Predicting Boston Housing Prices

```
#-------------------------------------------------------------------------------------------

BostonHousing.df <- read.csv("BostonHousing.csv")
Housing.df <- BostonHousing.df[1:500,]
selected.var <- c(1,4,6,13)
set.seed(1)
train.index <- sample(c(1:500),300)
train.df <- BostonHousing.df[train.index,selected.var]
valid.df <- BostonHousing.df[-train.index,selected.var]

Housing.lm <- lm(MEDV ~ ., data = train.df)
Housing.lm
```

```
##
## Call:
## lm(formula = MEDV ~ ., data = train.df)
##
## Coefficients:
## (Intercept)         CRIM         CHAS           RM
##    -28.2694      -0.2174       4.3180       8.2116
```

```
options(scipen=999)
summary(Housing.lm)
```

```
##
## Call:
## lm(formula = MEDV ~ ., data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.491  -2.781  -0.518   2.289  38.523
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -28.26937    3.49497  -8.089   0.0000000000000157 ***
## CRIM         -0.21744    0.04552  -4.777   0.0000028103408703 ***
## CHAS          4.31799    1.44165   2.995              0.00297 **
## RM            8.21161    0.54811  14.982 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.493 on 296 degrees of freedom
## Multiple R-squared:  0.5199, Adjusted R-squared:  0.515
## F-statistic: 106.8 on 3 and 296 DF,  p-value: < 0.00000000000000022
```

```
# MEDV=y, intercept coefficient(a): -28.2694, CRIM=x1 , CHAS = x2, RM = x3,
# --> MEDV = -28.8107 -0.2174*CRIM + 4.3180*CHAS + 8.2116*RM + ei
```

```
#-------------------------------------------------------------------------------------------

library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
Housing.lm.pred <- predict(Housing.lm,valid.df)
options(sipen=999,digits=4)
residuals <- valid.df$MEDV[1:10] - Housing.lm.pred[1:10]
data.frame("Predicted" = Housing.lm.pred[1:10], "Actual"= valid.df$MEDV[1:10], "Residuals" =
residuals)
```

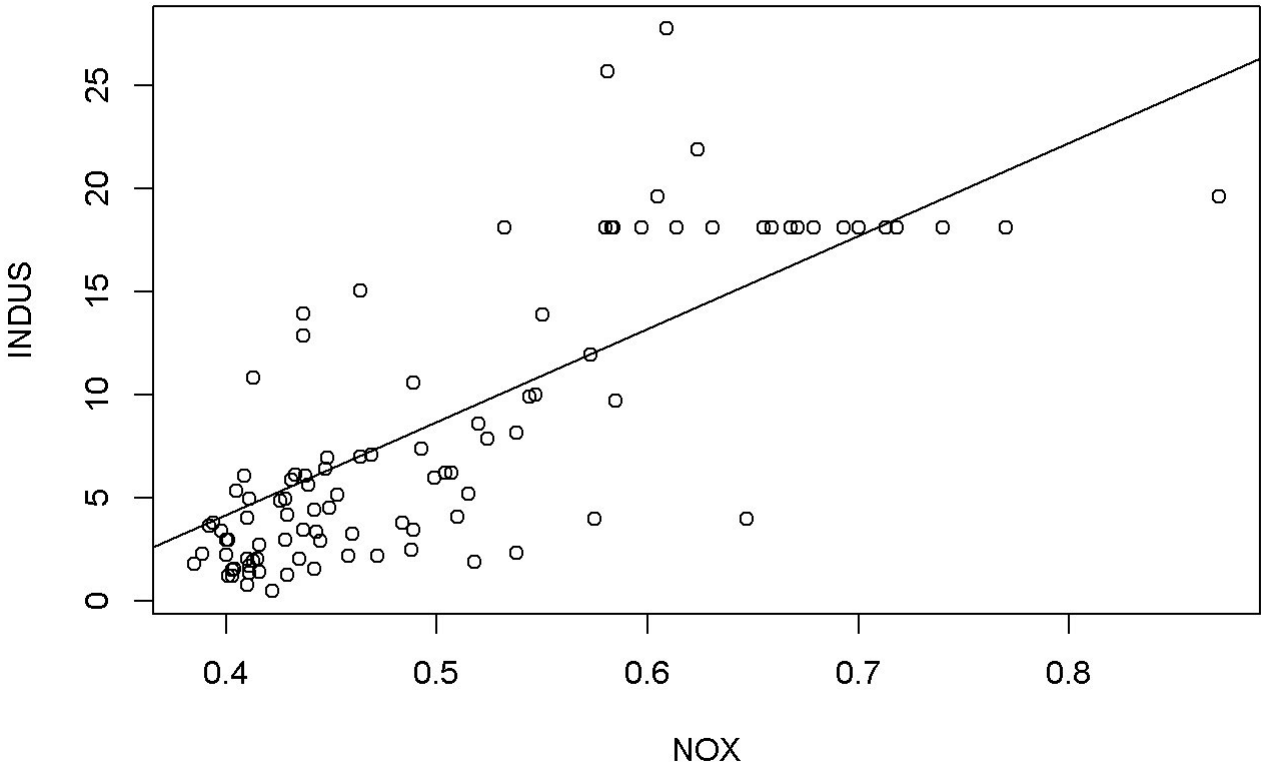
```
