Section 7 | Linear Regression

Mohammad Saqib Ansari

2023-12-04

Simple Linear Regression Analysis

This analysis will demonstrate a simple linear regression using the `cars` dataset.

Step 1: Reading and Understanding the Data

```
data(cars)
head(cars, n = 10)  # Display the top 10 rows of the dataset
```

```
##    speed dist
## 1      4    2
## 2      4   10
## 3      7    4
## 4      7   22
## 5      8   16
## 6      9   10
## 7     10   18
## 8     10   26
## 9     10   34
## 10    11   17
```

```
str(cars)           # Display the structure and variables in the dataset
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
##  $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

```
summary(cars)       # Display summary statistics and information
```

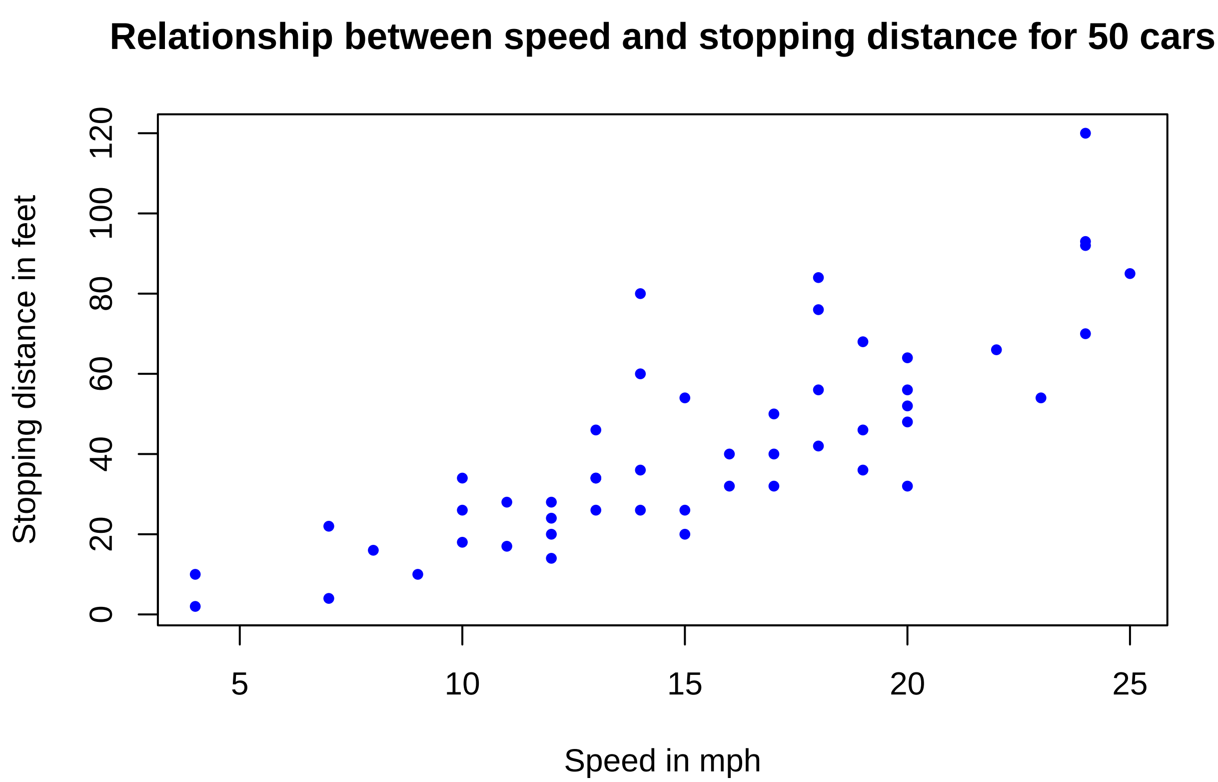
```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

```r
table(is.na(cars))   # Check for missing values
```

```
##
## FALSE
##   100
```

Step 2: Scatter Plot Representation

```r
plot(cars, col = "blue", pch = 16, cex = 0.75,
     main = "Relationship between speed and stopping distance for 50 cars",
     xlab = "Speed in mph", ylab = "Stopping distance in feet")
```



