1. Consider the bodyfat dataset and consider fitting a linear model for the response variable BODYFAT in terms of the explanatory variables AGE, WEIGHT, HEIGHT, ADIPOSITY, NECK, CHEST, ABDOMEN, HIP, THIGH, KNEE, ANKLE, BICEPS, FOREARM, and WRIST.

(a) Using each of the following methods, perform variable selection to select a subset of the explanatory variables for modeling the response:

i. Backward elimination using the individual $p$-values

ii. Forward selection using $p$-values

iii. Adjusted $R^2$

iv. AIC

v. BIC

vi. Mallow's $C_p$

*Solution.* Given below are functions I have written which return the model selected by each of these methods. The adjusted $R^2$ and Mallow's $C_p$ functions use a backward elimination method to select the best model according to these criteria.

```
setwd("~/Desktop/Stat 151")
body = read.csv("BodyFat.csv")
allvars = c('BODYFAT', 'AGE', 'WEIGHT', 'HEIGHT', 'ADIPOSITY', 'NECK', 'CHEST', 'ABDOMEN',
            'HIP', 'THIGH', 'KNEE', 'ANKLE', 'BICEPS', 'FOREARM', 'WRIST')
data = body[,allvars]

backward = function(vars = allvars, critical = .2) {
  model = lm(BODYFAT ~ ., data[,vars])
  pvalues = summary(model)$coefficients[-1,4]
  var = which(pvalues == max(pvalues))
  if(pvalues[var] > critical) {
    #print(var)
    vars = vars[-(var+1)]
    return(backward(vars, critical))
  } else {
    return(model)
  }
}

forward = function(vars = 1, critical = .2) {
  if (length(vars) == length(allvars))
  {
    return(lm(BODYFAT ~ ., data[,allvars]))
  }

  pvalues = sapply(1:length(allvars), function(var) {
    if(var %in% vars) {
      return(1)
    }
    model = lm(BODYFAT ~ ., data[,allvars[c(vars,var)]])
    p = rev(summary(model)$coefficients[,4])[1]
    return(p[length(p)])
  })
  var = which(pvalues == min(pvalues))
  if(pvalues[var] < critical) {
    #print(var)
    vars = c(vars,var)
    return(forward(vars = vars, critical = critical))
  } else {
    return(lm(BODYFAT ~ ., data[, allvars[vars]]))
  }
}

adjRsqr = function(vars = allvars) {
  ft = lm(BODYFAT ~., data[,vars])
  currentAdjR = summary(ft)$adj.r.squared
```

```
  adjR = sapply(vars[-1], function(var) {
    model = lm(BODYFAT ~., data[,vars[vars != var]])
    adjR = summary(model)$adj.r.squared
    return(adjR)
  })
  var = which(adjR == max(adjR))
  if(adjR[var] > currentAdjR)
  {
    #print(var)
    return(adjRsqr(vars[-(var+1)]))
  } else {
    return(ft)
  }
}

AIC = function() {
  model = lm(BODYFAT ~ ., data[,allvars])
  return(step(model, direction = "both", trace = 0))
}

BIC = function() {
  model = lm(BODYFAT ~ ., data[,allvars])
  return(step(model, direction="both", k = log(nrow(data)), trace = 0))
}

Mallow = function(vars = allvars) {
  n = nrow(data)
  p = length(vars) - 1
  ft = lm(BODYFAT ~ ., data[,vars])
  sigma = summary(ft)$sigma
  currentMallows = p+1

  p = p-1

  mallows = sapply(vars[-1], function(var) {
    model = lm(BODYFAT ~., data[,vars[vars != var]])
    RSS = deviance(model)
    mallow = (RSS / sigma^2) - (n-2-2*p)
    return(mallow)
  })
  var = which(mallows == min(mallows))
  if(mallows[var] < currentMallows)
  {
    #print(var)
    return(Mallow(vars[-(var+1)]))
  } else {
    return(ft)
  }
}
```

