Machin Learning Methods with R

Least Squares (LS)

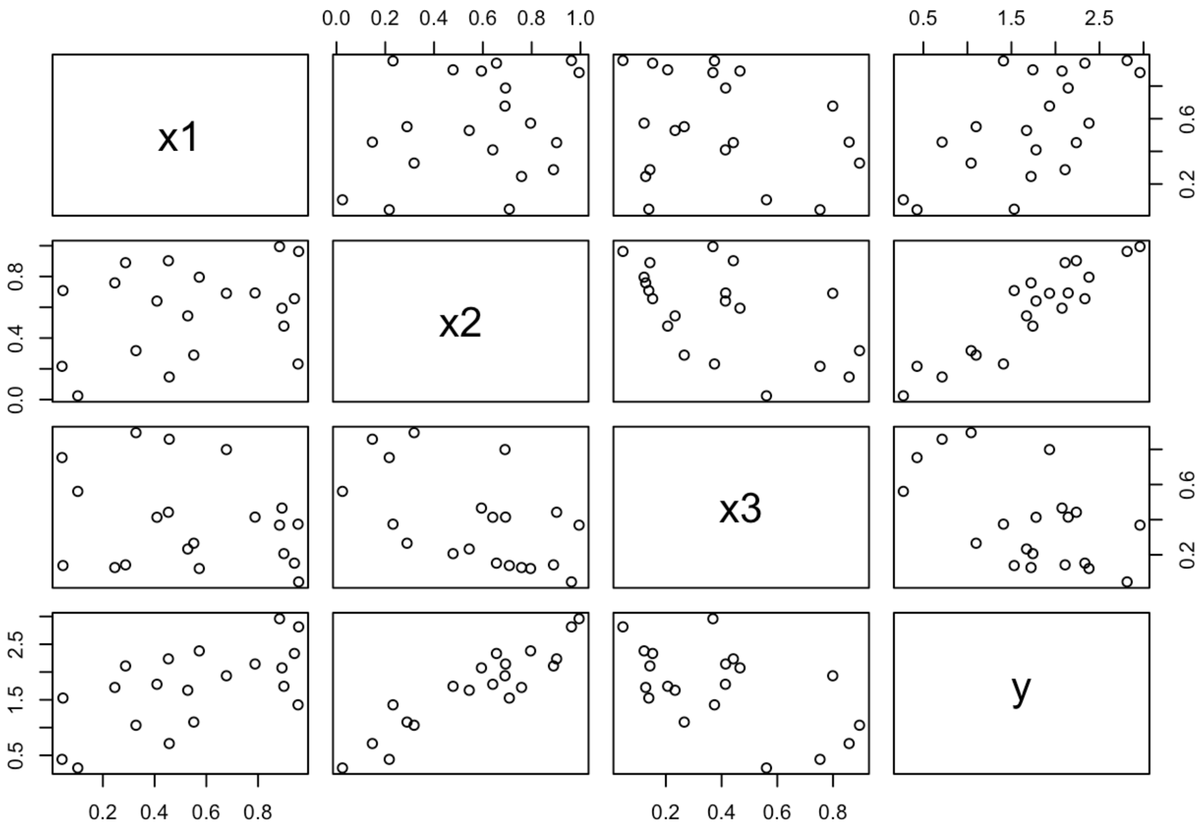
Generation of the Data

Here, we are generating synthetic data:

- We initialize a reproducible random generation using set.seed(123).
- x is a matrix of 60 random numbers uniformly distributed between 0 and 1. These numbers are reshaped into a 20x3 matrix (20 rows and 3 columns).
- We then calculate y by multiplying matrix x with a vector of coefficients and adding some normally distributed noise.
- Column names for the matrix x are set as “x1”, “x2”, and “x3”.
- The matrix is then combined with vector y into a dataframe d.
- A scatterplot matrix of d is plotted to visualize relationships between variables.

```
set.seed(123)
x <- matrix(runif(60), ncol = 3)
y <- x %*% c(1, 2, 0) + 0.1 * rnorm(20)
colnames(x) <- paste("x", 1:3, sep = "")
```

```
d <- data.frame(x, y = y)
plot(d)
```



