Ch3_Fall2020_Soln

Rong Fan

10/8/2020

```
library(styler)
```

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

1. Residual analysis and plots function WITH intercept

```
# To use this function, make sure your data satisfies the following conditions:
# 1. The first column is 1's vector.
# 2. The last column is the response variable (y).
# 3. The other columns are independant variables (X) which are involved in the regression model.

res.fun <- function(data) {
  data <- as.matrix(data)
  n <- nrow(data)
  p <- ncol(data) - 1
  X <- data[, -ncol(data)] # Predictors/ regressor matrix / independant variables
  hat.mat <- X %*% solve(t(X) %*% X) %*% t(X) # hat matrix
  y <- data[, ncol(data)] # Response variable / dependant variable
  reg <- lm(y ~ X) # Linear regression model
  sum.reg <- summary(reg)
  CI.coef <- confint(reg, level = .95)
  e.i <- reg$residuals # Residuals
  sigma.hat <- sigma(reg) # estimator of standard deviation/ sqrt(MSE)
  d.i <- e.i / sigma.hat # d_i / Standardized Residuals
  h.ii <- diag(hat.mat)
  r.i <- e.i / sigma.hat / sqrt(1 - h.ii) # r_i / Studentized residuals
  PRESS <- sum((e.i / (1 - h.ii))^2) # Prediction Error Sum of Squares
  anova.sat <- anova(reg) # ANOVA to obtain SST, SSE.
  R.2.pred <- 1 - PRESS / sum(anova.sat$`Sum Sq`) # R squared for prediction
  S.2.i <- (((n - p) * anova.sat$`Mean Sq`[2]) - e.i^2 / (1 - h.ii)) / (n - p - 1)
  t.i <- e.i / sqrt(S.2.i * (1 - h.ii)) # R-Student
  index.leverage <- which(h.ii > 2 * p / n) # Return index of the high leverage observations
  high.leverage <- h.ii[h.ii > 2 * p / n] # Return the h.ii values for high leverage
  cook.dis <- cooks.distance(reg) # Return the cook's distance
  index.inf <- which(cook.dis > 1)
  inf.out <- cook.dis[cook.dis > 1]

  Sati <- as.data.frame(cbind(
    round(e.i, 3), # Combine all statistics into a data frame
    round(r.i, 3),
    round(t.i, 3),
```

```r
    round(h.ii, 3),
    round(cook.dis, 3)
  ))
  names(Sati) <- c(
    "Residuals",
    "Studentized Residuals", "R-Student", "h_ii", "Cook's Distance"
  )
  # print(cbind(e.i, r.i, t.i, h.ii,cook.dis))
  names(PRESS) <- c("PRESS")
  names(R.2.pred) <- c("Prediction R-squared")

  leverage <- as.data.frame(cbind(index.leverage, high.leverage))
  names(leverage) <- c("High-leverage-index", "h_ii")

  influ <- as.data.frame(cbind(index.inf, inf.out))
  names(influ) <- c("Influential-outlier-index", "cook's distance")

  mylist <- list(sum.reg$coefficients, Sati, PRESS, R.2.pred, leverage, influ, CI.coef)

  return(mylist)
}
```

```r
# Residual Plots.
# To use this function, make sure your data satisfies the following conditions:
# 1. The first column is 1's vector.
# 2. The last column is the response variable (y).
# 3. The other columns are independant variables (X) which are involved in the regression model.
resid.plot <- function(data) {
  data <- as.matrix(data)
  X <- data[, -ncol(data)] # Predictors/ regressor matrix / independant variables
  y <- data[, ncol(data)] # Response variable / dependant variable
  fit.data <- lm(y ~ X)
  par(mfrow = c(2, 2), oma = c(0, 0, 0, 0))
  qqnorm(fit.data$residuals, datax = T, pch = 16, xlab = "Residual", main = "")
  qqline(fit.data$residuals, data = T)
  plot(fit.data$fitted.values, fit.data$residuals, pch = 16, xlab = "Fitted Value", ylab = "Residual")
  abline(h = 0)
  hist(fit.data$residuals, col = "grey", xlab = "Residual", main = "")
  plot(fit.data$residuals, type = "l", xlab = "Observation Order", ylab = "Residual")
  points(fit.data$residuals, pch = 16, cex = .5)
  abline(h = 0)
}
```