The summaries of each model are given below. As expected, the AIC method gave the same results as Mallows' CP, and the BIC method returned a model with less variables than the AIC method.

```
> summary(backward())

Call:
lm(formula = BODYFAT ~ ., data = data[, vars])

Residuals:
     Min       1Q   Median       3Q      Max
-10.0574  -2.7411  -0.1912   2.6929   9.4977

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.06213   10.84654  -1.850  0.06558 .
AGE           0.05922    0.02850   2.078  0.03876 *
WEIGHT       -0.08414    0.03695  -2.277  0.02366 *
NECK         -0.43189    0.20799  -2.077  0.03889 *
ABDOMEN       0.87721    0.06661  13.170  < 2e-16 ***
HIP          -0.18641    0.12821  -1.454  0.14727
```

```
THIGH         0.28644    0.11949   2.397  0.01727 *
FOREARM       0.48255    0.17251   2.797  0.00557 **
WRIST        -1.40487    0.47167  -2.978  0.00319 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 3.965 on 243 degrees of freedom
Multiple R-squared:  0.7467,  Adjusted R-squared:  0.7383
F-statistic: 89.53 on 8 and 243 DF,  p-value: < 2.2e-16


> summary(forward())

Call:
lm(formula = BODYFAT ~ ., data = data[, allvars[vars]])

Residuals:
     Min       1Q   Median       3Q      Max
-10.0574  -2.7411  -0.1912   2.6929   9.4977

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.06213   10.84654  -1.850  0.06558 .
ABDOMEN       0.87721    0.06661  13.170  < 2e-16 ***
WEIGHT       -0.08414    0.03695  -2.277  0.02366 *
WRIST        -1.40487    0.47167  -2.978  0.00319 **
FOREARM       0.48255    0.17251   2.797  0.00557 **
NECK         -0.43189    0.20799  -2.077  0.03889 *
AGE           0.05922    0.02850   2.078  0.03876 *
THIGH         0.28644    0.11949   2.397  0.01727 *
HIP          -0.18641    0.12821  -1.454  0.14727
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 3.965 on 243 degrees of freedom
Multiple R-squared:  0.7467,  Adjusted R-squared:  0.7383
F-statistic: 89.53 on 8 and 243 DF,  p-value: < 2.2e-16


> summary(adjRsqr())

Call:
lm(formula = BODYFAT ~ ., data = data[, vars])

Residuals:
     Min       1Q   Median       3Q      Max
-10.0574  -2.7411  -0.1912   2.6929   9.4977

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.06213   10.84654  -1.850  0.06558 .
AGE           0.05922    0.02850   2.078  0.03876 *
WEIGHT       -0.08414    0.03695  -2.277  0.02366 *
NECK         -0.43189    0.20799  -2.077  0.03889 *
ABDOMEN       0.87721    0.06661  13.170  < 2e-16 ***
HIP          -0.18641    0.12821  -1.454  0.14727
THIGH         0.28644    0.11949   2.397  0.01727 *
FOREARM       0.48255    0.17251   2.797  0.00557 **
WRIST        -1.40487    0.47167  -2.978  0.00319 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 3.965 on 243 degrees of freedom
Multiple R-squared:  0.7467,  Adjusted R-squared:  0.7383
F-statistic: 89.53 on 8 and 243 DF,  p-value: < 2.2e-16


> summary(AIC())

Call:
lm(formula = BODYFAT ~ AGE + WEIGHT + NECK + ABDOMEN + HIP +
    THIGH + FOREARM + WRIST, data = data[, allvars])
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-10.0574  -2.7411  -0.1912   2.6929   9.4977

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.06213   10.84654  -1.850  0.06558 .
AGE           0.05922    0.02850   2.078  0.03876 *
WEIGHT       -0.08414    0.03695  -2.277  0.02366 *
NECK         -0.43189    0.20799  -2.077  0.03889 *
ABDOMEN       0.87721    0.06661  13.170  < 2e-16 ***
HIP          -0.18641    0.12821  -1.454  0.14727
THIGH         0.28644    0.11949   2.397  0.01727 *
FOREARM       0.48255    0.17251   2.797  0.00557 **
WRIST        -1.40487    0.47167  -2.978  0.00319 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 3.965 on 243 degrees of freedom
Multiple R-squared:  0.7467,  Adjusted R-squared:  0.7383
F-statistic: 89.53 on 8 and 243 DF,  p-value: < 2.2e-16


> summary(BIC())

Call:
lm(formula = BODYFAT ~ WEIGHT + ABDOMEN + FOREARM + WRIST, data = data[,
    allvars])

Residuals:
    Min      1Q  Median      3Q     Max
-9.8002 -2.8728 -0.1545  2.8980  8.3845

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -31.29679    6.70886  -4.665 5.06e-06 ***
WEIGHT       -0.12557    0.02292  -5.479 1.05e-07 ***
ABDOMEN       0.92137    0.05192  17.747  < 2e-16 ***
FOREARM       0.44638    0.16822   2.654 0.008480 **
WRIST        -1.39177    0.40991  -3.395 0.000799 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 4.021 on 247 degrees of freedom
Multiple R-squared:  0.7351,  Adjusted R-squared:  0.7308
F-statistic: 171.4 on 4 and 247 DF,  p-value: < 2.2e-16


> summary(Mallow())

Call:
lm(formula = BODYFAT ~ ., data = data[, vars])

Residuals:
     Min       1Q   Median       3Q      Max
-10.0574  -2.7411  -0.1912   2.6929   9.4977

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.06213   10.84654  -1.850  0.06558 .
AGE           0.05922    0.02850   2.078  0.03876 *
WEIGHT       -0.08414    0.03695  -2.277  0.02366 *
NECK         -0.43189    0.20799  -2.077  0.03889 *
ABDOMEN       0.87721    0.06661  13.170  < 2e-16 ***
HIP          -0.18641    0.12821  -1.454  0.14727
THIGH         0.28644    0.11949   2.397  0.01727 *
FOREARM       0.48255    0.17251   2.797  0.00557 **
WRIST        -1.40487    0.47167  -2.978  0.00319 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1
```

```
Residual standard error: 3.965 on 243 degrees of freedom
Multiple R-squared:  0.7467,  Adjusted R-squared:  0.7383
F-statistic: 89.53 on 8 and 243 DF,  p-value: < 2.2e-16
```