##    Predicted Actual Residuals
## 3      30.73   34.7     3.975
## 4      29.19   33.4     4.212
## 5      30.40   36.2     5.796
## 6      24.52   28.7     4.175
## 7      21.08   22.9     1.820
## 8      22.38   27.1     4.719
## 9      17.92   16.5    -1.424
## 10     21.00   18.9    -2.096
## 11     24.05   15.0    -9.047
## 12     21.05   18.9    -2.149
```

```
# Y = -28.2694 -0.2174*0.1 + 4.3180*0 + 8.2116*6 --> Predicted price = 21(20.97846)
# line 10 --> 21(Predicted),Actual(18.9), Prediction Error(-2.096)
```

```
#-------------------------------------------------------------------------------------------
model <- lm(INDUS~NOX,data=BostonHousing.df)
model
```

```
##
## Call:
## lm(formula = INDUS ~ NOX, data = BostonHousing.df)
##
## Coefficients:
## (Intercept)          NOX
##       -13.9         45.2
```

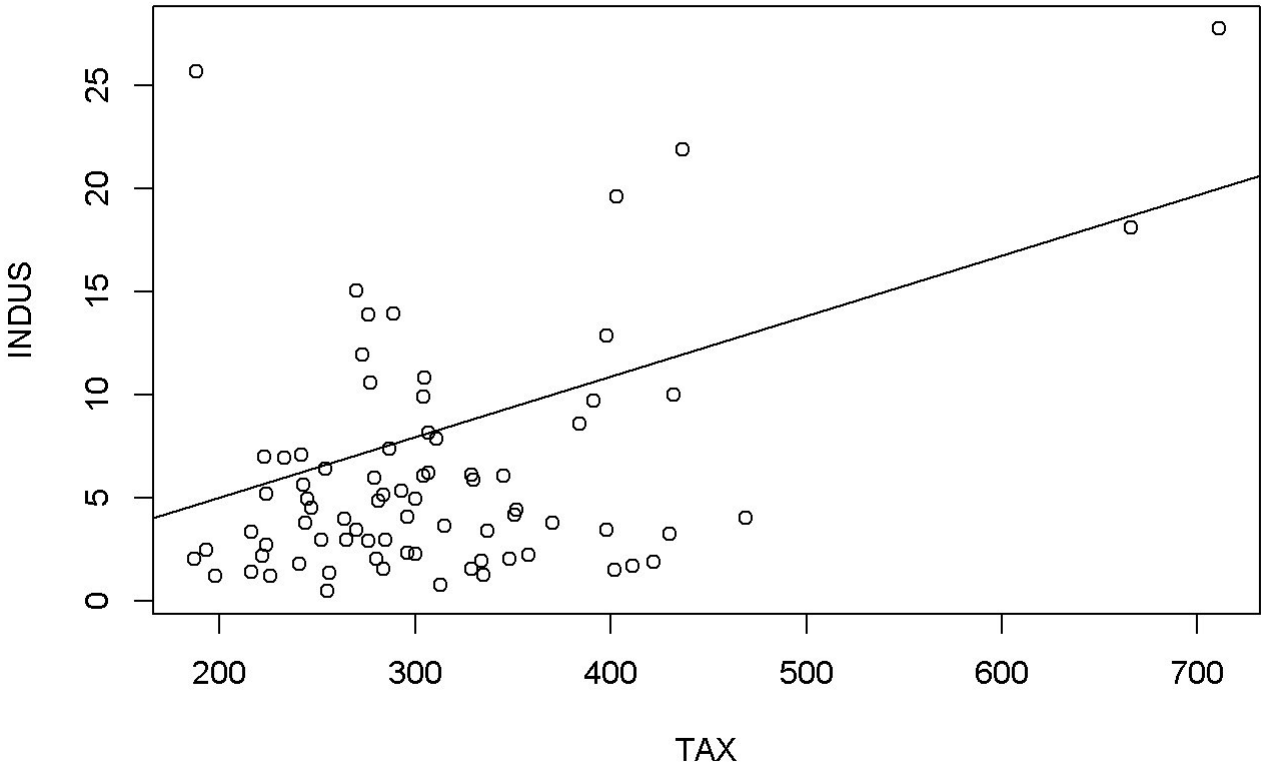
```
plot(INDUS~NOX,data=BostonHousing.df)
abline(model)
```




```
model <- lm(INDUS~TAX,data=BostonHousing.df)
model
```

```
##
## Call:
## lm(formula = INDUS ~ TAX, data = BostonHousing.df)
##
## Coefficients:
## (Intercept)          TAX
##     -0.8404       0.0293
```

```
plot(INDUS~TAX,data=BostonHousing.df)
abline(model)
```




```
Housing.lm <- lm(MEDV ~ INDUS + NOX + TAX, data = BostonHousing.df)
Housing.lm
```

```
##
## Call:
## lm(formula = MEDV ~ INDUS + NOX + TAX, data = BostonHousing.df)
##
## Coefficients:
## (Intercept)        INDUS          NOX          TAX
##     34.8234      -0.3494      -5.8453      -0.0126
```

```
options(scipen=999)
summary(Housing.lm)
```

```
##
## Call:
## lm(formula = MEDV ~ INDUS + NOX + TAX, data = BostonHousing.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12.45  -4.71  -1.95   3.21  33.82
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept) 34.82335    2.04944   16.99 < 0.0000000000000002 ***
## INDUS       -0.34943    0.08834   -3.96             0.000087 ***
## NOX         -5.84532    4.87195   -1.20                 0.23
## TAX         -0.01263    0.00312   -4.05             0.000060 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.9 on 502 degrees of freedom
## Multiple R-squared:  0.266,  Adjusted R-squared:  0.262
## F-statistic: 60.6 on 3 and 502 DF,  p-value: <0.0000000000000002
```

