Chapter 3: Linear Regression

3.1 Simple Linear Regression (SLR)

Simple Linear Regression is a straightforward way to quantify Y on the basis of a single predictor variable X assuming an approximately linear relationship between the two. We denote it as,

$$Y \approx \beta_0 + \beta_1 X \tag{3.1}$$

Where $\approx$ denotes regressing Y on X. β_0 and β_1 are unknown constants that represent the slope and intercept and are known as the model coefficients or parameters. We can predict Y given X using,

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{3.2}$$

The hat symbol signifies the predicted value of the response.

Estimating Coefficents

Let n observations be represented by,

$$(x_1, y_1), \dots, (x_n, y_n)$$

We want to find such parameters such that the resulting line given 3.2 is as close as possible to the n observations. To minimize we use the *least squares* criterion.

Let $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the predictor for Y based on the ith value of X. Then $e_i = y_i - \hat{y}_i$ represents the ith *residual*. This is the difference between the observed and the predicted response ith value. We define this as the *residual sum of squares* (RSS),

$$RSS = e_1^2 + \cdots + e_n^2 = (y_1 - (\hat{\beta}_0 + \hat{\beta}_1 x_1))^2 + \cdots + (y_n - (\hat{\beta}_0 + \hat{\beta}_1 x_n))^2 \tag{3.3}$$

The least squares approach chooses $\hat{\beta}_0, \hat{\beta}_1$ to minimize RSS. We can show using calculus that,

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{3.4}$$

Where $\bar{x}$ and $\bar{y}$ is the sample means such that $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. 3.4 defines the *least squares coefficient estimates* for simple linear regression.

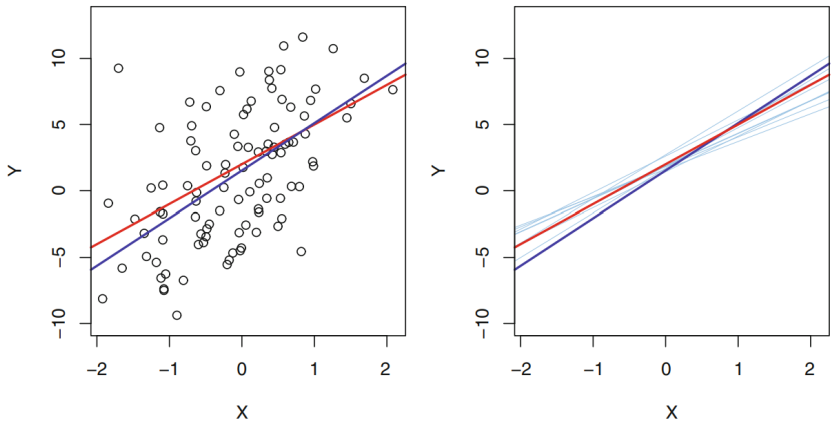
Assessing the Accuracy of the Coefficient Estimates

If f is to be approximated by a linear function, we can write this as,

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{3.5}$$

The error term is a catch all when we miss with this simple model. In other words it represents the variation in Y as a measurement error. 3.5 is also known as the *population regression line* which is the best linear approximation to the true relationship between X and Y.

3.4 characterizes the least squares line 3.2. The image below illustrates the true relationship and the predicted relationship $f(X) = 2 + 3X$.The image on the right shows the population regression line computed on sets of random observations. Each least squares line is different, but on average converge to the populaiton regression line.



Bias

The analogy between linear regression and estimation of the mean of a random variable is an apt one based on *bias*. If we use the sample mean $\hat{\mu}$ to estimate μ this is *unbiased*. We expect $\hat{\mu}$ to equal μ.

This means that given a set of observations $\hat{\mu}$ might overestimate μ but given a huge number of estimates the average would exactly equal μ.

Unbiased estimators *systematically* over or underestimate the true parameter. Averaging estimates over huge number of data sets would garner accurate parameters.

Variance

The natural question arises: *how accurate is the sample mean $\hat{\mu}$ as an estimate of μ*? We answer this using the *standard error* of $\hat{\mu}$ written as $SE(\hat{\mu})$. We can now introduce the variance formula as,

$$Var(\hat{\mu}) = SE(\hat{\mu}^2) = \frac{\sigma^2}{n} \tag{3.7}$$

where σ is the std. dev. of each of the realizations fo y_i of Y. 3.7 tells us that the larger the n observations, the smaller the variance.

How close are $\hat{\beta_0}, \hat{\beta_1}$ to the actual parameters? We can comppute the standard errors associated witht he predicted paramters using the forumals,

$$SE(\hat{\beta_0})^2 = \sigma^2\Big[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\Big] SE(\hat{\beta_1})^2 = \frac{2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.8}$$

where $\sigma^2 = Var(\epsilon)$. For each formula we need to assume that the errors ϵ_i for each observation are uncorrelated with common variance σ^2.

Intuitively $SE(\hat{\beta_1})^2$ is smaller when x_i are more spread out. We have more *leverage* to estimate a slope when this is the case.

We can also see that $SE(\hat{\beta_0})^2$ would be the same $SE(\hat{\mu})$ if $\bar{x}$ were 0.

While σ^2 is not known, it can be estimated from the data. This is called *residual standard error* and is given by the formula $RSE = \sqrt{RSS/(n-1)}$. When σ^2 is estimated from the data we should write $\hat{SE}(\hat{\beta_1})$ to indicate an estimate has been made but for simplicity of notion we omit the hat.

Confidence Interval

Standard errors can be used to compute *confidence intervals*. A 95% confidence interval as a range of values that with a *95% probability the range will contain the true unknown value of the parameter*. The range is defined in terms of lower of lower and upper limits computed from the sample data.

For linear regression, the 95% confidence interval for β_1 takes form,

$$\hat{\beta_1} \pm 2 \cdot SE(\hat{\beta_1}) \tag{3.9}$$

That is, there is approx a 95% chance that the interval

$$\Big[\hat{\beta_1} - 2 \cdot SE(\hat{\beta_1}), \hat{\beta_1} + 2 \cdot SE(\hat{\beta_1})\Big] \tag{3.10}$$

will contain the true value of β_1. Similarly, a confidence interval for β_0 takes form as,

$$\hat{\beta_0} \pm 2 \cdot SE(\hat{\beta_0}) \tag{3.11}$$

Hypothesis Tests Standard errors can be used to perform hypothesis tests on the coefficients. The most popular test is called the *null hypothesis* where,
H_0: There is no relationship between X and Y
H_a: There is some relationship between X and Y

Mathematically, this means that, $H_0 : \beta_1 = 0$
$H_a : \beta_1 \neq 0$

If $\beta_0 = 0$ then the model is reduced to $Y = \beta_0 + \epsilon$ and X is not associated wit Y. Testing the null hypothesis tell us how far away β_1 is such that β_1 is nonzero.

The T Statistic If $SE(\hat{\beta_1})$ is small then $\hat{\beta_1}$ provide strong evidence that $\beta_1 \neq 0$ and hence there is a relationship between X and Y. If $SE(\hat{\beta_1})$ is large, then $\hat{\beta_1}$ must be large in order for us to reject the null hypothesis.

For this procedure, we use the *t statistic* given by,

$$t = \frac{\hat{\beta_1}}{SE(\hat{\beta_1})} \tag{3.14}$$

The t statistic measures the numbers of standard deviation that $\hat{\beta_1}$ is away from 0.

If there is *no* relationship between X and Y then we expect 3.14 to have a t-distribution with *n-2* degrees of freedom.

t-distribution has a bell shaped and for values n greater than approx 30 is quite similar to the normal distribution. It's simple to calculate the probability of observing any value equal to $|t|$ or larger assuming $\beta_1 = 0$. This is called the *p-value*.

The interpretation of the p value is as follows:
* a small p-value indicates that it is unlikely to observe a substation association between the predictor and the response due to chance in the absence of any real association between the predictor and the response.

If we have a small p-value, then we can infer that there is an association and we can *reject the null hypothesis*. We can declare a relationship between X and Y. Typical p-values are 0.05 or 0.01.

Assessing the Accuracy of the Model

We want to quantify *the extent to which the model fits the data*. The quality of a linear regression fit is typically assessed using the *residual standard error*(RSE) and the R^2 statistic.

Residual Standard Error

Even if we know the true regression line we would not be able to perfectly predict Y from X. The RSE is an estimate of the standard deviation of ϵ.

It is the average amount that the response will deviate form the true regression line. It's computed as,

$$RSE = \sqrt{\frac{1}{n-2}RSS} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{3.15}$$

RSE is considered a measure of the *lack of fit* of the model to the data.