□

(b) Let $M_1, \ldots, M_6$ denote the six models selected by each of the six variable selection methods of the previous part. Select one of these models by cross-validation.

*Solution.* Of the 6 models resulting from the methods in part (a), 5 were the same. The only model that differed from the others was that obtained using the BIC criterion. These two models are

$$\begin{aligned} \text{BODYFAT} &= \text{AGE} + \text{WEIGHT} + \text{NECK} + \text{ABDOMEN} + \text{HIP} \\ &\quad + \text{THIGH} + \text{FOREARM} + \text{WRIST} \\ \text{BODYFAT} &= \text{WEIGHT} + \text{ABDOMEN} + \text{FOREARM} + \text{WRIST} \end{aligned}$$

where the second model comes from the BIC method and the first is from the other five methods. The GCV values for these models are 4092.841 and 4152.815, respectively. This suggests that the first model is better. I have also written a script, shown below, that simulates general cross-validation. This was mainly out of curiosity, to check how muc the results would differ from those obtained using GCV. This method gave the same answer, that the first model is better.

Pure cross-validation suggests that the first model is better, so I will choose this model for the next part. However, the scores of the two models are very similar, and in practice it might be useful to choose the smaller model, instead.

```
models = list()
models[[1]] = lm(BODYFAT ~ AGE + WEIGHT + NECK + ABDOMEN + HIP + THIGH + FOREARM + WRIST, data)
models[[2]] = lm(BODYFAT ~ WEIGHT + ABDOMEN + FOREARM + WRIST, data)

GCV = function(model) {
  deviance(model) + 2*(length(model$coefficients))*deviance(model)/252
}

variables = sapply(models, function(x) names(x$coefficients[-1]))
test = list(1:126,127:252)
errs = c(0,0)

prederror = function(trainSet, whichmodel) {
  testSet = (1:252)[-trainSet]
  vars = variables[[whichmodel]]
  model = lm(BODYFAT ~ ., data[trainSet, c("BODYFAT",vars)])
  beta = model$coefficients
  err = sapply(testSet, function(obs){
    x = c(1,as.numeric(data[obs,vars]))
    pred = sum(x*beta)
    actual = data[obs,"BODYFAT"]
    return(pred - actual)
    })
  return(sum(err^2))
}

errs[1] = prederror(test[[1]],1) + prederror(test[[2]],1)
errs[2] = prederror(test[[1]],2) + prederror(test[[2]],2)
#  GCV: 4092.841 4152.815
# errs: 4219.102 4324.321
```

