Linear Regression

ES/STT 7140: Statistical Modeling for Environmental Data

Brandon M. Greenwell

A good modelling tool

At a minimum:

- Universally applicable in classification and regression
- Unexcelled accuracy
- Capable of handling large data sets
- Handle missing values effectively

A good modelling tool

At a minimum:

- Universally applicable in classification and regression
- Unexcelled accuracy
- Capable of handling large data sets
- Handle missing values effectively

It would also be a plus to know:

- Which variables are important?
- How do variables interact?
- What is the shape of the data (i.e., how does it cluster?)
- How do the features separate classes?
- Are their novel cases and outliers?

A good modelling tool

At a minimum:

- Universally applicable in classification and regression
- Unexcelled accuracy
- Capable of handling large data sets
- Handle missing values effectively

It would also be a plus to know:

- Which variables are important?
- How do variables interact?
- What is the shape of the data (i.e., how does it cluster?)
- How do the features separate classes?
- Are their novel cases and outliers?

-Leo Breiman

Statistical models

Introduction

- In the one-sample t-test, we are interested in learning about the mean of a normal distribution/population

$$y_i \sim N\left(\mu, \sigma^2\right), \quad i = 1, 2, \ldots, n$$

- For example, y might represent the shell length of a randomly selected zebra mussel from a stream or lake in Michigan

- It is often convenient to think of the data y_i in terms of a statistical model:

$$y_i = \mu + \epsilon_i, \quad \epsilon_i \sim N\left(0, \sigma^2\right)$$

Introduction

- In the one-sample t-test, we are interested in learning about the mean of a normal distribution/population

$$y_i \sim N\left(\mu, \sigma^2\right), \quad i = 1, 2, \ldots, n$$

- For example, y might represent the shell length of a randomly selected zebra mussel from a stream or lake in Michigan

- It is often convenient to think of the data y_i in terms of a statistical model:

$$y_i = \mu + \epsilon_i, \quad \epsilon_i \sim N\left(0, \sigma^2\right)$$

  - *data = mean + remainder*

  - The above two expressions are mathematically equivalent

  - The remainder is the difference between the observed values and the mean, often refered to as the residuals

Introduction

- In the two-sample t-test problem, we are interested in the difference between the means of two populations (or groups):

$$y_{1i} \sim N\left(\mu_1, \sigma^2\right), \quad i = 1, 2, \ldots, n_1$$

$$y_{2j} \sim N\left(\mu_2, \sigma^2\right), \quad j = 1, 2, \ldots, n_2$$

$$\delta = \mu_2 - \mu_1$$

- As a linear model, we could use

$$y_k = \mu_1 + \delta g_k + \epsilon_k, \quad \sim N\left(0, \sigma^2\right), \quad k = 1, 2, \ldots, n_1 + n_2$$

  - Here, g_k is a *dummy variable* (one for `intake` group and zero for `discharge` group)

Introduction

- In the two-sample t-test problem, we are interested in the difference between the means of two populations (or groups):

$$y_{1i} \sim N\left(\mu_1, \sigma^2\right), \quad i = 1, 2, \ldots, n_1$$

$$y_{2j} \sim N\left(\mu_2, \sigma^2\right), \quad j = 1, 2, \ldots, n_2$$

$$\delta = \mu_2 - \mu_1$$

- As a linear model, we could use

$$y_k = \mu_1 + \delta g_k + \epsilon_k, \quad \sim N\left(0, \sigma^2\right), \quad k = 1, 2, \ldots, n_1 + n_2$$

  - Here, g_k is a *dummy variable* (one for `intake` group and zero for `discharge` group)

- Illustration: `clams.R`

- A similar approach can be used for ANOVA procedures as well

Introduction

- The primary focus of this chapter is to introduce more complicated models

Introduction

- The primary focus of this chapter is to introduce more complicated models
- In particular, we will focus on

Introduction

- The primary focus of this chapter is to introduce more complicated models
- In particular, we will focus on
    - *simple linear regression* (SLR) models (i.e., linear regression with a single predictor)

Introduction

- The primary focus of this chapter is to introduce more complicated models
- In particular, we will focus on
    - *simple linear regression* (SLR) models (i.e., linear regression with a single predictor)
    - *multiple linear regression* (MLR) models (i.e., linear regression with multiple predictors)

Introduction

- The primary focus of this chapter is to introduce more complicated models
- In particular, we will focus on
    - *simple linear regression* (SLR) models (i.e., linear regression with a single predictor)
    - *multiple linear regression* (MLR) models (i.e., linear regression with multiple predictors)
    - *nonlinear regression* (NLR) models

Introduction

- The primary focus of this chapter is to introduce more complicated models
- In particular, we will focus on
    - *simple linear regression* (SLR) models (i.e., linear regression with a single predictor)
    - *multiple linear regression* (MLR) models (i.e., linear regression with multiple predictors)
    - *nonlinear regression* (NLR) models
- In the next chapter, we will look at a more general class of regression models called *generalized linear models* (GzLMs)
    - GzLMs include both *logistic regression* and *Poisson regression* models

Simple linear regression

The SLR model

- A regression model is a formal means of expressing the two essential ingredients of a statistical model:

    1. A tenancy of the **response variable**, y, with a **predictor variable**, x, in some systematic fashion

    2. A scattering of points around the hypothesized curve of statistical relationship

- These two characteristics are embodied in a regression model by postulating that:

    1. There is a **probability distribution** of y for each level of x

    2. The means of these probability distributions vary in some systematic fashion with x

The SLR model

- There are many reasons for the vast popularity of regression models in statistical practice
- Regression models allow us to relate variables together in a mathematical form which can provide insight into the relationships between the variables of interest