2. Residual analysis and plots function WITHOUT intercept

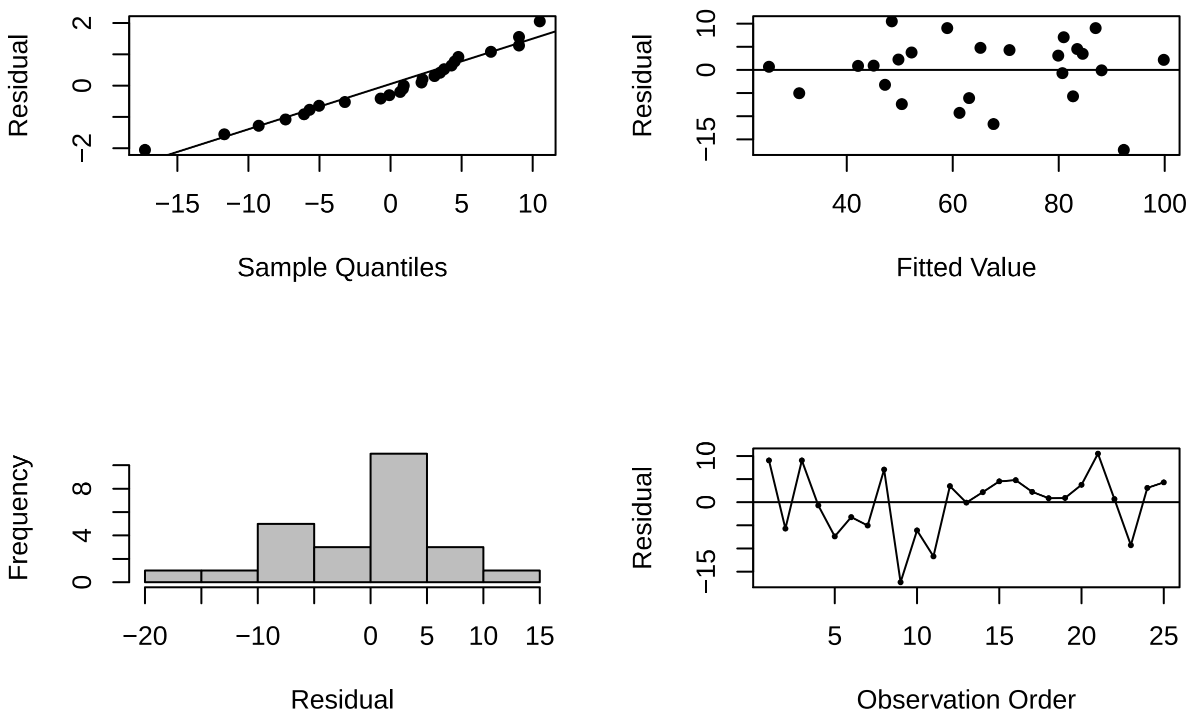
```r
# The calculation of residual analysis WITHOUT intercept in the model
# To use this function, make sure your data satisfies the following conditions:
# 1. The first column does NOT NOT NOT NOT NOT NOT NOT contain 1's vector.
# 2. The last column is the response variable (y).
# 3. The other columns are independant variables (X) which are involved in the regression model.

res.fun.no.int <- function(data) {
  data <- as.matrix(data)
  n <- nrow(data)
  p <- ncol(data) - 1
  X <- data[, -ncol(data)] # Predictors/ regressor matrix / independant variables
  hat.mat <- X %*% solve(t(X) %*% X) %*% t(X) # hat matrix
  y <- data[, ncol(data)] # Response variable / dependant variable
  reg <- lm(y ~ X - 1) # Linear regression model without intercept
  sum.reg <- summary(reg)
  CI.coef <- confint(reg, level = .95)
  e.i <- reg$residuals # Residuals
  sigma.hat <- sigma(reg) # estimator of standard deviation/ sqrt(MSE)
  d.i <- e.i / sigma.hat # d_i / Standardized Residuals
  h.ii <- diag(hat.mat)
  r.i <- e.i / sigma.hat / sqrt(1 - h.ii) # r_i / Studentized residuals
  PRESS <- sum((e.i / (1 - h.ii))^2) # Prediction Error Sum of Squares
  anova.sat <- anova(reg) # ANOVA to obtain SST, SSE.
  R.2.pred <- 1 - PRESS / sum(anova.sat$`Sum Sq`) # R squared for prediction
  S.2.i <- (((n - p) * anova.sat$`Mean Sq`[2]) - e.i^2 / (1 - h.ii)) / (n - p - 1)
  t.i <- e.i / sqrt(S.2.i * (1 - h.ii)) # R-Student
  index.leverage <- which(h.ii > 2 * p / n) # Return index of the high leverage observations
  high.leverage <- h.ii[h.ii > 2 * p / n] # Return the h.ii values for high leverage
  cook.dis <- cooks.distance(reg) # Return the cook's distance
  index.inf <- which(cook.dis > 1)
  inf.out <- cook.dis[cook.dis > 1]

