Homework 3

Brandon Amaral, Monte Davityan, Nicholas Lombardo, Hongkai Lu

2022-09-16

1 Iterative Technique for Finding Roots of `glm` Coefficients

We can use an iterative technique, such as the Fisher Scoring method to find the Maximum Likelihood Estimates for the coefficients of the logistic regression model.

Assuming $y_i \sim Bin(n_i, \pi_i)$ and $\log\left(\frac{\pi_i}{1-\pi_i}\right) = x_i^T\boldsymbol{\beta}$, i.e. applying the logistic regression model to predict y, we find the likelihood with respect to $\boldsymbol{\beta}$ to be proportional to

$$\prod_{i=1}^{N}\left(e^{x_i^T\boldsymbol{\beta}y_i}\right)\left(1+e^{x_i^T\boldsymbol{\beta}}\right)^{-n_i}.$$

Then, to maximize this with respect to the vector $\boldsymbol{\beta}$, we can use the iterative Fisher Scoring method as follows:

1. Set initial values $\boldsymbol{\beta}_0$ for $\boldsymbol{\beta}$
2. Compute the Observed Fisher Information $\mathbf{I}$ (i.e. the negative of the Hessian $\nabla^2\ell$) and gradient $\nabla\ell$ of the log-likelihood $\ell(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ using the initial values $\boldsymbol{\beta}_0$
3. Compute
$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 + \mathbf{I}^{-1}(\boldsymbol{\beta}_0)\nabla\ell(\boldsymbol{\beta}_0)$$
4. Perform a local line search to find $\boldsymbol{\beta}_{1_L}$ such that $\ell(\boldsymbol{\beta}_{1_L}) > \ell(\boldsymbol{\beta}_1)$, set $\boldsymbol{\beta}_1$ equal to $\boldsymbol{\beta}_{1_L}$
5. Repeat until convergence, i.e. compute
$$\boldsymbol{\beta}_{m+1} = \boldsymbol{\beta}_m + \mathbf{I}^{-1}(\boldsymbol{\beta}_m)\nabla\ell(\boldsymbol{\beta}_m)$$
   repeatedly until a desired amount of accuracy is achieved.

2 Forest Fires

Below we have the first 6 rows of our data set. Note that we have 10 numerical predictors and 2 categorical predictors for the response `area`.

X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0
7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0
7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0
8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0
8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0
8	6	aug	sun	92.3	85.3	488.0	14.7	22.2	29	5.4	0.0	0

Below, we show a few summary statistics for the data as well.

X	Y	month	day	FFMC	DC	ISI
Min. :1.000	Min. :2.0	aug :184	fri:85	Min. :18.70	Min. : 7.9	Min. : 0.000
1st Qu.:3.000	1st Qu.:4.0	sep :172	mon:74	1st Qu.:90.20	1st Qu.:437.7	1st Qu.: 6.500
Median :4.000	Median :4.0	mar : 54	sat:84	Median :91.60	Median :664.2	Median : 8.400
Mean :4.669	Mean :4.3	jul : 32	sun:95	Mean :90.64	Mean :547.9	Mean : 9.022
3rd Qu.:7.000	3rd Qu.:5.0	feb : 20	thu:61	3rd Qu.:92.90	3rd Qu.:713.9	3rd Qu.:10.800
Max. :9.000	Max. :9.0	jun : 17	tue:64	Max. :96.20	Max. :860.6	Max. :56.100
		(Other): 38	wed:54			

temp	RH	wind	rain	area	DMC
Min. : 2.20	Min. : 15.00	Min. :0.400	Min. :0.00000	Min. : 0.00	Min. : 1.1
1st Qu.:15.50	1st Qu.: 33.00	1st Qu.:2.700	1st Qu.:0.00000	1st Qu.: 0.00	1st Qu.: 68.6
Median :19.30	Median : 42.00	Median :4.000	Median :0.00000	Median : 0.52	Median :108.3
Mean :18.89	Mean : 44.29	Mean :4.018	Mean :0.02166	Mean : 12.85	Mean :110.9
3rd Qu.:22.80	3rd Qu.: 53.00	3rd Qu.:4.900	3rd Qu.:0.00000	3rd Qu.: 6.57	3rd Qu.:142.4
Max. :33.30	Max. :100.00	Max. :9.400	Max. :6.40000	Max. :1090.84	Max. :291.3

(a) We fit a linear regression with `area` as the response with all other variables as predictors. Note that we use a log-transformed response for the model since the original data is heavily skewed toward 0.

```r
# Response area (multiple regression) (Log-Transformed Model)
lm.fit.transformed <- lm(log(area + 1) ~ ., data = fire )
```

Based on the residual plots below, this model seems to be a poor model. The residual vs fit plot shows a trend in predictions, normality doesn't seem to be met especially at the tails, the scale location plot provides evidence for non- constant variance and some points are higher in leverage.

In general, the individual predictor variable effects the response (area) by leaving all else constant, a one unit increase in that predictor, effects the response (area) multiplicatively by e^{β_i}. For example: monthaug, leaving all else constant, if the month is august, will multiplicativly increase the area by $e^{0.3274391}$. Another example: ISI, leaving all else constant, a one unit increase in ISI will decrease area by a factor of $e^{0.0147970}$.

Table 4: Linear regression model coefficients

	Coefficient		Coefficient
(Intercept)	-0.5705460	daymon	0.1457734
X	0.0524204	daysat	0.3099153
Y	-0.0184700	daysun	0.2109897
monthaug	0.3274391	daythu	0.0722394
monthdec	2.2050797	daytue	0.3222933
monthfeb	0.1886078	daywed	0.1978808
monthjan	-0.3163816	FFMC	0.0074547
monthjul	0.0991694	DMC	0.0041790
monthjun	-0.2862231	DC	-0.0020052
monthmar	-0.3416243	ISI	-0.0147970
monthmay	0.7175267	temp	0.0360374
monthnov	-1.1031443	RH	0.0006673
monthoct	0.8232625	wind	0.0603127
monthsep	0.9934196	rain	0.0309440

```
plot(lm.fit.transformed)
```