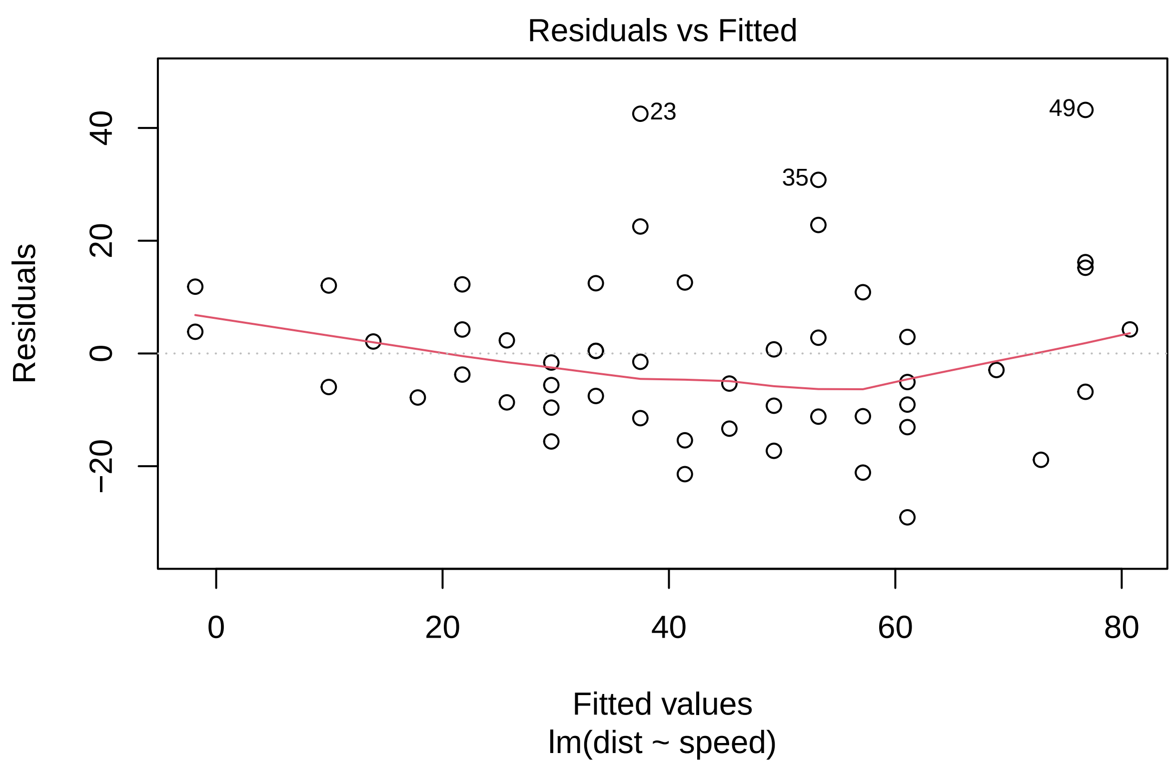
Step 3: Parameter Estimation and Regression Line