  Sati <- as.data.frame(cbind(
    round(e.i, 3), # Combine all statistics into a data frame
    round(r.i, 3),
    round(t.i, 3),
    round(h.ii, 3),
    round(cook.dis, 3)
  ))
  names(Sati) <- c(
    "Residuals",
    "Studentized Residuals", "R-Student", "h_ii", "Cook's Distance"
  )
  # print(cbind(e.i, r.i, t.i, h.ii,cook.dis))
  names(PRESS) <- c("PRESS")
  names(R.2.pred) <- c("Prediction R-squared")

  leverage <- as.data.frame(cbind(index.leverage, high.leverage))
  names(leverage) <- c("High-leverage-index", "h_ii")

  influ <- as.data.frame(cbind(index.inf, inf.out))
  names(influ) <- c("Influential-outlier-index", "cook's distance")

  mylist <- list(sum.reg$coefficients, Sati, PRESS, R.2.pred, leverage, influ, CI.coef)

  return(mylist)
}
```

```r
# Residual Plots.
# To use this function, make sure your data satisfies the following conditions:
# 1. The first column does NOT NOT NOT NOT NOT NOT NOT contain 1's vector.
# 2. The last column is the response variable (y).
# 3. The other columns are independant variables (X) which are involved in the regression model.
resid.plot.no.int <- function(data) {
  data <- as.matrix(data)
  X <- data[, -ncol(data)] # Predictors/ regressor matrix / independant variables
  y <- data[, ncol(data)] # Response variable / dependant variable
  fit.data <- lm(y ~ X - 1)
  par(mfrow = c(2, 2), oma = c(0, 0, 0, 0))
  qqnorm(fit.data$residuals, datax = T, pch = 16, xlab = "Residual", main = "")
  qqline(fit.data$residuals, data = T)
  plot(fit.data$fitted.values, fit.data$residuals, pch = 16, xlab = "Fitted Value", ylab = "Residual")
  abline(h = 0)
  hist(fit.data$residuals, col = "grey", xlab = "Residual", main = "")
  plot(fit.data$residuals, type = "l", xlab = "Observation Order", ylab = "Residual")
  points(fit.data$residuals, pch = 16, cex = .5)
  abline(h = 0)
}
```