The SLR model

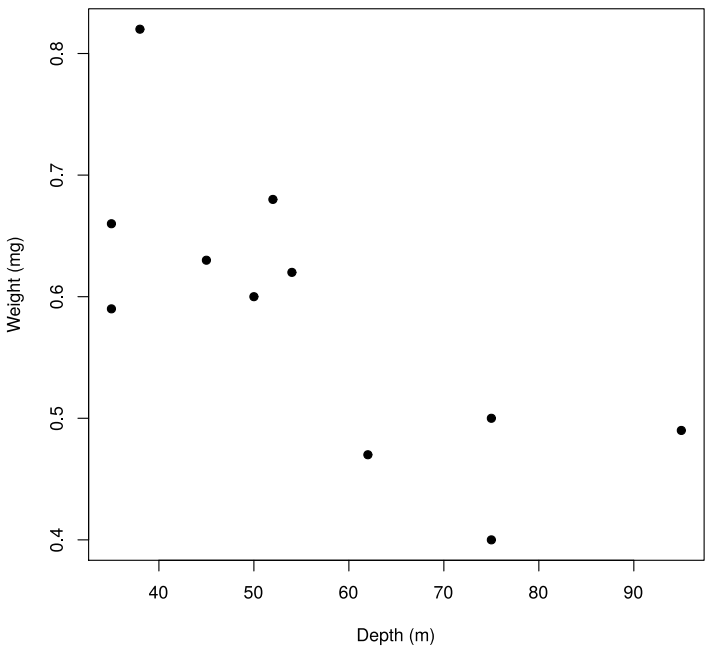
- There are many reasons for the vast popularity of regression models in statistical practice
- Regression models allow us to relate variables together in a mathematical form which can provide insight into the relationships between the variables of interest
- Regression models allow us to **determine statistically** if a *response variable* is related to one or more other *predictor variables*
    - For instance, we may want to determine the effect of increasing levels of DDT on eggshell thicknesses. How does increasing levels of DDT effect eggshell thickness?

The SLR model

- There are many reasons for the vast popularity of regression models in statistical practice
- Regression models allow us to relate variables together in a mathematical form which can provide insight into the relationships between the variables of interest
- Regression models allow us to **determine statistically** if a *response variable* is related to one or more other *predictor variables*
    - For instance, we may want to determine the effect of increasing levels of DDT on eggshell thicknesses. How does increasing levels of DDT effect eggshell thickness?
- Another common use of regression models is to **predict a response**
    - For instance, if water is contaminated with a certain level of toxin, can we predict the amount of accumulation of this toxin in a fish that lives in the water?

The SLR model

- Many statistical applications deal with modeling how a single response variable, denoted y, depends on a single predictor, denoted x
- Illustration: `diporeia.R`



The SLR model

- From the previous *scatterplot*, one can see a fairly strong relationship between between the weight of the *Diporeia* and the depth of water where the Diporeia are found
- The scatterplot suggests that a straight line relationship between weight of Diporeia (y) and water depth (x) may be a reasonable way to model the data:

$$weight = \beta_0 + \beta_1 depth$$

- Here, β_0 is the y-intercept of the regression line and β_1 is the slope (i.e., *rate of change*)
- Of course, the Diporeia data do not lie perfectly on a straight line and a probabilistic model is needed to account for the variability of points about the line:

The SLR model

- From the previous *scatterplot*, one can see a fairly strong relationship between between the weight of the *Diporeia* and the depth of water where the Diporeia are found
- The scatterplot suggests that a straight line relationship between weight of Diporeia (y) and water depth (x) may be a reasonable way to model the data:

$$weight = \beta_0 + \beta_1 depth$$

- Here, β_0 is the y-intercept of the regression line and β_1 is the slope (i.e., *rate of change*)
- Of course, the Diporeia data do not lie perfectly on a straight line and a probabilistic model is needed to account for the variability of points about the line:

$$weight_i = \beta_0 + \beta_1 depth_i + \epsilon_i, \quad i = 1, \ldots, 11$$

Assumptions of the SLR model

- The SLR model

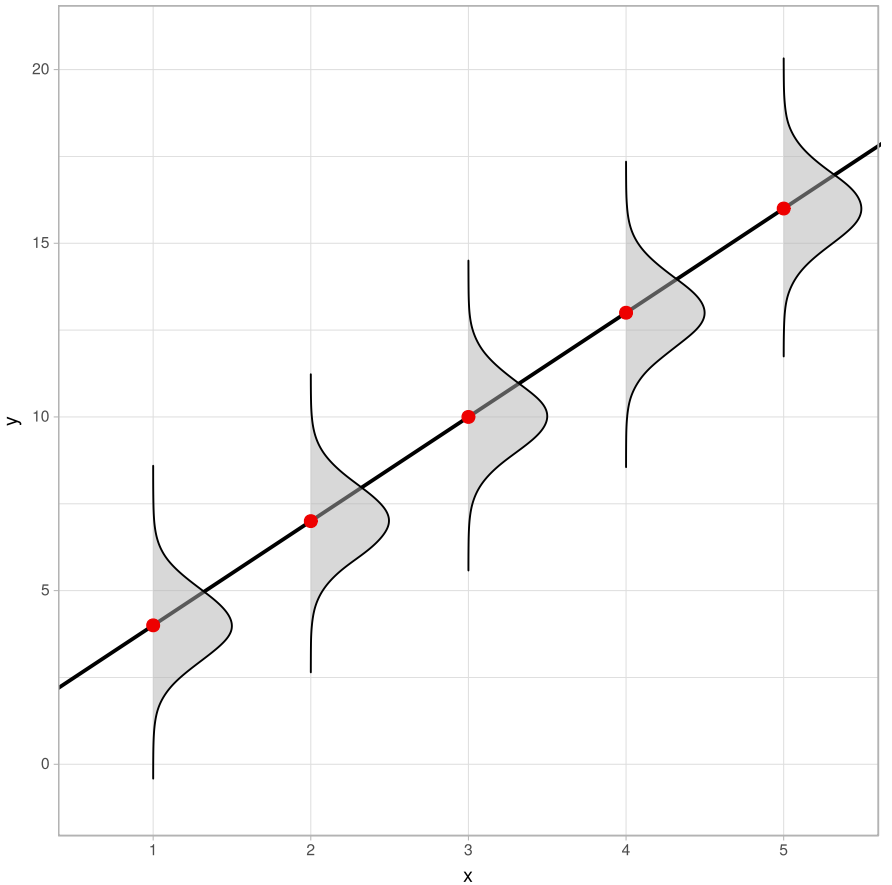
$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \ldots, n$$