Residual SUm Squared

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{3.16}$$

R^2 statistic

The RSE provides an absolute measure of the lack of fit of a model. But because it's measured in the units of Y, it is not always clear what constitutes a good RSE. The R^2 provides an alternative measure of fit. It takes the form of a *proportion* of variance explained. The R^2 is a value between 0 and 1.

The formula is seen below,

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} \tag{3.17}$$

where $TSS = \sum(y_i - \bar{y})^2$ is the *total sum of squares*.

TSS measures the total variance in the response Y and can be though of as the amount of variability inherent in the response before the regression is performed.

TSS - RSS measures the amount of variability in the response that is explained by performing the regression.

RSE is the measure of lack of fit of the model.

RSS is the residual sum squared that is eplained by the model.

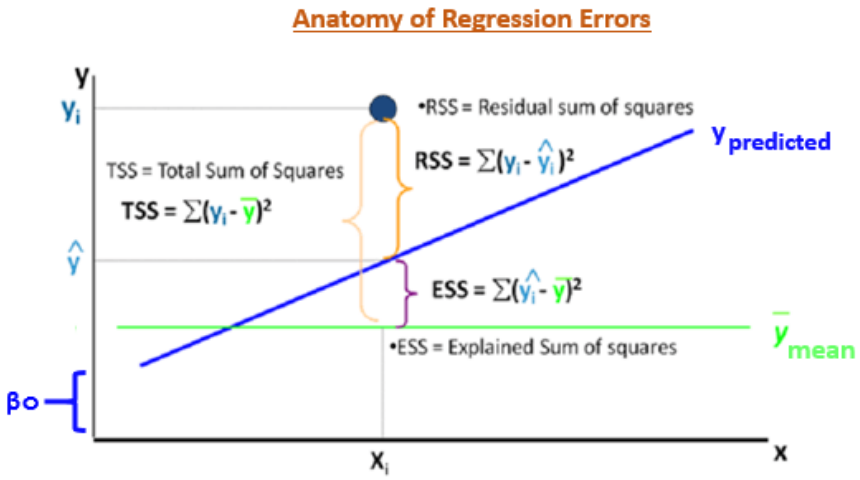


Figure 1: Regression Errors

Below is an image that illustrates the relaitonship between TSS, RSS and ESS where $TSS = RSS + ESS$,

R^2 measures the proportion of variability in Y that can be explained using X.

R^2 statistic has an interpretation advantage over the RSE since unlike the RSE it always lies between 0 and 1. However it can still be challenging to determine what is a *good* R^2 value.

The R^2 statistics is a measure of a linear relationship between X and Y. The *correlation* is defined as

$$Cor(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} \tag{3.18}$$

Note: The squared correlation and the $R^{@}$ coefficient is the same.

Multiple Linear Regression

Instead of fitting a separate simple linear regression model for each predictor, a better approach is to extend the SLR so that it can accommodate multiple predictors.

We can do this by giving each predictor a separate slope coefficient in a single model. We have p distinct predictors, then

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon \tag{3.19}$$

where X_j represents the jth predictor and β_j quantifies the association between the variable and the response. We interpret β_j as the *average* effect on Y of one unit increase in X_j *holding all predictors fixed.*

Estimating the Regression Coefficients

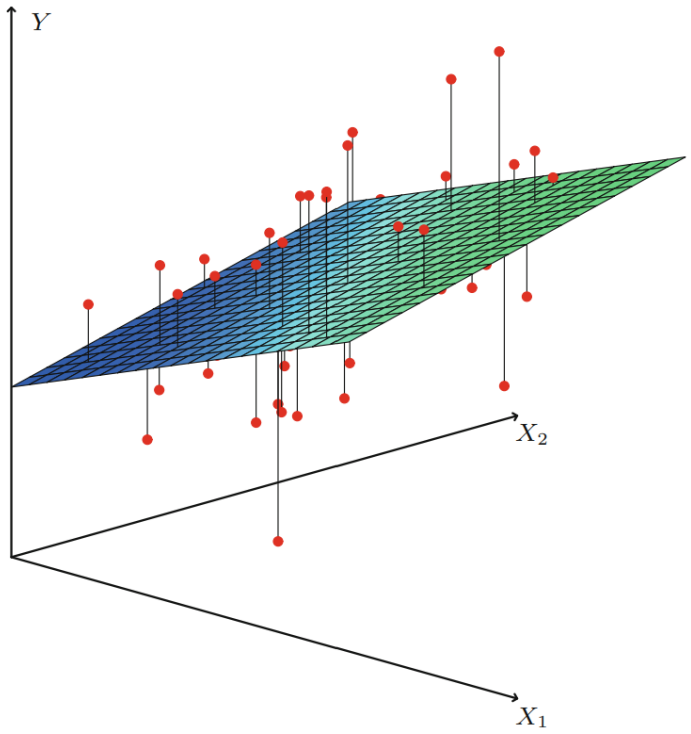
The regression coefficients are unknown and need to be estimated. Given $\hat{beta}_0 \ldots \hat{beta}_p$ we can make predictions using the formula,

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p \tag{3.21}$$

The parameters are estimated using the same least squared approach in SLR> We choose $\beta_0, \ldots, \beta_p$ such that we *minimize* the *residual sums squared* RSS, which can be written as,

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p))^2 \tag{3.22}$$

A visualization of two predictors is seen below. The line in the 2D now becomes a plane in 3D.



The values $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ that minimize the multiple least squares regression coefficient estimates are represented in matrix algebra.

But sometimes MLR suggests no relationship between two predictors given p is greater than 2 yet the SLR does. Why is this?

The correlation matrix for the three predictors reveals a tendency for one predictor to get credit for the effect of the other correlation. This is slightly counterproductive but proves to show some co-linearity.

Some important questions

1. Is at least one of the predictors $X_1, \ldots X_p$ useful in predicting the response?
2. Do all predictors help to explain Y or is only a subset of the predictors useful?
3. How well does the model fit the data?
4. Given a set of predictor values, what response value should we predict and how accurate is out prediction?

Question 1 In SLR we could determine the relationship between the response and the predictor by checking whether $\beta_1 = 0$. In MLR with p predictors, we need to ask whether all of te regression coefficients are 0, which is the null hypothesis. We also look at the alternative hypothesis, which states that at least one of the predictors in non-zero.

$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$ H_a : at least one β_j is non-zero

This hypothesis test is performed by computing the F statistic,

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)} \tag{3.23}$$

where, $TSS = \sum(y_i - \bar{y})^2$ $RSS = \sum(y_i - \hat{y_i})^2$

If the linear model assumptions are correct, one can show that the expected value of $RSS/(n-p-1)$ is the variance.

$$E\{RSS/(n-p-1)\} = \sigma^2$$

We can then provide H_0 is true,

$$E\{(TSS - RSS)/p\} = \sigma^2$$

When there is *no relationship* between the response and the predictors, one would expect that the F statistic to take a value *close to 1.* if H_a *is true*, then $E\{(TSS-RSS)/p\} > \sigma^2$ so we expect F to be *greater than 1.*

But how large does the F statistic need to be to reject H_0 and conclude that there is a relationship?

This depends on n and p. When n is large an F statistic that is just a little larger than 1 might still provide evidence against H_0. In contrast, a larger F stat is needed is needed to reject H_0 if n is small.

When H_0 is true and errors ϵ_i have a normal distribution, the F stat follows an F-distribution. For any given value of n and p, the p value associated with the F statistic can be calculated.

Based on the p value we can determine whether to reject or not reject the null hypothesis. ($p < 0.05$ or 0.01).

Subsets of predictors Sometimes we want to test subset q of the coefficients are 0. THis corresponds to a null hypothesis of,

$H_0 : \beta p - q + 1 = \beta_{p-q+2} = \cdots = \beta_p = 0$

In this case we fit a second model that uses all predictors except in the last Q. Suppose the residual sum of squares for that model is $RSS - 0$. The the F statistic is,

$$F = \frac{(RSS_0 - RSS)/q}{RSS/(n-p-1)} \tag{3.24}$$

Note that for the numerator we compare the subset RSS with the total predictors RSS.

The individual p-values of the predictors are the *partial effect* of adding that variable to the model. They are equivalent to the F stat that omits a single variable for the model leaving all others in.

Given individual p-values why do we need to look at the overall F stat? This logic is flawed: That if there is a small p value form all the predictors, then there is at least one predictor related to the response.

When P is large, let's say 100, and no variable is related to the response, about 5% of the p-values associated with each variable will be below 0.05 by chance. In other words we expect approx five small p-values in the absence of true association.

Remember that the p value represents a common threshold for significance, meaning that there is only a 5% chance that the observed effect is due to random variation rather than a true effect.

The F stat does not suffer from this because it adjusts for the number of predictors. If H_0 is true, then there is only a 5% chance the F stat will result in a p value below 0.05 regardless of the number of predictors or the number of observations.