3. Patient Example residual analysis.

```
library(readxl)
sat <- read_excel("satisfaction.xlsx") # Import the patient satisfaction data. It can be found at D2L.
sat.mat <- as.matrix(sat) # Convert the data structure into a matrix
```

```
resid.plot(sat.mat)
```



```r
res.fun(sat.mat)
```

```
## [[1]]
##                 Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 143.4720118  5.9548379 24.093353 2.632652e-17
## XAge         -1.0310534  0.1156112 -8.918286 9.284959e-09
## XSeverity    -0.5560378  0.1314103 -4.231312 3.429369e-04
## 
## [[2]]
##    Residuals Studentized Residuals R-Student  h_ii Cook's Distance
## 1      9.038                 1.299     1.321 0.045           0.026
## 2     -5.699                -0.882    -0.878 0.176           0.056
## 3      9.037                 1.381     1.412 0.155           0.117
## 4     -0.695                -0.104    -0.102 0.118           0.000
## 5     -7.390                -1.080    -1.084 0.076           0.032
## 6     -3.215                -0.473    -0.465 0.089           0.007
## 7     -5.032                -0.774    -0.767 0.165           0.040
## 8      7.062                 1.030     1.032 0.073           0.028
## 9    -17.280                -2.658    -3.151 0.166           0.467
## 10    -6.086                -0.875    -0.870 0.045           0.012
## 11   -11.697                -1.702    -1.785 0.068           0.071
## 12     3.482                 0.516     0.508 0.102           0.010
## 13    -0.086                -0.013    -0.012 0.101           0.000
## 14     2.179                 0.337     0.330 0.177           0.008
## 15     4.513                 0.669     0.661 0.102           0.017
## 16     4.770                 0.685     0.676 0.042           0.007
## 17     2.247                 0.332     0.325 0.097           0.004
## 18     0.870                 0.137     0.134 0.204           0.002
## 19     0.928                 0.138     0.135 0.104           0.001
## 20     3.769                 0.586     0.577 0.182           0.025
## 21    10.499                 1.624     1.691 0.175           0.187
## 22     0.680                 0.107     0.105 0.207           0.001
## 23    -9.278                -1.469    -1.511 0.212           0.194
## 24     3.093                 0.450     0.442 0.067           0.005
## 25     4.291                 0.618     0.609 0.049           0.007
## 
## [[3]]
##    PRESS 
## 1484.934 
## 
## [[4]]
## Prediction R-squared 
##            0.8622286 
## 
## [[5]]
## [1] High-leverage-index h_ii
## <0 rows> (or 0-length row.names)
## 
## [[6]]
## [1] Influential-outlier-index cook's distance
## <0 rows> (or 0-length row.names)
## 
## [[7]]
##                  2.5 %      97.5 %
## (Intercept) 131.122434 155.8215898
## XInter              NA          NA
## XAge         -1.270816  -0.7912905
## XSeverity    -0.828566  -0.2835096
```

Ex 3.7

The quality of Pinot Noir wine is thought to be related to the properties of clarity, aroma, body, flavor, and oakiness. Data for 38 wines are given in Table E3.4.

```
library(readxl)
Ex3_7 <- read_excel("Ex3_7.xlsx")
wine <- as.data.frame(Ex3_7)
attach(wine)
```

```
wine.reg <- lm(Quality ~ Clarity + Aroma + Body + Flavor + Oakiness)
wine.reg$coefficients
```

3.7 a. Fit a multiple linear regression model relating wine quality to these predictors. Do not include the "Region" variable in the model.

```
## (Intercept)     Clarity       Aroma        Body      Flavor    Oakiness
##   3.9968648   2.3394535   0.4825505   0.2731612   1.1683238  -0.6840102
```

The fitted linear regression model is $\hat{Quality} = 4.00 + 2.34(Clarity) + 0.48(Aroma) + 0.27(Body) + 1.17(Flavor) - 0.68(Oakiness)$.

```
summary(wine.reg)
```

3.7 b. Test for significance of regression. What conclusions can you draw?