  assumes that

  1. Independent observations (i.e., the random errors are independent)
  2. The errors have constant variance (i.e., *homoscedasticity*)
  3. The errors are normally distributed (for statistical inference)

- If these assumptions are not met, then alternative methods need to be applied (e.g., *weighted least squares* or *mixed-effects models*)

Assumptions of the SLR model



Least squares estimation

- How do we estimate the model coefficients β_0 and β_1?
- There are an infinite number of lines passing through the data points $\{x_i, y_i\}_{i=1}^n$
- The *least squares* (LS) solution seeks to find β_0 and β_1 that minimize the *sum of squares*:

$$SS(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- Hence, the LS line is the "best" fitting line in terms of minimizing $SS(\beta_0, \beta_1)$

Least squares estimation

- How do we estimate the model coefficients β_0 and β_1?
- There are an infinite number of lines passing through the data points $\{x_i, y_i\}_{i=1}^n$
- The *least squares* (LS) solution seeks to find β_0 and β_1 that minimize the *sum of squares*:

$$SS(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- Hence, the LS line is the "best" fitting line in terms of minimizing $SS(\beta_0, \beta_1)$
- So, how do we minimize $SS(\beta_0, \beta_1)$?

Least squares estimation

- How do we estimate the model coefficients β_0 and β_1?
- There are an infinite number of lines passing through the data points $\{x_i, y_i\}_{i=1}^n$
- The *least squares* (LS) solution seeks to find β_0 and β_1 that minimize the *sum of squares*:

$$SS(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- Hence, the LS line is the "best" fitting line in terms of minimizing $SS(\beta_0, \beta_1)$
- So, how do we minimize $SS(\beta_0, \beta_1)$?
  - **CALCULUS!!**

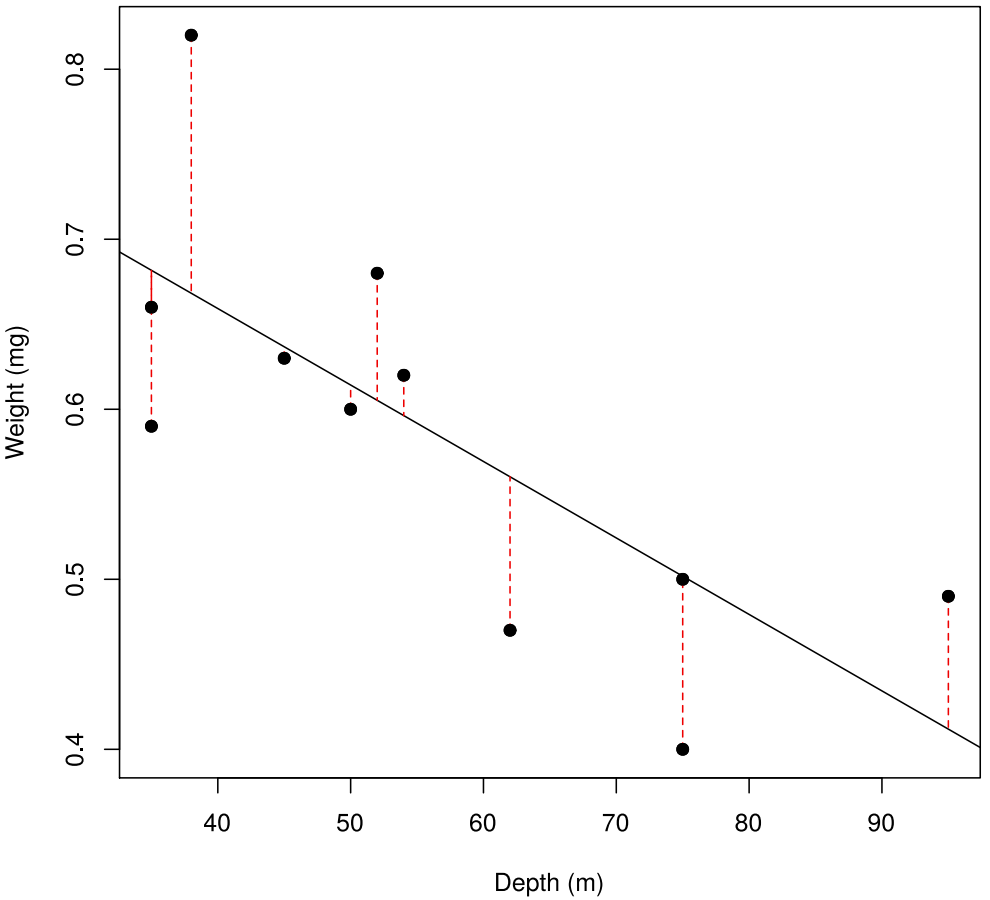
Least squares estimation

- The values of β_0 and β_1 that minimize $SS(\beta_0, \beta_1)$ are given by
  - $\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
  - $\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$

Least squares estimation

- The values of β_0 and β_1 that minimize $SS(\beta_0, \beta_1)$ are given by
  - $\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
  - $\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$
- These are called the LS estimators of β_0 and β_1
- Under the usual assumptions for the SLR model (normality not required), the LS estimators:
  - Are **unbiased** estimators of β_0 and β_1
  - Have **minimum variance** among all *linear* unbiased estimators of β_0 and β_1!
- How do we interpret $\widehat{\beta}_0$ and $\widehat{\beta}_1$ for a fitted SLR model?