□

(c) Let $M$ be the model selected in the previous part. Fit the model to the data. Perform regression diagnostics. Comment on the validity of the assumptions of the linear model. Identify influential observations and outliers. Delete them if necessary and re-fit the model.

*Solution.* The chosen model is

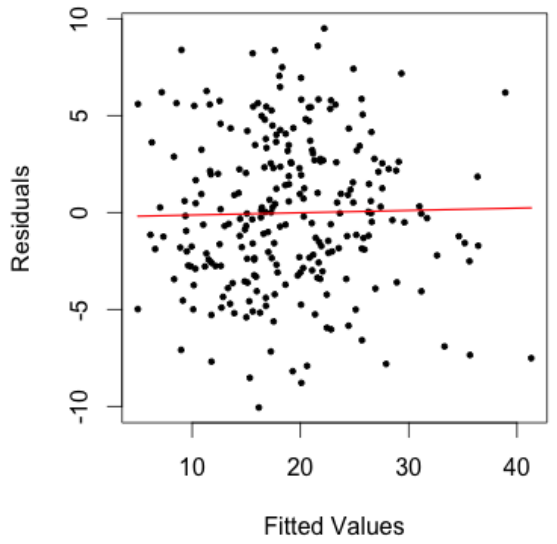
$$\text{BODYFAT} = \text{AGE} + \text{WEIGHT} + \text{NECK} + \text{ABDOMEN} + \text{HIP} + \text{THIGH} + \text{FOREARM} + \text{WRIST}.$$

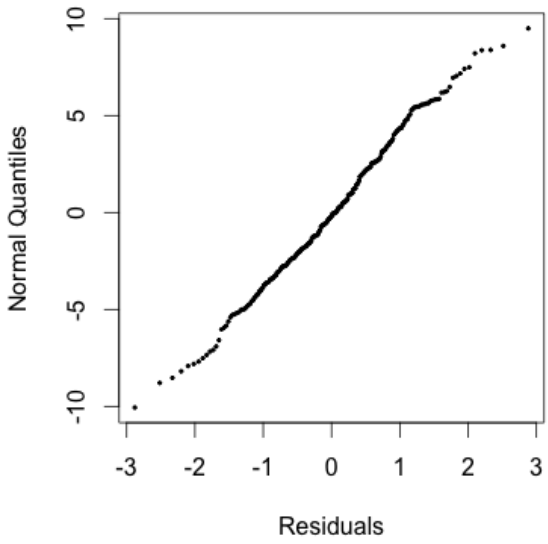
In the script below, I fit the model to the data and performed regression diagnostics. The left plot below, as well as the 14 plots below it, show evidence for constant variance across the fitted values as well as across each explanatory variable. The right plot below shows supports the assumption of the normality of the errors. I have also provided the code used to produce these plots and determine any other observations about the model and the data.

Testing for outliers, using the Bonferroni correction, does not yield any outliers at the .05 significance level. Observations 39 and 224 are influential observations, however. I have given the summary for the model where these observations have been excluded. I would use this model rather than the original.

