Agenda

2. Simple Linear Regression: Mathematical Form
3. Assessing Conditions
4. Making Predictions
5. Transformations
6. Outliers and Influential Points

The four-step process

1. Choose
2. Fit
3. Assess
4. Use

Why do we build statistical models?

- To understand the relationships between variables: Are two variables related? Do they tend to move in the same direction? Are they linearly related? Does knowing something about something mean anything about something else?
- To predict future outcomes: What will happen in the future?
- To quantify differences between groups or treatments: Does this drug work? How much better than a placebo? Does it affect men and women differently?

\[ Y = f(X) + \epsilon \]

- \( Y \): response variable
- \( f \): a function that makes up the model – there are infinitely many possible functions!
- \( X \): explanatory variables (i.e. the data)
- \( \epsilon \): error or noise term

How good is my model?

“\text{All statistical models} are wrong, but some are useful” –George Box

- Every model makes assumptions that aren’t true – understanding those assumptions is critical!
- Every model only estimates the expected value of the response variable
- Delicate balance between simplicity & accuracy
- There is no “correct” model and probably not a “best” one either!!
**Simple Linear Regression**  Simple linear regression is a special case of the model above, wherein:

- \( Y \) is a quantitative variable
- \( f \) makes a line!
- \( X \) is a single quantitative variable
- \( \epsilon \sim N(0, \sigma_\epsilon) \), where \( \sigma_\epsilon \) is fixed

Notation for this may vary. Note that:

- \( Y, X \): the (idea of the) quantities (e.g. Age, Price)
- \( y, x \): specific values of those quantities (e.g. 22, $35,748)
- \( y_i, x_i \): the \( i \)th specific value of a collection
- \( \bar{y}, \bar{x} \): average (mean) value of a collection of values
  \[
  \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i
  \]
- \( \hat{y} \): an estimate of an unknown value based on the model

**Activity: regression and you**  We are going to choose some variables we want to study about our class as a whole. This isn’t going to be a sampling activity (hopefully, we will capture data about the entire class) but we will use linear modeling as a descriptive method. First, we need to brainstorm some variables to collect. Last year, the class collected variables about students credits for the semester, number of siblings, number of pairs of shoes, number of facebook friends, and distance travelled to Smith after winter break. But, we don’t have to use those!

Go to this spreadsheet and enter your information.

```r
require(mosaic)
# install.packages("googlesheets")
require(googlesheets)
url = gs_url("https://docs.google.com/spreadsheets/d/1B6GHk8ekRY41Yjhfo0jIZtISe8cjUlYJu-TEYwIPQZmNDA/pub?gid=0&single=true&output=csv")
ds = gs_read_csv(url)

xyplot(numFBFriends ~ numShoes, data=ds, type = c("p", "r"))
mod = lm(numFBFriends ~ numShoes, data=ds)
summary(mod)
```

1. **Before you import the data**, consider these variables with your neighbor. Identify two bivariate relationships that you expect to be strongest, and two that you expect to be weakest.

2. Import the data and explore these relationships. Were you right? What criteria did you use to assess? [Hint: use \texttt{cor()} and \texttt{xyplot()}]
3. Use `lm()` to fit a linear regression model to one pair of variables and examine the output using `summary()`. What conclusions can you draw?

4. Assess the conditions of your model. Are they met?

**Model Visualization**  Compare the least squares regression line (right) with the average (left). We can think of the latter as a the null model where $\beta_1 = 0$ and $\hat{y} = \bar{y}$.

```r
panel.regplot = function (x, y, mod, ...) {
  panel.xyplot(x, y, type = c("p"), lwd=3, pch=19, ...)
  panel.arrows(x0=x, y0=y, x1=x, y1=fitted.values(mod), code=2, length=0.1, col="darkgray")
  panel.abline(mod)
}
mod.mean = lm(numFBFriends ~ 1, data=ds)
mod.line = lm(numFBFriends ~ numShoes, data=ds)
xyplot(numFBFriends ~ numShoes, data=ds, panel=panel.regplot, mod = mod.mean)
xyplot(numFBFriends ~ numShoes, data=ds, panel=panel.regplot, mod = mod.line)
```

Which model was better?

- Residuals: $y - \hat{y}$

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad \hat{\sigma} = \sqrt{\frac{SSE}{n-2}}$$

- Least Squares: technique for minimizing SSE
- Finds unique straight line between scatterplot of points

```r
sum(residuals(mod.mean)^2)
sum(residuals(mod.line)^2)
sqrt(sum(residuals(mod.line)^2) / (nrow(ds) - 2))
```

**Activity: Conditions for Regression**  Write, in your own words, definitions for each of the following:

- Linearity:

- Independence:
• Normality of Residuals:

• Equal Variance of Residuals:

Assessing Conditions  How loosely should we allow the conditions for regression to be violated? To get a feel for this, consider the following simulation. Note that, by construction, the data are generated from the model:

\[ Y = 10 + 3 \cdot X + \epsilon, \quad \epsilon \sim N(0,1) \]

Run this chunk several times, with several different choices of \( n \). What do you notice?

```r
n = 20
data <- data.frame(x = rnorm(n))
data = mutate(data, y = 10 + 3 * x + rnorm(n))
xyplot(y ~ x, data=data, type=c("p", "r"))
mod.ex <- lm(y ~ x, data=data)
plot(mod.ex)
```

Making Predictions  Use the function `predict` to use your model to make predictions.

```r
new.ds = data.frame(numShoes = c(3, 8))
predict(mod.line, newdata = new.ds)
```