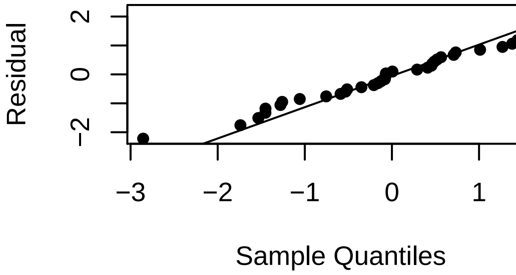
```
##
## Call:
## lm(formula = Quality ~ Clarity + Aroma + Body + Flavor + Oakiness)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.85552 -0.57448 -0.07092  0.67275  1.68093
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9969     2.2318   1.791 0.082775 .
## Clarity       2.3395     1.7348   1.349 0.186958
## Aroma         0.4826     0.2724   1.771 0.086058 .
## Body          0.2732     0.3326   0.821 0.417503
## Flavor        1.1683     0.3045   3.837 0.000552 ***
## Oakiness     -0.6840     0.2712  -2.522 0.016833 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.163 on 32 degrees of freedom
## Multiple R-squared:  0.7206, Adjusted R-squared:  0.6769
## F-statistic: 16.51 on 5 and 32 DF,  p-value: 4.703e-08
```

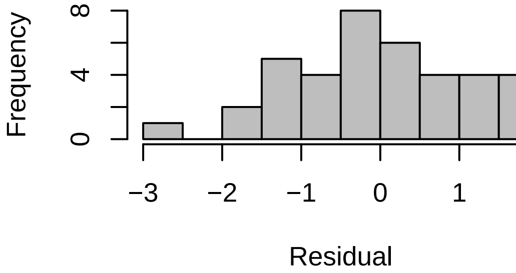
The F test statistics is $F = 16.51$ which gives a very small p-value $p = 4.7 \times 10^{-8}$. Therefore the linear regression model is significant.

3.7 c. Use t-tests to assess the contribution of each predictor to the model. Discuss your findings. See the output in 3.7 b.

The t-test implies that the coefficients for Flavor and Oakiness are significant given significant level $\alpha = 0.05$. We fail to reject the hypothesis that the other coefficients are zeros.

```
nrow <- dim(wine)[1]
wine.mat <- as.matrix(Ex3_7)
resid.plot(wine.mat[, 1:6])
```