```
#-------------------------------------------------------------------------------------------

round(cor(BostonHousing.df),2)
```

```
##             CRIM    ZN INDUS  CHAS   NOX    RM   AGE   DIS   RAD   TAX PTRATIO
## CRIM        1.00 -0.20  0.41 -0.06  0.42 -0.22  0.35 -0.38  0.63  0.58    0.29
## ZN         -0.20  1.00 -0.53 -0.04 -0.52  0.31 -0.57  0.66 -0.31 -0.31   -0.39
## INDUS       0.41 -0.53  1.00  0.06  0.76 -0.39  0.64 -0.71  0.60  0.72    0.38
## CHAS       -0.06 -0.04  0.06  1.00  0.09  0.09  0.09 -0.10 -0.01 -0.04   -0.12
## NOX         0.42 -0.52  0.76  0.09  1.00 -0.30  0.73 -0.77  0.61  0.67    0.19
## RM         -0.22  0.31 -0.39  0.09 -0.30  1.00 -0.24  0.21 -0.21 -0.29   -0.36
## AGE         0.35 -0.57  0.64  0.09  0.73 -0.24  1.00 -0.75  0.46  0.51    0.26
## DIS        -0.38  0.66 -0.71 -0.10 -0.77  0.21 -0.75  1.00 -0.49 -0.53   -0.23
## RAD         0.63 -0.31  0.60 -0.01  0.61 -0.21  0.46 -0.49  1.00  0.91    0.46
## TAX         0.58 -0.31  0.72 -0.04  0.67 -0.29  0.51 -0.53  0.91  1.00    0.46
## PTRATIO     0.29 -0.39  0.38 -0.12  0.19 -0.36  0.26 -0.23  0.46  0.46    1.00
## LSTAT       0.46 -0.41  0.60 -0.05  0.59 -0.61  0.60 -0.50  0.49  0.54    0.37
## MEDV       -0.39  0.36 -0.48  0.18 -0.43  0.70 -0.38  0.25 -0.38 -0.47   -0.51
## CAT..MEDV -0.15  0.37 -0.37  0.11 -0.23  0.64 -0.19  0.12 -0.20 -0.27   -0.44
##            LSTAT  MEDV CAT..MEDV
## CRIM        0.46 -0.39     -0.15
## ZN         -0.41  0.36      0.37
## INDUS       0.60 -0.48     -0.37
## CHAS       -0.05  0.18      0.11
## NOX         0.59 -0.43     -0.23
## RM         -0.61  0.70      0.64
## AGE         0.60 -0.38     -0.19
## DIS        -0.50  0.25      0.12
## RAD         0.49 -0.38     -0.20
## TAX         0.54 -0.47     -0.27
## PTRATIO     0.37 -0.51     -0.44
## LSTAT       1.00 -0.74     -0.47
## MEDV       -0.74  1.00      0.79
## CAT..MEDV -0.47  0.79      1.00
```

```
Housing.df <- BostonHousing.df[1:500,]
selected.var <- c(1,2,4,6,11,12,13)
set.seed(1)
train.index <- sample(c(1:500),300)
train.df <- BostonHousing.df[train.index,selected.var]
