Homework 5 Solution

Stephanie DeGraaf

11/08/2018

1. (a) Residuals against fitted values: Check for linearity and heteroskedasticity. We would like to see a random scatter around the zero line. A curved pattern, for example, would indicate a non-linear, curve pattern in the data. We would like to see that the spread of the residuals does not change with the fitted values. A cone-shaped pattern would indicate strong heteroskedasticity.

   (b) Scale-location (square root standardized residuals against fitted values): Check for heteroskedasticity. We would like to see the lowess curve be flat through most of the data. Upward or downward slopes indicate heteroskedasticity.

   (c) Normal Q-Q: Check for normality. If the residuals are normally distributed, they will follow the straight diagonal line.

   (d) Standardized residuals against leverage: Identify influential points. These would be points in the upper and lower right corners, which have large residuals and large leverages. Cook's distance is included to indicate thresholds for classifying potential points as outliers.

2.

```
bodyfat<- read.csv("Bodyfat.csv")
bodyfat<- bodyfat[,setdiff(colnames(bodyfat), "Density")]

# backward elimination:
sub_bodyfat = bodyfat
stop = 0
while(stop == 0){
  model<- lm(bodyfat ~ ., data = sub_bodyfat)
  pvals=summary(model)$coefficients[-1,4] # get pvalues just for the explanatory variables
  max_var<- names(which.max(pvals))
  max_pval<- max(pvals)
  if (max_pval<0.05){
    stop =1
  } else{
    sub_bodyfat<- sub_bodyfat[, setdiff(colnames(sub_bodyfat), max_var)]
  }
}
be_model<- model
# backward elimination variables:
back_elim_vars<- names(be_model$coefficients)[-1]
back_elim_vars
```

```
## [1] "Weight"  "Abdomen" "Forearm" "Wrist"
```

```
# forward selection:
candidate_variables<- setdiff(colnames(bodyfat), "bodyfat")
selected_variables<- NULL
stop = 0
while(stop == 0){
  var_pvals = rep(NA, length(candidate_variables))
  names(var_pvals) = candidate_variables
  for (var in candidate_variables){
    dataset = bodyfat[,c("bodyfat", selected_variables, var)]
```

```
    var_model<- lm(bodyfat ~ ., data = dataset)
    var_pvals[var]<- summary(var_model)$coefficients[,4][var]
  }
  if(min(var_pvals) > 0.05){
    stop = 1
  }else{
    best_var<- names(which.min(var_pvals))
    selected_variables<- c(selected_variables, best_var)
    candidate_variables<- setdiff(candidate_variables, best_var)
  }
}
# forward selection variables:
forward_sel_vars<- selected_variables
forward_sel_vars
```

```
## [1] "Abdomen" "Weight"  "Wrist"   "Forearm"
```

```
library(leaps)
# adjusted Rsquared
leaps_output<- leaps(x=bodyfat[,setdiff(colnames(bodyfat), "bodyfat")],
                     y = bodyfat$bodyfat,
                     method = "adjr2", nbest = 1)
which_vars<- leaps_output$which[which.max(leaps_output$adjr2),]
# adjusted Rsquared variables:
adj_rsq_vars<- setdiff(colnames(bodyfat), "bodyfat")[which_vars]
adj_rsq_vars
```

```
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Hip"     "Thigh"   "Biceps"
## [8] "Forearm" "Wrist"
```

```
# AIC
n = nrow(bodyfat)
p = ncol(bodyfat)-1
# get all 2^p possible models:
possible_models <- expand.grid(rep(list(0:1), p))
x_variables<- colnames(bodyfat)[-1]
aic<- rep(NA, nrow(possible_models))
model1<- lm(bodyfat ~ 1, data = bodyfat)
aic[1]<- n*log(deviance(model1)/n) + 2*(1 + 0)
for (m in 2:nrow(possible_models)){
  sel_vars<- x_variables[as.logical(possible_models[m,])]
  model_input<- paste("bodyfat ~", paste(sel_vars, collapse=" + "))
  model<- lm(model_input, data = bodyfat)
  p_model<- length(sel_vars)
  aic[m]<- n*log(deviance(model)/n) + 2*(1 + p_model)
}
best_model<- possible_models[which.min(aic),]
# AIC selected variables:
aic_vars<- x_variables[as.logical(best_model)]
aic_vars
```

```
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Hip"     "Thigh"   "Forearm"
## [8] "Wrist"
```

```
# BIC
bic<- rep(NA, nrow(possible_models))
```

```
model1<- lm(bodyfat ~ 1, data = bodyfat)
bic[1]<- n*log(deviance(model1)/n) + (log(n))*(1 + 0)
for (m in 2:nrow(possible_models)){
  sel_vars<- x_variables[as.logical(possible_models[m,])]
  model_input<- paste("bodyfat ~", paste(sel_vars, collapse=" + "))
  model<- lm(model_input, data = bodyfat)
  p_model<- length(sel_vars)
  bic[m]<- n*log(deviance(model)/n) + (log(n))*(1 + p_model)
}
best_model<- possible_models[which.min(bic),]
# BIC selected variables:
bic_vars<- x_variables[as.logical(best_model)]
bic_vars
```

```
## [1] "Weight"  "Abdomen" "Forearm" "Wrist"
```

```
# Mallow's Cp
leaps_output<- leaps(x=bodyfat[,setdiff(colnames(bodyfat), "bodyfat")],
                     y = bodyfat$bodyfat,
                     method = "Cp", nbest = 1)
which_vars<- leaps_output$which[which.min(leaps_output$Cp),]
# Mallow's Cp variables:
mallows_cp_vars<- setdiff(colnames(bodyfat), "bodyfat")[which_vars]
mallows_cp_vars
```

```
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Thigh"   "Forearm" "Wrist"
```

```
library(caret)
folds = createFolds(bodyfat$bodyfat, k = 10)
get_CV<- function(vars){
  MSE_folds<- rep(NA, length(folds))
  for (f in 1:length(folds)){
    mod<- lm(paste("bodyfat ~", paste(vars, collapse=" + ")), data = bodyfat[-folds[[f]],])
    pred<- predict(mod, bodyfat[folds[[f]],])
    true_y<- bodyfat[folds[[f]], "bodyfat"]
    MSE_folds[f] = 1/length(folds[[f]]) * sum((pred-true_y)^2)
  }
  MSE = mean(MSE_folds)
  return(MSE)
}

vars_list<- list("M1" = back_elim_vars, "M2" = forward_sel_vars, "M3" = adj_rsq_vars,
                 "M4" = aic_vars, "M5" = bic_vars, "M6" = mallows_cp_vars)
model_cvs<- unlist(lapply(vars_list, get_CV))
model_cvs
```