Using the F stat to test for any association between the predictors and the response works well when p is small and certainty small compared to $n>$

If $p > n$ where there are more coefficients than observations, we can not fit a MLR using least squares so the F stat can not be used. When p is large, *forward selection* and *high dimensional setting* can be used.

Deciding on important variables The first step of MLR is to compute the F stat and examine the associated p-value. If the p-value is related to the response whcih one is it?

The task of determining which predictors are associated with the response in order to fit a single model involving only those predictors is called *variable selection*.

We would like to perform variable selection by trying out a lot of different models, each containing subsets of the predictors. We then can use some models, like, 1. Mallow's C's 2. Akaike information criterion (AIC) 3. Bayesian information criterion (BIC) 4. adjusted R^2

There are a total of 2^p models that contain subsets of p variables. We need an automated and efficient approach to choose a smaller set of models to consider:

- *Forward selection* We begin with a *null model* that only contains an intercept but no predictors. We fit p simple linear regressions and add to the null model the variable that results in the lowest RSS. We then add to that model the variable that result sin the lowest RSS for the new two variable model. WE continue this until some stopping rule. No restriction for use but greedy and redundant.
- *Backward selection* We start with all variables in the model and remove variables with the largest p-value. The new $(p-1)$ variable model is fit and the variable with the largest p-value is removed. This continues until some stopping rule. Used when $p > n$.
- *Mixed selection* This is a combo of both forward and backwards selection. We start with no variables in the model and then with forward selection we add the variable that provides the best fit. We continue to add variables one by one. At any point the variables rise above a certain threshold, we remove them from the model. Remedies inefficiencies of forward and restrictions of backwards.

Model fit The most common numerical measures of model fit are the RSE and R^2 (the fraction of variance explained).

The R^2 is the square of the correlation of the response and the variable and can be computed by $Corr(X, Y)^2$.

But R^2 will always increase when more variables are added to the model even if those variables are only weakly associated with the reps one. This is because adding another variable to the least squares equation must allow us to fit the training data more accurately. Thus the R^2 statistic which is computed on the training data will increase.

How can RSE increase when RSS must decrease? In general RSE is defined as

$$RSE = \sqrt{\frac{1}{n-p-1}RSS} \tag{3.25}$$

which simplifies to 3.15 for simple linear regression. It can also be useful to plot the data which can suggest *synergy* or *interaction* effects between predictors.

Predictions There are three sorts of uncertainty associated with this prediction,

1. the coefficient estimates $\hat{\beta_0}, \dots, \hat{\beta_p}$ are estimates for $\beta_0, \dots, \beta_p$. That is, the least squares plane

$$\hat{Y} = \hat{\beta_0} + \hat{\beta_1} X_1 + \cdots + \hat{\beta_p} X_p$$

   is only an estimate for the true population regression plane

$$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

   The inaccuracy in the coefficient estimates is related to the reducible error. We can compute the confidence interval in order to determine how close $\hat{Y}$ will be to $f(x)$.

Because a linear model is an approximation of reality, we have an additional source of reducible error called *model bias*.

Because of the intrinsic irreducible error ϵ from out model that prohibits perfect predictions, how can we measure the variation of $\hat{Y}$ versus Y. We can use *prediction intervals*.

Prediction intervals are wider than confidence intervals because of the incorporated irreducible and reducible error.

Other considerations in the regression model

Qualitative predictors

Predictors with only two levels If a qualitative predictor (also known as a *factor*) only has two *levels* or possible values, they can be incorporated to linear models through a dummy variable. This dummy variable takes on two possible numerical values such that

$$x_i = \begin{cases} 0 & female \\ 1 & male \end{cases}$$

We can then use the variable as a predictor in the regression model,

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i == \begin{cases} \beta_0 + \beta_1 & female \\ \beta_0 + \epsilon_i & male \end{cases}$$

Qualitative Predictors with $>$ 2 levels When a qualitative predictor has mote than two levels, a single dummy variable cannot represent all possible values. We then create additinal variables such that,

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i 2\epsilon_i = \begin{cases} \beta_0 + \beta_1 & f1 \\ \beta_0 + \beta_2 & f2 \\ \beta_0 + \epsilon_i & f3 \end{cases}$$

Extensions of the linear model SLR makes several highly restrictive assumptions that are often violated in practice. The model assumes *additive* and *linear* relationships between predictors and responses. The additive assumption means that the effect of changes in X_j on Y is independent of the values of the other predictors. The linear assumptions concludes constant rate of change.

Removing the additive assumptions If we add an *interaction term* as a third predictor we can illustrate the interactions of two predictors that are not independent.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon \tag{3.31}$$

The inclusion of the term relaxes the additive assumption. We can see this in the newly constructed LR model where we rewrite 3.31 such that,

$$Y = \beta_0 + (\beta_1 + \beta_3 X_2) + \beta_2 X_2 + \epsilon = \beta_0 + \tilde{\beta}_1 X_1 + \beta_2 X_2 + \epsilon \tag{3.32}$$

The effect on X_1 is no longer constant. Adjusting X_2 will change the impact on X_1.

The Hierarchical principle Sometimes the case that an interaction term has a very small p-value, but the associated main effects do not. The *hierarchical principle* sates that if *we include an interaction in a model, we should also include the main effects even if the p-value associated with their coefficients are not significant.*

In other words, if the interaction between X_1 and X_2 seem important, we should include both X_1 and X_2 in the model even if their coefficient estimates have big p-values.

The rationale behind this is that if $X_1 x X_2$ is related to the response, then whether the coefficients X_1 or X_2 are exactly zero is of little interest. $X_1 x X_2$ is typically correlated with X_1 and X_2 so leaving them out tends to alter the meaning of the interaction.

For example, if we wish to predict the balance using income and student variables, the absence of an interaction term takes for as,

$$balance_i \approx \beta_0 + \beta_1(income_i) + \begin{cases} \beta_2 & student \\ 0 & n_student \end{cases} = \beta_1(income_i) + \begin{cases} \beta_0 + \beta_2 & student \\ \beta_0 & n_student \end{cases}$$

This fits two parallel lines to the data one for students and the other for non students. They have different intercepts but this illustrates that the average effect on balance is independent from income and student predictors. This presents a serious limitation - income may have a very different effect on balance of a student vs non studuent.

By adding a iteration variable, our model becomes

$$balance_i \approx \beta_0 + \beta_1(income_i) + \begin{cases} \beta_2 + \beta_3 & student \\ 0 & n_student \end{cases} = \beta_1(income_i) + \begin{cases} (\beta_0 + \beta_2) + (\beta_1 + \beta_3) income_i & student \\ \beta_0 + \beta_1(income_i) & n_student \end{cases}$$

We still have two different regression lines with different intercepts but now with different slopes. This allow s for possibilities in changes in income that may effect balances of students and non students.

Non linear relationships In come cases true relationships between response and predictors may be non linear. We can use a *polynomial regression* to directly extend the linear model to accommodate for non-linear relationships.

A simple approach for incorporating non linear associations in a linear model is to include transformed versions of the predictors in the model. Int he visual above, the data points seem to have a quadratic shape and therefore suggest a model form of

$$mpg = \beta_0 + \beta_1(horsepower) + \beta_2(horsepower^2) + \epsilon$$

This is still a linear model and therefore can use standard linear regression computations just with a transformed variable of $X_2 = horsepower^2$.

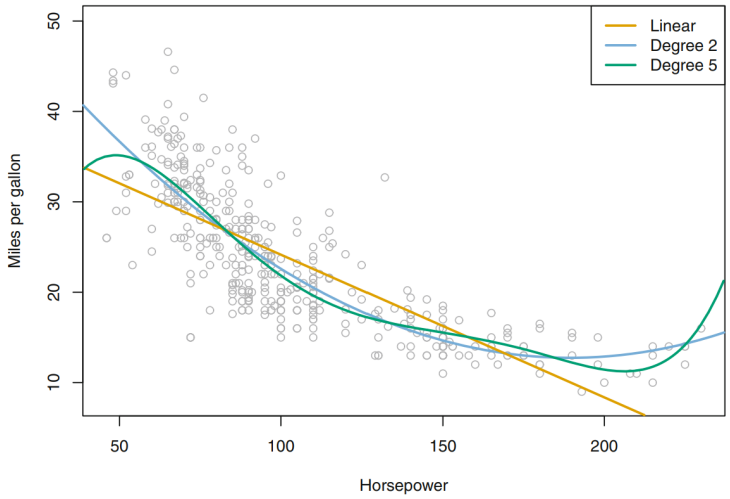


Figure 2: Visualization of Poly reg

Potential Problems

1. non linearity of response-predictor relationships
2. correlation of error terms
3. non-constant variance of error terms
4. outlines
5. high - leverage points
6. col-linearity

1. non linearity of data

Linear regression assumes a straight line fit between predictors and response. If the relationship is far from linear, then conclusions are suspect. Use *residual plots* to identify non linearity.