valid.df <- BostonHousing.df[-train.index,selected.var]

housing.lm <- lm(MEDV ~ ., data = train.df)
housing.lm
```

```
##
## Call:
## lm(formula = MEDV ~ ., data = train.df)
##
## Coefficients:
## (Intercept)         CRIM           ZN         CHAS           RM      PTRATIO
##     16.0030      -0.0327      -0.0177       3.8217       4.5327      -0.7726
##       LSTAT
##     -0.5818
```

```
options(sipen=999,digits=10)
summary(housing.lm)
```

```
##
## Call:
## lm(formula = MEDV ~ ., data = train.df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -18.924543  -3.162235  -1.007182   1.990467  29.132072
##
## Coefficients:
##                Estimate  Std. Error  t value               Pr(>|t|)
## (Intercept) 16.00296901  5.68366830  2.81561              0.0051987 **
## CRIM        -0.03265811  0.04290058 -0.76125              0.4471194
## ZN          -0.01771804  0.01596637 -1.10971              0.2680341
## CHAS         3.82171621  1.24439243  3.07115              0.0023325 **
## RM           4.53265254  0.59153211  7.66256    0.00000000000026851 ***
## PTRATIO     -0.77256228  0.17277472 -4.47150    0.00001111900684105 ***
## LSTAT       -0.58177891  0.06535875 -8.90132 < 0.000000000000000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.569239 on 293 degrees of freedom
## Multiple R-squared:  0.6503937,  Adjusted R-squared:  0.6432345
## F-statistic: 90.84759 on 6 and 293 DF,  p-value: < 0.00000000000000022204
```

```
housing.lm.step <-step(housing.lm,direction = "forward")
```

```
## Start:  AIC=1037.27
## MEDV ~ CRIM + ZN + CHAS + RM + PTRATIO + LSTAT
```

```
summary(housing.lm.step)
```

```
##
## Call:
## lm(formula = MEDV ~ CRIM + ZN + CHAS + RM + PTRATIO + LSTAT,
##     data = train.df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -18.924543  -3.162235  -1.007182   1.990467  29.132072
##
## Coefficients:
##                Estimate  Std. Error  t value               Pr(>|t|)
## (Intercept) 16.00296901  5.68366830  2.81561              0.0051987 **
## CRIM        -0.03265811  0.04290058 -0.76125              0.4471194
## ZN          -0.01771804  0.01596637 -1.10971              0.2680341
## CHAS         3.82171621  1.24439243  3.07115              0.0023325 **
## RM           4.53265254  0.59153211  7.66256    0.00000000000026851 ***
## PTRATIO     -0.77256228  0.17277472 -4.47150    0.00001111900684105 ***
## LSTAT       -0.58177891  0.06535875 -8.90132 < 0.000000000000000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.569239 on 293 degrees of freedom
## Multiple R-squared:  0.6503937,  Adjusted R-squared:  0.6432345
## F-statistic: 90.84759 on 6 and 293 DF,  p-value: < 0.00000000000000022204
```

```
housing.lm.step.pred <- predict(housing.lm.step,valid.df)
accuracy(housing.lm.step.pred,valid.df$MEDV)
```

```
##                      ME        RMSE         MAE           MPE        MAPE
## Test set -0.6924045393 4.571737279 3.475846241 -7.756619093 19.15449117
```

```
housing.lm.step <-step(housing.lm,direction = "backward")
```

```
## Start:  AIC=1037.27
## MEDV ~ CRIM + ZN + CHAS + RM + PTRATIO + LSTAT
##
##           Df  Sum of Sq        RSS       AIC
## - CRIM     1   17.97411  9105.7873 1035.8649
## - ZN       1   38.19535  9126.0085 1036.5304
## <none>                   9087.8132 1037.2721
## - CHAS     1  292.54583  9380.3590 1044.7773
## - PTRATIO  1  620.15232  9707.9655 1055.0759
## - RM       1 1821.12593 10908.9391 1090.0666
## - LSTAT    1 2457.53776 11545.3510 1107.0767
##
## Step:  AIC=1035.86
## MEDV ~ ZN + CHAS + RM + PTRATIO + LSTAT
##
##           Df  Sum of Sq        RSS       AIC
## - ZN       1   38.74483  9144.5321 1035.1387
## <none>                   9105.7873 1035.8649
## - CHAS     1  293.97234  9399.7597 1043.3971
## - PTRATIO  1  663.75407  9769.5414 1054.9727
## - RM       1 1805.01975 10910.8071 1088.1180
## - LSTAT    1 3006.30967 12112.0970 1119.4533
##
## Step:  AIC=1035.14
## MEDV ~ CHAS + RM + PTRATIO + LSTAT
##
##           Df  Sum of Sq        RSS       AIC
## <none>                   9144.5321 1035.1387
## - CHAS     1  313.31719  9457.8493 1043.2453
## - PTRATIO  1  625.43946  9769.9716 1052.9859
## - RM       1 1782.52855 10927.0607 1086.5645
## - LSTAT    1 3007.03561 12151.5678 1118.4293
```

