Linear Regression

36-600

The Setting

- Linear regression is an inferential (read: inflexible) model in which we assume that Y is related to the predictor variables $\mathbf{x}$ via the model

$$Y|\mathbf{x} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

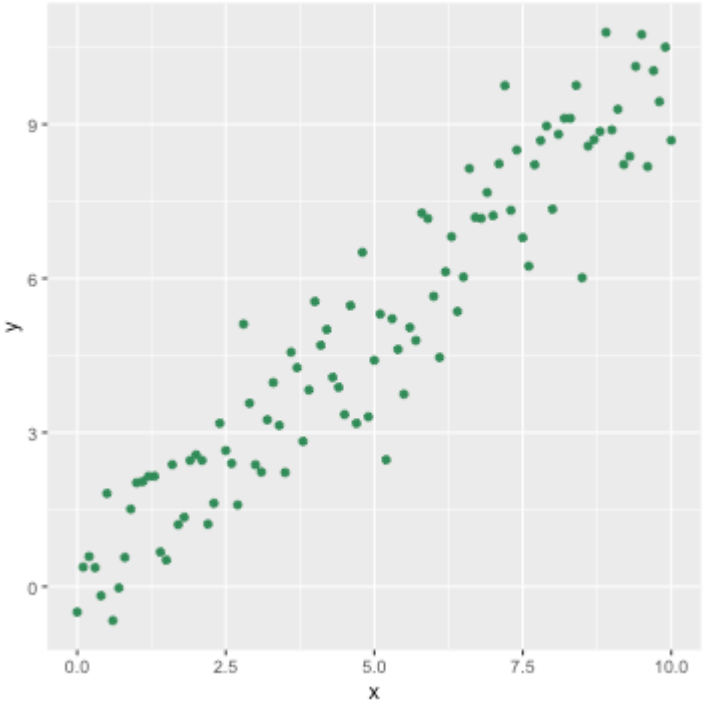
  - ϵ is a random variable that is assumed (1) to have mean zero, (2) to have variance σ^2 that is constant as a function of x, and (3) (optional...we will return to this point) to be normally distributed

- Why would we use linear regression?
  - while it is inflexible, it is also readily interpretable
  - it is a fast model to learn: the β's can be computed via formula
  - to be clear: it is not necessarily the case that there is an *a priori* belief that $\mathbf{x}$ and Y are exactly linearly related

Linear Regression: Example

- Here are some observed data:

```
set.seed(303)
x        <- seq(0,10,by=0.1)
y        <- x + rnorm(length(x),sd=1)
df       <- data.frame(x,y)
s        <- sample(length(x),round(0.7*length(x)))
df.train <- df[s,]  # training set
df.test  <- df[-s,] # test set
```



Linear Regression: Example

```
# that's a tilde between y and x
lm.out <- lm(y~x,data=df.train)
summary(lm.out)
```

```
##
## Call:
## lm(formula = y ~ x, data = df.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7207 -0.6929  0.0499  0.7333  2.3347
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.04017    0.22962  -0.175    0.862
## x            1.00559    0.04098  24.536   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9554 on 69 degrees of freedom
## Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8957
## F-statistic:   602 on 1 and 69 DF,  p-value: < 2.2e-16
```

- ...and here we regress the response variable Y upon x
- the estimated coefficients are
  - $\hat{\beta}_0 = -0.040$
  - $\hat{\beta}_1 = 1.006$
- the estimated probability that one would observe a value of 1.006 or larger (or -1.006 or smaller) is $\leq 10^{-16}$
  - this is (much) less than 0.05, so we conclude that the true value of β_1 is not zero
  - i.e., there is a significant association between x and y!