Train

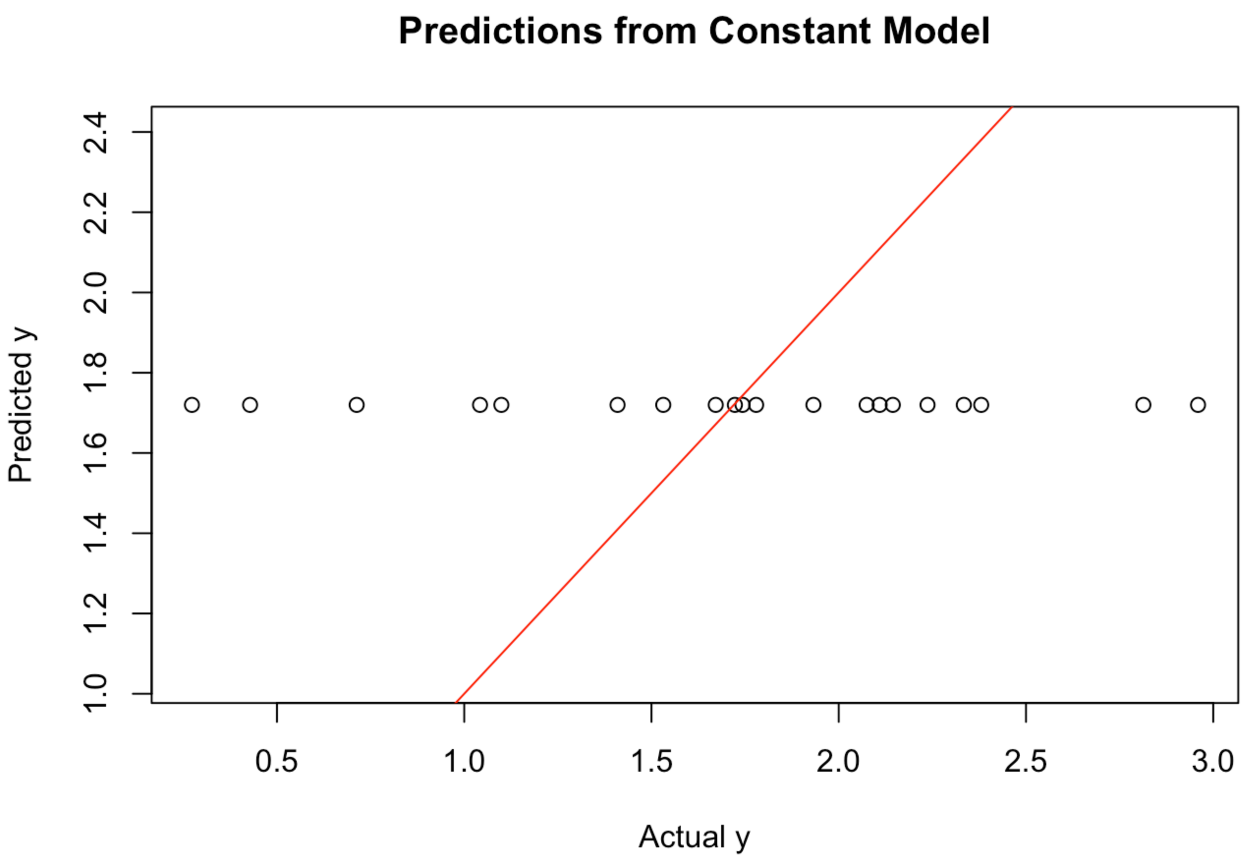
Here, we are training several linear regression models:

1. lm0: A constant model where the only predictor is the intercept.
2. lm1: A simple linear regression model with x1 as the predictor.
3. lm3: A multiple linear regression model using all three predictors (x1, x2, and x3). For each of these models, predictions are plotted against the actual y values. The red line represents a perfect prediction line where actual equals predicted.

```
lm0 <- lm(y~1, data = d)
lm0
```

```
##
## Call:
## lm(formula = y ~ 1, data = d)
##
## Coefficients:
## (Intercept)
##        1.72
```