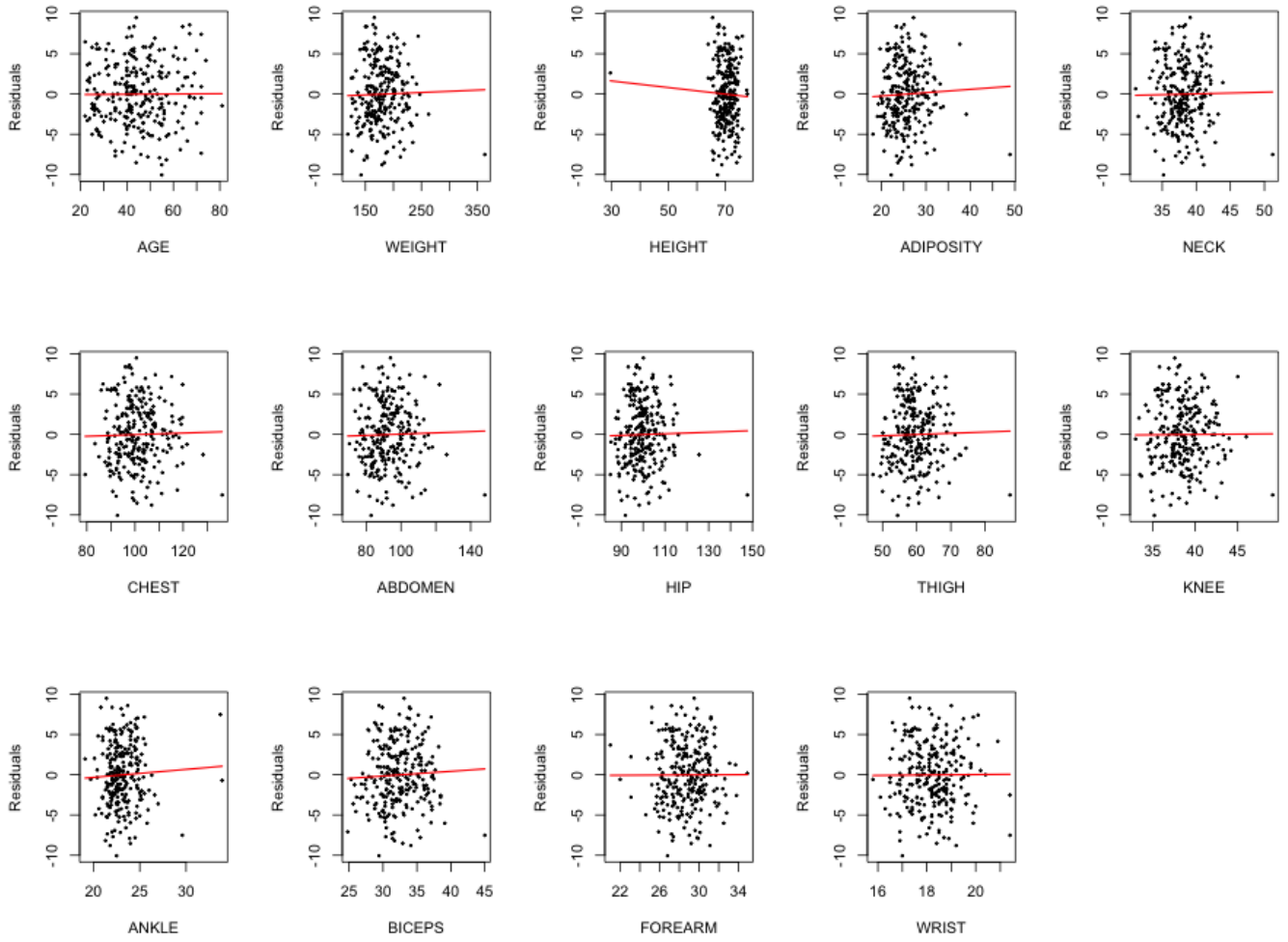
On the last homework, 39 was considered an outlier according to the Bonferroni correction at the .05 significance level. The reason it is not an outlier in this model is that the design matrix has changed, thus its leverage and residual are different. Perhaps the categories in which it was most different from the other observations are not being used in this model.

Overall, the model appears strong. The assumption that the errors are normal with constant variance is not contradicted by this evidence.