```
summary(housing.lm.step)
```

```
##
## Call:
## lm(formula = MEDV ~ CHAS + RM + PTRATIO + LSTAT, data = train.df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -18.748006  -3.115446  -1.059867   1.885316  29.131460
##
## Coefficients:
##                Estimate  Std. Error  t value               Pr(>|t|)
## (Intercept) 15.52157132  5.52748428  2.80807              0.0053163 **
## CHAS         3.94218638  1.23998028  3.17923              0.0016336 **
## RM           4.46817348  0.58922548  7.58313    0.00000000000044184 ***
## PTRATIO     -0.74032186  0.16481536 -4.49183    0.00001014683885636 ***
## LSTAT       -0.58446374  0.05934143 -9.84917 < 0.000000000000000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.567622 on 295 degrees of freedom
## Multiple R-squared:  0.6482117,  Adjusted R-squared:  0.6434417
## F-statistic: 135.8931 on 4 and 295 DF,  p-value: < 0.00000000000000022204
```

```
housing.lm.step.pred <- predict(housing.lm.step,valid.df)
accuracy(housing.lm.step.pred,valid.df$MEDV)
```

```
##                      ME        RMSE         MAE          MPE        MAPE
## Test set -0.6975702803 4.565717637 3.474073895 -7.953675503 19.28480577
```

```
housing.lm.step <-step(housing.lm,direction = "both")
```

```
## Start:  AIC=1037.27
## MEDV ~ CRIM + ZN + CHAS + RM + PTRATIO + LSTAT
##
##           Df  Sum of Sq        RSS       AIC
## - CRIM     1   17.97411  9105.7873 1035.8649
## - ZN       1   38.19535  9126.0085 1036.5304
## <none>                   9087.8132 1037.2721
## - CHAS     1  292.54583  9380.3590 1044.7773
## - PTRATIO  1  620.15232  9707.9655 1055.0759
## - RM       1 1821.12593 10908.9391 1090.0666
## - LSTAT    1 2457.53776 11545.3510 1107.0767
##
## Step:  AIC=1035.86
## MEDV ~ ZN + CHAS + RM + PTRATIO + LSTAT
##
##           Df  Sum of Sq        RSS       AIC
## - ZN       1   38.74483  9144.5321 1035.1387
## <none>                   9105.7873 1035.8649
## + CRIM     1   17.97411  9087.8132 1037.2721
## - CHAS     1  293.97234  9399.7597 1043.3971
## - PTRATIO  