Plot residuals $e_i = y_i - \hat{y}_i$ vs. predictor x_i. In the case of multiple predictors, plot the residuals vs. the predicted/fitted values of $\hat{y}_i$.

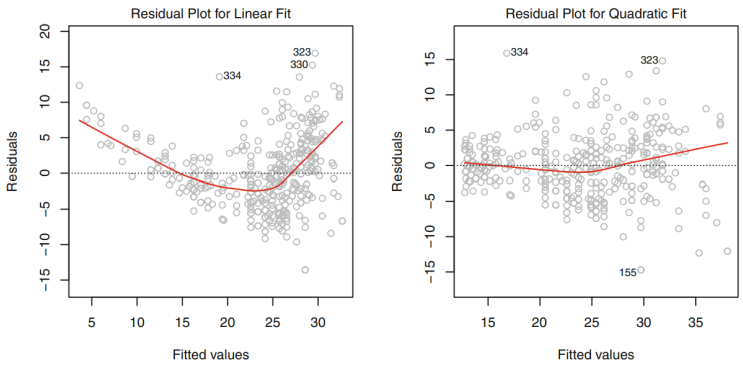


Figure 3: Residual Plot

Ideally the plot should show no pattern where the red line is a smooth fit to the residuals. A horizontal red line shows a random pattern.

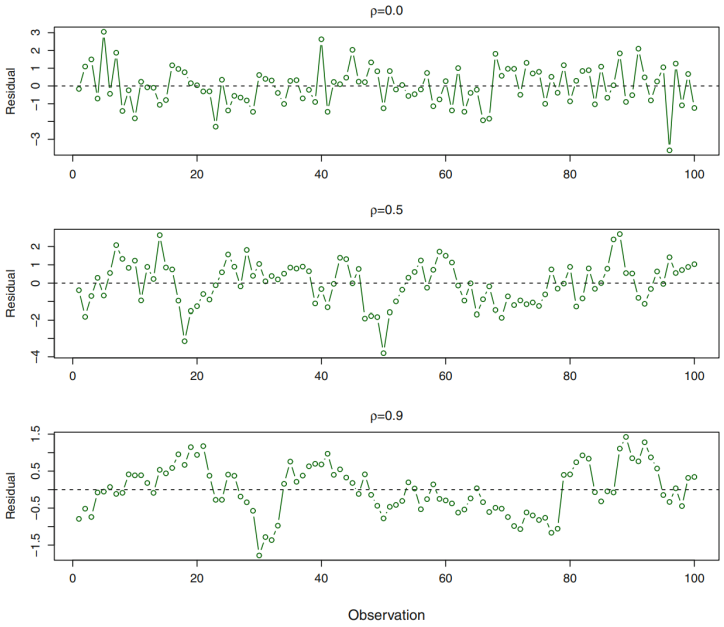
For the plot on the right, transformed variables such as $logX, \sqrt{X}, X^2$ could help.

2. Correlation of error terms An important assumption of the linear regression model is that the error terms $\epsilon 1, \epsilon_2, \ldots, \epsilon_n$ are uncorrelated. This means that ϵ_i is positive provides little to no information about the sign of ϵ_{i+1}.

The standard errors that are computed for the estimated regression coefficients of the fitted values are based on the assumptions of uncorrelated error terms. *If there is a correlation, the estimated standard errors will tend to underestimate the true standard errors.*

Suppose we double our data, then observations and error terms are identical pairs. Our standard error calculations would be if we had sample size fo $2n$ and our estimated parameters would be the same for the $2n$ samples as for the n samples. Our estimated parameters would be the same for the $2n$ samples as for the n samples but the confidence intervals would be narrower by a factor of $\sqrt{2}$.

If error terms are positively correlated, then we may see *tracking* in the residuals - adjacent residual may have similar values.

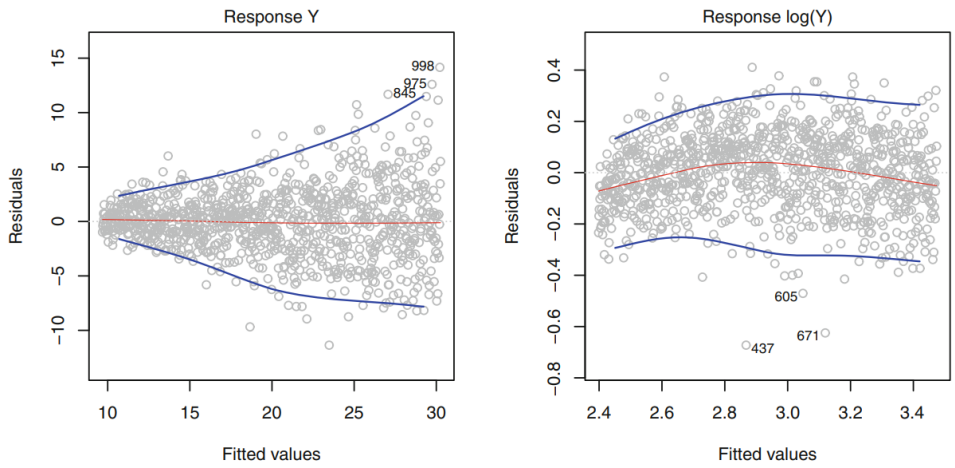


The above graph shows uncorrelated error terms and at the bottom correlated error terms with a coefficient of 0.9. We can see tracking of the residuals.

3. non constant variance of error terms Another assumption is that error terms have a constant variance $Var(\epsilon_i) = \sigma^2$. Standard errors, confidence intervals, and hypothesis tests associated with the linear model all rely on this assumption.

But many times error terms are non-constant. We can identify this by looking at *heteroscedasticity* from the presence of a *funnel shape* in the residual plot. The funnel shape exhibits an increase in variances withe the increase of values of the response.

To remedy this we can transform Y using a concave function like $logY, \sqrt{Y}$. The transformation results in greater mount of shrinkage of the larger responses leading to a reduction in heteroscedasticity.



The image above shows a log transformation of Y given heteroscedasticity.

Sometimes we have a good idea of the variance fo each response. An example of this could be that the ith response could be an average of n_i raw observations. If each of these raw observations is uncorrelated with the variance σ^2 then their average has variance $\sigma_i^2 = \sigma^2/n_i$. We can use *weighted least squares* with weight proportional to the inverse variances.

4. outiers An *outlier* is a point for which y_i is far from the value predicted by the model. Even if an outline does not have much effect on the least squares fit, if can cause issues with the RSE, R^2 and other measures of fit. Residual plots can be used to identify outliers but it can be hard to decide how large an outlier must be to deem as an outlier. WE can then use *standardized residuals* by dividing each residual e_i by its estimated standard error.

Observations who standardized residuals are greater than 3 in absolute value are possible outliers. Sometimes a solution is to just remove the outlier, but we have to be careful f doing this because it could point to a deficiency with the model like a missing predictor.

5. high leverage points

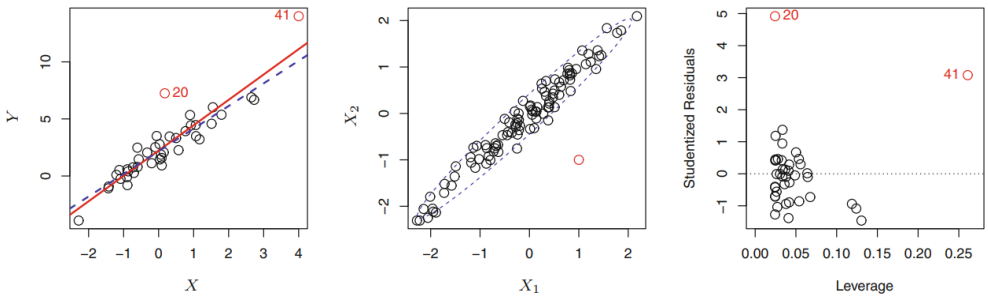
Observations with *high leverage* have an unusual value for x_i. High leverage observations tend to have a sizable impact on the estimated regression line. It's a cause of concern if the least squares line is heavily affected by just a couple of observations because any problems with these points may invalidate the entire fit.

High leverage points can be visually identified as they lay outside th ormal range of observations. To quantify if a observation is a leverage, use the *leverage statistic*,

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2} \tag{3.37}$$

h_i increases with the distance of $x_i - \bar{x}$. The leverage stat is always between $1/n$ and 1 and the average leverage for all observations is always equal to $(p+1)/n$. So if a leverage stat has a value greater than $(p+1)/n$ we can suspect a high leverage.