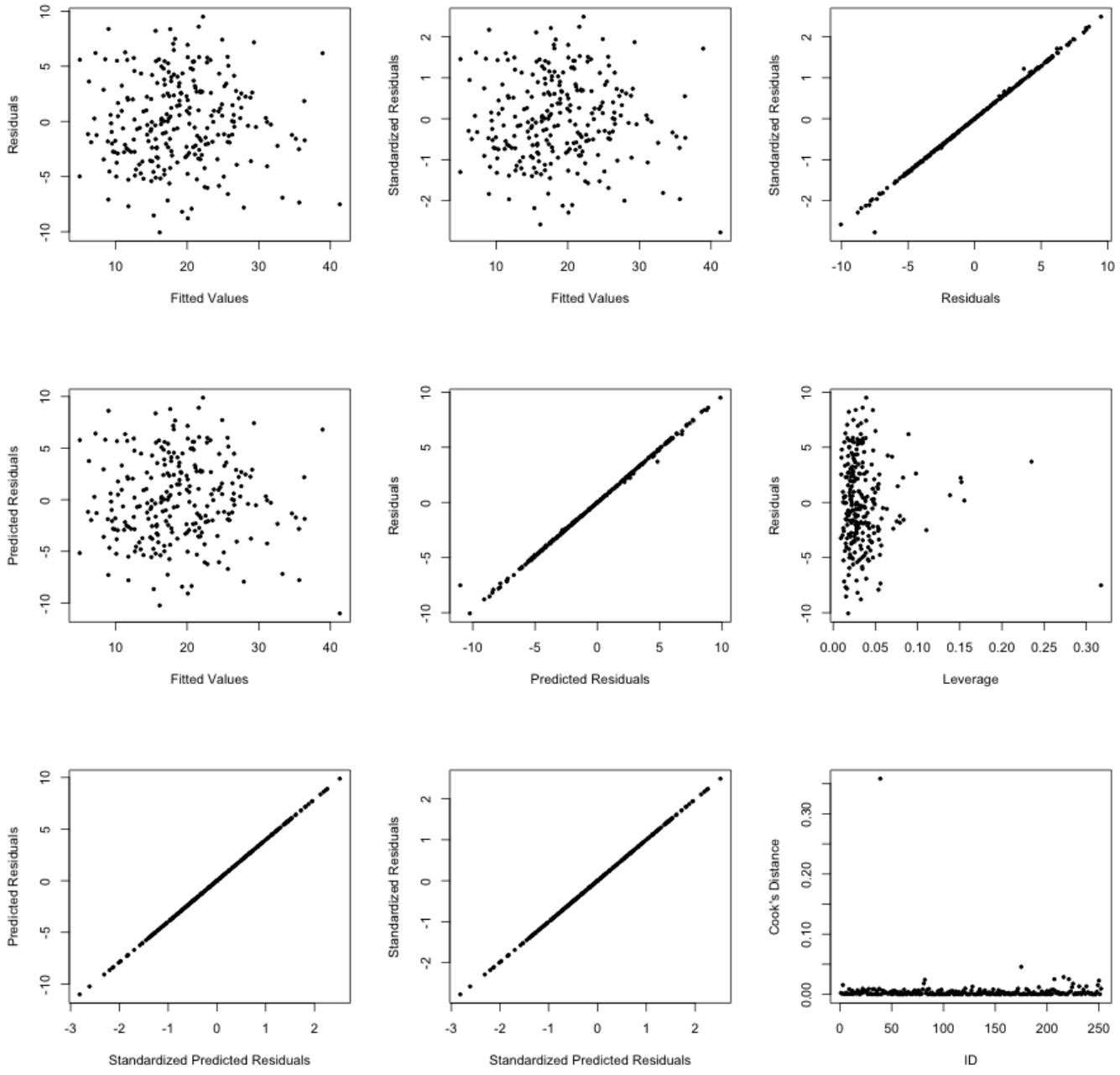

```
ft = lm(BODYFAT ~ AGE + WEIGHT + NECK + ABDOMEN + HIP + THIGH + FOREARM + WRIST, data)
par(mfrow = c(1,2))
scatter.smooth(ft$fitted.values, ft$residuals, span = 1000000, lpars = list(col = 'red', lwd = 1.5)
    ,
```




```
                pch = 19, cex = .5, ylab = "Residuals", xlab = "Fitted Values")
qqnorm(ft$residuals, pch = 19, cex = .3, main = "", ylab = "Normal Quantiles", xlab = "Residuals")
par(mfrow = c(3,5))
for(i in allvars[-1])
{
  scatter.smooth(data[,i], ft$residuals, span = 1000000, lpars = list(col = 'red', lwd = 1.5),
                 pch = 19, cex = .3, ylab = "Residuals", xlab = i)
}

n = nrow(data)
p = 9
x = matrix(c(rep(1, n), as.numeric(as.matrix(data[,names(ft$coefficients)[-1]]))), nrow = n)
h = x %*% solve(t(x)%*% x) %*% t(x)

fit = unname(ft$fitted.values)
res = unname(ft$residuals)
sigma = sqrt(deviance(ft)/df.residual(ft))
stdRes = unname(res / (sigma*(1-diag(h))))
predRes = unname(res / (1 - diag(h)))
stdPredRes = unname(stdRes*sqrt((n-p-1) / (n-p-stdRes^2)))
cook = stdRes^2 * diag(h) / ((1-diag(h)) * (p+1))
par(mfrow = c(3,3))
# Residuals against fitted values
plot(fit, res, pch = 19, cex = .5, xlab = "Fitted Values", ylab = "Residuals")
# Standardized residuals against fitted values
plot(fit, stdRes, pch = 19, cex = .5, xlab = "Fitted Values", ylab = "Standardized Residuals")
# Residuals against Standardized Residuals
plot(res, stdRes, pch = 19, cex = .5, xlab = "Residuals", ylab = "Standardized Residuals")
```




```
# Predicted residuals against fitted values
plot(fit, predRes, pch = 19, cex = .5, xlab = "Fitted Values", ylab = "Predicted Residuals")
# Residuals against predicted residuals
plot(predRes, res, pch = 19, cex = .5, xlab = "Predicted Residuals", ylab = "Residuals")
# Residuals against leverage
plot(diag(h), res, pch = 19, cex = .5, xlab = "Leverage", ylab = "Residuals")
