Data Mining
W4240 Sections 001, 003/004

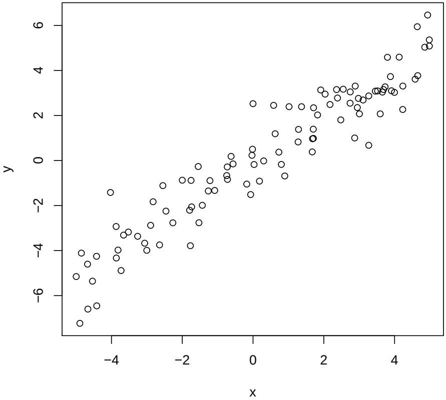
Lauren A. Hannah

Columbia University, Department of Statistics

September 30, 2014

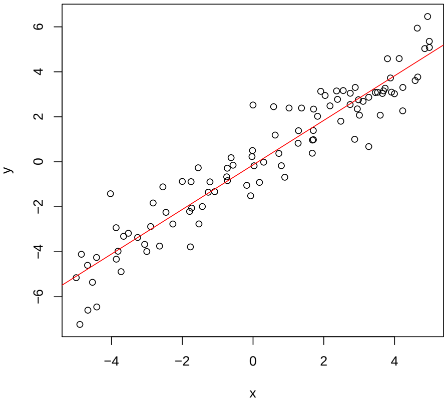
Outline

Linear Regression



Training data are the set of inputs and outputs, $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{n}$

Linear Regression



In *linear regression*, the goal is to predict y from x using a linear function

Categorical Covariates

Linearity assumption: Y increases (or decreases) at a constant rate as X increases in value

So what happens when X is categorical?

$$y = \beta_0 + \beta_1 \times (\text{red or blue})$$

does not make sense

Can look at *marginal increase for category*: how much more would blue give me than red?

Categorical Covariates

We can estimate the marginal increase by transforming X with a *dummy variable*

$$x_i = \begin{cases} 1 & \text{if } i^{th} \text{ item is blue} \\ 0 & \text{if } i^{th} \text{ item is red} \end{cases}$$

Then we get the linear relationship

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 x_i + \epsilon_i & \text{if } i^{th} \text{ item is blue} \\ \beta_0 + \epsilon_i & \text{if } i^{th} \text{ item is red} \end{cases}$$

Categorical Covariates

So what happens if X has more than two levels? For example, X could be red, green or blue.

Solution: pick one as a baseline and compare against that (here, red is the baseline)

$$x_{i1} = \begin{cases} 1 & \text{if } i^{th} \text{ item is blue} \\ 0 & \text{if } i^{th} \text{ item is not blue} \end{cases}$$

$$x_{i2} = \begin{cases} 1 & \text{if } i^{th} \text{ item is green} \\ 0 & \text{if } i^{th} \text{ item is not green} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 x_{i1} + \epsilon_i & \text{if } i^{th} \text{ item is blue} \\ \beta_0 + \beta_2 x_{i2} + \epsilon_i & \text{if } i^{th} \text{ item is green} \\ \beta_0 + \epsilon_i & \text{if } i^{th} \text{ item is red} \end{cases}$$

Covariate Interactions

Simple linear regression assumes that the interactions are *additive*:

$$f(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

Adding one additional unit of x_1 does not change the value of one additional unit of x_2

The simplest way to ease the additivity assumption is to include an *interaction term*:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

Outline

Linear Regression

Linear regression also assumes that the relationships are *linear*:

$$f(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

However, our model says that the relationship is linear only in the covariates. What if we used different covariates?