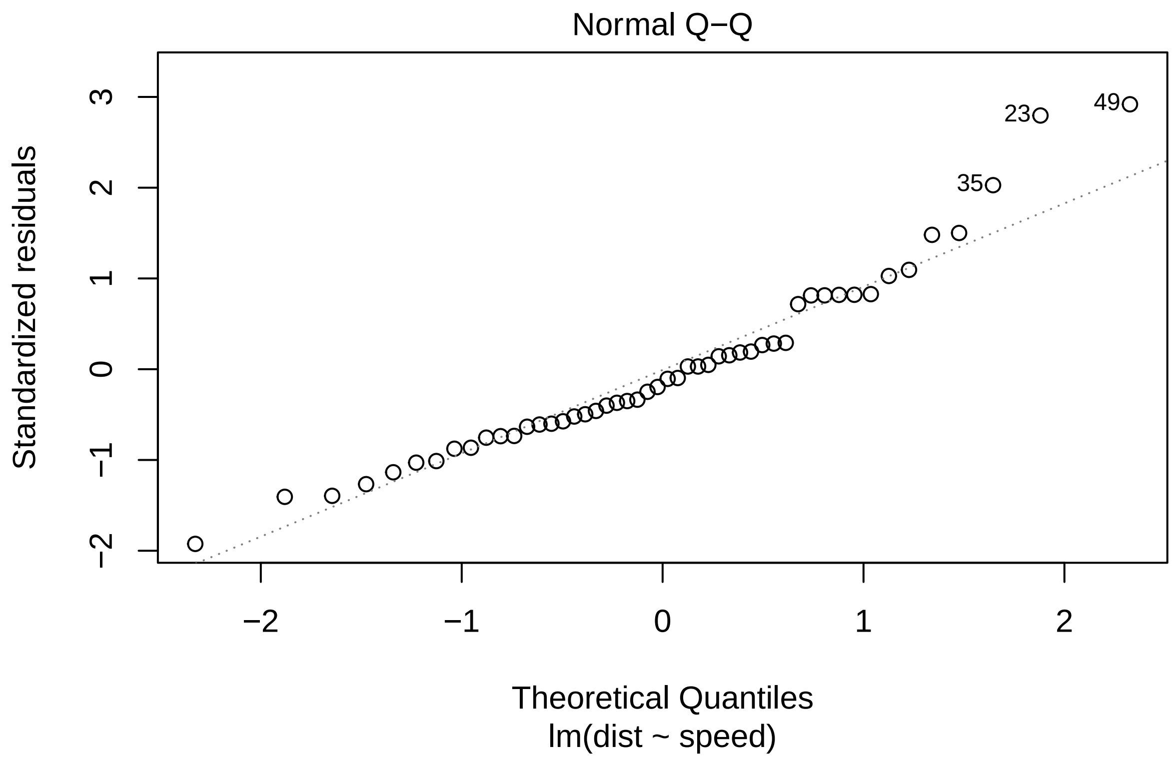
```r
m0 <- lm(formula = dist ~ speed, data = cars)  # Fitting a linear regression model
summary(m0)                                    # Summary of the regression model
```