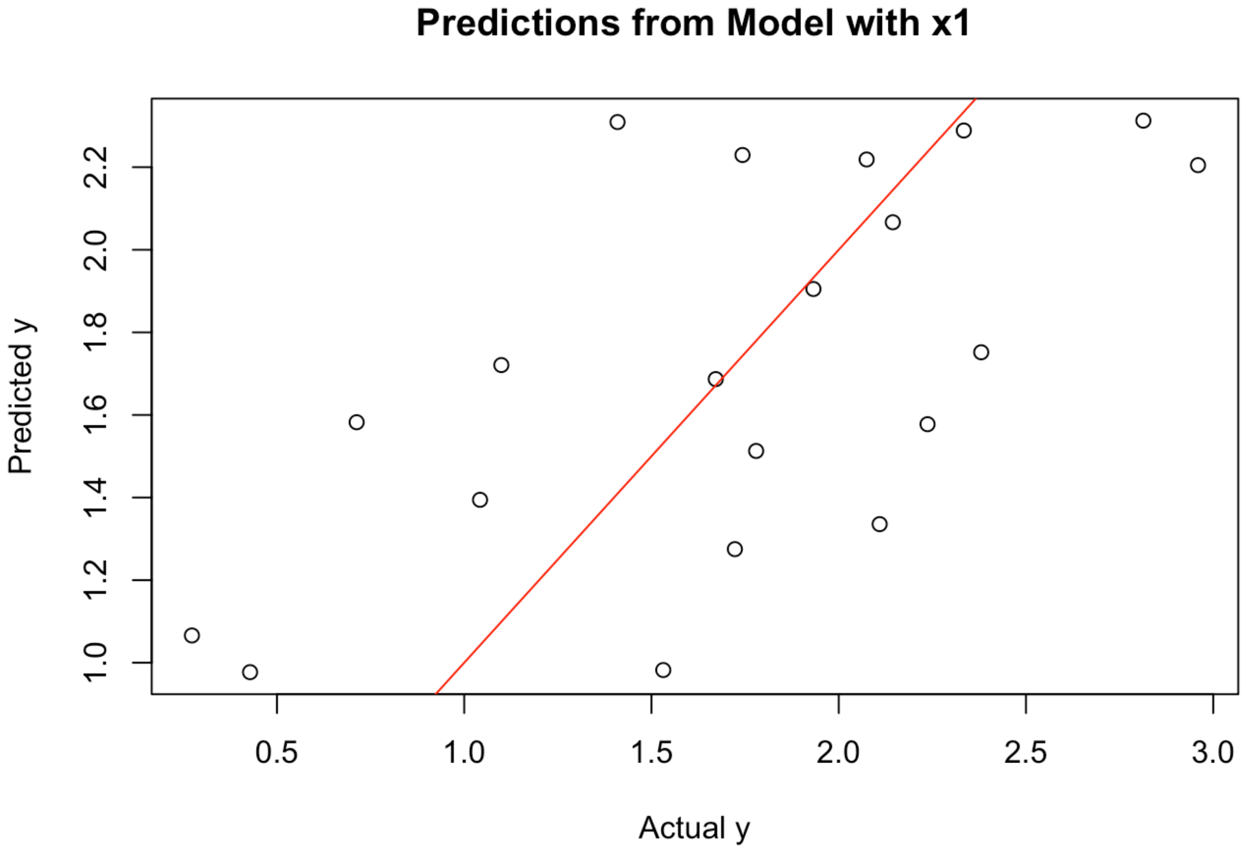
```
plot(d$y, predict(lm0), xlab="Actual y", ylab="Predicted y",
abline(a=0, b=1, col="red") # Line of perfect prediction
```




```
lm1 <- lm(y~x1, data = d)
lm1
```

```
##
## Call:
## lm(formula = y ~ x1, data = d)
##
## Coefficients:
## (Intercept)           x1
##      0.9157       1.4600
```

```
plot(d$y, predict(lm1), xlab="Actual y", ylab="Predicted y",
abline(a=0, b=1, col="red") # Line of perfect prediction
```

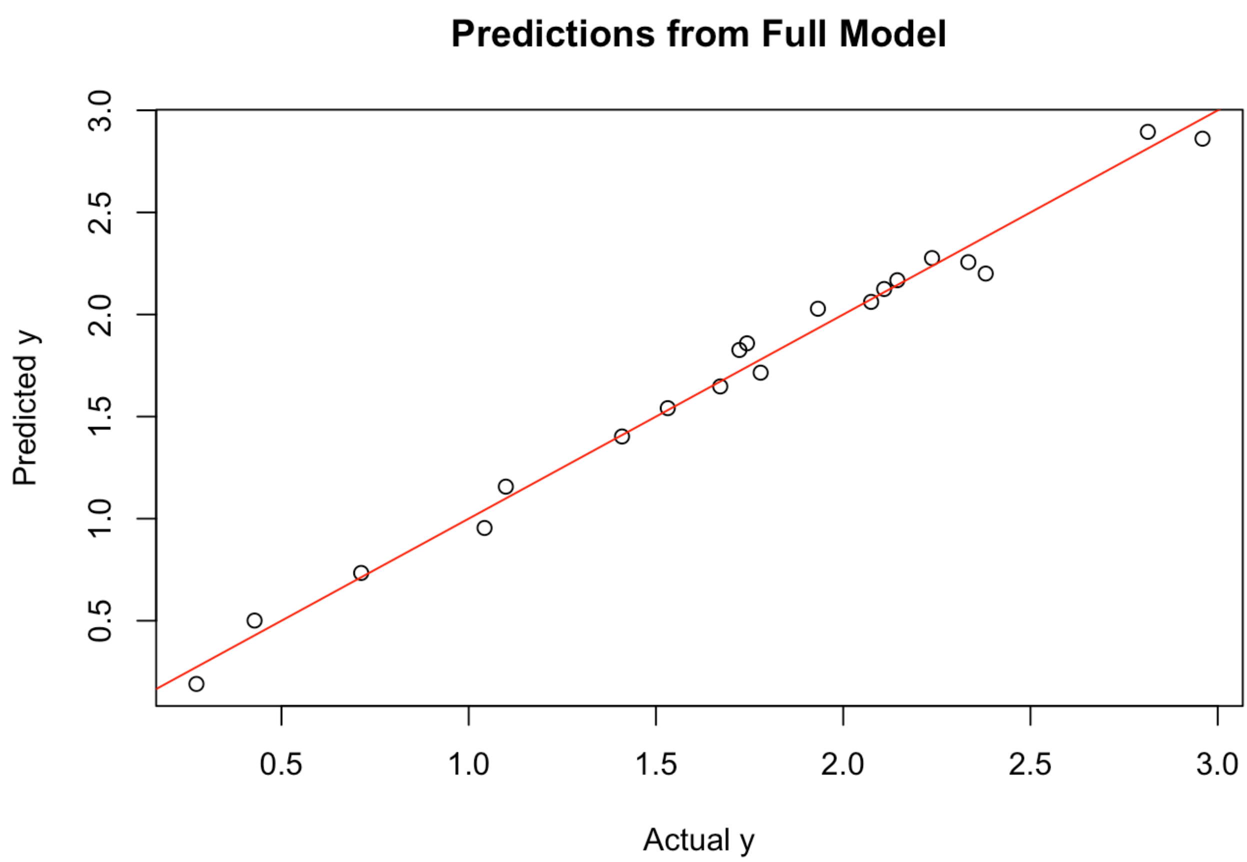



```
lm3 <- lm(y~x1+x2+x3, data = d)
lm3
```

```
##
```

```
## Call:
## lm(formula = y ~ x1 + x2 + x3, data = d)
##
## Coefficients:
## (Intercept)           x1           x2           x3
##     0.09585      0.91834      1.99804     -0.08761
```

```
plot(d$y, predict(lm3), xlab="Actual y", ylab="Predicted y",
abline(a=0, b=1, col="red") # Line of perfect prediction
```




```
summary(lm3)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3, data = d)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11566 -0.06133 -0.01260  0.06785  0.18004
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09585    0.08200   1.169    0.260
## x1           0.91834    0.06623  13.867 2.47e-10 ***
## x2           1.99804    0.08453  23.637 7.18e-14 ***
## x3          -0.08761    0.09060  -0.967    0.348
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
##
## Residual standard error: 0.08621 on 16 degrees of freedom
## Multiple R-squared:  0.9882, Adjusted R-squared:  0.986
## F-statistic: 446.5 on 3 and 16 DF,  p-value: 1.251e-15
```