3.7 d. Analyze the residuals from this model. Is the model adequate?

```
wine.residual <- cbind(matrix(1, nrow, 1), wine.mat[, 1:6])
res.fun(wine.residual)
```

```
## [[1]]
##                 Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)  3.9968648  2.2317701  1.7908945 0.0827745402
## XClarity     2.3394535  1.7348271  1.3485226 0.1869578145
## XAroma       0.4825505  0.2724473  1.7711703 0.0860575292
## XBody        0.2731612  0.3325606  0.8213876 0.4175033004
## XFlavor      1.1683238  0.3044807  3.8371025 0.0005522334
## XOakiness   -0.6840102  0.2711928 -2.5222287 0.0168327190
```

```
##
## [[2]]
##    Residuals Studentized Residuals R-Student  h_ii Cook's Distance
## 1       0.289                 0.273     0.269 0.170           0.003
## 2       1.380                 1.296     1.311 0.161           0.054
## 3      -0.159                -0.150    -0.148 0.168           0.001
## 4       1.012                 0.989     0.989 0.226           0.048
## 5      -0.069                -0.062    -0.061 0.076           0.000
## 6      -0.077                -0.068    -0.067 0.057           0.000
## 7      -0.514                -0.468    -0.462 0.106           0.004
## 8       0.410                 0.374     0.369 0.110           0.003
## 9       1.441                 1.349     1.367 0.155           0.056
## 10      1.600                 1.436     1.461 0.081           0.030
## 11      1.633                 1.461     1.488 0.075           0.029
## 12      1.505                 1.663     1.713 0.395           0.301
## 13      0.454                 0.435     0.430 0.196           0.008
## 14     -0.350                -0.376    -0.371 0.359           0.013
## 15      0.564                 0.556     0.549 0.238           0.016
## 16      1.681                 1.492     1.523 0.061           0.024
## 17      0.487                 0.457     0.451 0.159           0.007
## 18     -0.073                -0.066    -0.065 0.111           0.000
## 19     -1.279                -1.171    -1.178 0.117           0.030
## 20     -2.856                -2.743    -3.087 0.198           0.310
## 21      0.007                 0.006     0.006 0.125           0.000
## 22     -0.531                -0.479    -0.473 0.089           0.004
## 23     -1.533                -1.482    -1.511 0.208           0.096
## 24     -1.260                -1.195    -1.203 0.178           0.051
## 25     -1.451                -1.299    -1.314 0.077           0.023
## 26     -0.207                -0.185    -0.183 0.075           0.000
## 27     -0.081                -0.077    -0.076 0.187           0.000
## 28     -0.589                -0.525    -0.519 0.069           0.003
## 29     -1.452                -1.329    -1.346 0.116           0.039
## 30     -1.739                -1.654    -1.702 0.182           0.101
## 31     -1.058                -1.019    -1.020 0.203           0.044
## 32     -0.123                -0.121    -0.119 0.231           0.001
## 33      1.269                 1.192     1.200 0.161           0.045
## 34      0.463                 0.434     0.428 0.154           0.006
## 35      0.519                 0.472     0.466 0.104           0.004
## 36      0.709                 0.652     0.647 0.126           0.010
## 37      0.732                 0.797     0.792 0.375           0.064
## 38     -0.754                -0.691    -0.686 0.119           0.011
##
## [[3]]
##    PRESS
## 63.92662
##
## [[4]]
## Prediction R-squared
##            0.5870065
##
## [[5]]
##   High-leverage-index      h_ii
## 1                  12 0.3946823
## 2                  14 0.3592445
## 3                  37 0.3752668
##
## [[6]]
## [1] Influential-outlier-index cook's distance
## <0 rows> (or 0-length row.names)
##
## [[7]]
##                   2.5 %     97.5 %
## (Intercept) -0.54910206  8.5428317
## X                    NA         NA
## XClarity    -1.19427368  5.8731807
## XAroma      -0.07240642  1.0375075
## XBody       -0.40424262  0.9505650
## XFlavor      0.54811681  1.7885307
## XOakiness   -1.23641174 -0.1316086
```

The Normal probability plots of residuals does not violate the normality assumption. The PRESS=63.93 is relative small. The fitted value vs residual plot does not look like randomly scattered. There are three high-leverage observations. These outputs indicate that the model is not adequacy.

3.7 e. Calculate R^2 and the adjusted R^2 for this model. Compare these values to the R^2 and adjusted R^2 for the linear regression model relating wine quality to only the predictors "Aroma" and "Flavor." Discuss your results. For the full model, the multiple R-squared: $R^2 = 0.7206$ and Adjusted R-squared: $R^2 = 0.6769$. (See output in 3.7 b.)

For the reduced model which contains only "Aroma" and "Flavor" as predictors, the multiple R-squared: $R^2 = 0.6586$ and Adjusted R-squared: $R^2 = 0.639$. (See output below.)

The difference between the two adjusted R^2 is only 0.0379 which is pretty small. But the reduced model used only two predictors. Therefore the reduced model is better than the full model.

```r
wine.reg.redu <- lm(Quality ~ Aroma + Flavor)
summary(wine.reg.redu)
```

```
##
## Call:
## lm(formula = Quality ~ Aroma + Flavor)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.19048 -0.60300 -0.03203  0.66039  2.46287
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.3462     1.0091   4.307 0.000127 ***
## Aroma         0.5180     0.2759   1.877 0.068849 .
## Flavor        1.1702     0.2905   4.027 0.000288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.229 on 35 degrees of freedom
## Multiple R-squared:  0.6586, Adjusted R-squared:  0.639
## F-statistic: 33.75 on 2 and 35 DF,  p-value: 6.811e-09
```

```
confint(wine.reg, level = 0.95, "Flavor")
```