Linear Regression: Example

- Caveats to keep in mind regarding p values:
  - if the true value of a coefficient β_i is equal to zero, then the p-value will be sampled from a Uniform(0,1) distribution (so there's a 5% chance that you'll conclude there's a significant association between a predictor variable and the response even when there is none)
  - as the sample size n gets large, the estimated coefficient uncertainty goes to zero, and so *all* associations are eventually deemed statistically significant, even if they do not have practical significance

$\Rightarrow$ Rather than using p-values, use variable selection methods (covered soon) to determine which subset of the predictor variables should be included in your final model

Linear Regression: Example

- To compare the linear regression model against other models, we compute the mean-squared error for the data in the *test* set

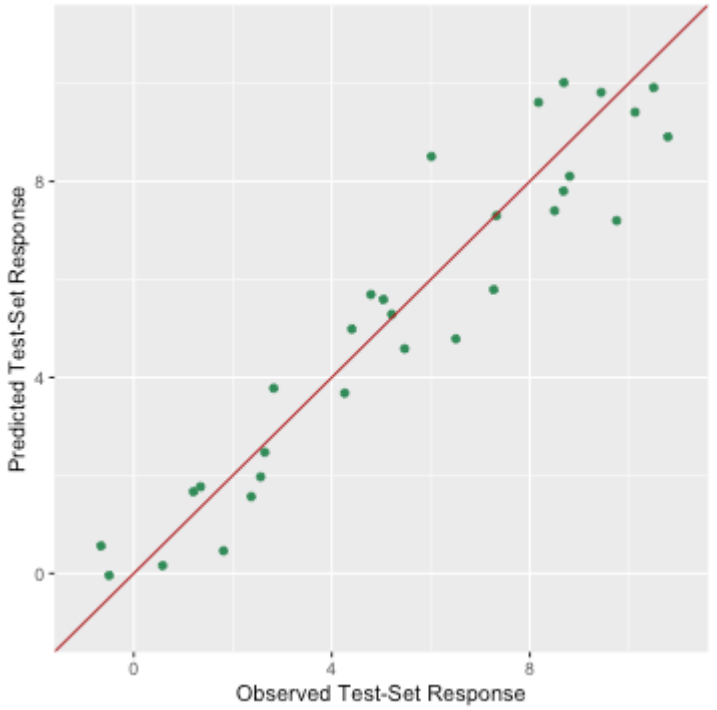
```
y.hat <- predict(lm.out,newdata=df.test)
mean((df.test$y-y.hat)^2)
```

```
## [1] 1.244703
```

- To determine the quality of the linear regression model as a representation of the data-generating process in absolute terms, we use "Adjusted R-squared" (which has value 0.8957)
  - it is an estimate of the proportion of the variance of the data along the y-axis that is explained by the linear regression model
  - values near zero indicate that there is no apparent association between the predictor variables and the response variable
  - values near one indicate a strong linear association between the predictors and the response
  - values in between? the model has value, perhaps, but doesn't tell the whole story about the data-generating process

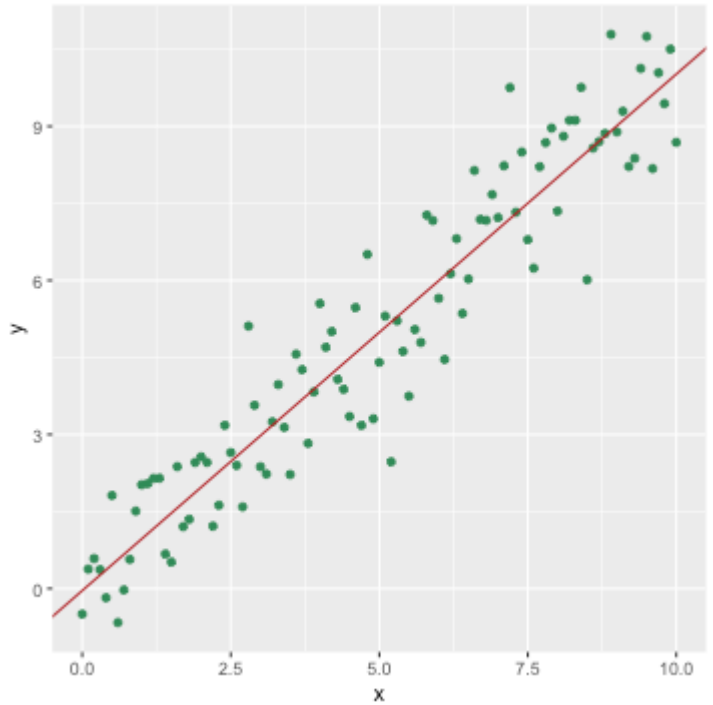
Linear Regression: Example

- A useful diagnostic (for *any* regression model, not just a linear regression model!) is to plot predicted responses (for the test-set data only) versus the observed responses
    - if there is no association between the predictor variable and the response variable, the points will stretch horizontally across the plot
    - if there is a deterministic association between the predictor variable and the response variable, the points will follow the red line
    - a "good" model follows the red line, but with random scatter!