Model Comparison with anova()

The anova() function is employed to compare the models. First, the analysis of variance table for lm3 is displayed. After that, a comparison of all four models (lm0, lm1, lm2, and lm3) is done.

```
print(anova(lm3))
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## x1         1 3.9799  3.9799 535.4639 9.991e-14 ***
## x2         1 5.9693  5.9693 803.1073 4.199e-15 ***
```

```
## x3          1 0.0070  0.0070   0.9351    0.3479
## Residuals 16 0.1189  0.0074
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

```
lm2 <- lm(y~x1+x2, data=d)
print(anova(lm0, lm1, lm2, lm3))
```

```
## Analysis of Variance Table
##
## Model 1: y ~ 1
## Model 2: y ~ x1
## Model 3: y ~ x1 + x2
## Model 4: y ~ x1 + x2 + x3
##   Res.Df     RSS Df Sum of Sq        F    Pr(>F)
## 1     19 10.0751
## 2     18  6.0951  1    3.9799 535.4639 9.991e-14 ***
## 3     17  0.1259  1    5.9693 803.1073 4.199e-15 ***
## 4     16  0.1189  1    0.0070   0.9351    0.3479
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

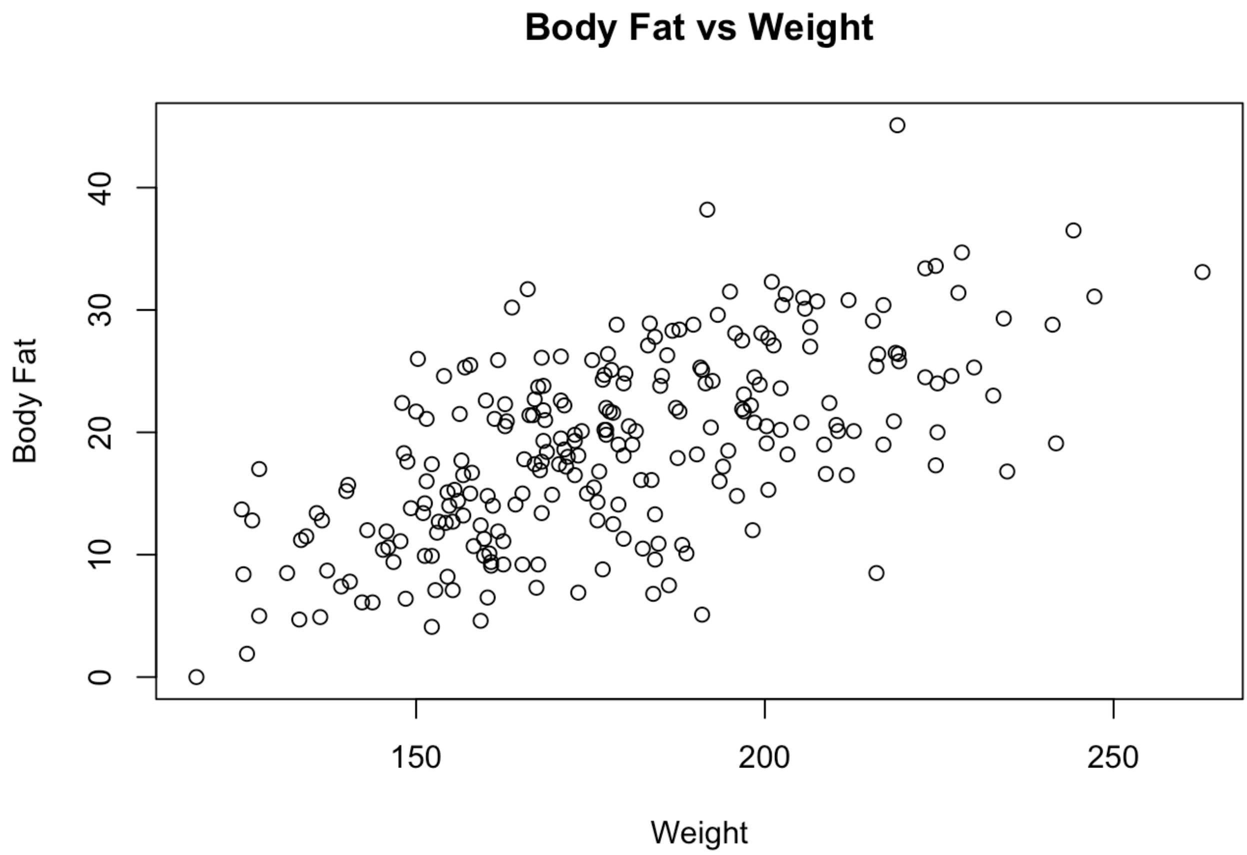
Body Fat Data Analysis

We load the fat dataset from the UsingR package. Some data points and variables deemed as anomalies or unused are removed. A scatter plot is then generated to visualize the relationship between weight and body fat.

```
library("UsingR")
```

```
## Loading required package: MASS

## Loading required package: HistData

## Loading required package: Hmisc

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
data(fat)
fat <- fat[-c(31,39,42,86), -c(1,3,4,9)]# omitting strange va
attach(fat)

plot(fat$weight, fat$body.fat, xlab="Weight", ylab="Body Fat"
```



Linear Model for Body Fat Data

A linear regression model model.lm is built on a subset (2/3) of the data. The rest 1/3 is reserved for testing. The summary of this model is displayed.

```
set.seed(123)
n <- nrow(fat)
train <- sample(1:n,round(n*2/3))
test <- (1:n)[-train]
model.lm <- lm(body.fat~., data = fat, subset=train)
summary(model.lm)
```

```
##
## Call:
## lm(formula = body.fat ~ ., data = fat, subset = train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4688 -2.7421 -0.1162  2.7285  9.0751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -41.81344   52.38032  -0.798   0.4260
## age           0.08386    0.03911   2.144   0.0336 *
## weight       -0.12932    0.14604  -0.886   0.3773
## height        0.56531    0.67794   0.834   0.4057
## BMI           1.25203    0.90522   1.383   0.1687
## neck         -0.45496    0.28652  -1.588   0.1144
## chest        -0.19395    0.13505  -1.436   0.1531
## abdomen       0.79287    0.10772   7.360 1.12e-11 ***
## hip          -0.19868    0.17020  -1.167   0.2449
## thigh         0.08344    0.17164   0.486   0.6276
## knee          0.05469    0.29236   0.187   0.8519
## ankle        -0.21770    0.42515  -0.512   0.6094
## bicep         0.19942    0.19193   1.039   0.3005
## forearm       0.31561    0.24968   1.264   0.2082
## wrist        -1.40770    0.66446  -2.119   0.0358 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
##
## Residual standard error: 4.017 on 150 degrees of freedom
## Multiple R-squared:  0.7649, Adjusted R-squared:  0.743
## F-statistic: 34.86 on 14 and 150 DF,  p-value: < 2.2e-16
```

Model Evaluation

The performance of model.lm is evaluated on the test data in terms of R-squared and Mean Squared Error (MSE). Predicted body fat values are then plotted against actual values to visualize the model's predictions.