Least squares estimation



The SLR model

- The SLR model is belongs to a broad class of models called *linear models* (LMs)
    - In an LM, the response is a **linear function of the coefficients**
    - Later on we'll see how to deal with nonlinear models where the response is not linearly related to the model parameters
- The classic two-sample t-test and ANOVA are linear models where the predictors are indicators for the levels of the factors involved
- In the R software, the `lm()` function can be used to fit regression models
    - Illustration `diporeia.R`

The SLR model

- The SLR model is belongs to a broad class of models called *linear models* (LMs)
    - In an LM, the response is a **linear function of the coefficients**
    - Later on we'll see how to deal with nonlinear models where the response is not linearly related to the model parameters
- The classic two-sample t-test and ANOVA are linear models where the predictors are indicators for the levels of the factors involved
- In the R software, the `lm()` function can be used to fit regression models
    - Illustration `diporeia.R`
- For the Diporeia example, we have $\hat{y} = 0.839135 - 0.004498x$, where $\hat{y}$ is the *predicted value* of y

Inference in the SLR model

- Typically, the parameter of primary interest in SLR is the slope, β_1
- The slope measures the average rate of change in y relative to x
- Occasionally, interest also lies in the y-intercept, β_0, but usually only in cases where x values are collected **near the origin**
- Otherwise, the y-intercept may not have any practical meaning
- For the Diporeia example, the estimate slope is $\widehat{\beta}_1 = -0.0045$ (**how do we interpret this number?**)
- In SLR, it is natural to ask whether or not the **slope differs significantly from zero**.
    - If the slope equals zero and the model is correctly specified, then y will not depend on x (i.e., a horizontal regression line)
    - If the relation is quadratic, then one could fit a straight line and get an estimated slope near zero which could be very misleading (**always plot your data**)

Inference in the SLR model

- If $\epsilon \sim N\left(0, \sigma^2\right)$, then $\widehat{\beta}_1 \sim N\left(\beta, \sigma^2_{\widehat{\beta}_1}\right)$ (**why?**)
- The formulas for $\widehat{SE}\left(\widehat{\beta_0}\right)$ and $\widehat{SE}\left(\widehat{\beta_1}\right)$ are messy, but are provided my most statistical software
    - In R, these are located in the column labeled `Std. Error` after applying the `summary()` function (e.g., `summary(slr)`)

Inference in the SLR model

```
summary(slr)
```

```
##
## Call:
## lm(formula = weight ~ depth)
##
## Residuals:
##       Min         1Q    Median         3Q       Max
## -0.101819 -0.056004 -0.006746  0.049235  0.151772
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.839135   0.081394  10.310 2.77e-06 ***
## depth       -0.004498   0.001382  -3.254  0.00993 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08363 on 9 degrees of freedom
## Multiple R-squared:  0.5406,    Adjusted R-squared:  0.4896
## F-statistic: 10.59 on 1 and 9 DF,  p-value: 0.009926
```

Inference in the SLR model

- In SLR, one can test the following hypothesis:

$$H_0 : \beta_1 = \beta_{10} \quad vs. \quad H_1 : \beta_1 \neq \beta_{10}$$

  - Typically, $\beta_{10} = 0$

- If the null hypothesis is true, then

$$t_{obs} = \frac{\widehat{\beta}_1 - \beta_{10}}{\widehat{SE}\left(\widehat{\beta_1}\right)}$$

  will have a t-distribution with $n - 2$ degrees of freedom

- From this result,we can conduct various tests of hypotheses regarding β_1 and construct $100\,(1-\alpha)\%$ confidence intervals for β_1

Inference in the SLR model

- In SLR, one can test the following hypothesis:

$$H_0 : \beta_1 = \beta_{10} \quad vs. \quad H_1 : \beta_1 \neq \beta_{10}$$

  - Typically, $\beta_{10} = 0$

- If the null hypothesis is true, then

$$t_{obs} = \frac{\widehat{\beta}_1 - \beta_{10}}{\widehat{SE}\left(\widehat{\beta_1}\right)}$$

  will have a t-distribution with $n - 2$ degrees of freedom

- From this result,we can conduct various tests of hypotheses regarding β_1 and construct $100\,(1 - \alpha)\%$ confidence intervals for β_1

- Similar results hold for the y-intercept, β_0

The coefficient of determination

- One of the most important statistics in regression analysis is the coefficient of determination, better known as the R^2 ("R-squared")
- In the SLR model, R^2 is just the square of the (Pearson) correlation between x and y
- In MLR (i.e., when we have more than one predictor), a more general definition of R^2 is required
- The value of R^2 is always between zero and one and **represents the proportion of variability in the response that is explained by the regression model**

The coefficient of determination

- One of the most important statistics in regression analysis is the coefficient of determination, better known as the R^2 ("R-squared")
- In the SLR model, R^2 is just the square of the (Pearson) correlation between x and y
- In MLR (i.e., when we have more than one predictor), a more general definition of R^2 is required
- The value of R^2 is always between zero and one and **represents the proportion of variability in the response that is explained by the regression model**
- Illustration: `diporeia.R`

"All models are wrong, but some are useful."