1  663.75407  9769.5414 1054.9727
## - RM       1 1805.01975 10910.8071 1088.1180
## - LSTAT    1 3006.30967 12112.0970 1119.4533
##
## Step:  AIC=1035.14
## MEDV ~ CHAS + RM + PTRATIO + LSTAT
##
##           Df  Sum of Sq        RSS       AIC
## <none>                   9144.5321 1035.1387
## + ZN       1   38.74483  9105.7873 1035.8649
## + CRIM     1   18.52360  9126.0085 1036.5304
## - CHAS     1  313.31719  9457.8493 1043.2453
## - PTRATIO  1  625.43946  9769.9716 1052.9859
## - RM       1 1782.52855 10927.0607 1086.5645
## - LSTAT    1 3007.03561 12151.5678 1118.4293
```

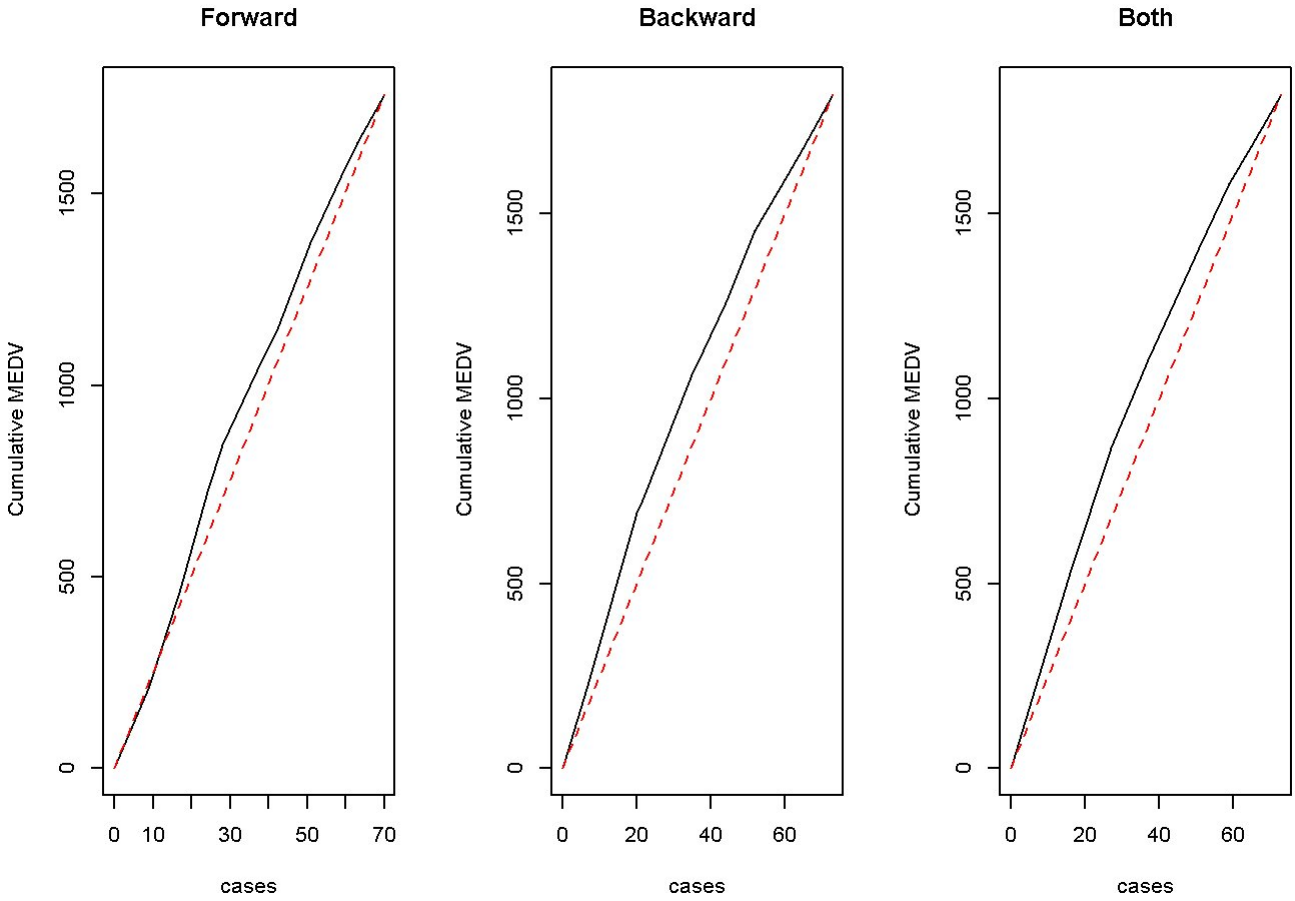
```
summary(housing.lm.step)
```

```
##
## Call:
## lm(formula = MEDV ~ CHAS + RM + PTRATIO + LSTAT, data = train.df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -18.748006  -3.115446  -1.059867   1.885316  29.131460
##
## Coefficients:
##                Estimate  Std. Error  t value               Pr(>|t|)
## (Intercept) 15.52157132  5.52748428  2.80807              0.0053163 **
## CHAS         3.94218638  1.23998028  3.17923              0.0016336 **
## RM           4.46817348  0.58922548  7.58313    0.00000000000044184 ***
## PTRATIO     -0.74032186  0.16481536 -4.49183    0.00001014683885636 ***
## LSTAT       -0.58446374  0.05934143 -9.84917 < 0.000000000000000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.567622 on 295 degrees of freedom
## Multiple R-squared:  0.6482117,  Adjusted R-squared:  0.6434417
## F-statistic: 135.8931 on 4 and 295 DF,  p-value: < 0.00000000000000022204
```

```
housing.lm.step.pred <- predict(housing.lm.step,valid.df)
accuracy(housing.lm.step.pred,valid.df$MEDV)
```

```
##                      ME        RMSE         MAE          MPE        MAPE
## Test set -0.6975702803 4.565717637 3.474073895 -7.953675503 19.28480577
```

```r
# forward lift chart

par(mfrow=c(1,3))

library(forecast)
training <- sample(BostonHousing.df$MEDV,300)
validation <- sample(setdiff(BostonHousing.df$CRIM,training),200)
liftchart.df <- BostonHousing.df[!is.na(BostonHousing.df[validation,]$MEDV),]