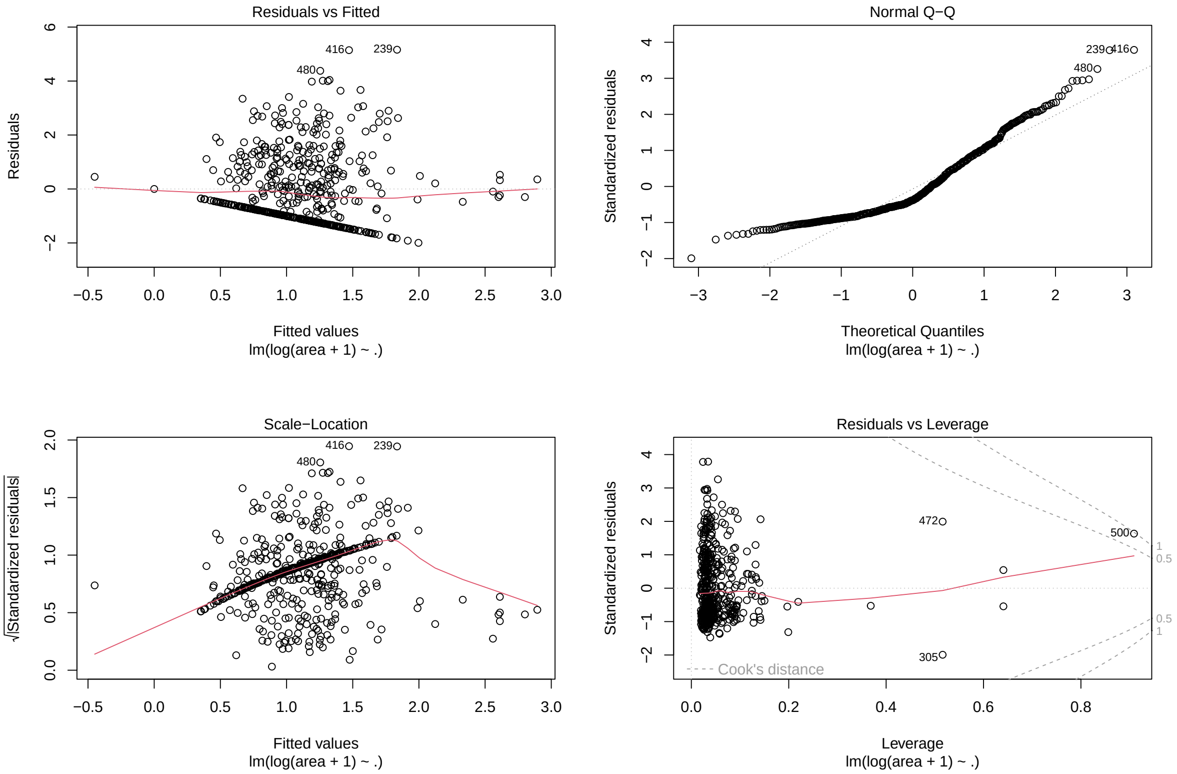


Table 5: GLM Coefficients

	Coefficient		Coefficient
(Intercept)	4.6436182	daymon	-0.1330900
X	-0.0583771	daysat	-0.0663623
Y	-0.0413436	daysun	0.0126375
monthaug	0.2073997	daythu	0.0036450
monthdec	-16.8244177	daytue	-0.2725271
monthfeb	-0.4220374	daywed	-0.3474402
monthjan	15.0496593	FFMC	-0.0314613
monthjul	0.1292420	DMC	0.0011376
monthjun	0.3762216	DC	-0.0004078
monthmar	0.4896616	ISI	0.0159075
monthmay	-0.0085826	temp	-0.0486107
monthnov	16.3096513	RH	-0.0058511
monthoct	1.0053354	wind	-0.0803579
monthsep	-0.0050524	rain	-0.0068858

(b) We fit a logistic regression in which the binarized version of `area` is the response.

```
# Binary area
fire <- fire %>%
  mutate(binary_area = if_else(area != 0, "Not zero", "Zero")) %>%
  mutate(binary_area = as.factor(binary_area))

glm.fit <- glm(binary_area ~ ., family = "binomial", data = fire %>% select(-area))

kable(list(glm.fit$coefficients[1:14],glm.fit$coefficients[15:28]),
      col.names = "Coefficient", booktabs = TRUE, caption = "GLM Coefficients")
```

In general, the individual predictor variable effects the response (area) by leaving all else constant, a one unit increase in that predictor, effects the response (area) by its coefficient estimate. For example: wind, leaving all else constant, with a one unit increase in wind, will decrease the log odds of area being non-zero by 0.08036.

(c) We use the linear model to predict responses for rows 15-25 and report the MSE:

```
# log transformed response model
pred.transformed <- predict(lm.fit.transformed, newdata = fire[15:25,])

actual <- fire[15:25,] %>% select(area)

MSE.transformed <- mean(unlist((pred.transformed - actual)^2))
MSE.transformed
```

```
## [1] 1.573042
```