-George Box

Accessing the model fit

- In using the SLR to model the Diporeia data, we assumed that the relationship between `weight` and `depth` is linear
- The true *functional relationship* between `weight` and `depth` is probably not linear 😱
- So, does that mean that the SLR model is wrong in this example?
- Of course not, **statistical models are only approaximations to the truth!!** (it's rather odd to call an approximation wrong)
- For the Diporeia data, the SLR seems to provide a **reasonable** approximation!!
    - Other statistical models may be just as, if not more, reasonable
- Of course, the SLR may not provide a reasonable approximation to the truth in many cases
- The most common way to assess the fit of a regression models is through a thorough examination of the *residuals*

Accessing the model fit

- Given a fitted SLR model, we can compute a prediction for the i-th observation: $\hat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$ for $i = 1, 2, \ldots, n$ (these are called the *fitted values*)
- Ideally, if the model fit is good, then the differences between the observed and fitted values should be small:

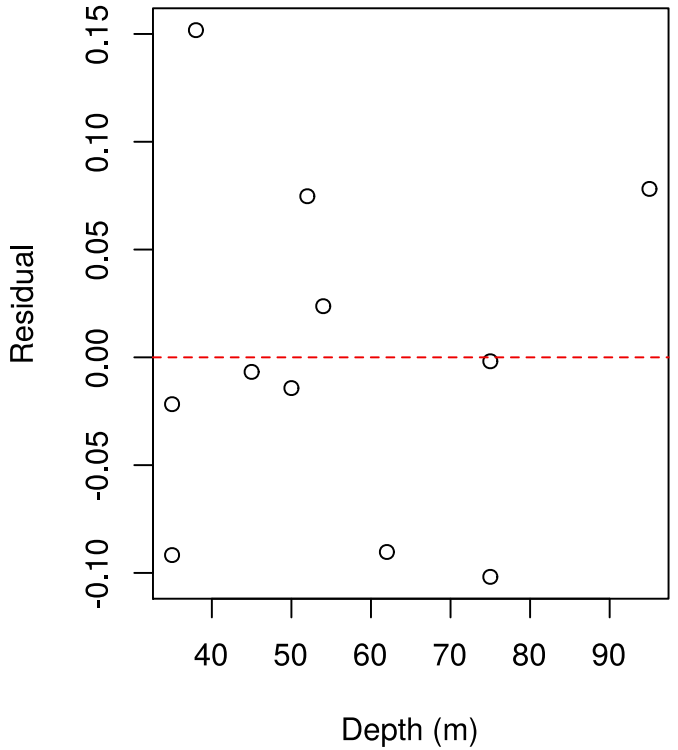
$$r_i = y_i - \hat{y}_i, \quad i = 1, 2, \ldots, n$$

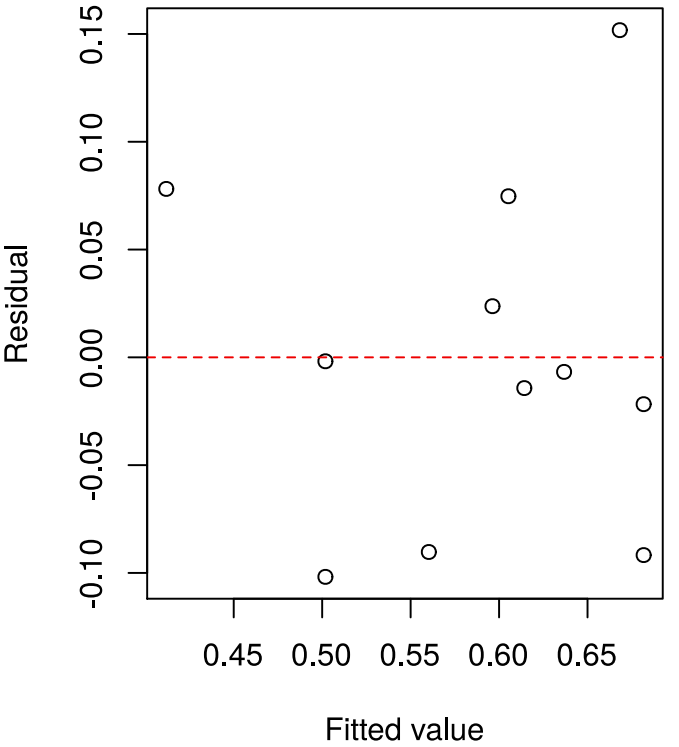
- These differences are referred to as the *residuals*
- One can think of the i-th residual as an estimated of the i-th error term: $\epsilon_i = y_i - \beta_0 - \beta_1 x_i$
- Recall that the errors in the SLR model are assumed to be random and have constant variance; hence, scatterplots of the residuals versus x_i or $\hat{y}_i$ should look like a random scatter of points!
    - In other words, if the model is specified correctly, scatterplots of the residual should not show any structure

Accessing the model fit

```
# Compute residuals
resids <- weight - fitted(slr)

# Setup for side-by-side plots
par(mfrow = c(1, 2))

# Residual vs. depth
plot(depth, resids, xlab = "Depth (m)", ylab = "Residual")
abline(h = 0, lty = 2, col = "red2")  # reference line

# Residual vs. fitted value
plot(fitted(slr), resids, xlab = "Fitted value", ylab = "Residual")
abline(h = 0, lty = 2, col = "red2")  # reference line
```

Accessing the model fit





Accessing the model fit

- Most often, it is better to work with the *standardized residuals* and *studentized residuals*
- The *mean squared error* (MSE) provides an estimate of the constant variance, σ^2:

$$MSE = \sum_{i=1}^{n} r_i^2/(n-2)$$

  - The *root-mean squared error* (RMSE), $RMSE = \sqrt{MSE}$, is often used to compare fitted regression models (e.g., using k-fold cross-validation)

- The standardized residuals are defined as $r_i/\sqrt{MSE}$
- The studentized residuals are defined as $r_i/\sqrt{MSE\,(1-h_{ii})}$, where h_{ii}, called a *hat value*, measures how far x_i is from the other x values (to be discussed more in-depth later!)