Linear Regression: Example

- Here are the data with the best-fit model overlaid:

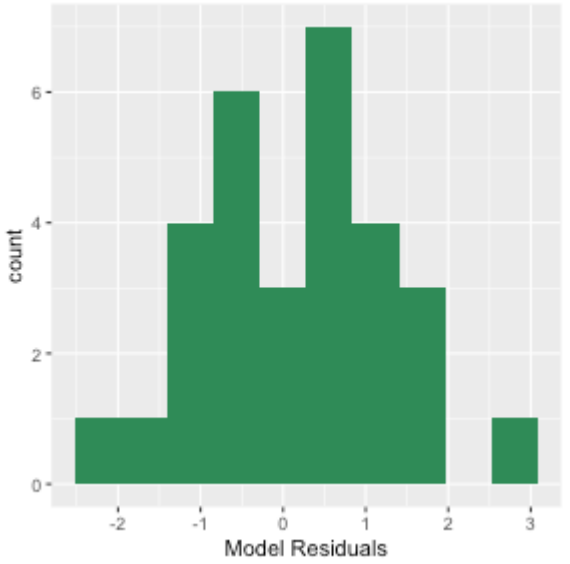


Linear Regression: Assumptions

- Let's go back to the "optional" assumption (3): ϵ is assumed to be normally distributed
  - to check this, plot a histogram of the fitted model residuals $e = Y - \hat{Y}$
- To test for normality, e.g., pass the values of the residuals into the Shapiro-Wilk test
  - the p-value is $0.983 > 0.05$
  - we fail to reject the null hypothesis that the data are normally distributed
- If we *reject* normality, it means that we "cannot take the numbers output by `R` at face value"
  - however, if assumptions (1) and (2) still hold, then the model is still perfectly valid...we just cannot "trust" the estimated uncertainties on the coefficients and the associated p-values (i.e., inference is affected, but not prediction)
  - it may make sense to attempt a transformation of the response values, e.g., $Y \rightarrow \log Y$

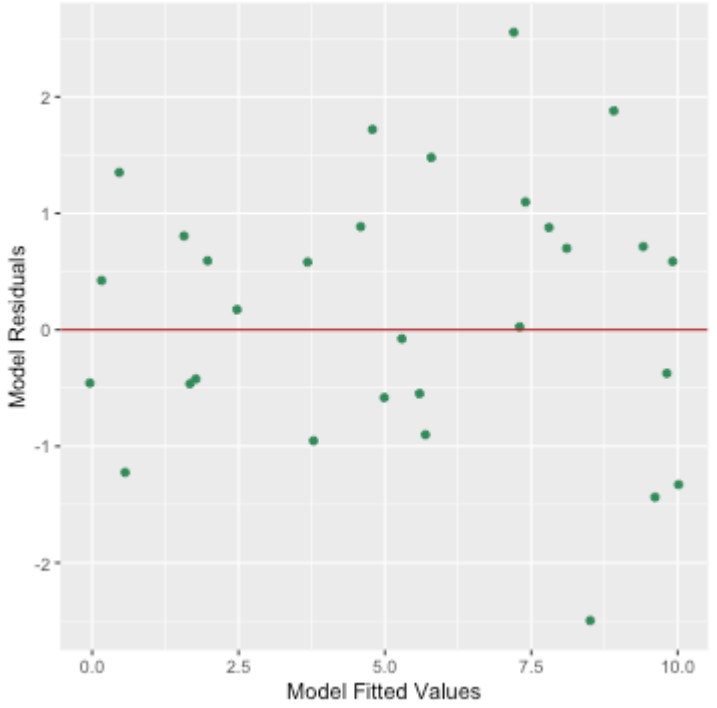
```
y.hat <- predict(lm.out,newdata=df.test)
e     <- df.test$y - y.hat
shapiro.test(e)$p.value
```

```
## [1] 0.9835445
```