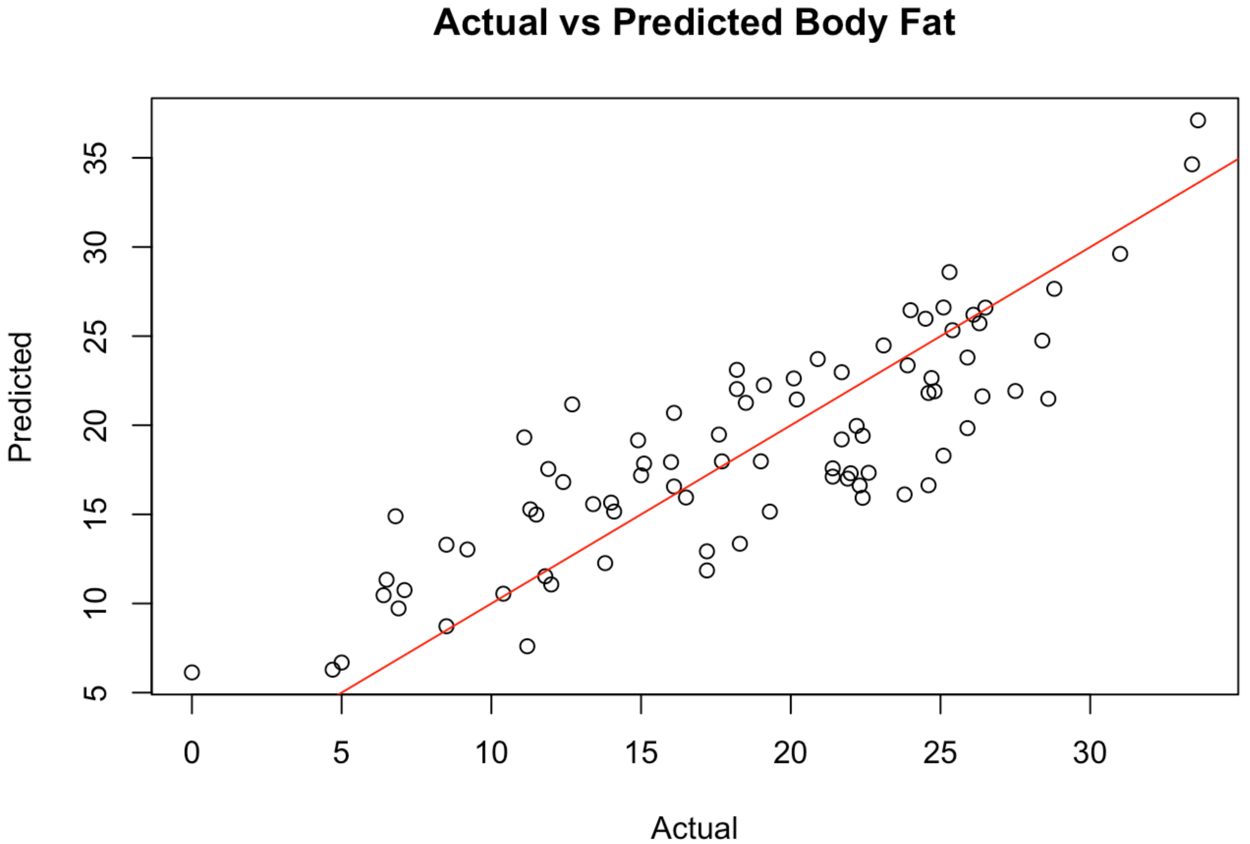
```
pred.lm <- predict(model.lm,newdata = fat[test,])
cor(fat[test,"body.fat"],pred.lm)^2 # R^2 for test data
```

```
## [1] 0.705793
```

```
mean((fat[test,"body.fat"]-pred.lm)^2) # MSE_test
```

```
## [1] 15.22816
```

```
plot(fat[test,"body.fat"], pred.lm, xlab="Actual", ylab="Pred
abline(a=0, b=1, col="red") # Line of perfect prediction
```



Automatic model search with step()