(d) We select the best model from all the submodels of the linear model with four predictors, using randomly split data with 80% training and 20% validation.

```
fire2 <- fire %>% mutate(area = log1p(area))

training_ind <- sample(1:517, floor(.8*517), replace = FALSE)

train <- fire2[training_ind, c("temp","RH","wind","rain", "area")]
val <- fire2[-training_ind, c("temp","RH","wind","rain", "area")]

model.combos <- function(model,
                         response,
                         predictors,
                         training,
                         validation,
                         null.model = TRUE,
                         debug = FALSE) {
  "This function iterates through all possible combos (2^n)-1 of
    predictor variables and outputs the model with the best validation
    MSE

    Inputs:
      - model: A supervised regression learning model object (ie. lm)
          Expecting the function to have formula and data parameters
      - response: A string. The name of the response variable
      - predictors: A vector of strings. The list of names of the
          predictor variables
      - training: A data frame. The training set
      - validation: A data frame. The validation set
      - null.model: A boolean. (Optional). If TRUE, will compare the initial best
          model as the null model (which is just the average). If set to FALSE,
          the intial best model will be null and only the (2^n)-1 combos will be
          considered
      - debug: A boolean. (Optional). If TRUE print out the best.MSE and
          best.predictions as they are being calculated

    Output:
      - best.predictors: A list. Containing the names of the best
        predictors (as measured by lowest validation MSE)
      - best.mse: A number. The best MSE among the models
  "
  # Combinations is made from this code:
  # https://stackoverflow.com/questions/40049313/generate-all-combinations-of-all-
  # lengths-in-r-from-a-vector
  # It generates the combinations of predictors as a list of strings
  combinations <- do.call("c", lapply(seq_along(predictors),
                              function(i) combn(predictors, i, FUN = list)))

  if (debug) {
    print(combinations)
  }

  # The validation response
  actual <- validation[[response]]
```

```r
  # If null.model is TRUE, compare to the null model (the average)
  if (null.model) {
    average.model <- mean(training[[response]])
    best.MSE <- mean((average.model - actual)^2)
    best.predictions <- c("null model")

    if (debug) {
      print(best.MSE)
      print("null")
    }
  }
  # If null.model is FALSE, compare only the (2^n)-1 models
  else {
    best.MSE <- NA
    best.predictions <- NA
  }


  # Loop through the combinations
  for (combo in combinations) {
    # Make the formula from the combos and the response
    formula <- reformulate(combo, response = response)
    # Fit the model using the given model and training data
    model.fit <- model(formula = formula, data = training)
    # Make predictions on the validation set
    preds <- predict(model.fit, newdata = validation)
    # Calculate validation MSE using the predictions
    MSE.validation <- mean(unlist((preds - actual)^2))
    # Update the best.predictions if the MSE is lower then the prior
      # best
    if (debug) {
      print(MSE.validation)
      print(combo)
    }

    if (is.na(best.MSE) || MSE.validation < best.MSE) {
      best.MSE <- MSE.validation
      best.predictions <- combo
    }
  }

  # Return best.MSE and best.predictions
  return(list(Best.MSE = best.MSE, Best.Combination = best.predictions))
}

model.combos(lm, "area", c("temp","RH","wind","rain"),
             train, val, TRUE, FALSE)
```

```
## $Best.MSE
## [1] 1.674681
##
## $Best.Combination
## [1] "wind"
```

```r
# Run the function above multiple times to showcase based on different train/
# test splits, the output combination of models is different (ie. the predictor
# variables used may not be relevant (according to this model) to the response)

library(hash)

best.combinations.all <- hash()

numIters <- 100
numCombosMadeAsBest <- 0

preds <- c("temp","RH","wind","rain", "area")

for (i in 1:numIters) {
  training_ind <- sample(1:517, floor(.8*517), replace = FALSE)

  train <- fire2[training_ind, preds]
  val <- fire2[-training_ind, preds]
  res <- model.combos(lm, "area", c("temp","RH","wind","rain"),
                      train, val, TRUE, FALSE)

  combo <- paste(res$Best.Combination, collapse=" ")
  if (is.null(best.combinations.all[[combo]])) {
    best.combinations.all[[combo]] <- TRUE
    numCombosMadeAsBest = numCombosMadeAsBest + 1
  }
}
```

There are 13 combos of predictors that were selected as the best out of 16 possible combinations.

Chapter 3 Summary:

Section 3.1:

We study linear regression as a jumping off point for supervised learning. Although linear regression is a very simple approach which has been around for a long time, it remains a useful tool in most arsenals for predicting quantitative responses. Linear regression can answer questions such as, “Is there a relationship between predictors and response,” “How strong is the relationship between predictors and response,” “Which predictors are associated with the response,” “How large is the association between each predictor and the response,” “How accurately can we predict the response,” “Is the relationship linear,” and “Is there synergy/interaction between predictors?”

Simple linear regression is a “straightforward approach for predicting a quantitative response Y on the basis of a single predictor variable X.” It assumes there is approximately a linear relationship between X and Y, notated $Y \approx \beta_0 + \beta_1 X$. Sometimes, we use the wording “regressing Y onto X. We use β_0 and β_1 as unknown constants that represent the intercept and slope terms of this linear model. We call them model”coefficients” or “parameters.” We use training data to product estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, we can predict future responses for a particular set of inputs x by computing $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$, where the hat ˆsymbol denotes an estimated value of the coefficient or response.

Estimating the Coefficients

In real situations, our model coefficients are unknown, so we must use data to estimate the coefficients. We want a pair of $\hat{\beta}_0$ and $\hat{\beta}_1$ so the linear model fits the data well, such that $y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x$ for each i. How do we measure this “closeness”? The most common approach is minimizing the “least squares” criterion. To do this, let $\hat{y} \approx \hat{\beta}_0 + \hat{\beta}_1 x$ be the prediction for Y based on the ith value of X. Then $e_i = y_i - \hat{y_i}$ represents the ith residual, the difference between the ith observed response value and the ith response value that is predicted by our linear model. Then, we define the “residual sum of squares” (RSS) as $RSS = e_1^2 + e_2^2 + ... + e_n^2$. Our Least Squares approach seeks to minimize RSS by choosing the best $\hat{\beta}_0$ and $\hat{\beta}_1$. Calculus can show us that the minimizers are

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_1 - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

Assessing the Accuracy of the Coefficient Estimates

Since, in a linear model, we assume that the true relationship between X and Y takes the form $Y = f(X) + \epsilon$ for an unknown function f, where ϵ is a mean-zero error term, we use the following relationship to approximate f:

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

This model “defines the population regression line, which is the best linear approximation to the true relationship between X and Y. We use β_0 as the intercept term, the expected value of Y when $X = 0$, and we use β_1 as the slope, the average increase in Y associated with a single unit increase in X. The least squares line is characterized by the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ defined above.

Linear regression includes an unbiased process of estimating the population mean, since on average, we expect $\hat{\mu}$ to equal μ. Although one particular set of observations $y_1, ..., y_n$ might lead $\hat{\mu}$ to over- or underestimate μ, if we take the average of a large number of $\hat{\mu}$ estimates, the average would tend toward exactly μ. “Hence, an unbiased estimator does not systematically over- or underestimate the true parameter.” The least squares coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ defined above benefit from being unbiased as well.

Even though we have established that the average of many $\hat{\mu}$’s will tend toward exactly μ over many data sets, we still want to know how “far off” any single estimate of $\hat{\mu}$ will be. We generally use the standard error to answer this question, written $SE(\hat{\mu})$. The variance formula $Var(\hat{\mu}) = SE(\hat{\mu})^2 = \frac{\sigma^2}{n}$ where σ is the standard deviation of each individual realization y_i of Y. This standard error tells us roughly how far each estimate of $\hat{\mu}$ will deffer from the actual value of μ. Notice that as n, increases, the standard error shrinks, so we expect our predictions to become increasingly accurate with larger sample sizes. Similarly, we can compare how close $\hat{\beta}_0$ and $\hat{\beta}_1$ are to the “true” values β_0 and β_1. The have their own formulas which depend on $\sigma^2 = Var(\epsilon)$.

$$SE(\hat{\beta}_0) = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right]$$

$$SE(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

We should notice that $SE(\hat{\beta}_1)$ becomes smaller as x_i become more spread out; “we have more leverage to estimate a slope when this is the case.”