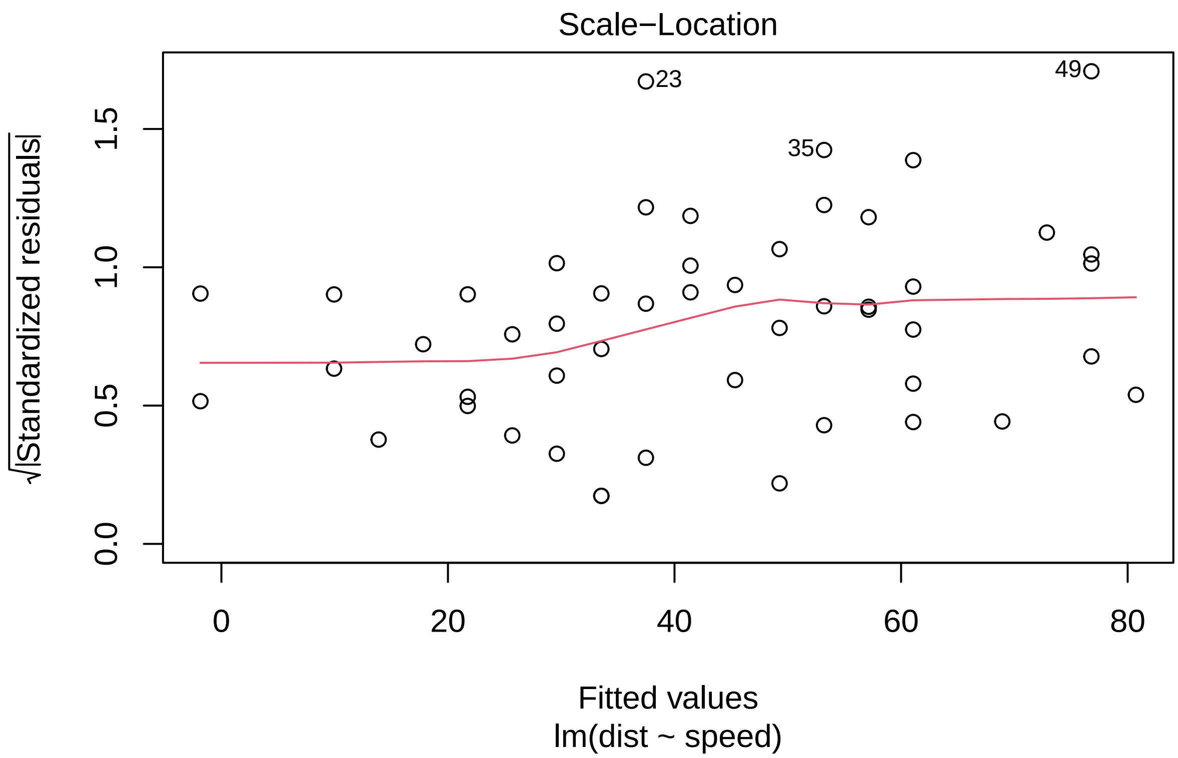
```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

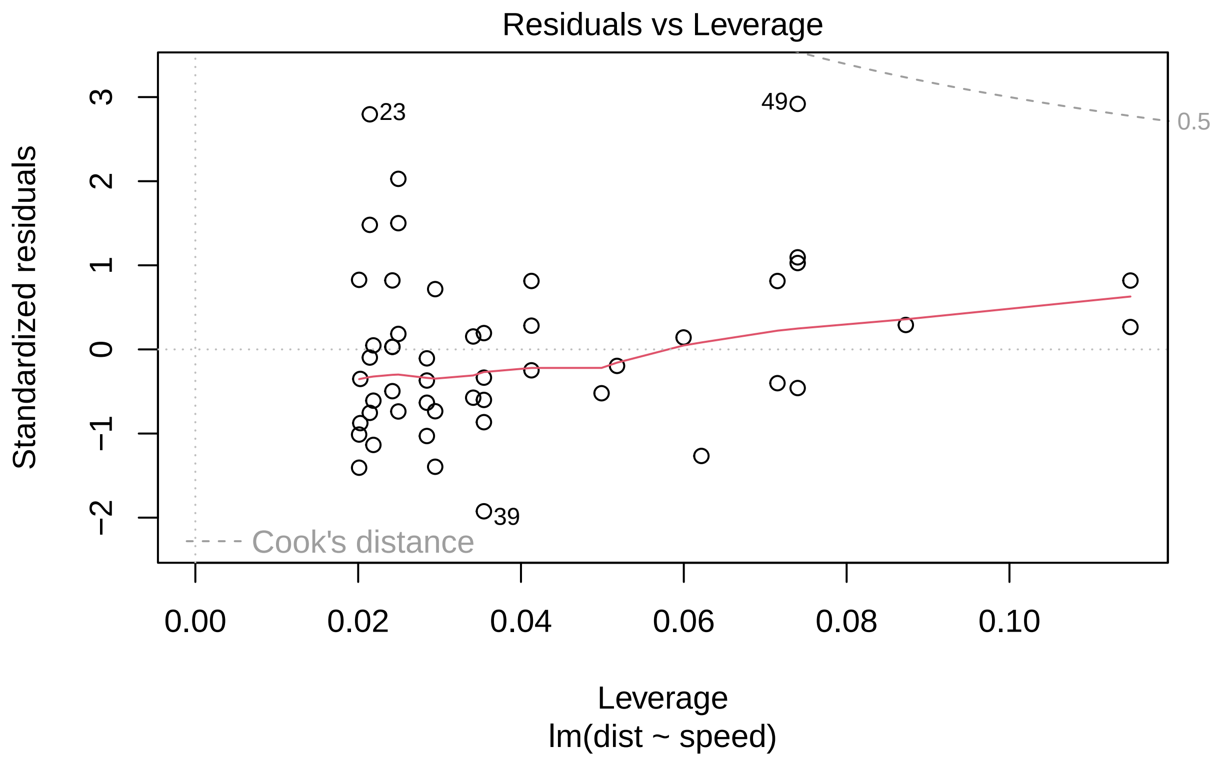
```r
plot(m0)  # Drawing the regression line
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(dist ~ speed)
−2
−1
0
1
2
3
23
49
35