Accessing the model fit

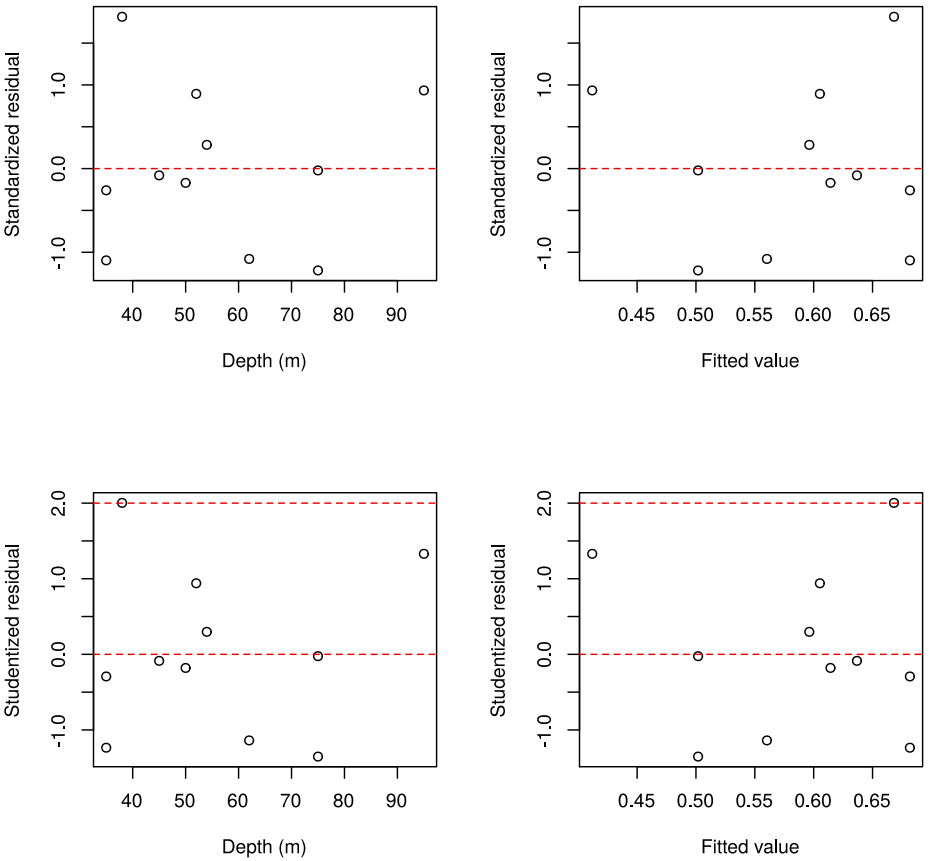
```
# Compute MSE
mse <- sum(resids ^ 2) / (length(resids) - 2)

# Compute standardized and studentized residuals
(r_stan <- resids / sqrt(mse))
```

```
##           1           2           3           4           5           6
## -1.09676754 -0.25973316  1.81482819 -0.08066253 -0.17049172  0.89367894
##           7           8           9          10          11
##  0.28378082 -1.07962457 -1.21751940 -0.02175600  0.93426697
```

```
(r_stud <- resids / sqrt(mse * (1 - hatvalues(slr))))
```

```
##           1           2           3           4           5           6
## -1.23500390 -0.29246987  2.00339174 -0.08618023 -0.17978796  0.93955891
##           7           8           9          10          11
##  0.29781080 -1.13849222 -1.35237291 -0.02416571  1.32959703
```

Accessing the model fit



Accessing the model fit

- The studentized residuals, as defined on the previous slide, are sometimes referred to as standardized residuals 😱:

```
r_stud
```

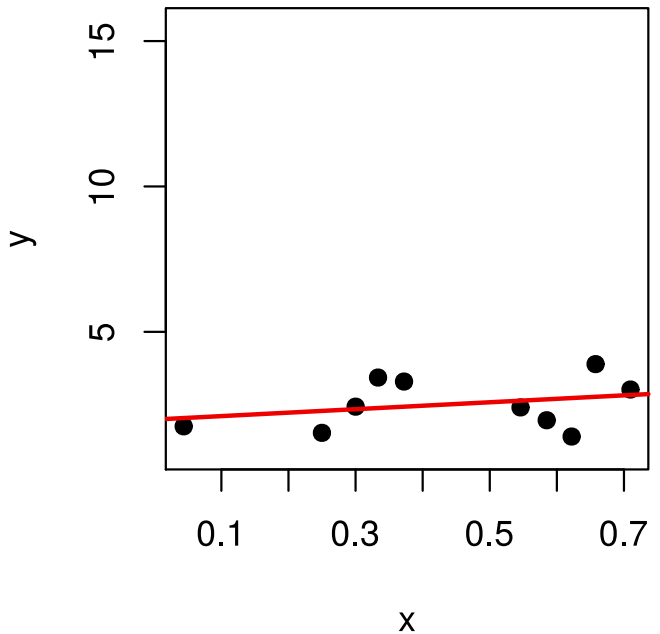
```
##           1           2           3           4           5           6
## -1.23500390 -0.29246987  2.00339174 -0.08618023 -0.17978796  0.93955891
##           7           8           9          10          11
##  0.29781080 -1.13849222 -1.35237291 -0.02416571  1.32959703
```