Although we need to make a typically untrue assumption that each error term ϵ_i has a common variance σ^2 for each observation, the above formulas are typically good approximations even if the assumption isn’t particularly valid. In general though, σ^2 is not known, but can be estimated from the data. We call this estimate the residual standard error RSE, given by the formula $RSE = \sqrt{RSS/(n-2)}$.

The standard errors defined above can be used for computing ranges of values where the estimate has a given likelihood to lie, called confidence intervals. For example, a 95% condience intervals is the range of values that will contain the true unknown value of a parameter with 95% probability. The following property also holds for a 95% confidence interval: “if we take repeated samples and construct the confidence interval for each sample, 95% of the intervals will contain the true unknown value of the parameter.”

We can also use standard errors to perform a hypothesis test on the coefficients. Hypothesis tests usually include testing a "null hypothesis" H_0 versus "alternative hypothesis" H_a. Typically, we have H_0: "There is no relationship between X and Y, so $\beta_1 = 0$" and H_a: "There is some relationship between X and Y, so $\beta_1 \neq 0$." We typically use a p-value derived from a t-statistic which measures the number of standard deviations that $\hat{\beta_1}$ is away from 0 (or the value used in the hypotheses):

$$t = \frac{\hat{\beta_0} - 0}{SE(\hat{\beta_1})}$$

The p-value is the probability of observing any number equal to $|t|$ or larger, assuming $\beta_1 = 0$. When we see a notably small p-value (decided ahead of time, often less than 5% or 1%), we infer there is an association between the predictor and the response and subsequently reject the null hypothesis; we can declare a relationship exists between X and Y.

Assessing the Accuracy of the Model

If we have rejected the null hypothesis in favor of the alternative, we often want to quantify "the extent to which the model fits the data." We measure the quality of a linear regression fit using two related quantities: the residual square error (RSE) and the R^2 statistic. First, RSE is an estimate of the standard deviation of ϵ. It is essentially the average amount of deviation between the actual response and the hypothetically "true" regression line. We consider RSE to be a measure of the "lack of fit" of the model to the data. Since RSE is measured in terms of units of Y, it may not be readily clear what a "good" RSE is.

By contrast, the R^2 statistic takes the form of a proportion, so it is always between 0 and 1. This scale tells us the proportion of variance explained by the model. R^2 is calculated using,

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS},$$

where $TSS = \sum(y_i - \bar{y})^2$ is the total sum of squares, and RSS is the residual sum of squares defined previously. Since TSS measures the total variance in the response Y, it can be though of as the "amount of variability inherent in the response before the regression is performed." By contrast, RSS measure the variable which remains unexplained after the regression is performed. So, $TSS - RSS$ measures the "amount variability in the response that is explained (or removed) by performing the regression, and R^2 measures the proportion of variability in Y that can be explained using X."

Unlike RSE which has to be interpreted contextually based on units of Y, R^2 has consistent interpretation from model to model. An R^2 statistic close to 1 indicates "a large proportion of the variability in the response is explained by the regression. A number near 0 indicates the regression does not explain much of the variability in the response; this might occur because the linear model is wrong, or the error variance σ^2 is high," or even both. Even with this interpretation advantage, it can still be difficult to decide what is a "good" R^2 value without knowing about its application.

Section 3.2:

Simple linear regression is useful if there is only one predictor:

$$Y = \beta_0 + \beta_1 X$$

If we had more than one predictors, we may need to use multiple linear regression for prediction:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

3.2.1 Estimating the Regression Coefficients

How do we estimate the coefficients of multiple linear regression? Similar to simple linear regression, we has below formula to make prediction:

$$\hat{y} = \hat{\beta_0} + \hat{\beta_1} x_1 + \hat{\beta_2} x_2 + \cdots + \hat{\beta_p} x_p$$

To estimate the parameter of $\beta_0, \beta_1, \ldots, \beta_p$, we use the same least squares approach for simple linear regression:

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_p x_{ip})^2$$

3.2.2 Some Important Questions

Answering below questions will help us to better understand multiple linear regression:

1. Is at least one of the predictors useful in predicting the response? This raises a new question: Is There a Relationship Between the Response and Predictors? We perform a hypothesis test to find out the relationship between the response and predictors:

   $$\begin{aligned} H_0 :& \beta_1 = \beta_2 = \cdots = \beta_p = 0 \\ H_1 :& \text{at least one } \beta_i \text{ is nonzero} \end{aligned}$$

   This test is performed by computing the F-statistic, which allow us to determine the p-value. If there is no relationship between the response and predictors, we may expect the F-statistic to be close to 1.If there is a relationship, we may expect the F-statistic to be greater than 1.

   $$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}$$

   p-value would be more general than F-statistic for determine the relation between the response and predictors. To reject the null hypothesis, the p-values need to be less than 0.05 typically.

2. Do all of the predictors help explain Y, or only a few of them? This raises a new question: how to decide on Important Variables? After computing the F-statistic and to examine the associated p-value, we use variable selection to determine the significance of predictors. There are three approaches for variable selection, which are forward selection, backward selection, and mixed selection. Forward selection begins with a model that contains no variable, then we start adding the most significant variables one after the other, at last, we will stop selection if it reach a point or until all the variables are selected. Backward selection begins with a model that contains all variables, then we start removing the least significant variables one after the other, at last, we will stop selection if it reach a point or until no variable is left. Mixed selection is a combination of forward and backward selection.

3. How well does the model fit the data? This raise a new question: Is the Model Fit? Similar to the simple regression setting, we calculate RSE and R^2 to determine how well the model fits the data.

   $$RSE = \sqrt{\frac{RSS}{(n-p-1)}}$$

4. How accurate is our prediction? This raises a new question: How to assess prediction accuracy? We use confidence intervals to determine the error between $\hat{Y}$ and $f(X)$. And we use prediction intervals to the utmost extent reduce both reducible and irreducible error.

Section 3.3:

To incorporate qualitative predictors of only two levels, we can use an indicator or dummy variable that takes on two possible numerical values e.g. 0 and 1. For example, consider a regression model based only on one 2 level qualitative predictor. We have

$$x_i = \begin{cases} 1 & \text{if } i\text{th person owns a house} \\ 0 & \text{if } i\text{th person does not own a house,} \end{cases}$$

and therefore, the model is given by

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person owns a house} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person does not own a house,} \end{cases}.$$

This is also known as one-hot encoding. The encoding is arbitrary, and we could also flip the 0 and 1 encoding or use a 1 and -1 encoding. The only difference is the interpretation of the coefficients. When a qualitative predictor has more than two levels, a single dummy variable cannot represent all possible values, so we create additional dummy variables until there are 1 fewer dummy variables than the total amount of factors. The level with no dummy variable is known as the baseline. Again, the baseline level is chosen arbitrarily, and the final predictions will be the same regardless of the choice of baseline. The coefficients and their p-values do depend on the choice of coding, however. We can use and F-test to test $H_0 : \beta_1 = \beta_2 = 0$, which does not depend on the coding.