reg <- lm(MEDV ~ ., data = BostonHousing.df[,-c(1,2,4,6,11,12,14)],subset = training)
pred_v <- predict(reg,newdata=BostonHousing.df[validation,-c(1,2,4,6,11,12,14)])
library(gains)
gain <- gains(BostonHousing.df[validation,]$MEDV[!is.na(pred_v)],pred_v[!is.na(pred_v)])
options(scipen=999)
MEDV <- BostonHousing.df[validation,]$MEDV[!is.na(BostonHousing.df[validation,]$MEDV)]
plot(c(0,gain$cume.pct.of.total*sum(MEDV)) ~ c(0,gain$cume.obs), xlab = "cases",ylab = " Cumu
lative MEDV", main = "Forward",type="l")
lines(c(0,sum(MEDV)) ~ c(0,dim(BostonHousing.df[validation,])[1]), col = "red",lty=2)

# backward lift chart

library(forecast)
liftchart.df <- BostonHousing.df[!is.na(BostonHousing.df[validation,]$CRIM),]
training <- sample(BostonHousing.df$CRIM,300)
validation <- sample(setdiff(BostonHousing.df$CRIM,training),200)


reg <- lm(MEDV~., data = BostonHousing.df[,-c(4,6,11,12)],subset = training)
pred_v <- predict(reg,newdata=BostonHousing.df[validation,-c(4,6,11,12)])
library(gains)
gain <- gains(BostonHousing.df[validation,]$MEDV[!is.na(pred_v)],pred_v[!is.na(pred_v)])
options(scipen=999)
MEDV <- BostonHousing.df[validation,]$MEDV[!is.na(BostonHousing.df[validation,]$MEDV)]
plot(c(0,gain$cume.pct.of.total*sum(MEDV)) ~ c(0,gain$cume.obs), xlab = "cases",ylab = " Cumu
lative MEDV", main = "Backward",type="l")
lines(c(0,sum(MEDV)) ~ c(0,dim(BostonHousing.df[validation,])[1]), col = "red",lty=2)

# Both lift chart

reg <- lm(MEDV ~ ., data = BostonHousing.df[,-c(1,2,4,6,11,12,14)],subset = training)
pred_v <- predict(reg,newdata=BostonHousing.df[validation,-c(1,2,4,6,11,12,14)])
library(gains)
gain <- gains(BostonHousing.df[validation,]$MEDV[!is.na(pred_v)],pred_v[!is.na(pred_v)])
options(scipen=999)
MEDV <- BostonHousing.df[validation,]$MEDV[!is.na(BostonHousing.df[validation,]$MEDV)]
plot(c(0,gain$cume.pct.of.total*sum(MEDV)) ~ c(0,gain$cume.obs), xlab = "cases",ylab = " Cumu
lative MEDV", main = "Both",type="l")
lines(c(0,sum(MEDV)) ~ c(0,dim(BostonHousing.df[validation,])[1]), col = "red",lty=2)
```


Forward
Backward
Both
Cumulative MEDV
0
500
1000
1500
0 10 30 50 70
0 20 40 60
cases