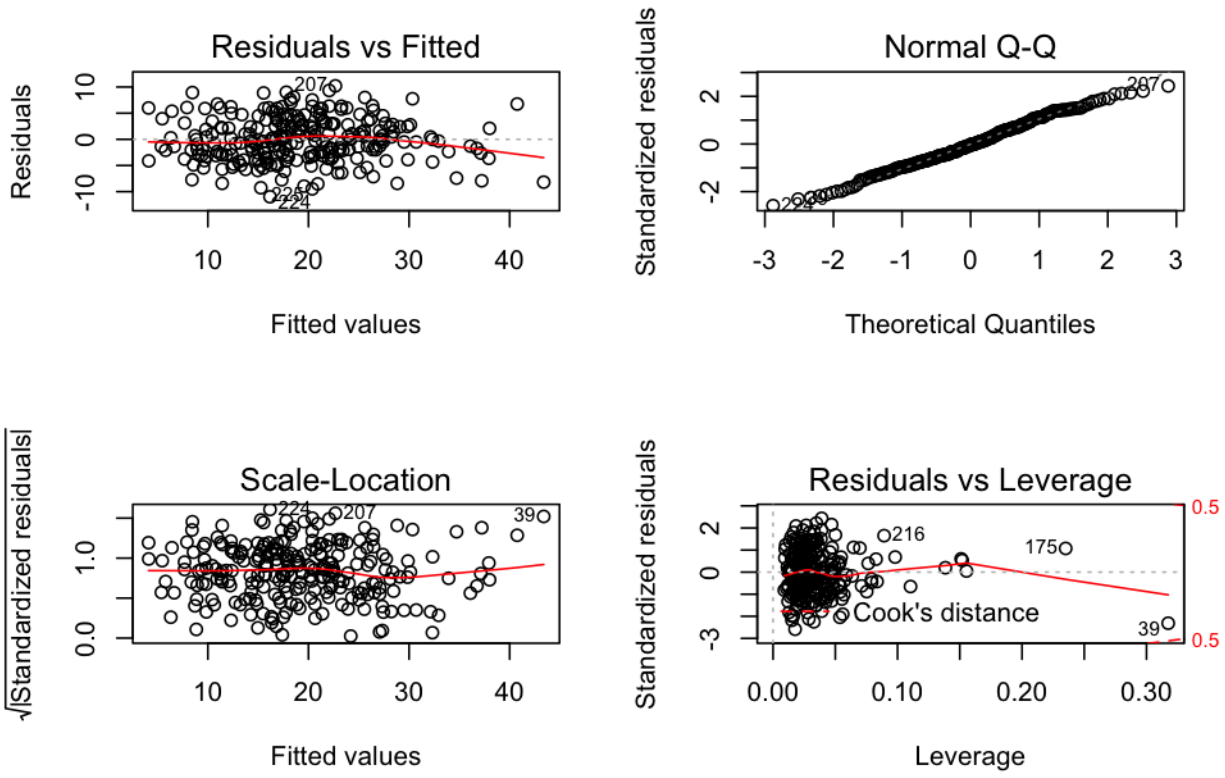
```
##       M1       M2       M3       M4       M5       M6
## 19.39743 19.39743 19.37306 19.22093 19.39743 19.32421
```

```
which.min((model_cvs))
```