We can use a dummy variable approach even when incorporating both quantitative and qualitative predictors.

The standard linear regression model provides interpretable results, but makes several highly restrictive assumptions that are often violated in practice. Two of the most important are that the relationship between predictors and response are additive and linear, i.e. that the association between predictor X_j and the response Y does not depend on the values of the other predictors, and that the change in the response Y associated with a one-unit change in X_j is constant, regardless of the value of X_j.

One way to extend the standard linear regression model (with two variables for example) is to include a third predictor, called an interaction term, which is constructed by computing the product of X_1 and X_2, which results in the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon.$$

Note that this can be rewritten as

$$Y = \beta_0 + \tilde{\beta}_1 X_1 + \beta_2 X_2 + \epsilon,$$

where $\tilde{\beta}_1 = \beta_1 + \beta_3 X_2$. Hence the association between X_1 and Y is no longer constant.

The hierarchical principle sates that if we include an interaction in a model, we should also include the main effects, even if the p-values associated with their coefficients are not significant. $X_1 \times X_2$ is typically correlated with X_1 and X_2, so leaving them out tends to alter the meaning of the interaction.

In some cases, the true relationship between the response and the predictors may be non-linear. A simple way to directly extend the linear model to accommodate non-linear relationships is to use polynomial regression. A simple approach for incorporating non-linear associations in a linear model is to include transformed versions of the predictors, i.e.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2^2 + \epsilon.$$

This is still a linear model with respect to the β parameters. This approach is known as polynomial regression since we have included polynomial functions of the predictors in the regression model.

There are a few common problems that occur when fitting a linear regression model:

1. Non-linearity of the response-predictor relationships

   The linear regression model assumes a straight-line relationship between the predictors and the response. If the true relationship isn't linear, then the conclusions drawn from a linear model are suspect, and the prediction accuracy can be poor. Residual plots are useful visual tools to identify non-linearity. We can plot the residuals $e_i = y_i - \hat{y}_i$, versus the predictor x_i or in the multiple regression case, we instead plot the residuals versus the predicted values $\hat{y}_i$. Ideally, the residual plot will show no pattern. If it does, it may indicate a problem with the linear model. If the residual plot indicates a non-linear association, a simple approach is to use non-linear transformations of the predictors, such as $\log X$, $\sqrt{X}$, and X^2 in the model.

2. Correlation of error terms

The linear regression model assumes the error terms $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ are uncorrelated since if there is a correlation, the estimated standard errors will tend to underestimate the true standard errors. As a result, confidence and prediction intervals and will be narrower than they should be and p-values may be lower than they should be.

Such correlations frequently occur in the context of time series data, which consists of observations for which measurements are obtained at discrete points in time. Observations that are obtained at adjacent time points will have positively correlated errors in many cases. To determine this, we can use a residual plot as a function of time. If the errors are uncorrelated, there should be no pattern, but if they are positively correlated, there may be tracking in the residuals, i.e. adjacent residuals may have similar values.

Correlation among the error terms can also occur outside of time series data. The assumption of uncorrelated errors is extremely important for linear regression and for other statistical methods. Good experimental design is crucial to mitigate the risk.

3. Non-constant variance of error terms

   Standard errors, confidence intervals, and hypothesis tests associated with the linear model rely upon the assumption of constant variance, $Var(\epsilon_i) = \sigma^2$. It's often the case that the variances of the error terms are non-constant. One can identify non-constant variances in the errors from the presence of a funnel shape in the residual plot. One possible solution is to transform the response Y using a concave function such as $\log Y$ or $\sqrt{Y}$ which leads to a greater amount of shrinkage of the larger responses, and hence a reduction in heteroscedasticity. If we have a good idea of the variance of each response, we can fit our model by weighted least squares with weights proportional to the inverse variances.

4. Outliers

   An outlier is a point for which y_i is far from the value predicted by the model. Residual plots can be used to identify outliers, but it can be difficult to decide how large a residual needs to be before we consider the point to be an outlier. To address this, we can plot the studentized residuals, computed by dividing each residual e_i by its estimated standard error. Observations whose studentized residuals are greater than 3 in absolute value are possible outliers. One solution is to simply remove the observation, but an outlier may also indicate a deficiency of the model.

5. High-leverage points

   Observations with high leverage have an unusual value for x_i. High leverage observations tend to have a sizable impact on the estimated regression line. Any problems with high leverage points may invalidate the entire fit, so it is important to identify high leverage observations.

   In a multiple linear regression, it is possible to have an observation that is well within the range of each individual predictor's values, but is unusual in terms of the full set of predictors. In order to quantify an observation's leverage, we compute the leverage statistic. For a simple linear regression, it is

$$h_i = \frac{1}{n} + \frac{(x_i + \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}.$$

   The leverage statistic is always between $1/n$ and 1, and the average leverage for all the observations is always equal to (p+1)/n, so if an observation has a leverage statistic that greatly exceeds this, it may be a high leverage point.

6. Collinearity

   Collinearity refers to the situation in which two or more predictor variables are closely related to one another. The presence of collinearity can pose problems in the regression context, since it can be difficult to separate out the individual effects of collinear variables on the response.

   Since collinearity reduces the accuracy of the estimates of the regression coefficients, it causes the standard error for $\hat{\beta}_j$ to grow. As a result, if there is collinearity, we may fail to reject $H_0 : \beta_j = 0$. The power of the hypothesis test (the probability of correctly detecting a non-zero coefficient) is reduced by collinearity.

A simple way to detect collinearity is to look at the correlation matrix of the predictors. If there is an element that is large in absolute value, that indicates a pair of highly correlated variables. It is possible for collinearity to exist between three of more variables even if no pair of variables has a high correlation. This is called multicollinearity. A better way to assess multicollinearity is to compute the variance inflaction factor (VIF),

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}},$$

where $R^2_{X_j|X_{-j}}$ is the R^2 from a regression of X_j onto all of the other predictors. The VIF is the ratio of the variance of $\hat{\beta}_j$ when fitting the full model divided by the variance of $\hat{\beta}_j$ if fit on its own. As a rule of thumb, a value that exceeds 5 or 10 indicates a high amount of collinearity.