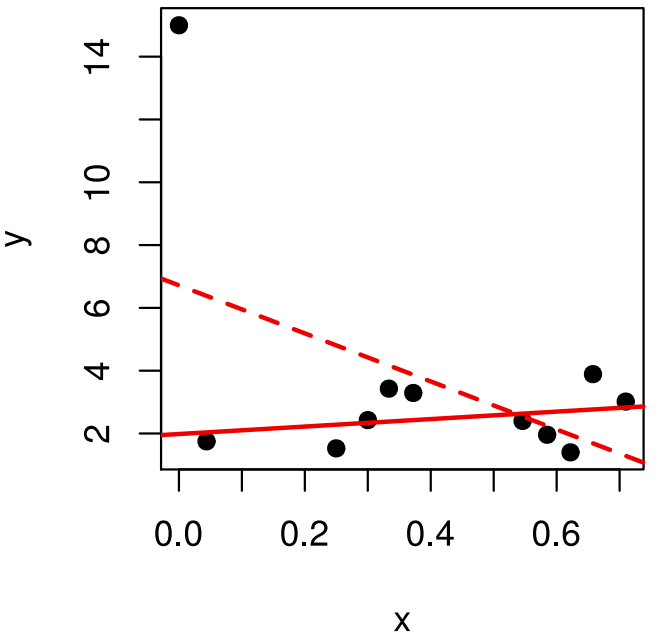
```
rstandard(slr)  # built-in R function
```

```
##           1           2           3           4           5           6
## -1.23500390 -0.29246987  2.00339174 -0.08618023 -0.17978796  0.93955891
##           7           8           9          10          11
##  0.29781080 -1.13849222 -1.35237291 -0.02416571  1.32959703
```

Leave-one-out

- An *extreme point* in the predictor space tends to pull the regression lines towards itself!


y
5
10
15
0.1
0.3
0.5
0.7
x


y
2
4
6
8
10
14
0.0
0.2
0.4
0.6
x

Leave-one-out

- An *extreme point* in the predictor space tends to pull the regression lines towards itself!
- This influential effect can mask points that have a strong influence on the estimated slope of the line (especially in the case where there are many predictors) since this effect will tend to make the residual less extreme
- An effective way to address this problem is to use the *jacknife*:
  - For $i = 1, 2, \ldots, n$:

    1) Remove the i-th observation

    2) Recompute the LS estimates, denoted $\widehat{\beta}_{0(i)}$ and $\widehat{\beta}_{1(i)}$.

    3) Obtain the corresponding fitted value $\hat{y}_{(i)} = \widehat{\beta}_{0(i)} + \widehat{\beta}_{1(i)} x_i$

    4) Obtain the *leave-one-out* residual $r_{(i)} = y_i - \hat{y}_{(i)}$
- The $r_{(i)}$, for $i = 1, 2, \ldots, n$, are known as the *PRESS* residuals

Leave-one-out

- Fortunately, the PRESS residuals can be computed without having to refit the model n times:

$$r_{(i)} = r_i / \left(1 - h_{ii}\right)$$

- The PRESS statistic

$$PRESS = \sum_{i=1}^{n} r_{(i)}^2$$

  provides a useful measure of the predictive performance of the model

  - It is equivalent to *n-fold cross-validation* or *leave-one-out-cross-valudation*
  - k-fold cross validation is more general (5 and 10 are typical choices for k)

- Why bother use k-fold cross-validation? What's wrong with using $\sum_{i=1}^{n} r_i^2$?

Walleye example

Data on Walleye fish that were caught in Butternut Lake, Wisconsin at three different periods:

```
# Load the data
url <- paste0("https://raw.githubusercontent.com/bgreenwell/stt7140-e
              "master/data/walleye.csv")
walleye <- read.csv(url)
```

```
## <simpleError in parse_block(g[-1], g[1], params.src): duplicate lab
```

```
head(walleye)
```

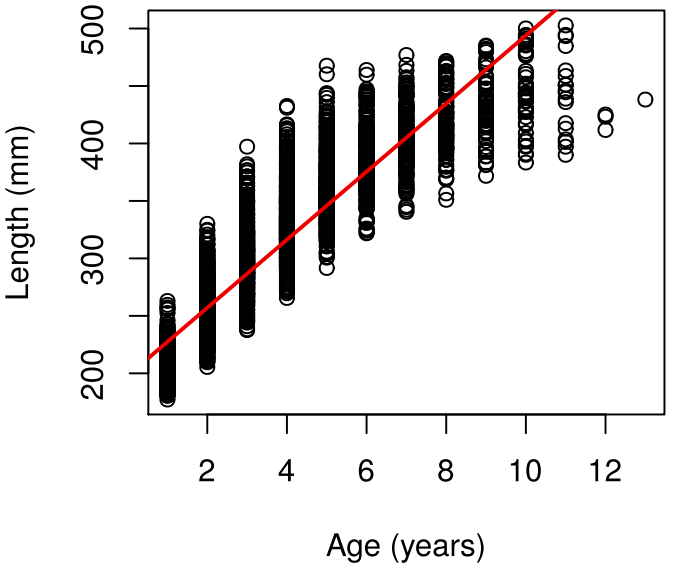
```
##   age   length period
## 1   1 215.2540      1
## 2   1 193.2576      1
## 3   1 202.5781      1
## 4   1 201.4597      1
## 5   1 232.0309      1
## 6   1 191.0207      1
```

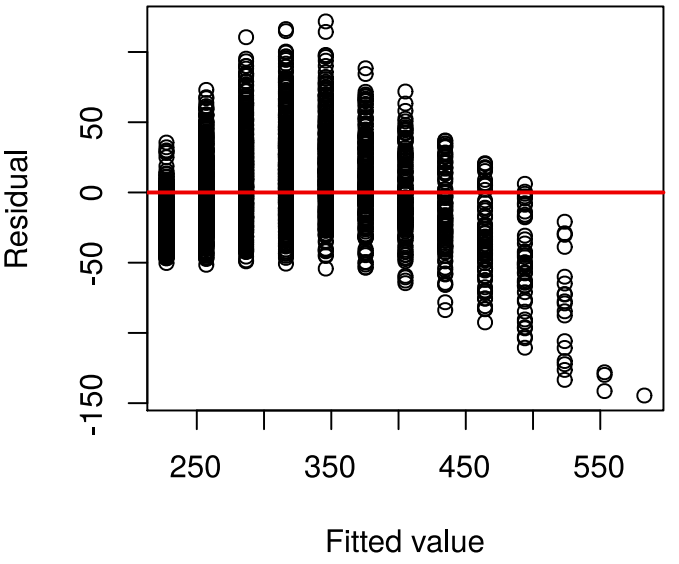
Walleye example

```r
# Fit an SLR model
walleye_slr <- lm(length ~ age, data = walleye)
summary(walleye_slr)  # print model summary
```

```
##
## Call:
## lm(formula = length ~ age, data = walleye)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -144.562  -21.966   -3.468   20.148  121.770
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 197.8332     1.0593   186.8   <2e-16 ***
## age          29.6099     0.2474   119.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.82 on 3196 degrees of freedom
```