A data point can have high leverage without being an outlier if it lies far from the mean of the predictor variables but still follows the general trend of the data.



Observation 41 has high leverage and is an outlier, which is a dangerous combination.

6. col-linearity Col-linearity refers to the situation which two or more predictor variables are closely related to one another ie. the predictors are highly correlated. This can cause problems in the regression context and it can be hard to separate individual effects of col-linear variables on the response.

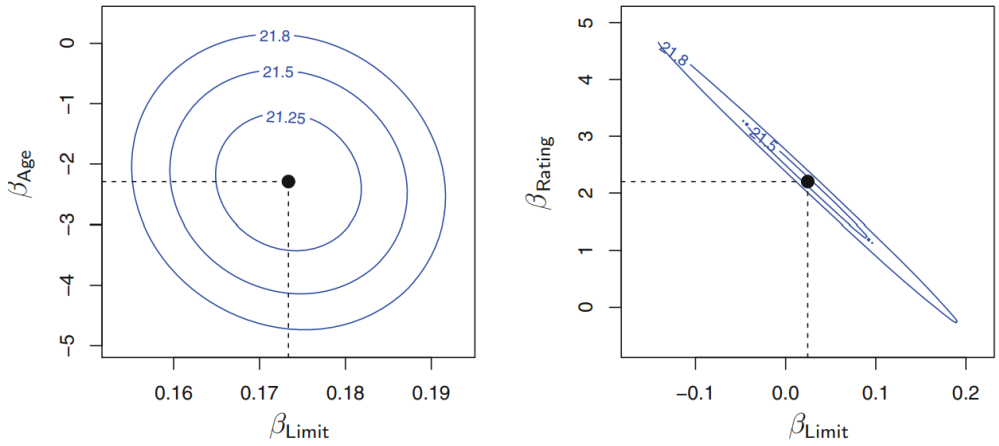


Figure 4: Contour plot

The contour plot on the left shows that there are many pairs that correspond to similar values of RSS. The right contour plot shows distinct minimum RSS and well defined values.

Col-linearity reduces the accuracy of the estimates of the regression coefficient it causes the standard error for $\hat{\beta}_j$ to grow. The t-stat for each predictor uses $\hat{\beta}_j$ and therefore declines the t-stat. The *power* or the probability of detecting a nonzero coefficient decreases as we fail to reject the null hypothesis more frequently.

To identify col linearity look at a correlation matrix of all the predictors. Col-linearity can also exist for three different predictors - this is called *multi-collinearity* and instead of using a matrix we use a *variance inflation factor* or VIF.

A VIF is a ratio of the variance of $\hat{\beta}_j$ when fitting the full model divided by the variance of $\hat{\beta}_j$ if fit on it's own. The smallest value of VIF is 1 and indicates the complete absence of col-linearity.

A value of VIF that exceeds 5 or 10 indicates a problematic amount of col-linearity. The VIF can be computed using the formula,

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

$R^2_{X_j|X_{-j}}$ is the R^2 from a regression of X_j onto all of the other predictors. If $R^2_{X_j|X_{-j}}$ is close to 1, then collinearity is present and the VIF will be large.

Faced with collinearity we can either, 1. drop the problematical variable from the regression 2. combine collinear variables into a single predictor

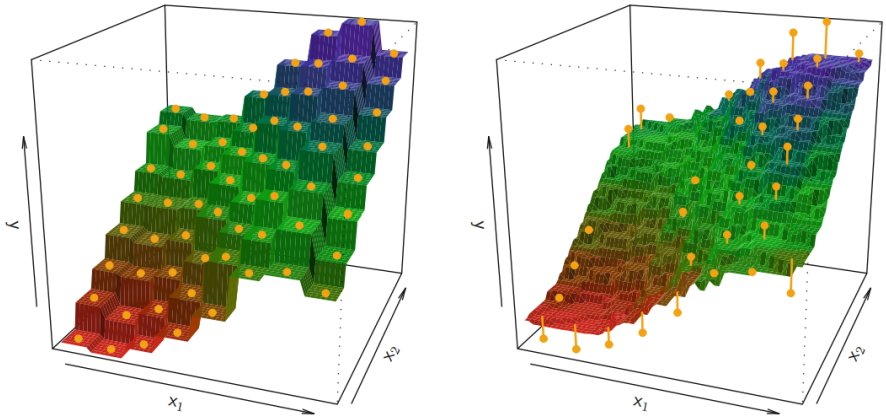
Comparison of linear regression with K-nearest neighbors

Linear regression is an example of a parametric approach because it assumes a linear function form of $f(X)$. Parametric methods have the advantage of easier fit with smaller number of coefficients, simple interpretations, and easy statistical tests. But the downside is that they make strong assumptions of $f(X)$ and therefore prediction accuracy is susceptible to functional forms far from the truth.

non-parametric methods do not explicitly assume a parametric form of $f(X)$ and thereby provide an alternative and more flexible approach for performing regression.

KNN regression is related to KNN classification. Given a value for K and a prediction point x_0, KNN regression first identifies the K training observations that are closet to x_0 represented by $\mathcal{N}_0$. It then estimates $f(x_0)$ using the average of all the training responses in $\mathcal{N}_0$. In other words,

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i$$



Pots of $f(\widehat{X})$ using KNN regression on 2d data. Left plot has value $K = 1$ while right plot has $K = 9$, producing a smoother fit.

$K = 1$ perfectly interpolates the training observations and takes form of a step function.

The optimal value of K depends on a *bias-variance tradeoff*. Too much smoothness can cause bias by masking some of the strucutre of $f(X)$.

IN waht settings will the parametric approahc such as least squares linear regression outperform a non-parametric approach such as KNN?

The parametric approach will outperform the nonparametric approach if the parametric form that has been selected is close to the true form of f.

KNN regression has a problem where as p predictors increase then there is a decrease in performance as there is a reduction in sample size. When we have 100 observations over $p = 20$ prediction this causes the *curse of dimensional* where a given observation has no neighboring. K observations that are nearest to a given test observation x_0 may be very far away from x_0 in p-dimensional space when p is large, leading to a poor KNN fit.

As a general rule: *parametric models will tend to outperform non-parametric approaches when there is a small number of observations per predictor.*

R-lab

```r
# simple linear regression
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```r
# model fit
lm.fit<-lm(medv~lstat, data = Boston)

# same thing
attach(Boston)
lm.fit<-lm(medv~lstat)

lm.fit
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

```r
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
# coefficient quantities
coef(lm.fit)
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

```r
# confidence interval
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

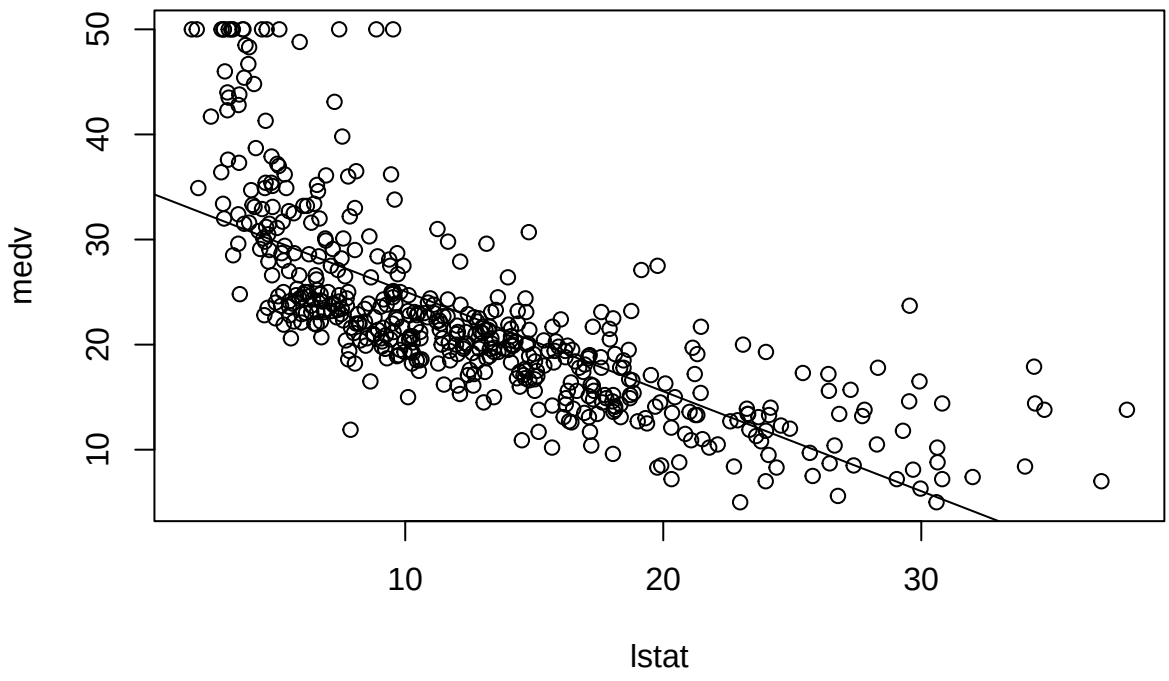
```r
# conf intervals and pred intervals
predict(lm.fit, data.frame(lstat=(c(5, 10, 15))), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