There are two simple solutions, drop one of the problematic variables from the regression, or combine the collinear variables together into a single predictor.

Section 3.4:

Exploring the relationship between sales and advertising budget

Since there are three variables that make up advertising budget (TV, radio, and newspaper), a multiple linear regression model is used with TV, radio and newspaper as the predictors and sales as the response. An F-statistic can be used to test if $H_0 : \beta_{TV} = \beta_{radio} = \beta_{newspaper} = 0$. For this dataset, the pvalue was low providing significant evidence of the relationship between sales and advertising budget.

Strength of the relationship

RSE which estimates the standard deviation of the response (in this case the sales) between the population regression line; and, R^2 which is the percentage of variability of the response explained by the predictors are two good measures for the strength of a relationship between response and predictors.

Which media are association with sales?

Individual pvalues of the predictors can be viewed to identify the significance of any individual predictor to the response.

How large is the association between each medium and sales?

Constructing confidence intervals for the individual $\hat{\beta}$ coefficients is a good way to go about measure the size of association between the predictors and the response. If the confidence interval includes 0, then it may suggest a predictor has non- significant impact on the response. However, collinearity could be an issue which results in large confidence bounds. VIF scores can be viewed to ensure collinearity is not an issue. Another technique to identify individual relationships between the predictors and response is to perform simple linear regression models between each predictor and response and identify size of association.

How accurately can we predict future sales?

If we are interested in an individual response then we can use a prediction interval, but if interested in an average response we can use a confidence interval. The bounds for prediction intervals are always wider then confidence intervals due to taking into account irreducible error.

Is the relationship linear?

A residual diagnostic plot can be used to identify linearity. We expect no pattern (centered at 0) among the residuals if the relationship is linear. Transformations can be used if patterns are found.

Is there synergy among the advertising media?

Interactions among variables should be explored.

Section 3.5:

Linear regression is a parametric method due to its linearity assumption. The advantage to parametric methods is they are easy to fit because of the small number of parameters to estimate, tests for significance are easy to perform, and interpretation is usually clear. The disadvantage of parametric methods are their strong assumptions. If the true model deviates from our assumed model, then the results will be poor.

Non- parametric methods do not have assumptions of the model form and are generally more flexible then parametric methods.

KNN Regression:

Is very similar to KNN classifier (but for when the response is continuous) in that the model will identify the K closest points to a point of prediction (denoted as N_0) and averages those to form the prediction.

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

The best value of K is based on the bias- variance trade off. Smaller K will have lower bias but higher variance whereas larger values of K will have higher bias but lower variance. KNN can suffer from the curse of dimensionality where in higher dimensions, the K observations closest to the point of prediction may be very off.

In the choice between parametric or non- parametric methods, if the true form of the relationship matches the parametric models assumptions, then it will outperform the non- parametric method.

"In general, parametric methods will tend to outperform non- parametric approaches when there is a small number of observations per predictor".

4 ISLR Problems

3.8

(a) We use the `lm()` function to perform a simple linear regression with `mpg` as the response and `horsepower` as the predictor.

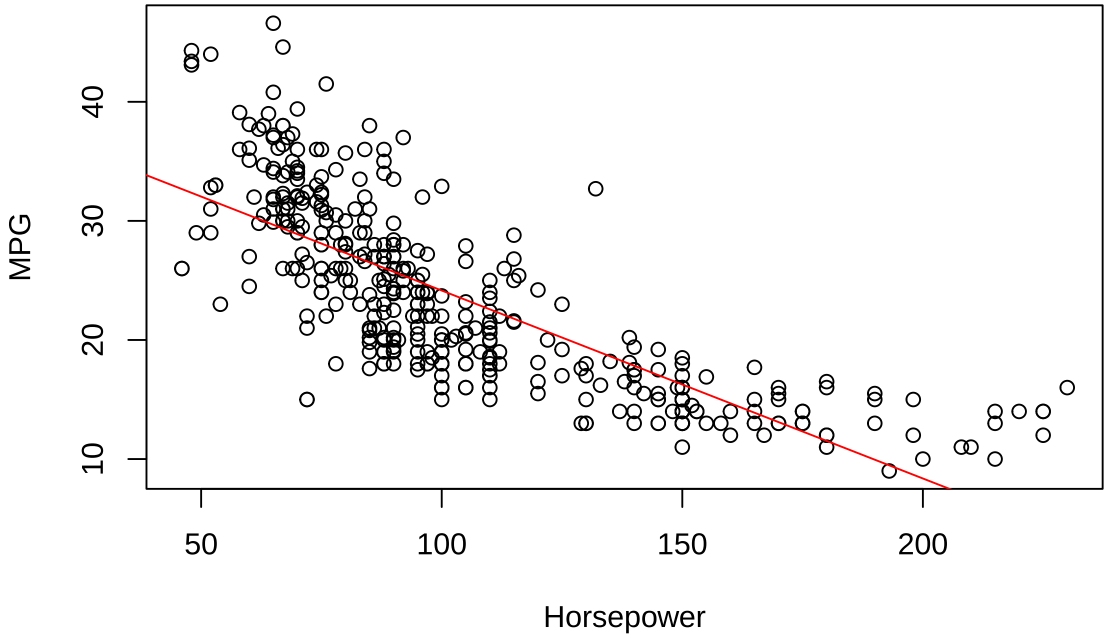
```
Auto <- ISLR::Auto

model <- lm(mpg ~ horsepower, data = Auto)

summary <- summary(model)

CI <- predict(model,newdata = data.frame(horsepower = 98),interval = "confidence")
PI <- predict(model,newdata = data.frame(horsepower = 98),interval = "prediction")

results <- round(summary$coefficients,5)
results[,"Pr(>|t|)"] <- c("<2e-16","<2e-16") # originally 1.220362e-187, 7.031989e-81

knitr::kable(results)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	39.93586	0.7175	55.65984	<2e-16
horsepower	-0.15784	0.00645	-24.48914	<2e-16

The model does indicate that there is a weak negative relationship between `mpg` and `horsepower`. In particular, for every unit increase in `horsepower`, we only see a 0.157 decrease in `mpg`. The predicted `mpg` associated with a horsepower of 98 is 24.467, with confidence interval $[23.973, 24.961]$ and prediction interval $[14.809, 34.125]$.

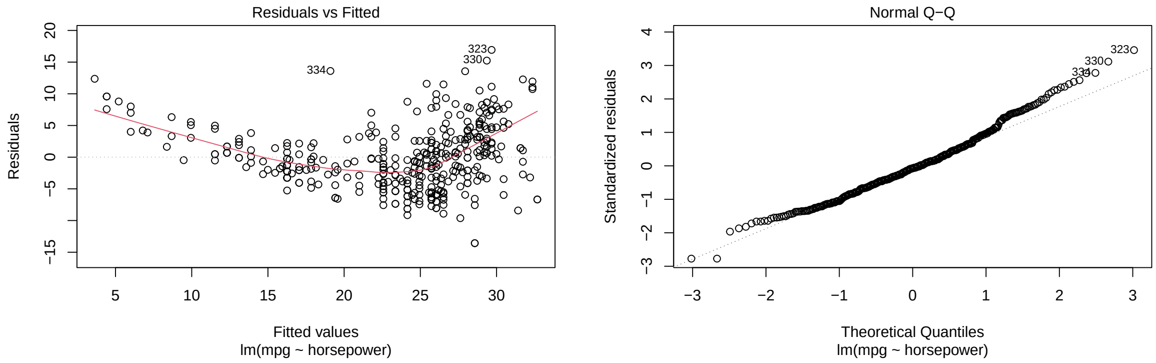
(b) We plot the response against the predictor and use `abline()` to display the least squares regression line

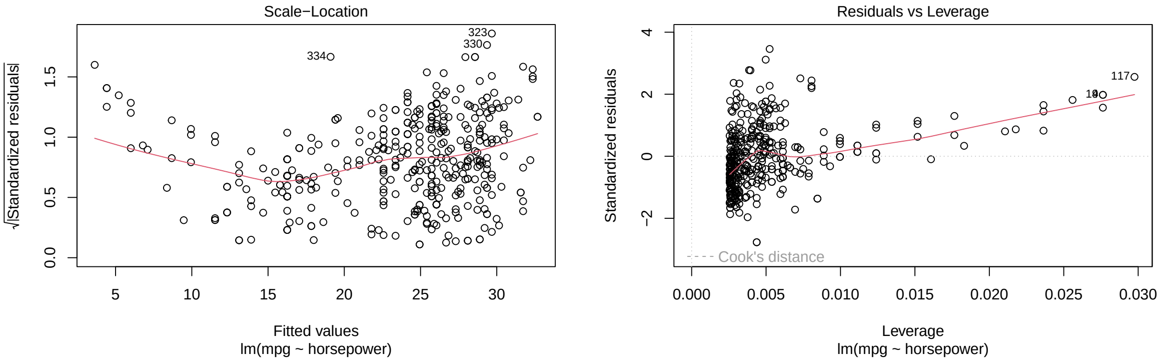
```
plot(Auto$horsepower, Auto$mpg, xlab = "Horsepower", ylab = "MPG")
abline(coef = model$coefficients, col = "red")
```



(c) We use the `plot()` function to produce the diagnostic plots of the least squares regression fit.