Walleye example





Estimation versus prediction

- Regression models are often used to predict a new response or estimate a mean response for a given value of the predictor x
- We have seen how to compute a predicted value $\hat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x$
- However, as with any estimate, we need a measure of reliability associated with $\hat{y}_0$.
- Returning to the Diporeia example, the fitted regression line is given by

$$\hat{y} = 0.83914 - 0.0045x$$

  where x is the water depth and y is the weight of the Diporeia
- Suppose we want to predict the weight of a Diporeia at a depth of 40 meters. Then we would simply plug $x = 40$ into the estimated regression equation to get a predicted value of $\hat{y} = 0.83914 - 0.0045(40) = 0.65914$ mg.

Estimation versus prediction

- Regression analysis is really a problem of estimating a conditional mean or expectation, denoted

$$E\left[y|x\right] = \beta_0 + \beta_1 x$$

- Here, $E\left[y|x\right]$ corresponds to the average value of the response y for all units in the population with a specific x value
- Suppose we want to estimate the mean weight of the Diporeia found at a depth of 40 meters
    - For the prediction problem, we want to predict the weight of a **single Diporeia**
    - For the estimation problem, we want to estimate the mean of a conditional population (i.e., the population of Diporeia found at a depth of 40 meters)
    - In both cases, we use $\hat{y} = 0.65914$ as the predicted weight and as the estimate of the mean weight of Diporeia found at a depth of 40 meters
- In other words, the point estimate is th same for prediction and estimation, the difference lies in the estimated standard error of each!

Estimation versus prediction

- There is more uncertainty associated with prediction a single new observation (**why?**)
- For a given value of x, it is customary to compute **confidence intervals for an estimated mean response** and a **prediction interval for a single new response value**
- The idea of a prediction interval is to determine an interval that will contain a certain percentage of the population
    - Because a prediction interval is attempting to capture a single, random future response, as opposed to the mean of the conditional population, it will be wider than the associated confidence interval
- Formulas on board (and in text)

Examples

- Using the `predict()` function in R
- Anscombe's quartet of "identical" simple linear regressions

Cautionary notes

- Extrapolation
- Outliers and robust regression

Multiple linear regression

Background

- The multiple linear regression (MLR) model generalizes the SLR model to a model with $p > 1$ predictors:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_p x_p + \epsilon$$

- Typical assumptions include:

    1. Independent observations
    2. Constant error variance
    3. The error term has a normal distribution

- The coefficients can still be estimated using the least-squares method
- The interpretation of the regression coefficients is similar to that of the slope in SLR, but trickier

Coefficient of determination

- Similar to ANOVA, the total variability in y (SST) can be partitioned into two components:
  - The variability due to the regression (SSR)
  - The variability due to error (SSE)
- SST = SSR + SSE
- The general definition for the coefficient of determination is

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

  - Generally, we want R^2 to be "high", but the definition of "high" depends on the application
  - R^2 can be arbitrarily inflated by adding more predictors (regardless of their association with y)
  - $R^2_{adj} = 1 - \left(\frac{n-1}{n-p}\right)\frac{SSE}{SST}$

The regression F-test

- Just as in ANOVA, an F-test can be conducted by comparing SSR and SSE
  - $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0 \quad vs. \quad H_1$: not so
  - $F_{obs} = \frac{SSR/p}{SSE/(n-p-1)}$
  - If H_0 is true, then F_{obs} will have come from an F-distribution with p numerator degrees of freedom and $n - p - 1$ denominator degrees of freedom
- The individual coefficients can also be tested using the usual t-test approach (on the board)
- Beware of the *multiple testing problem*
  - Whenever tests (or confidence intervals) for multiple parameters are considered, some sort of multiplicity correction should be made (i.e., *Bonferroni* or *Scheffes method*)
  - This is common when constructing confidence/prediction bands for a fitted regression curve

Partial F-tests

- We can also use F-tests to test if a specific subset of the regression coefficients are all zero
    - This can be accomplished by using the sums of squares from both the full and reduced models (details omitted, but can be found in the online notes)
    - In R, this is trivial using the `anova()` function:

```
full_model <- lm(y ~ x1 + x2 + x3 + x4)
reduced_model <- lm(y ~ x1 + x4)
anova(reduced_model, full_model)
```

- This can be useful for model building, but better techniques are available (e.g., *forward*, *backward*, and *stagewise regression*)

Mercury example

```
url <- paste0("https://raw.githubusercontent.com/bgreenwell/",
              "eesR/master/R/Data/HgBass.txt")
hg_bass <- read.table(url, header = TRUE)
# pairs(hg_bass)  # scatterplot matrix

# Full and reduced models
fm1 <- lm(log(Avg.Mercury) ~ log(Alkalinity) + pH + log(Calcium) +
            log(Chlorophyll), data = hg_bass)
summary(fm1)
```

```
##
## Call:
## lm(formula = log(Avg.Mercury) ~ log(Alkalinity) + pH + log(Calciu
##     log(Chlorophyll), data = hg_bass)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56640 -0.34854  0.09526  0.25318  1.57607
##
```