# Predicted residuals against standardized predicted residuals
plot(stdPredRes, predRes, pch = 19, cex = .5, xlab = "Standardized Predicted Residuals", ylab = "
    Predicted Residuals")
# Standardized residuals against standardized predicted residuals
plot(stdPredRes, stdRes, pch = 19, cex = .5, xlab = "Standardized Predicted Residuals", ylab = "
    Standardized Residuals")
# Cook's distance against the ID number of the subjects
plot(body$IDNO, cook, pch = 19, cex = .5, xlab = "ID", ylab = "Cook's Distance")
```

```
bonferroni = .05 / n
pValues = sapply(stdPredRes, function(t) {
  2*(1-pt(abs(t), n-p-1))
})
which(pValues < .01)
# Returns 39 224
which(pValues < bonferroni)
### Returns integer(0)
ftdrop = lm(BODYFAT ~ AGE + WEIGHT + NECK + ABDOMEN + HIP + THIGH + FOREARM + WRIST, data[-c
    (39,224),])
```

This is the summary for the model obtained by removing the 39th and 224th observations:

```
Call:
lm(formula = BODYFAT ~ AGE + WEIGHT + NECK + ABDOMEN + HIP +
    THIGH + FOREARM + WRIST, data = data[-c(39, 224), ])

Residuals:
    Min      1Q  Median      3Q     Max
-8.8086 -2.7012 -0.2672  2.6823  9.4454

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.34883   10.65212  -1.910 0.057280 .
AGE           0.06947    0.02809   2.473 0.014076 *
WEIGHT       -0.07242    0.03658  -1.980 0.048890 *
NECK         -0.34846    0.20756  -1.679 0.094474 .
ABDOMEN       0.84245    0.06672  12.626  < 2e-16 ***
HIP          -0.13094    0.12821  -1.021 0.308135
THIGH         0.27692    0.11753   2.356 0.019269 *
FOREARM       0.35477    0.17874   1.985 0.048293 *
WRIST        -1.59434    0.46618  -3.420 0.000735 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 3.886 on 241 degrees of freedom
Multiple R-squared:  0.7523,  Adjusted R-squared:  0.7441
F-statistic:  91.5 on 8 and 241 DF,  p-value: < 2.2e-16
```

□