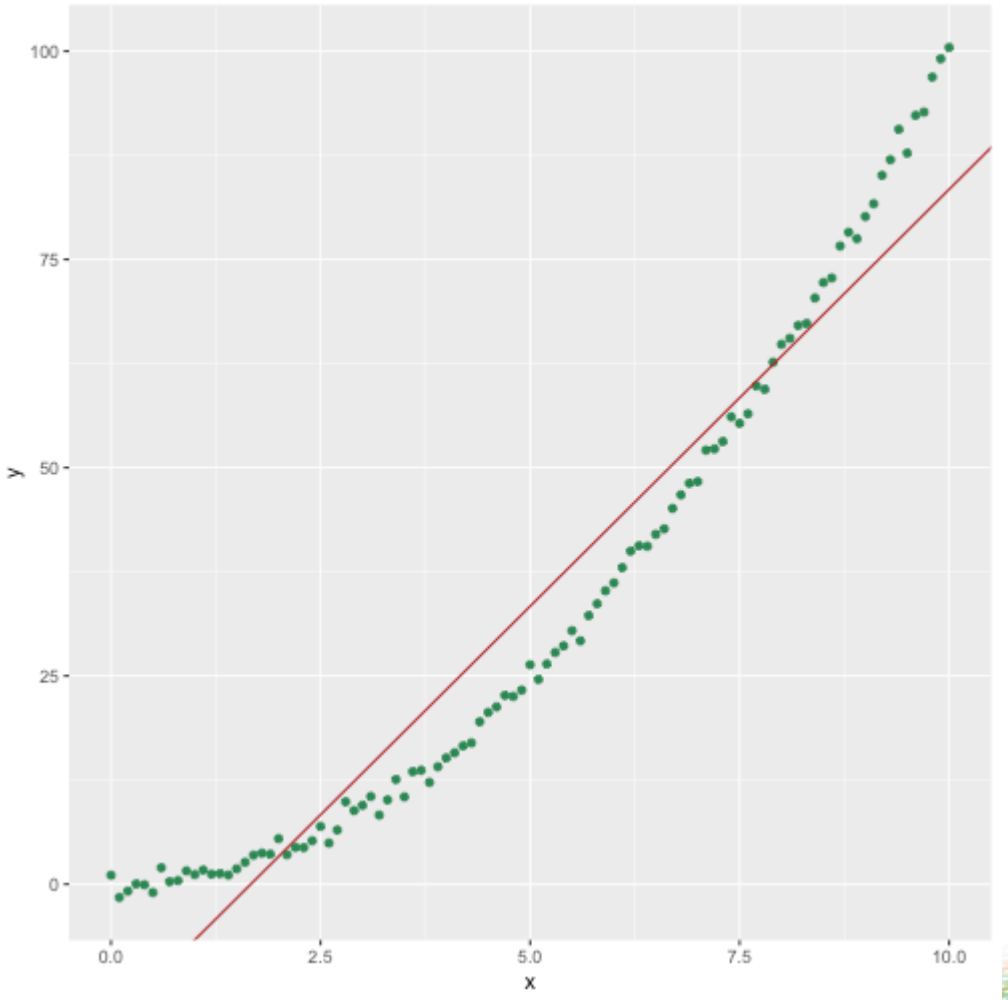
Linear Regression: Assumptions

- Now let's go back to assumption (2): ϵ has constant variance σ^2
    - to check this, plot the residuals e versus predictions $\hat{Y}$
- If the scatter around $e = 0$ appears constant as a function of $\hat{Y}$ appears constant, then assumption (2) holds
- If assumption (2) does not hold (but assumption (1) does), then the linear model is not the "best linear unbiased estimator" or BLUE model
    - it may make sense to attempt *weighted* linear regression, if we can estimate σ_i^2 for each datum
- Note again that if your goal is prediction, whether assumption (2) holds or not really doesn't matter...the model either makes competitive predictions, or it doesn't!