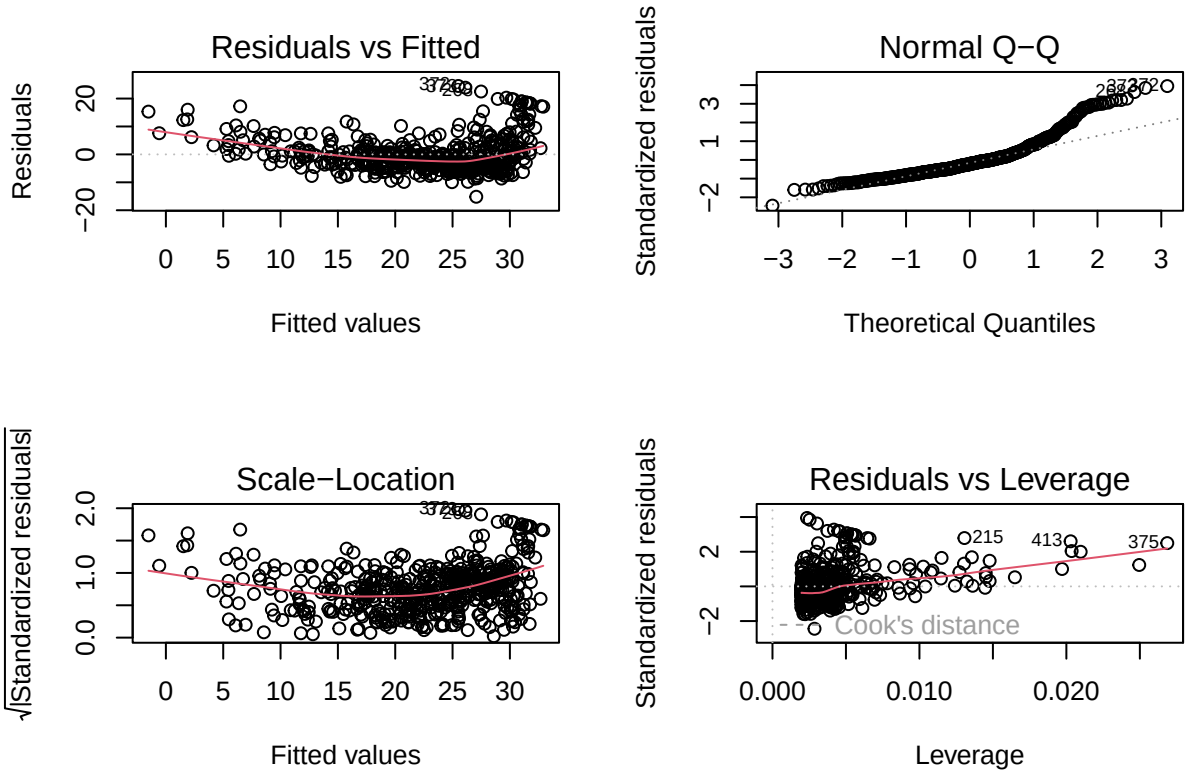
```r
predict(lm.fit, data.frame(lstat=(c(5, 10, 15))), interval = "prediction")
```

```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

```r
# plot
plot(lstat, medv)
abline(lm.fit)
```

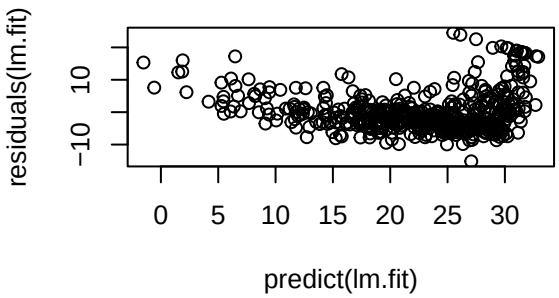


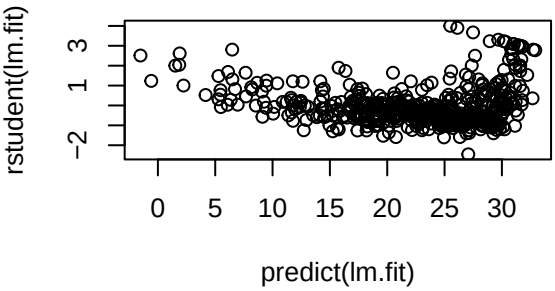
```
# diagnostic plots
par(mfrow=c(2,2))
plot(lm.fit)
```

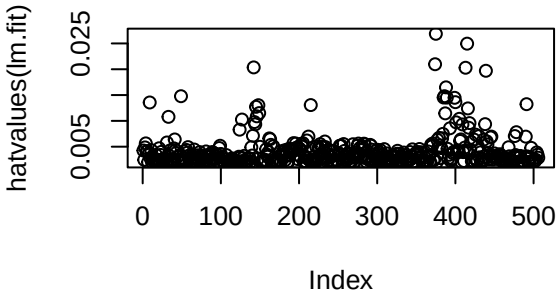



```
# compute residuals
plot(predict(lm.fit), residuals(lm.fit))

# studentized residual plot
plot(predict(lm.fit), rstudent(lm.fit))

# leverage statistics
plot(hatvalues(lm.fit))
which.max(hatvalues(lm.fit)) # identifies index of largest ele of vector
```

```
## 375
## 375
```







MLR Lab

```
# MLR
lm.fit <- lm(medv~lstat + age, data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
```

```
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

```
# short hand
lm.fit <- lm(medv~., data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
# indv components of summary
summary(lm.fit)$r.sq
```

```
## [1] 0.7406427
```

```
summary(lm.fit)$sigma
```

```
## [1] 4.745298
```

```
#VIF
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.3
```

```
## Loading required package: carData

## Warning: package 'carData' was built under R version 4.2.3
```

```
vif(lm.fit)
```

```
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.792192 2.298758 3.991596 1.073995 4.393720 1.933744 3.100826 3.955945
##      rad      tax  ptratio    black    lstat
## 7.484496 9.008554 1.799084 1.348521 2.941491
```

```
# subset regression
lm.fit1 <- lm(medv~.-age, data=Boston)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
## indus         0.020562   0.061433   0.335 0.737989
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
# same thing update()
lm.fit1 <- update(lm.fit, ~.-age)
```