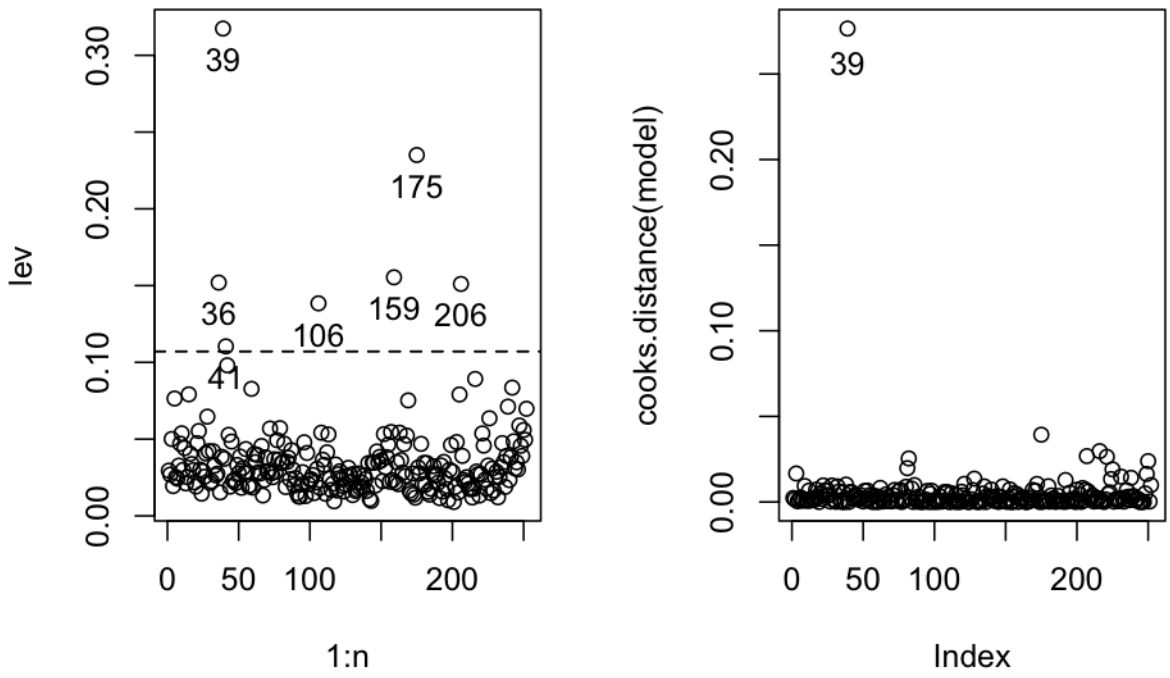
```
## M4
##  4
```

```
sel_vars<- vars_list[[which.min(model_cvs)]]
model = lm(paste("bodyfat ~", paste(sel_vars, collapse=" + ")), data = bodyfat)
```

```
par(mfrow = c(2,2))
plot(model)
```




```
lev<- hatvalues(model)
lev.sorted <- sort(lev, decreasing=T, index.return=T)
par(mfrow = c(1,2))
plot(1:n, lev)
abline(h=3*mean(lev), lty=2)
for(i in lev.sorted$ix[1:7]) {
  text(i, lev[i]-0.02, rownames(bodyfat)[i])
}
abline(h=3*mean(lev), lty=2)
plot(cooks.distance(model))
text(which.max(cooks.distance(model)), max(cooks.distance(model))-0.02,
     which.max(cooks.distance(model)))
```




```
# remove 39 and 175
outliers<- c(39,175)
new_model<- lm(paste("bodyfat ~", paste(aic_vars, collapse=" + ")), data = bodyfat[-outliers,])
summary(new_model)
```