- Covariates: x, x^2, x^3, x^4

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4$$

- Covariates: $x_1, x_2, x_1 x_2$

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

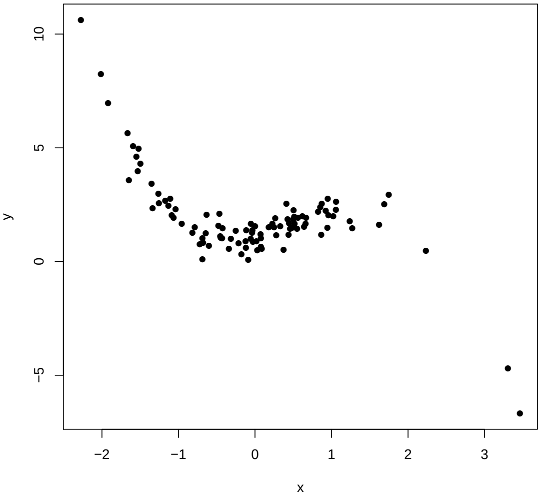
- Covariates: $\log(x_1), \log(x_2)$

$$f(x) = \beta_0 + \beta_1 \log(x_1) + \beta_2 \log(x_2)$$

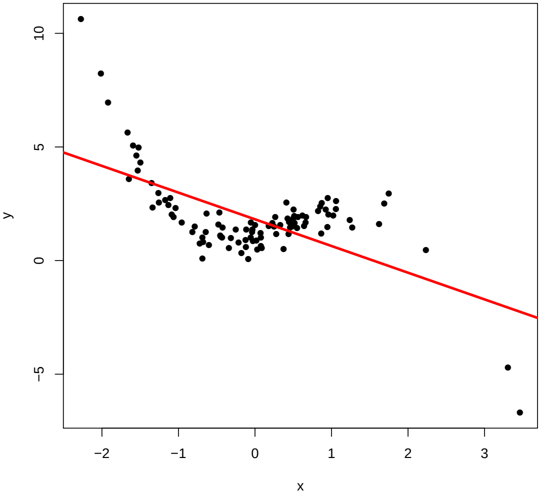
- Covariates: $x, \mathbf{1}_{\{-1 \leq x \leq 1\}}$

$$f(x) = \beta_0 + \beta_1 x + \beta_2 \mathbf{1}_{\{-1 \leq x \leq 1\}}$$

Polynomial Regression

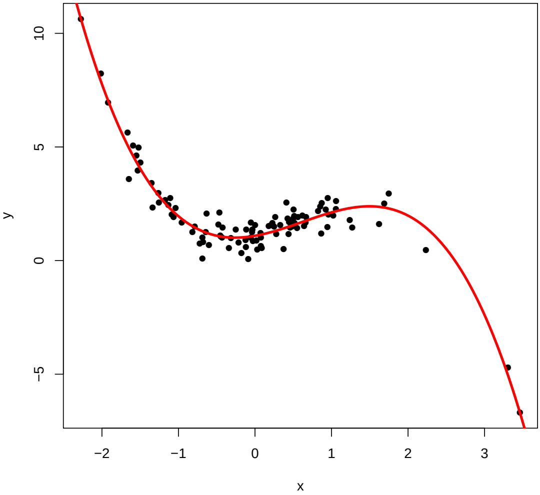


Polynomial Regression



$$f(x) = \beta_0 + \beta_1 x$$

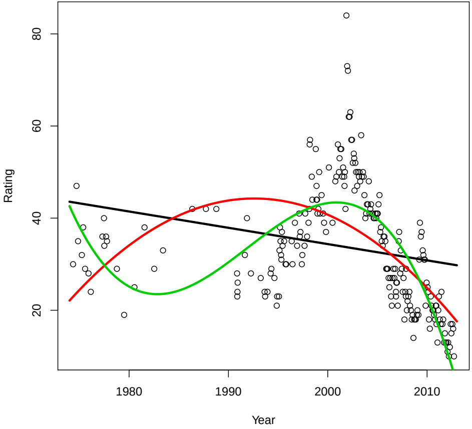
Polynomial Regression



$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

Polynomial Regression

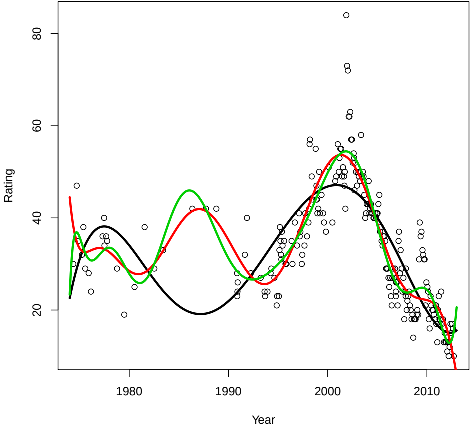
In general, as we let the powers get higher we can fit (almost) *any* function with $x, x^2, x^3, x^4, x^5, x^6, \ldots$.