```
plot(model)
```



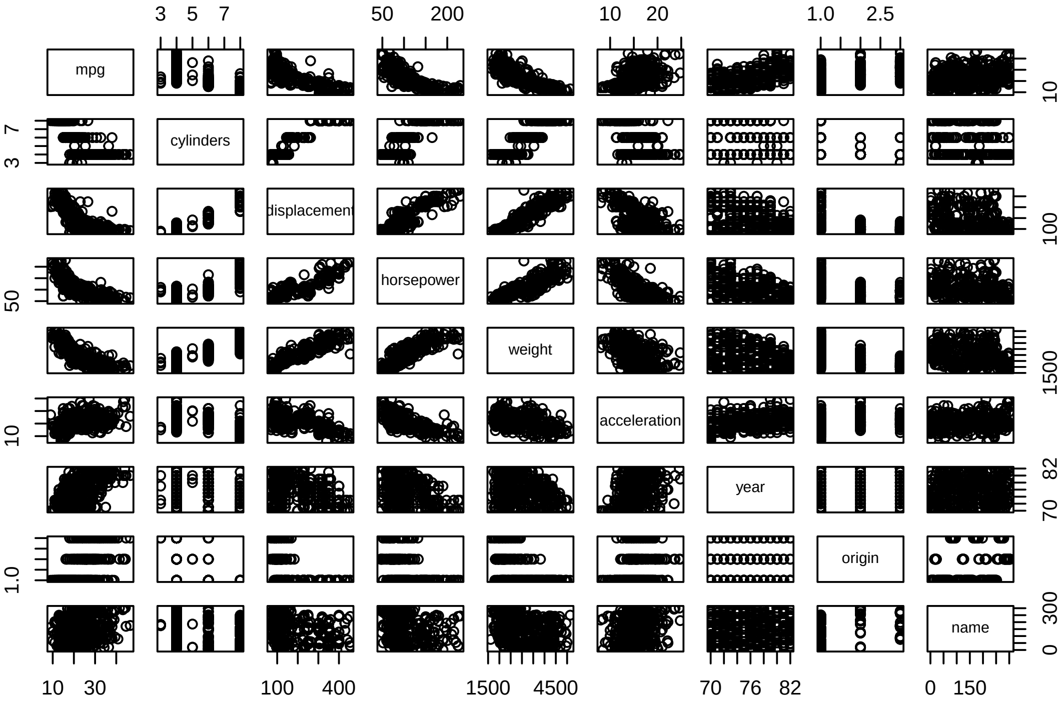


We notice an upward curve in the residual vs fit plot, suggesting a non-linear relationship between the predictor and the response, which we can also see in the above plot of the data. The Scale-location plot is also curved upward, further suggesting non-linearity and non-constant variance in the residuals.

3.9

(a) Produce a scatterplot matrix which includes all of the variables in the data set

```
library(ISLR)
pairs(Auto)
```



(b) Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable, which is qualitative.

