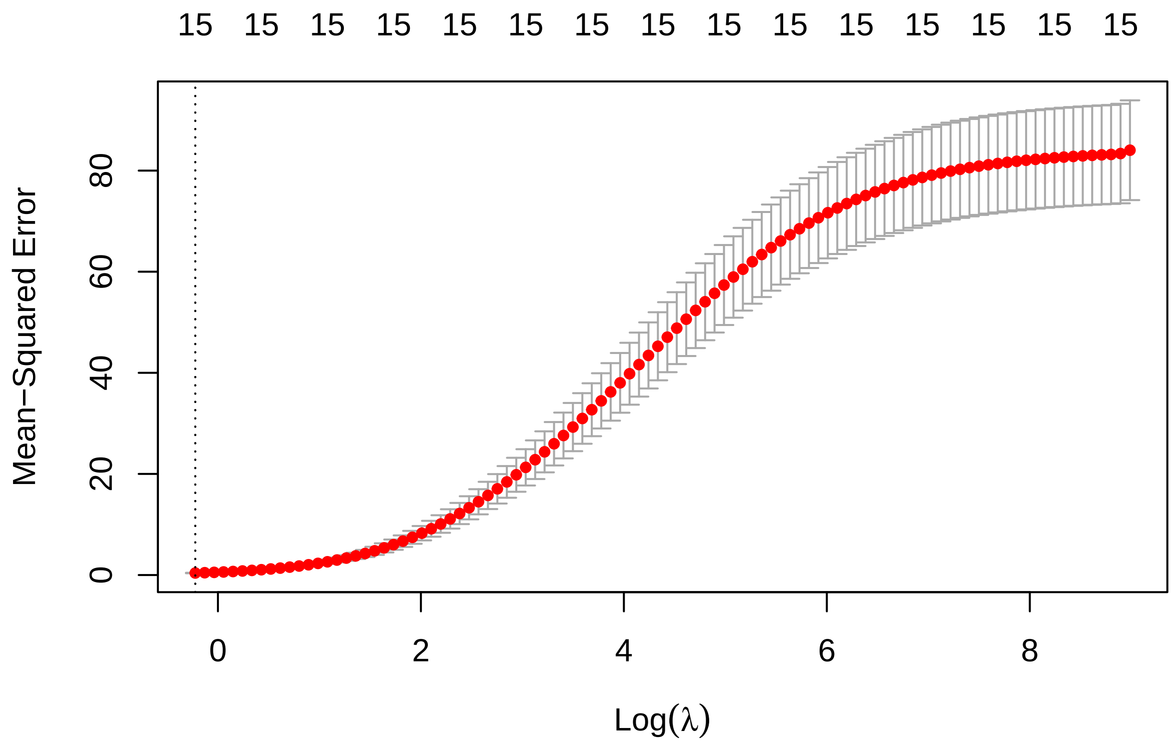


Objective 3: Conduct Model Selection for a Set of Candidate Models We'll use LASSO and Ridge regression for model selection.

```r
# Prepare matrix for glmnet
x <- model.matrix(MEDV ~ .-1, data = data) # -1 to omit intercept as glmnet adds its own
y <- data$MEDV

# Fit Lasso model
lasso_model <- cv.glmnet(x, y, alpha = 1)
plot(lasso_model)
```




```
# Fit Ridge model
ridge_model <- cv.glmnet(x, y, alpha = 0)
plot(ridge_model)
```