Linear Regression: Assumptions

- If assumption (1), that the mean of ϵ is zero, does not hold...
  - ...then linear regression is simply not a good representation of the data-generating process...full stop



Linear Regression: What if I Have Categorical Predictors?

```
summary(lm(y~.,data=df.ic))
```

```
##
## Call:
## lm(formula = y ~ ., data = df.ic)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69380 -0.72002 -0.00031  0.73428  2.53092
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02808    0.22111  -0.127    0.899
## x            1.03043    0.06891  14.953   <2e-16 ***
## icVanilla   -0.16815    0.40184  -0.418    0.677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.01 on 98 degrees of freedom
## Multiple R-squared:  0.8968,    Adjusted R-squared:  0.8947
## F-statistic: 425.8 on 2 and 98 DF,  p-value: < 2.2e-16
```

- Let's add a factor variable with favorite ice-cream flavor
- When a predictor variable is a factor variable with k levels, then $k-1$ so-called *dummy variables* are shown in the output
  - `ic` has levels `Chocolate` and `Vanilla`, and so `Chocolate` (the first level) becomes the "reference level" and `icVanilla` is what gets output
- Mathematically, the model is

$$\hat{Y}|\mathbf{x} = \beta_0 + \beta_1 x + \beta_2 \mathbb{I}_{vanilla}$$

  - $\mathbb{I}$ is an *indicator variable*: here, it takes on value 1 if the value of `ic` is `Vanilla` and 0 otherwise
  - for chocolate ice-cream eaters, the model is

$$\hat{Y}|\mathbf{x} = \beta_0 + \beta_1 x$$

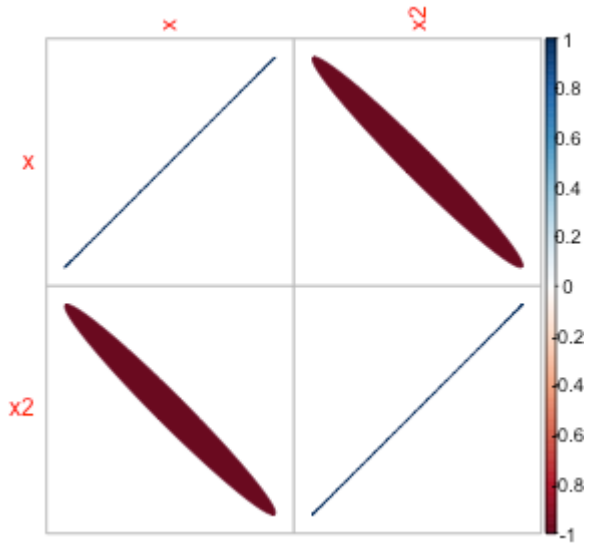
  - for vanilla ice-cream eaters, the model is

$$\hat{Y}|\mathbf{x} = \beta_0 + \beta_1 x + \beta_2$$

Linear Regression: Multicollinearity

- Let's now remove the factor variable and add a second predictor variable that is linearly related to the first one

```
suppressMessages(library(corrplot))
corrplot(cor(df.train[,1:2]),method="ellipse") # correlation between columns 1 and 2
```



- Visual inspection of this plot indicates clearly that multicollinearity is present
  - keep in mind that multicollinearity affects inference, but not prediction!

Linear Regression: Variance Inflation Factor

- The variance inflation factor, or vif, is the amount by which the estimated variance for a coefficient is inflated because of multicollinearity
- For instance, assume a modeled regression line is

$$\hat{Y} = 5 + 4x_1 - 2x_2$$

  - let the estimated standard deviations for $\hat{\beta}_1$ and $\hat{\beta}_2$ be 2, and let the vif's be 4 and 9, respectively
  - the actual estimate of the standard deviation for $\hat{\beta}_1$ should be $2 \times \sqrt{4} = 4$, and for $\hat{\beta}_2$, it should be $2 \times \sqrt{9} = 6$

- If a vif value is high, then the coefficient estimate by `lm()` for that variable can be much more uncertain than `R` indicates
  - the rule of thumb is to worry if vif values are above 5 (more conservative) or 10 (less conservative)

```
suppressMessages(library(car))
lm.out <- lm(y~.,data=df.train)
vif(lm.out)
```

```
##        x       x2
## 25.79774 25.79774
```

- A mitigation here is to remove one variable at a time until the vif values for those that remain fall below 5 or 10, but this can adversely affect the model's predictive abilities...a better (but not necessarily perfect) strategy is to pursue *principal components regression*