Here, we have a linear, quadratic and third degree fit.

Polynomial Regression

In general, as we let the powers get higher we can fit (almost) *any* function with $x, x^2, x^3, x^4, x^5, x^6, \ldots$.



Here, we have a 5^{th}, 10^{th} and 15^{th} degree fit.

Polynomial Regression

Let's see how that looks in R.

```
> congress <- read.csv("Congress.csv")
> attach(congress)
> fit1 <- lm(Rating ~ Year)
> fit2 <- lm(Rating ~ poly(Year,2,raw=T))
> fit3 <- lm(Rating ~ poly(Year,3,raw=T))
> data.test <- seq(1974,2013,0.1)
> df.test <- data.frame(Year = data.test)
> y.1 <- predict(fit1,newdata=df.test)
> y.2 <- predict(fit2,newdata=df.test)
> y.3 <- predict(fit3,newdata=df.test)
> plot(Year,Rating)
> lines(data.test,y.1,col=1,lwd=3)
> lines(data.test,y.2,col=2,lwd=3)
> lines(data.test,y.3,col=3,lwd=3)
```

Outline

Potential Problems

Linear regression has a lot of assumptions. They are:

- $f(X)$ is linear
- errors are iid Gaussian

So, we need to check that:

- residuals have a Gaussian distribution
- residuals are uncorrelated
- residuals have constant variance

(Note: if the assumptions are not met, you still have a valid predictive model. You just can't use it for things like confidence intervals.)

Potential Problems

Linear regression has a lot of assumptions. What if they are not met? What else could go wrong?

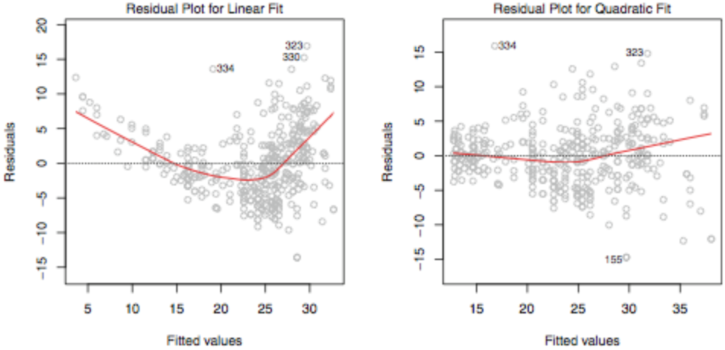
- non-linear covariate-response relationship
- correlation of errors
- non-constant variance of errors
- outliers
- high-leverage points
- collinearity

Outline

Nonlinear Relationships

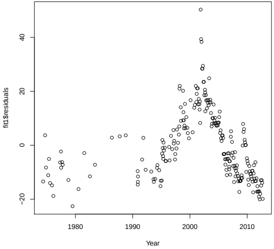
Linear models assume linear relationships between the covariates and response. How do I know if there is a nonlinear relationship?

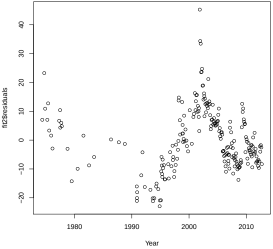
Simplest way: look at residuals.

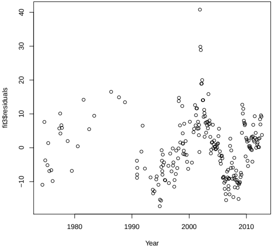


Nonlinear Relationships

Transforming your covariates can lead to more Gaussian-looking residuals.







Or not.

Nonlinear Relationships

Usually the easiest way to check for Gaussian errors is with a **Q-Q plot**. (This stands for quantile-quantile plot.) It compares:

$$\arg\min_z \quad P(Z \le z) \ge \alpha$$
$$vs.$$
$$\arg\min_z \quad P(R \le z) \ge \alpha$$

where Z has a standard normal distribution, R is the empirical distribution of your residuals and α is a number between 0 and 1.