Interaction term lab

Use syntax lstat:black to include an interaction term between lstat and black. lstat*age inlcudes lstat, age, and interaction term.

```r
summary(lm(medv~lstat*age, data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Non linear transformations of predictors

Use I(X^2) wraps X such that we can transform predictors.

Anova performs a hypothesis test between the two models to see which model is superior.

Use poly() function to create the polynomial.

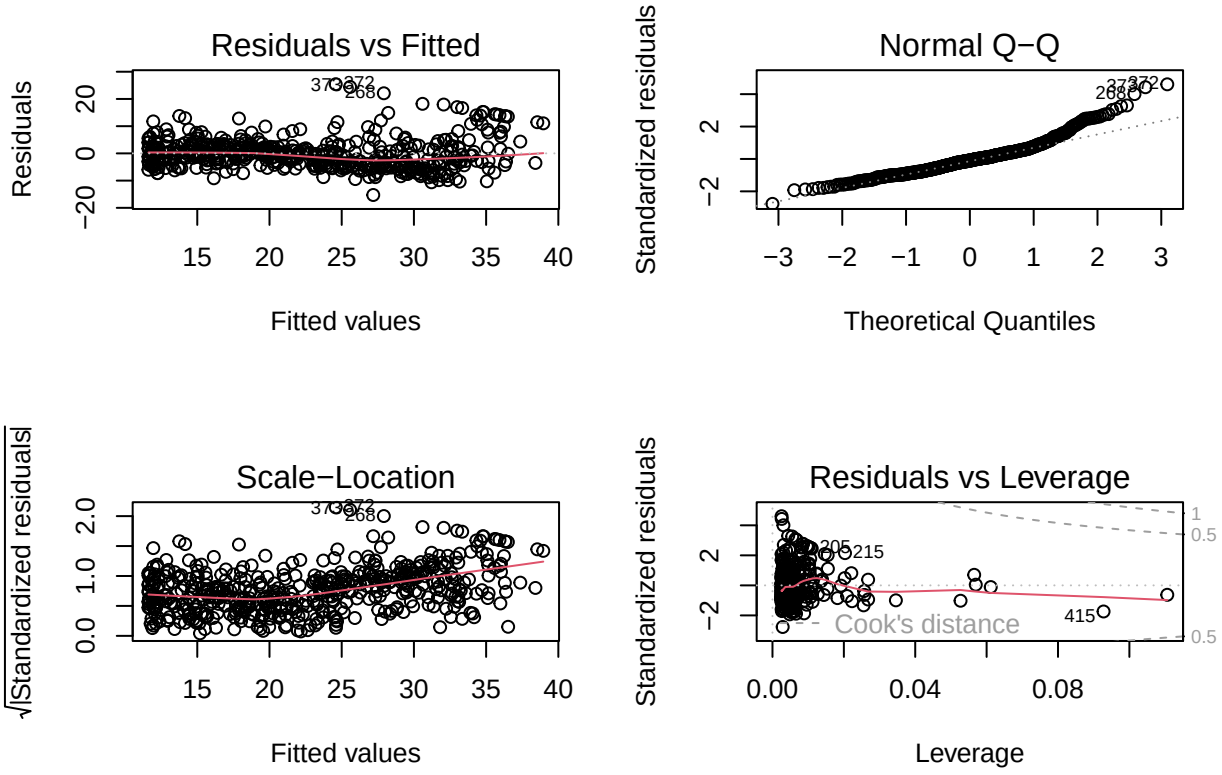
```r
# transformed
lm.fit2 <- lm(medv~lstat + I(lstat^2))
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

```r
# anova()
lm.fit <- lm(medv~lstat)
anova(lm.fit,lm.fit2)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# plot
par(mfrow = c(2, 2))
plot(lm.fit2)
```




```r
# poly reg
lm.fit5 <- lm(medv~poly(lstat, 5))
summary(lm.fit5)
```

```
##
## Call:
```

```
## lm(formula = medv ~ poly(lstat, 5))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5433  -3.1039  -0.7052   2.0844  27.1153
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       22.5328     0.2318  97.197  < 2e-16 ***
## poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
## poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
## poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
## poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
## poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.215 on 500 degrees of freedom
## Multiple R-squared:  0.6817, Adjusted R-squared:  0.6785
## F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

```
# log transformation
summary(lm(medv~log(rm), data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ log(rm), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.487  -2.875  -0.104   2.837  39.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -76.488      5.028  -15.21   <2e-16 ***
## log(rm)       54.055      2.739   19.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.915 on 504 degrees of freedom
## Multiple R-squared:  0.4358, Adjusted R-squared:  0.4347
## F-statistic: 389.3 on 1 and 504 DF,  p-value: < 2.2e-16
```

Qualitative predictors

contrasts() returns the coding that R uses for the dummy variables

```
names(Carseats)
```

```
##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"
```

```r
# MLR
lm.fit <- lm(Sales ~.+Income:Advertising + Price:Age, data = Carseats)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9208 -0.7503  0.0177  0.6754  3.3413
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.5755654  1.0087470   6.519 2.22e-10 ***
## CompPrice            0.0929371  0.0041183  22.567  < 2e-16 ***
## Income               0.0108940  0.0026044   4.183 3.57e-05 ***
## Advertising          0.0702462  0.0226091   3.107 0.002030 **
## Population           0.0001592  0.0003679   0.433 0.665330
## Price               -0.1008064  0.0074399 -13.549  < 2e-16 ***
## ShelveLocGood        4.8486762  0.1528378  31.724  < 2e-16 ***
## ShelveLocMedium      1.9532620  0.1257682  15.531  < 2e-16 ***
## Age                 -0.0579466  0.0159506  -3.633 0.000318 ***
## Education           -0.0208525  0.0196131  -1.063 0.288361
## UrbanYes             0.1401597  0.1124019   1.247 0.213171
## USYes               -0.1575571  0.1489234  -1.058 0.290729
## Income:Advertising   0.0007510  0.0002784   2.698 0.007290 **
## Price:Age            0.0001068  0.0001333   0.801 0.423812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.011 on 386 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8719
## F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16
```

```r
# contrast()
attach(Carseats)
contrasts(ShelveLoc)
```

```
##        Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```

Excersises

1. The null hypothesis states that the coefficents TV, radio, newspaper, sales are 0, ie that there is no relationship to the response such that TV = radio = newspaper = sales = 0.

2. The KNN classifier attempts to predict the class to which the output variable belongs to by computing local probability (using bayes). The KNN regression model predicts the value of the output by the local average.

3.

a) iv

b) 50 + 4.0(20) + 110(0.07) + 440(0.01) + -10(1) = 137,100
c) false, the coefficient of the interaction term does not determine the statistical significance of the interaction effect. Instead, we need to use a hypothesis test and p values.

8.

```
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

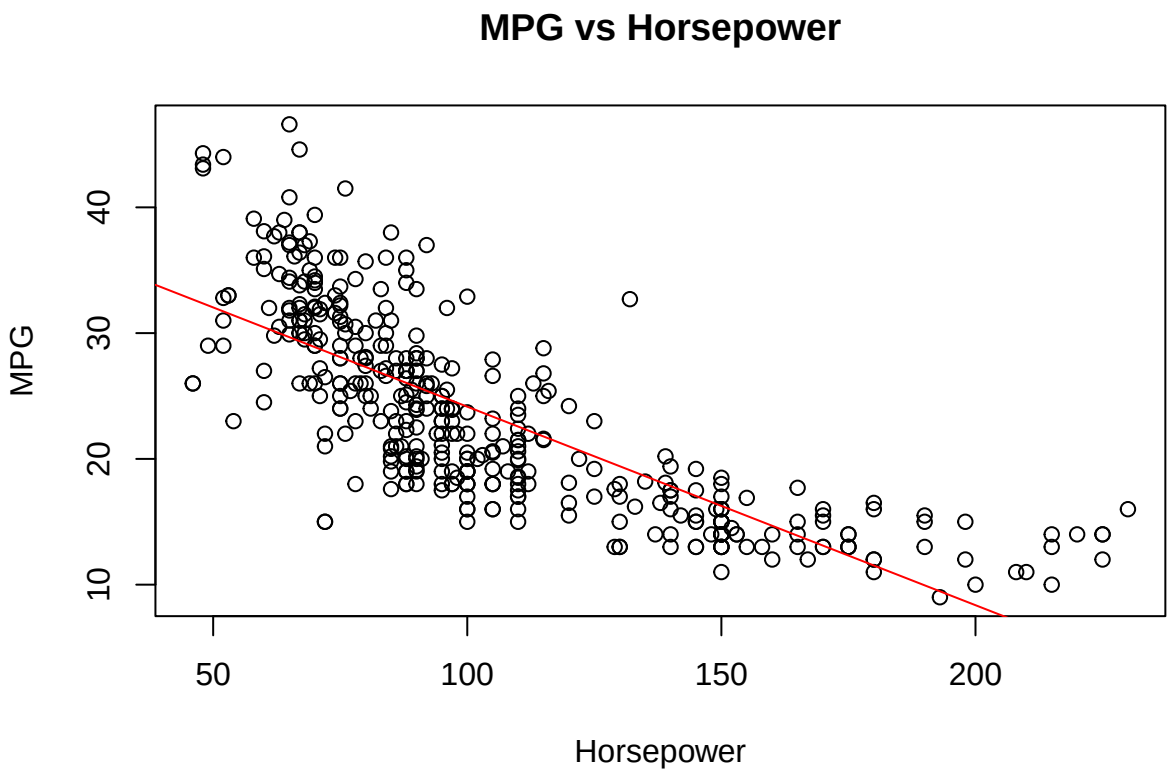
```
# SLR
lm.fit <- lm(mpg ~ horsepower, data = Auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