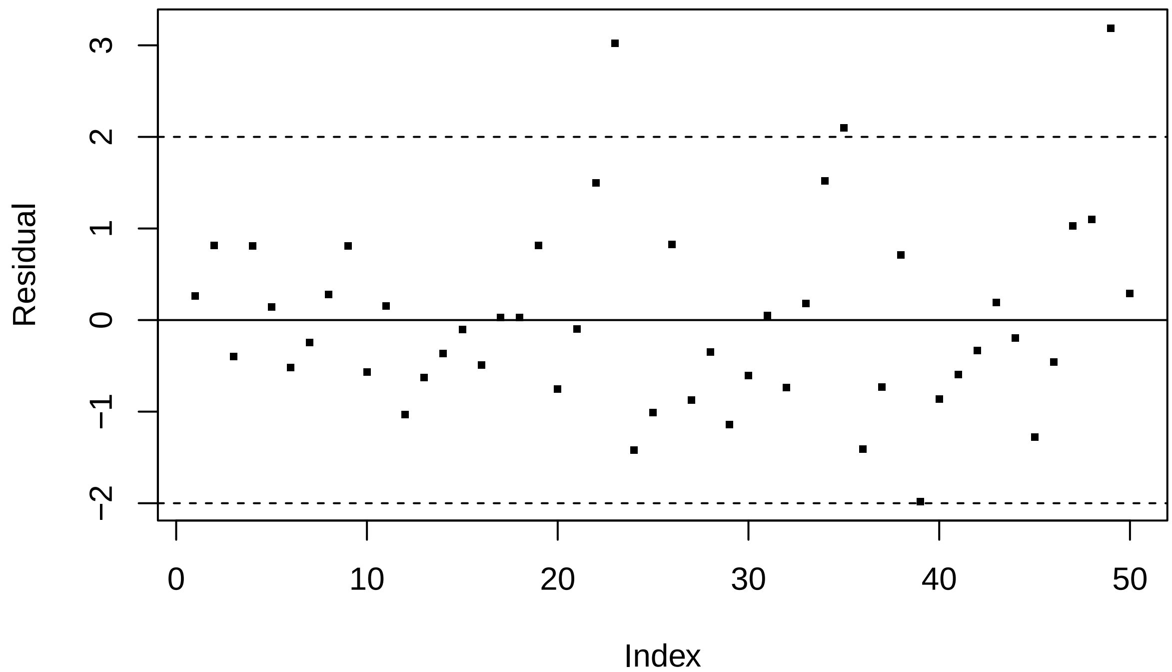
Scale−Location
23
49
35
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
20
40
60
80
Fitted values
lm(dist ~ speed)



Step 4: Residual Analysis

```
res.m0 <- rstudent(m0)  # Calculate studentized residuals

# Plotting residuals and lines for reference
plot(res.m0, pch = 15, cex = 0.5, ylab = "Residual")
abline(h = c(-2, 0, 2), lty = c(2, 1, 2))
```



Step 5: Prediction of a New Value

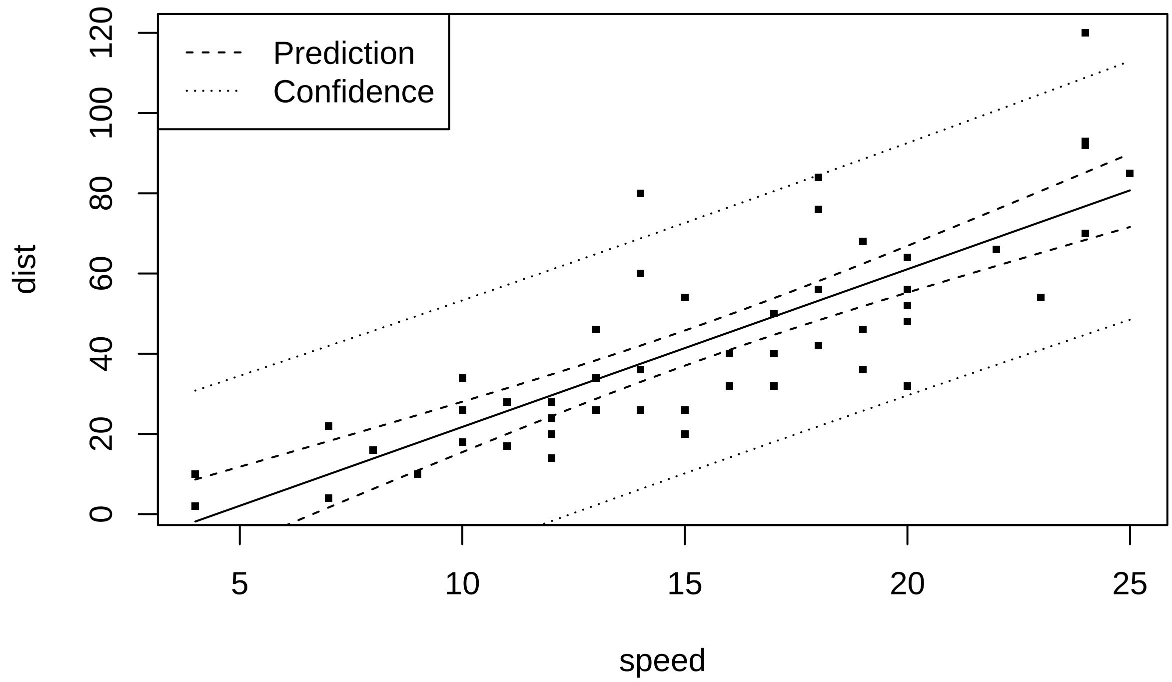
```
xnew <- 16
xnew <- as.data.frame(xnew)
colnames(xnew) <- "speed"
predict(m0, xnew, interval = "pred")  # Predicting a new value
```

```
##        fit      lwr     upr
## 1 45.33945 14.10499 76.5739
```