Let's understand why this is a sensible analysis. What should result?

It should be a straight line, with slope and intercept depending on residual mean and variance.

Non-linear Relationships

In R, we can use the functions qqnorm and qqline.

```
> x.norm <- rnorm(500)
> qqnorm(x.norm)
> qqline(x.norm)
> qqnorm(fit1$residuals)
> qqline(fit1$residuals)
```

What does it mean if the upper tail is above the line? Below the line?

Non-linear Relationships

Residuals in an multivariate setting are harder to analyze.

- There are tests to see if they have a multivariate Gaussian distribution.
- They cannot tell you where your model is systemically under-predicting or over-predicting.

Correlation of Error Terms

What are uncorrelated errors?[1]

- ϵ_i provides no information about ϵ_{i+1}

Why do we want uncorrelated errors?

- correlated errors lead to underestimation of standard error
- this means confidence intervals are too small

When do we have correlated errors?

- time series data
- when there are hidden factors

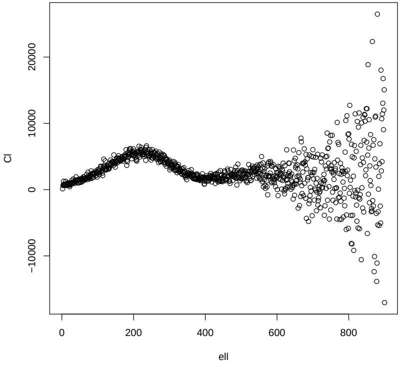
[1] Gaussian assumption means uncorrelated errors are independent as well... if the assumption holds.

Outline

Non-constant Variance of Errors

Linear models assume constant variance of error terms, $\mathrm{Var}(\epsilon_i) = \sigma^2$. This is called **homoscedasticity**.

Often, the variance changes with the covariates, $\mathrm{Var}(\epsilon_i) = \sigma^2(x_i)$. This is called **heteroscedasticity**.



Non-constant Variance of Errors

Why will this cause a problem?

Rewrite linear regression problem. Assume Gaussian errors around linear function:

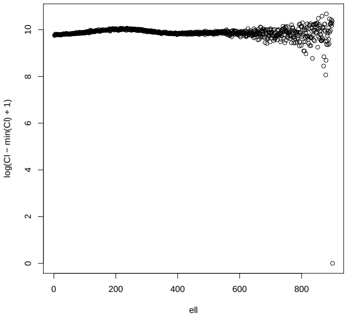
$$\hat{\beta} = \arg\min_{\beta} \prod_{i=1}^{n} p\left(y_i \,|\, \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2\right)$$
$$= \arg\min_{\beta} \sum_{i=1}^{n} -\log(\sigma^2) - \frac{1}{2\sigma^2}\left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_j\right)^2$$

Only equivalent to least squares when $\sigma_1^2 = \cdots = \sigma_n^2$. Otherwise data have different weights.

Non-constant Variance of Errors

What are some good ways to deal with heteroscedasticity?

- do regression on the $\log$ of your responses

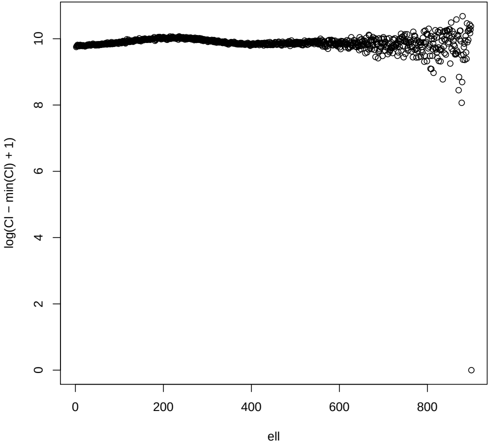


- re-weight if you know the variances (give data equal weight)
- fit a better model

Outline

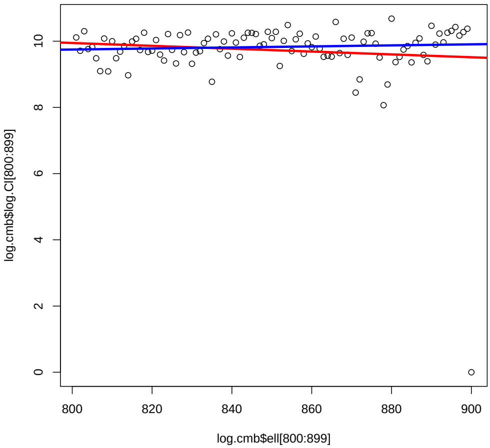
Outliers

Outliers are values far from predicted model. They may be real or bad data.