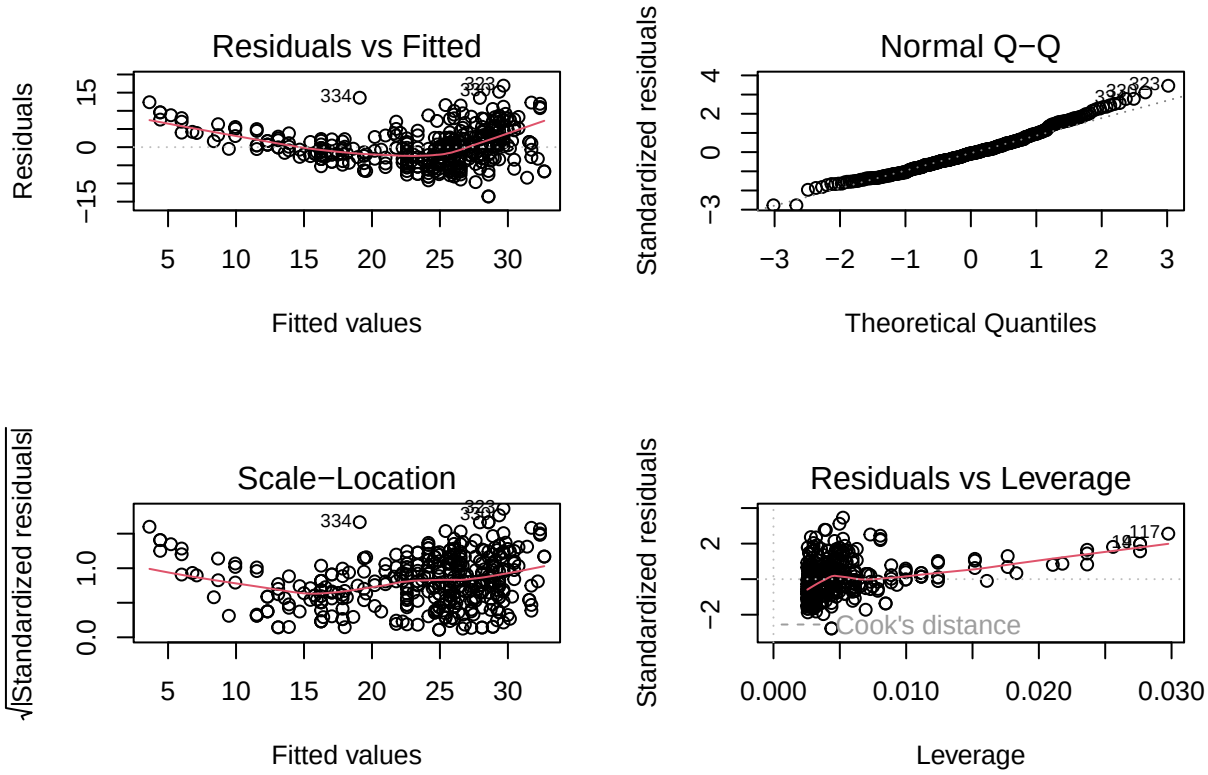
The p-value is sufficiently small such that we can conclude that there is a relationship between horsepower and mpg. The estimated coefficient of horsepower shows a negative correlation, ie that for every one unit increase of horsepower there is a 0.157 decrease in mpg. Our adjusted R^2 value shows a mid/strong correlation.

Given a horsepower value of 98, the predicted mpg is 24.47.

```
# plot
plot(Auto$horsepower, Auto$mpg, main = "MPG vs Horsepower", xlab="Horsepower", ylab="MPG" )
abline(lm.fit, col = "red")
```




```
# diagnostic plots
par(mfrow = c(2,2))
plot(lm.fit)
```



Th Residuals vs Fitted plot shows some non-linearity. We could further trasnform the predictors (log, power, ect).

The Normal QQ plot shoes a close fit to the reference line, supporting that the residuals follow a normal distribution.

The residuals vs leverage plot shows points that are both high leverage and high std. residuals.

10. a, b)

```
names(Carseats)
```

```
##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"
```

```
# MLR
mlr.fit <- lm(Sales~Price+Urban+US, data = Carseats)
summary(mlr.fit)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

```
# inspect classification
attach(Carseats)
```

```
## The following objects are masked from Carseats (pos = 3):
##
##     Advertising, Age, CompPrice, Education, Income, Population, Price,
##     Sales, ShelveLoc, Urban, US
```

```
contrasts(Urban)
```

```
##     Yes
## No    0
## Yes   1
```

```
contrasts(US)
```

```
##     Yes
## No    0
## Yes   1
```

For every one unit increase in Price, sales decrease by 0.05. Cars that are urban are expected to have 0.02 lesser sales compared to non urban cars. Cars made in the US are expected to have 1.2 greater sales compared to foreign cars.

c)

$$sales = 13.04 - 0.05(Price) - 0.02(Urban) + 1.2(US)$$

d) Given a large p value of 0.936, we can reject the null hypothesis of Urban.

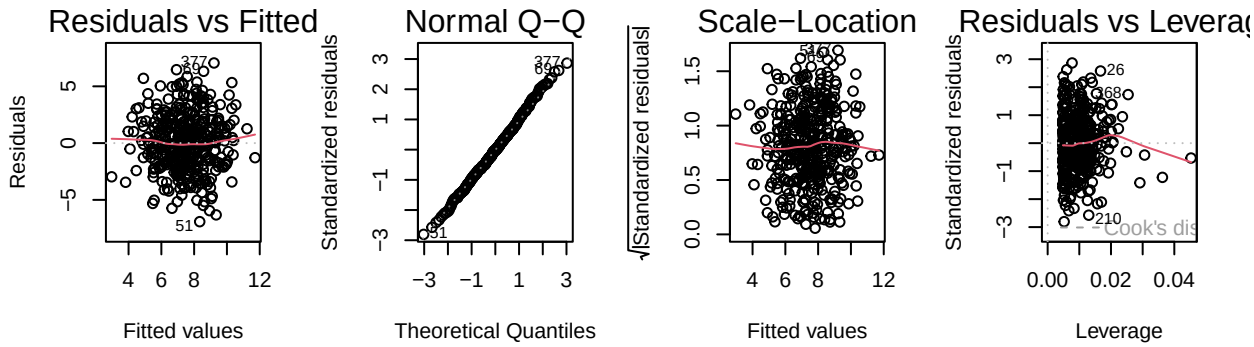
e)

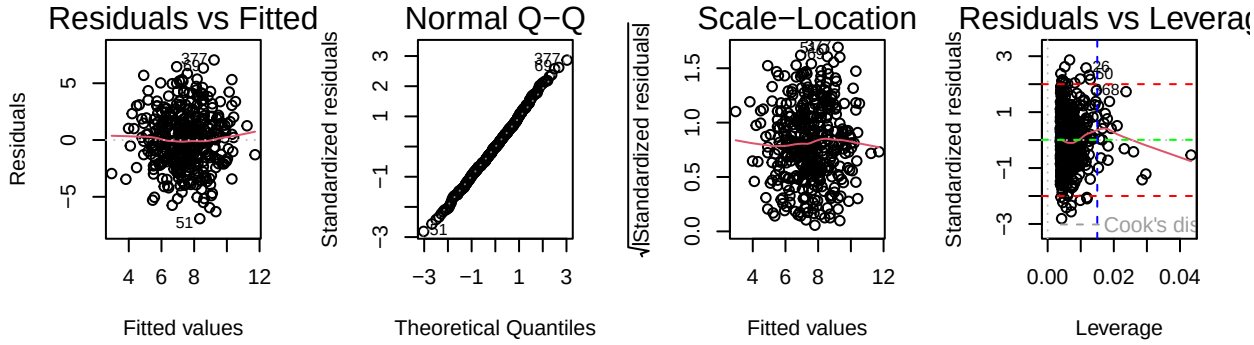
```
# MLR
mlr.fit <- lm(Sales~Price+US, data = Carseats)
summary(mlr.fit)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

f)

```r
model_e <- lm(Sales~Price+Urban+US, data = Carseats)
model_f <- lm(Sales~Price+US, data = Carseats)

par(mfrow = c(2, 4))
plot(model_e)
plot(model_f)

# lines for high leverage
leverage <- hatvalues(model_f)
std_residuals <- rstandard(model_f)

avg_leverage <- mean(leverage)
high_leverage <- 2 * avg_leverage

abline(h = c(-2, 2), col = "red", lty = 2)  # Lines for standardized residuals
abline(v = high_leverage, col = "blue", lty = 2)  # Line for high leverage threshold

# lines for influential points
n <- nrow(Carseats)
inf_line <- 4 / n
abline(h = inf_line, col = "green", lty = 2)
```





Model f is better given the reduction of predictors still produces the same results given the diagnostic plots. A reduced MLR with lesser predictors is better than a model with extraneous predictors.

g)

```
# conf interval
confint(model_f)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

h) yes, seen from the additionally plotted threshold lines, there are few points with high std. residuals and high leverage as well as a combination of the two.

12.

a) Given the equation in 3.38, we have

$$(\sum_{i=1}^{n} x_i y_i)/(\sum_{i=1}^{n} x_{i'}^2) = (\sum_{i=1}^{n} x_i y_i)/(\sum_{i=1}^{n} y_{i'}^2)(\sum_{i=1}^{n} x_{i'}^2) = (\sum_{i=1}^{n} y_{i'}^2)$$

Therefore the TSS of each varibale must be the same.