The step() function is an automated approach to select the best model by adding or dropping predictors. This optimized model's predictions are evaluated on the test set.

```
model.lmstep <- step(model.lm)
```

```
## Start:  AIC=473.19
## body.fat ~ age + weight + height + BMI + neck + chest + ab
##     hip + thigh + knee + ankle + bicep + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - knee     1      0.56 2421.5 471.22
## - thigh    1      3.81 2424.8 471.45
## - ankle    1      4.23 2425.2 471.47
## - height   1     11.22 2432.2 471.95
## - weight   1     12.66 2433.6 472.05
## - bicep    1     17.43 2438.4 472.37
## - hip      1     21.99 2442.9 472.68
## - forearm  1     25.79 2446.7 472.93
## <none>                 2421.0 473.19
## - BMI      1     30.88 2451.8 473.28
## - chest    1     33.29 2454.2 473.44
## - neck     1     40.69 2461.6 473.94
## - wrist    1     72.44 2493.4 476.05
## - age      1     74.19 2495.1 476.17
## - abdomen  1    874.35 3295.3 522.06
##
## Step:  AIC=471.22
## body.fat ~ age + weight + height + BMI + neck + chest + ab
##     hip + thigh + ankle + bicep + forearm + wrist
##
```

```
##           Df Sum of Sq    RSS    AIC
## - ankle    1      3.73 2425.2 469.48
## - thigh    1      5.06 2426.6 469.57
## - height   1     11.43 2432.9 470.00
## - weight   1     12.25 2433.8 470.06
## - bicep    1     17.57 2439.1 470.42
## - hip      1     21.87 2443.4 470.71
## - forearm  1     27.22 2448.7 471.07
## <none>                 2421.5 471.22
## - BMI      1     30.45 2452.0 471.29
## - chest    1     34.01 2455.5 471.53
## - neck     1     41.76 2463.3 472.05
## - wrist    1     73.29 2494.8 474.14
## - age      1     90.50 2512.0 475.28
## - abdomen  1    882.77 3304.3 520.51
##
## Step:  AIC=469.48
## body.fat ~ age + weight + height + BMI + neck + chest + ab
##     hip + thigh + bicep + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - thigh    1      4.01 2429.3 467.75
## - height   1     10.70 2435.9 468.20
## - weight   1     13.71 2439.0 468.41
## - hip      1     20.12 2445.4 468.84
## - bicep    1     20.63 2445.9 468.88
## - forearm  1     26.75 2452.0 469.29
## - BMI      1     28.64 2453.9 469.42
## <none>                 2425.2 469.48
## - chest    1     31.73 2457.0 469.62
## - neck     1     38.07 2463.3 470.05
## - age      1     94.06 2519.3 473.76
## - wrist    1    102.60 2527.8 474.31
## - abdomen  1    911.90 3337.1 520.14
```

```
##
## Step:  AIC=467.75
## body.fat ~ age + weight + height + BMI + neck + chest + ab
##     hip + bicep + forearm + wrist
##
##            Df Sum of Sq    RSS    AIC
## - height   1      8.35 2437.6 466.32
## - weight   1     10.87 2440.1 466.49
## - hip      1     16.26 2445.5 466.85
## - bicep    1     25.80 2455.1 467.49
## - forearm  1     26.03 2455.3 467.51
## - BMI      1     26.11 2455.4 467.51
## <none>                 2429.3 467.75
## - chest    1     36.37 2465.6 468.20
## - neck     1     37.64 2466.9 468.29
## - age      1     90.84 2520.1 471.81
## - wrist    1    104.26 2533.5 472.68
## - abdomen  1    909.34 3338.6 518.21
##
## Step:  AIC=466.32
## body.fat ~ age + weight + BMI + neck + chest + abdomen + h
##     bicep + forearm + wrist
##
##            Df Sum of Sq    RSS    AIC
## - weight   1      2.69 2440.3 464.50
## - hip      1     16.60 2454.2 465.44
## - bicep    1     22.51 2460.1 465.83
## - forearm  1     26.05 2463.7 466.07
## <none>                 2437.6 466.32
## - chest    1     35.07 2472.7 466.67
## - neck     1     42.31 2479.9 467.16
## - BMI      1     48.83 2486.4 467.59
## - age      1     90.86 2528.5 470.35
## - wrist    1    108.74 2546.3 471.52
```

```
## - abdomen  1    901.44 3339.0 516.24
##
## Step:  AIC=464.5
## body.fat ~ age + BMI + neck + chest + abdomen + hip + bice
##     forearm + wrist
##
##            Df Sum of Sq    RSS    AIC
## - bicep     1     20.42 2460.7 463.87
## - forearm   1     25.82 2466.1 464.23
## <none>                  2440.3 464.50
## - hip       1     38.79 2479.1 465.10
## - neck      1     51.45 2491.7 465.94
## - chest     1     60.45 2500.7 466.54
## - BMI       1     64.43 2504.7 466.80
## - age       1    126.05 2566.3 470.81
## - wrist     1    151.98 2592.3 472.47
## - abdomen   1    940.63 3380.9 516.29
##
## Step:  AIC=463.87
## body.fat ~ age + BMI + neck + chest + abdomen + hip + fore
##     wrist
##
##            Df Sum of Sq    RSS    AIC
## <none>                  2460.7 463.87
## - hip       1     32.11 2492.8 464.01
## - forearm   1     40.85 2501.6 464.59
## - neck      1     41.55 2502.3 464.64
## - chest     1     61.19 2521.9 465.93
## - BMI       1     78.43 2539.1 467.05
## - age       1    118.23 2578.9 469.62
## - wrist     1    147.95 2608.7 471.51
## - abdomen   1    935.00 3395.7 515.01
```

```
pred.lmstep <- predict(model.lmstep,newdata = fat[test,])
```

```
cor(fat[test,"body.fat"],pred.lmstep)^2 # R^2 for test data
```

```
## [1] 0.70701
```

```
mean((fat[test,"body.fat"]-pred.lmstep)^2) # MSE_test
```

```
## [1] 15.16469
```