Step 6: Confidence and Prediction Intervals

```
gridx <- data.frame(speed = seq(min(cars$speed), max(cars$speed), length = 100))
CIline <- predict(m0, new = gridx, interval = "conf", level = 0.95)  # Confidence interval
CIpred <- predict(m0, new = gridx, interval = "pred", level = 0.95)  # Prediction interval

# Plotting regression line with confidence and prediction intervals
plot(dist ~ speed, data = cars, pch = 15, cex = 0.5)
matlines(gridx, cbind(CIline, CIpred[, -1]), lty = c(1, 2, 2, 3, 3), col = 1)
legend("topleft", lty = 2:3, c("Prediction", "Confidence"))
```



Multiple Linear Regression

```
ozone <- read.table("ozone.txt", header = TRUE)
```

Step 1: Read the Data

```
dim(ozone)        # Display the dimensions of the dataset
```

Step 2: Variable Representation and Summary

```
## [1] 112  13
```

```
ozone.m <- ozone[, 1:11]  # Selecting columns for analysis
names(ozone.m)     # Display the variable names
```

```
##  [1] "maxO3"  "T9"     "T12"    "T15"    "Ne9"    "Ne12"   "Ne15"
##  [8] "Wx9"    "Wx12"   "Wx15"   "maxO3v"
```

```r
summary(ozone.m)  # Summary statistics of the selected variables
```

```
##      maxO3              T9             T12             T15       
##  Min.   : 42.00   Min.   :11.30   Min.   :14.00   Min.   :14.90  
##  1st Qu.: 70.75   1st Qu.:16.20   1st Qu.:18.60   1st Qu.:19.27  
##  Median : 81.50   Median :17.80   Median :20.55   Median :22.05  
##  Mean   : 90.30   Mean   :18.36   Mean   :21.53   Mean   :22.63  
##  3rd Qu.:106.00   3rd Qu.:19.93   3rd Qu.:23.55   3rd Qu.:25.40  
##  Max.   :166.00   Max.   :27.00   Max.   :33.50   Max.   :35.50  
##       Ne9             Ne12            Ne15           Wx9         
##  Min.   :0.000   Min.   :0.000   Min.   :0.00   Min.   :-7.8785  
##  1st Qu.:3.000   1st Qu.:4.000   1st Qu.:3.00   1st Qu.:-3.2765  
##  Median :6.000   Median :5.000   Median :5.00   Median :-0.8660  
##  Mean   :4.929   Mean   :5.018   Mean   :4.83   Mean   :-1.2143  
##  3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.00   3rd Qu.: 0.6946  
##  Max.   :8.000   Max.   :8.000   Max.   :8.00   Max.   : 5.1962  
##       Wx12             Wx15            maxO3v      
##  Min.   :-7.878   Min.   :-9.000   Min.   : 42.00  
##  1st Qu.:-3.565   1st Qu.:-3.939   1st Qu.: 71.00  
##  Median :-1.879   Median :-1.550   Median : 82.50  
##  Mean   :-1.611   Mean   :-1.691   Mean   : 90.57  
##  3rd Qu.: 0.000   3rd Qu.: 0.000   3rd Qu.:106.00  
##  Max.   : 6.578   Max.   : 5.000   Max.   :166.00
```

```r
reg.mul <- lm(maxO3 ~ ., data = ozone.m)  # Fitting multiple linear regression model
summary(reg.mul)  # Summary of the regression model
```