```
cor(subset(Auto, select = -name))
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

(c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results.

```
mlr = lm(mpg ~ . - name, data = Auto)
summary(mlr)
```

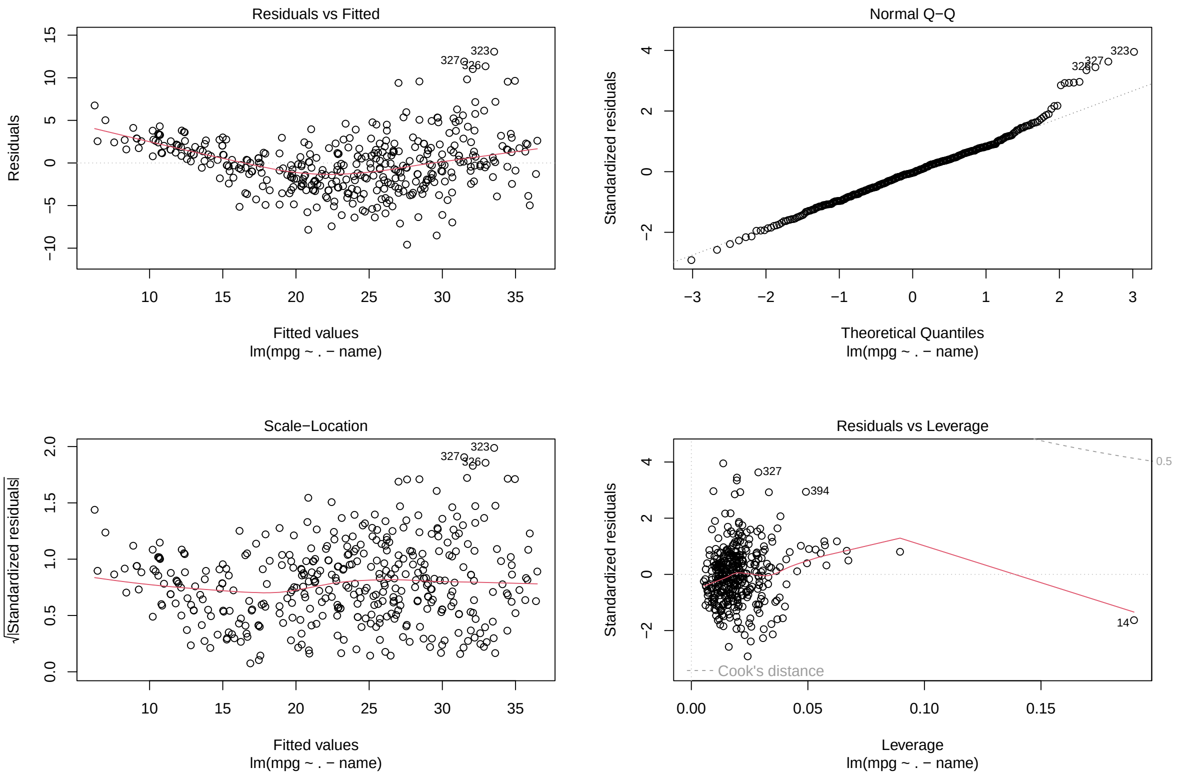
```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

Both F-statistic and p-value's value provide evidence against the null hypothesis. As a result, we can conclud there is a relationship between the predictors and the response. Based on the p-value, displacement, weight, year, and origin have a statistically significant relationship to the response. The estimate coefficient for year is 0.750773, which indicate the mpg goes up 0.750773 per year.

(d) Use the plot() function to produce diagnostic plots of the linear regression fit.

By the residual plot, we can conclude the data is the model is not fit well since there is non-linear pattern showing. Also, scale-location plot show there is no point located outside of range [-2,2], which indicates there may be no outliers. Point 14 is the high leverage point.

```
plot(mlr)
```



(e) Use the * and : symbols to fit linear regression models with interaction effects.

Cylinders * weight is statistically significant, whereas displacement * acceleration is not.

```
mlr.new <-  lm(mpg ~ displacement * acceleration + cylinders * weight, data = Auto)
summary(mlr.new)
```

```
##
## Call:
## lm(formula = mpg ~ displacement * acceleration + cylinders *
##     weight, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.113  -2.414  -0.288   1.889  17.269
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               55.8261201  5.7905043   9.641  < 2e-16 ***
## displacement               0.0020878  0.0119973   0.174  0.86194
## acceleration               0.4813134  0.1761558   2.732  0.00658 **
## cylinders                 -3.6528520  0.8431479  -4.332 1.88e-05 ***
## weight                    -0.0125830  0.0016461  -7.644 1.69e-13 ***
## displacement:acceleration -0.0010923  0.0008624  -1.267  0.20606
## cylinders:weight           0.0011551  0.0002312   4.995 8.94e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.097 on 385 degrees of freedom
## Multiple R-squared:  0.7287, Adjusted R-squared:  0.7245
## F-statistic: 172.4 on 6 and 385 DF,  p-value: < 2.2e-16
```

(f) Try a few different transformations of the variables.

The log(cylinders) transformation is more significant than cylinders, but the sqrt(acceleration) is less significant than acceleration. We assume that is depend on the variables selection.

```
summary(lm(mpg ~ . - name + log(cylinders), data=Auto))
```

```
##
## Call:
## lm(formula = mpg ~ . - name + log(cylinders), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.762  -2.093  -0.180   1.730  12.942
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.333e+00  6.639e+00  -0.201  0.84096
## cylinders       3.673e+00  1.299e+00   2.827  0.00494 **
## displacement    2.008e-02  7.420e-03   2.707  0.00710 **
## horsepower     -2.750e-02  1.398e-02  -1.967  0.04986 *
## weight         -6.393e-03  6.442e-04  -9.924  < 2e-16 ***
## acceleration    1.059e-01  9.789e-02   1.082  0.27981
## year            7.482e-01  5.033e-02  14.865  < 2e-16 ***
## origin          1.268e+00  2.787e-01   4.548 7.29e-06 ***
## log(cylinders) -2.287e+01  6.912e+00  -3.308  0.00103 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.285 on 383 degrees of freedom
## Multiple R-squared:  0.8264, Adjusted R-squared:  0.8228
## F-statistic:   228 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
summary(lm(mpg ~ . - name + sqrt(acceleration), data=Auto))
```

```
##
## Call:
## lm(formula = mpg ~ . - name + sqrt(acceleration), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8414 -1.9898 -0.0903  1.9205 13.1480
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.840e+01  1.715e+01   3.404 0.000733 ***
## cylinders          -2.865e-01  3.185e-01  -0.900 0.368911
## displacement        8.116e-03  7.768e-03   1.045 0.296786
## horsepower         -3.469e-02  1.399e-02  -2.479 0.013599 *
## weight             -5.343e-03  6.823e-04  -7.830 4.82e-14 ***
## acceleration        4.623e+00  9.986e-01   4.630 5.02e-06 ***
## year                7.554e-01  4.971e-02  15.196  < 2e-16 ***
## origin              1.327e+00  2.721e-01   4.876 1.59e-06 ***
## sqrt(acceleration) -3.738e+01  8.178e+00  -4.570 6.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.245 on 383 degrees of freedom
## Multiple R-squared:  0.8307, Adjusted R-squared:  0.8272
## F-statistic: 234.9 on 8 and 383 DF,  p-value: < 2.2e-16
```