3.7 f. Find a 95% CI for the regression coefficient for "Flavor" for both models in part e. Discuss any differences.

```
##            2.5 %   97.5 %
## Flavor 0.5481168 1.788531
```

```
confint(wine.reg.redu, level = 0.95, "Flavor")
```

```
##            2.5 %   97.5 %
## Flavor 0.5803295 1.760003
```

The 95% CI for the regression coefficient for "Flavor" is (0.55, 1.79) in the full model and (0.58, 1.76) in the reduced model. The length of CI in reduced model $1.760003 - 0.58032952 \approx 1.18$ is shorter than the length of CI in full model $1.7885307 - 0.54811681 \approx 1.24$. In other words, the CI for "Flavor" in the reduced model is more precise.

3.26 Variables selection Consider the wine quality data in Exercise 3.7. Use variable selection techniques to determine an appropriate regression model for these data.

Let's use the forward selection method to determine the model.

```
# Forward Method
fit0 <- lm(Quality ~ 1, data = wine)
step.for <- step(fit0, direction = "forward", scope = ~ Clarity + Aroma + Body + Flavor + Oakiness + Qua
```

```
## Start:  AIC=55.37
## Quality ~ 1
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts): the
## response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts):
## problem with term 6 in model.matrix: no columns are assigned
```

```
##            Df Sum of Sq     RSS    AIC
## + Flavor    1    96.615  58.173 20.182
## + Aroma     1    77.442  77.347 31.007
## + Body      1    46.603 108.186 43.758
## + Region    1    39.796 114.993 46.077
## <none>                  154.788 55.370
## + Oakiness  1     0.343 154.446 57.286
## + Clarity   1     0.125 154.663 57.339
##
## Step:  AIC=20.18
## Quality ~ Flavor
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts): the
## response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts):
## problem with term 6 in model.matrix: no columns are assigned
```

```
##            Df Sum of Sq    RSS    AIC
## + Oakiness  1    5.7174 52.456 18.251
## + Aroma     1    5.3212 52.852 18.537
## <none>                  58.173 20.182
## + Region    1    2.3974 55.776 20.583
## + Clarity   1    1.4286 56.745 21.237
## + Body      1    0.3803 57.793 21.933
##
## Step:  AIC=18.25
## Quality ~ Flavor + Oakiness
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts): the
## response appeared on the right-hand side and was dropped

## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts):
## problem with term 6 in model.matrix: no columns are assigned
```

```
##           Df Sum of Sq    RSS    AIC
## + Aroma    1    6.6026 45.853 15.139
## + Clarity  1    2.9416 49.514 18.058
## <none>                  52.456 18.251
## + Region   1    1.3049 51.151 19.294
## + Body     1    0.5356 51.920 19.861
##
## Step:  AIC=15.14
## Quality ~ Flavor + Oakiness + Aroma
```

```
## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts): the
## response appeared on the right-hand side and was dropped

## Warning in model.matrix.default(Terms, m, contrasts.arg = object$contrasts):
## problem with term 6 in model.matrix: no columns are assigned
```

```
##           Df Sum of Sq    RSS    AIC
## <none>                  45.853 15.139
## + Clarity  1   1.69358 44.160 15.709
## + Body     1   0.14769 45.706 17.016
## + Region   1   0.00048 45.853 17.138
```

Based on the forward selection method, the model with the least AIC should contain the predictor variables *Flavor*, *Oakiness*, and *Aroma*.

```r
lm(Quality ~ Flavor + Oakiness + Aroma)
```

```
##
## Call:
## lm(formula = Quality ~ Flavor + Oakiness + Aroma)
##
## Coefficients:
## (Intercept)       Flavor     Oakiness        Aroma
##      6.4672       1.1997      -0.6023       0.5801
```

Using the LSE estimor, we have the linear regression model

$$\hat{Quality} = 6.4672 + 1.1997(Flavor) - 0.6023(Oakiness) + 0.5801(Aroma).$$