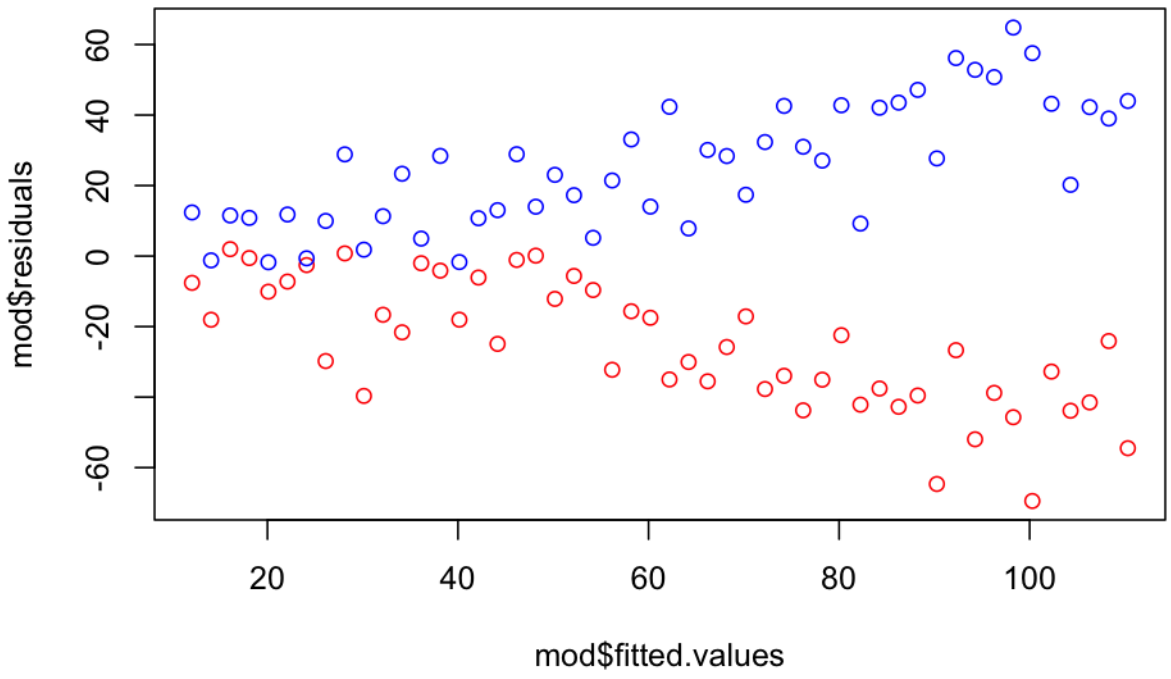
```
##
## Call:
## lm(formula = paste("bodyfat ~", paste(aic_vars, collapse = " + ")),
##     data = bodyfat[-outliers, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.765  -2.907  -0.280   2.902  10.185
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -23.61973   11.66629  -2.025 0.044010 *
## Age           0.07367    0.03090   2.384 0.017882 *
## Weight       -0.07655    0.04000  -1.914 0.056867 .
## Neck         -0.38378    0.22883  -1.677 0.094809 .
## Abdomen       0.91029    0.07296  12.476  < 2e-16 ***
## Hip          -0.13611    0.14051  -0.969 0.333664
## Thigh         0.27670    0.12854   2.153 0.032340 *
## Forearm       0.43052    0.22477   1.915 0.056631 .
## Wrist        -1.73658    0.51979  -3.341 0.000968 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.249 on 241 degrees of freedom
## Multiple R-squared:  0.7482, Adjusted R-squared:  0.7398
## F-statistic: 89.51 on 8 and 241 DF,  p-value: < 2.2e-16
```