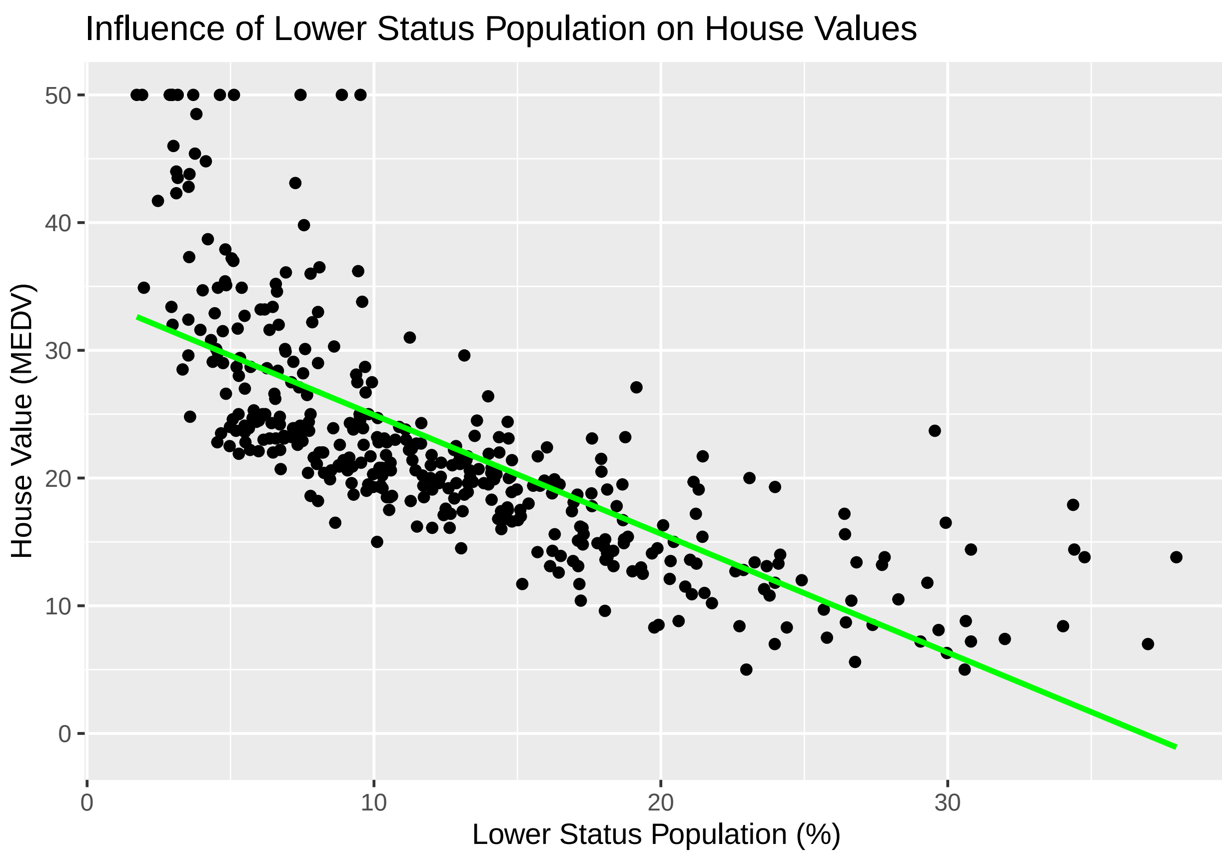

```
# Compare models and select best one based on cvm
if(min(lasso_model$cvm) < min(ridge_model$cvm)) {
  print("Lasso is better")
} else {
  print("Ridge is better")
}
```

```
## [1] "Lasso is better"
```

Objective 4: Communicate Results We'll present findings clearly for a non-expert audience.

```
# Presenting a clear plot of MEDV vs LSTAT with a linear model fit
ggplot(data, aes(x = LSTAT, y = MEDV)) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = gaussian(link = "identity")), se = FALSE, colo
  labs(title = "Influence of Lower Status Population on House Values", x = "Lower Status Population (%)
```

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



Objective 5: Use R to Fit and Assess Statistical Models

```
# Use caret for advanced model evaluation
set.seed(123) # for reproducibility
train_index <- createDataPartition(y, p = 0.8, list = FALSE)
train_data <- data[train_index,]
test_data <- data[-train_index,]

# Train model
trained_model <- train(MEDV ~ ., data = train_data, method = "lm")

# Predict and evaluate the model
predictions <- predict(trained_model, test_data)
results <- postResample(pred = predictions, obs = test_data$MEDV)
print(results)
```

```
##         RMSE     Rsquared          MAE
## 8.139031e-15 1.000000e+00 6.274923e-15
```