Step 3: Parameter Estimation

```
## 
## Call:
## lm(formula = maxO3 ~ ., data = ozone.m)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -53.566  -8.727  -0.403   7.599  39.458 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 12.24442   13.47190   0.909   0.3656    
## T9          -0.01901    1.12515  -0.017   0.9866    
## T12          2.22115    1.43294   1.550   0.1243    
## T15          0.55853    1.14464   0.488   0.6266    
## Ne9         -2.18909    0.93824  -2.333   0.0216 *  
## Ne12        -0.42102    1.36766  -0.308   0.7588    
## Ne15         0.18373    1.00279   0.183   0.8550    
## Wx9          0.94791    0.91228   1.039   0.3013    
## Wx12         0.03120    1.05523   0.030   0.9765    
## Wx15         0.41859    0.91568   0.457   0.6486    
## maxO3v       0.35198    0.06289   5.597 1.88e-07 ***
```

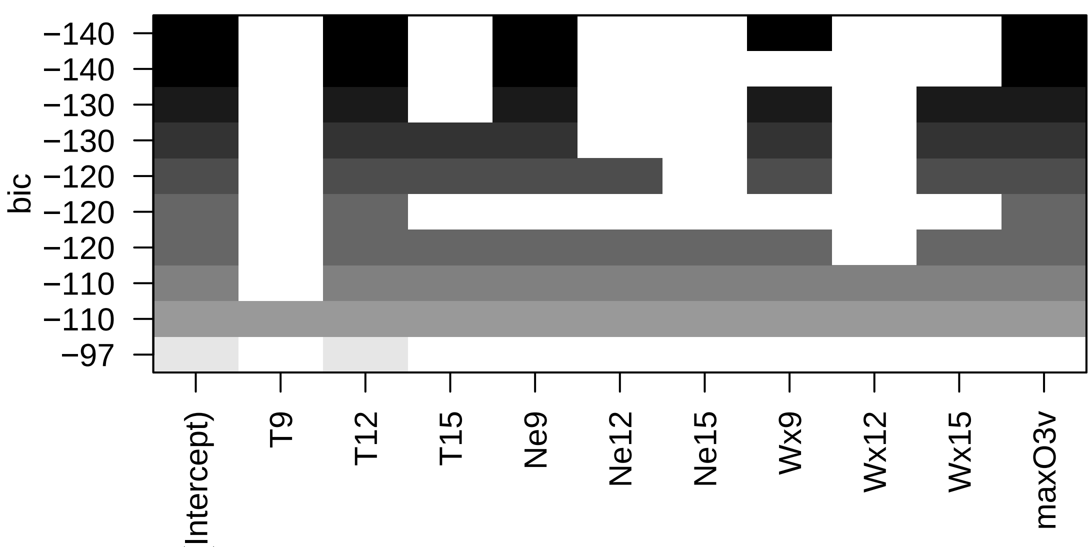
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.36 on 101 degrees of freedom
## Multiple R-squared:  0.7638, Adjusted R-squared:  0.7405
## F-statistic: 32.67 on 10 and 101 DF,  p-value: < 2.2e-16
```

```r
library("leaps")  # Load the leaps package for variable selection
```

Step 4: Variable Selection