3.

```
X = c(1:50,1:50)
D = rep(c(0,1), each = 50)
means<- 10 + X + D + 2*X*D
y<- sapply(means, function(mu) rnorm(1, mean = mu, sd = 10))

mod<- lm(y ~ X)
plot(mod$fitted.values, mod$residuals,
     col = rep(c("red","blue"), each = 50))
```



No, the variance of the residuals is not constant. This is because in the true model, we have two lines with different slopes that we are trying to estimate using only one line. The D variable here plays the role of a switch, so that conditional on D, Y will be one of two different lines. Our residuals are showing what happens when we fit one line between these two lines.

4. Let $y^{(j)} = Res(y, X^{-j})$ and $X^{(j)} = Res(X, X^{-j})$. We are interested in the regression $y^{(j)} \sim X^{(j)}$, which has slope $\hat{\beta}_j$ and residuals $e^{(j)}$. We want to show $e^{(j)} = \hat{e}$.

By definition, $e^{(j)} = y^{(j)} - X^{(j)}\hat{\beta}_j$. Let $H(-j) = X^{-j}(X^{-j^T}X^{-j})^{-1}X^{-j^T}$, the hat matrix of X without the jth column. Then $y^{(j)} = (I - H(-j))y$ and $X^{(j)} = (I - H(-j))X_j$, so we can rewrite

$$e^{(j)} = (I - H(-j))y - (I - H(-j))X_j\hat{\beta}_j.$$

We would like to write this in terms of Hy. Let $\hat{\beta}^{-j}$ be the OLS coefficients without $\hat{\beta}_p$. By definition,

$$\begin{aligned} Hy &= X\hat{\beta} \\ &= \hat{\beta}_0 1_n + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p \\ &= X^{-j}\hat{\beta}^{-j} + \hat{\beta}_j X_j. \end{aligned}$$

We know that $\hat{\beta}$ minimizes $|||y - X\beta||^2 = ||(y - \hat{\beta}_j X_j) - X^{-j}\beta^{-j}||$, so we can write

$$\hat{\beta}^{-j} = ({X^{-j}}^T X^{-j})^{-1} {X^{-j}}^T (y - \hat{\beta}_j X_j).$$

Substituting this equation for $\hat{\beta}^{-j}$ into the previous equation for Hy gives

$$\begin{aligned} Hy &= X^{-j}({X^{-j}}^T X^{-j})^{-1} {X^{-j}}^T (y - \hat{\beta}_j X_j) + \hat{\beta}_j X_j \\ &= H(-j)(y - \hat{\beta}_j X_j) + \hat{\beta}_j X_j \\ &= H(-j)y + \hat{\beta}_j (I - H(-j)) X_j \\ \Rightarrow (I - H(-j))y &= (I - Hy) + \hat{\beta}_j (I - H(-j)) X_j. \end{aligned}$$

Finally, we can substitute this equation into our equation for $e^{(j)}$:

$$\begin{aligned} e^{(j)} &= (I - H(-j))y - (I - H(-j))X_j\hat{\beta}_j \\ &= (I - Hy) + \hat{\beta}_j (I - H(-j)) X_j - (I - H(-j))X_j\hat{\beta}_j \\ &= I - Hy \\ &= \hat{e}. \end{aligned}$$

5. (a) False. This could happen, but it will not always happen. This can easily be shown by example.

   (b) False. By definition, $Cov(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$. An observation with high leverage increases the $X^T X$ term, thus decreasing $Cov(\hat{\beta})$. Therefore, removing a high leverage point will increase $Cov(\hat{\beta})$. Precision is the inverse of the covariance; thus, if the covariance increases, the precision decreases. Removing a high leverage term increases the covariance, so it will decrease precision.

   (c) True. For a fixed number of parameters, the best model is chosen by minimizing the RSS, either for AIC or Mallow's Cp. The two methods have different penalties for the number of covariates, but if the number of covariates is the same, then the AIC and Mallow's Cp will pick the same model with that many covariates.

   (d) False. R^2 always improves for models with higher numbers of covariates. We should use the adjusted R^2 instead, which penalizes for adding more covariates.

6. Ant colonies report.