Best subset regression with Leaps and Bound algorithm

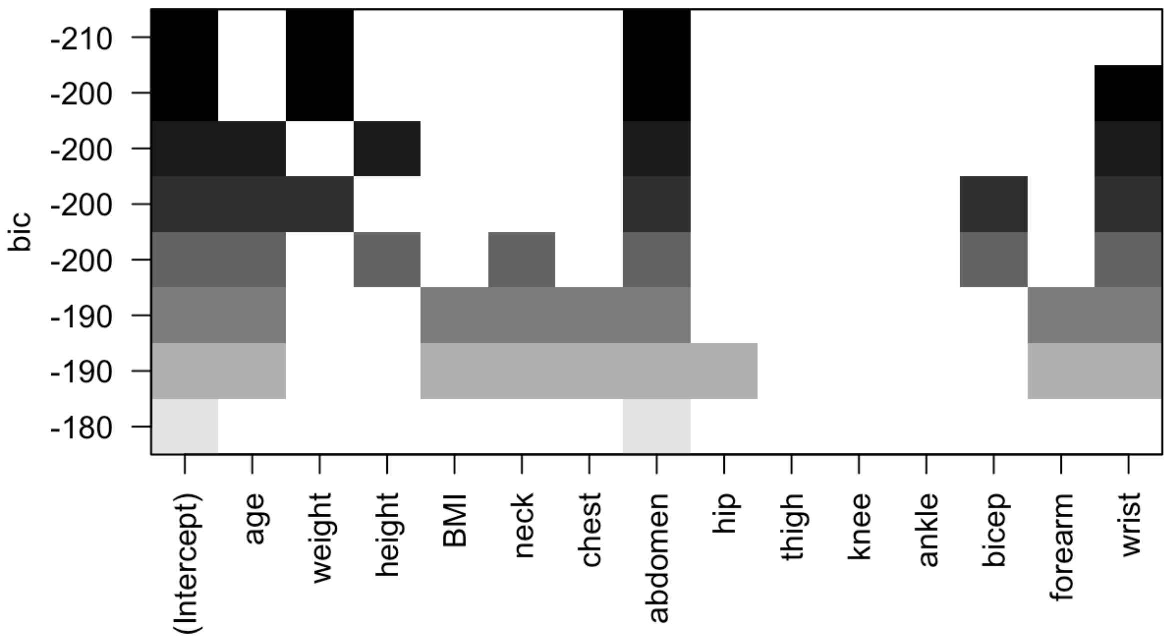
The regsubsets() function from the leaps package is used for best subset regression. This helps in identifying the best model using a specific number of predictors. The best model found here uses weight and abdomen as predictors. Its performance on the test set is then evaluated.

```
library(leaps)
lm.regsubset <- regsubsets(body.fat~., data=fat, nbest = 1, s
summary(lm.regsubset)
```

```
## Subset selection object
## Call: regsubsets.formula(body.fat ~ ., data = fat, nbest =
## 14 Variables  (and intercept)
##         Forced in Forced out
## age         FALSE      FALSE
## weight      FALSE      FALSE
## height      FALSE      FALSE
## BMI         FALSE      FALSE
```

```
## neck           FALSE      FALSE
## chest          FALSE      FALSE
## abdomen        FALSE      FALSE
## hip            FALSE      FALSE
## thigh          FALSE      FALSE
## knee           FALSE      FALSE
## ankle          FALSE      FALSE
## bicep          FALSE      FALSE
## forearm        FALSE      FALSE
## wrist          FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          age weight height BMI neck chest abdomen hip thig
## 1  ( 1 ) " " " "    " "    " " " "  " "   "*"     " " " "
## 2  ( 1 ) " " "*"    " "    " " " "  " "   "*"     " " " "
## 3  ( 1 ) " " "*"    " "    " " " "  " "   "*"     " " " "
## 4  ( 1 ) "*" " "    "*"    " " " "  " "   "*"     " " " "
## 5  ( 1 ) "*" "*"    " "    " " " "  " "   "*"     " " " "
## 6  ( 1 ) "*" " "    "*"    " " "*"  " "   "*"     " " " "
## 7  ( 1 ) "*" " "    " "    "*" "*"  "*"   "*"     " " " "
## 8  ( 1 ) "*" " "    " "    "*" "*"  "*"   "*"     "*" " "
##          forearm wrist
## 1  ( 1 ) " "     " "
## 2  ( 1 ) " "     " "
## 3  ( 1 ) " "     "*"
## 4  ( 1 ) " "     "*"
## 5  ( 1 ) " "     "*"
## 6  ( 1 ) " "     "*"
## 7  ( 1 ) "*"     "*"
## 8  ( 1 ) "*"     "*"
```

```
plot(lm.regsubset)
```




```
modregsubset.lm <- lm(body.fat~weight+abdomen,data=fat,subset
pred.regsubset <- predict(modregsubset.lm,newdata = fat[test,
cor(fat[test,"body.fat"],pred.regsubset)^2 # R^2 for test dat
```

```
## [1] 0.695192
```

```
mean((fat[test,"body.fat"]-pred.regsubset)^2) # MSE_test
```

```
## [1] 15.78438
```

Principal Component Regression (PCR) is a regression technique that first reduces the predictors using Principal Component Analysis (PCA) and then builds a regression model based on the principal components.