```
## Warning: package 'leaps' was built under R version 4.2.2
```

```r
choice <- regsubsets(maxO3 ~ ., data = ozone.m, nbest = 1, nvmax = 11)  # Choose variables
plot(choice, scale = "bic")  # Plot criteria for variable selection
```



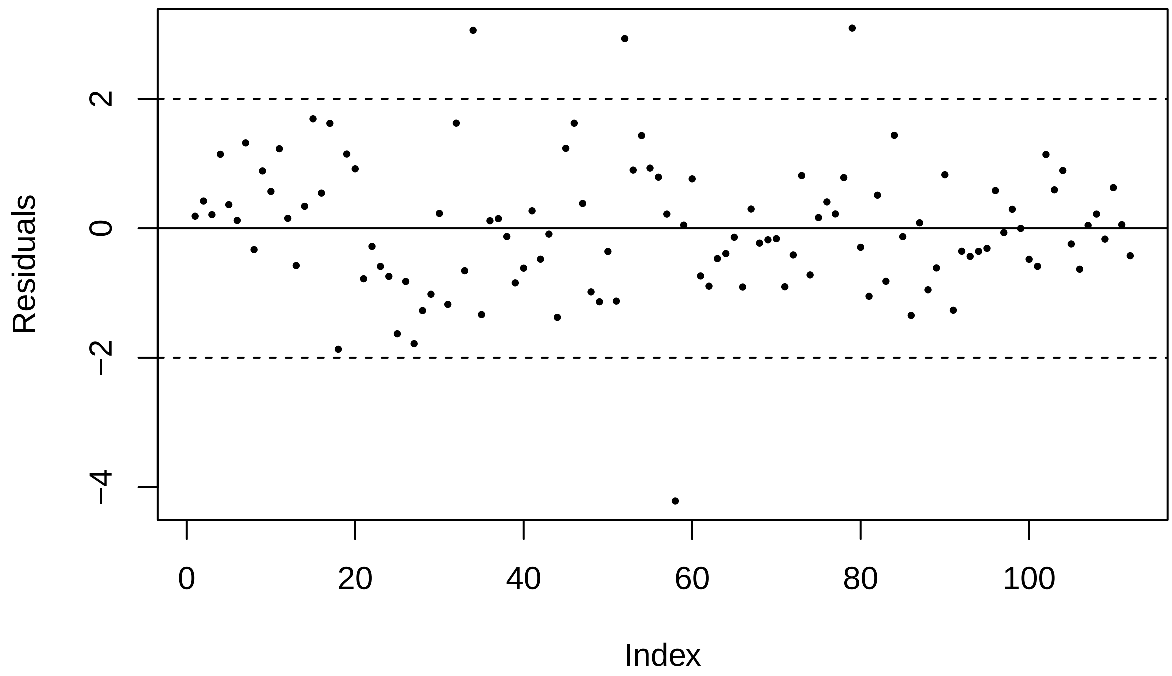

```r
# Select the best model based on BIC criteria
best_model <- summary(choice)$which[which.min(summary(choice)$bic), ]
```

```r
final.reg <- lm(maxO3 ~ T12 + Ne9 + Wx9 + maxO3v, data = ozone.m)  # Creating the final model
summary(final.reg)  # Summary of the final model
```

Step 5: Final Model and Residual Analysis

```
##
## Call:
## lm(formula = maxO3 ~ T12 + Ne9 + Wx9 + maxO3v, data = ozone.m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.396  -8.377  -1.086   7.951  40.933
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.63131   11.00088   1.148 0.253443
## T12          2.76409    0.47450   5.825 6.07e-08 ***
## Ne9         -2.51540    0.67585  -3.722 0.000317 ***
## Wx9          1.29286    0.60218   2.147 0.034055 *
## maxO3v       0.35483    0.05789   6.130 1.50e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14 on 107 degrees of freedom
## Multiple R-squared:  0.7622, Adjusted R-squared:  0.7533
## F-statistic: 85.75 on 4 and 107 DF,  p-value: < 2.2e-16
```

```r
res.m <- rstudent(final.reg)  # Calculate studentized residuals
plot(res.m, pch = 16, cex = 0.5, ylab = "Residuals")  # Plot residuals
abline(h = c(-2, 0, 2), lty = c(2, 1, 2))  # Add lines for reference
```




```
# Creating a matrix with new values
xnew <- matrix(c(19, 8, 2.05, 70), nrow = 1)
colnames(xnew) <- c("T12", "Ne9", "Wx9", "maxO3v")
xnew <- as.data.frame(xnew)

# Predicting new values and intervals
predict(final.reg, xnew, interval = "pred")
```

Step 6: Predict New Values

```
##        fit      lwr      upr
## 1 72.51437 43.80638 101.2224
```

```
# The predicted value is 72.5, and the 95% prediction interval is [43.8, 101.2]
```