Outliers

Since we are minimizing *squared error*, ordinary least squares tries to fit outliers.



Outliers

How to cope with outliers:

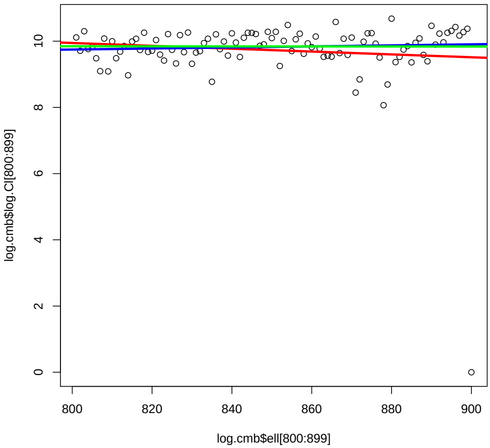
- if it is clearly bad data (e.g. a woman had 50 children), remove it
- fit another model, like **least absolute deviation linear regression**

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \left| y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right|$$

You can use the `rq()` function in the `quantreg` package just like `lm()` to do least absolute deviation regression.

Outliers

Let's compare the least absolute deviation fit to the least squares fits.



High Leverage Points

Outliers: unusual y_i values for a given x_i.

High Leverage Points: unusual x_i values.

Why are these a problem?

- they influence $\hat{\beta}$ much more than other points
- can compare *leverage statistics*:

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{k=1}^{n}(x_k - \bar{x})^2}$$

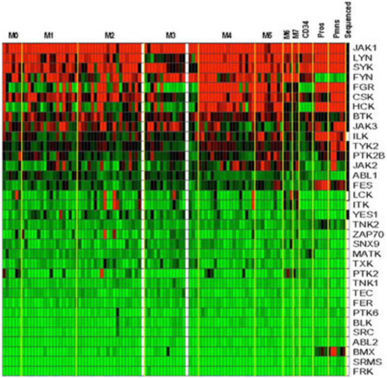
- average is $(p+1)/n$, so if any values are much greater, that point has high leverage

Outline

Collinearity

Collinearity is when two or more covariates have correlations close to 1 or -1. This happens in a number of settings:

- X_1 is grade on test 1, X_2 is grade on test 2
- $X_1, \ldots, X_p$ are gene expression values



Collinearity

Why is collinearity a problem?

- if you know X_1, then X_2 has little predictive value
- if truly collinear, $\hat{\beta}_1$ and $\hat{\beta}_2$ are not uniquely defined (and $(\mathbf{X}^T\mathbf{X})^{-1}$ doesn't exist)
- if close to collinear, small changes in data can lead to large changes in $\hat{\beta}_1$ and $\hat{\beta}_2$

How do I find it?

- look at the correlations between the covariates (close to 1 or -1 means high collinearity)
- look at the eigenvalues of $\mathbf{X}^T\mathbf{X}$ (close to 0 means high collinearity)

How do I fix it?

- select a subset of predictors
- put some constraints on $\hat{\beta}$ (regularization)

Example: Prostate Data

Data in Prostate.txt (also available on ESL website)

Predictors (columns 1–8): lcavol (log cancer volume), lweight (log weight), age, lbph (log amount of benign prostatic hyperplasia), svi (seminal vesicle inversion), lcp (log capsular penetration), gleason, pgg45 (percentage of Gleason scores 4 or 5)

outcome (column 9): lpsa (level of prostate-specific antigen)

train/test indicator (column 10)

```
> prostate <- read.table("Prostate.txt",header=TRUE, sep="\t")
> names(prostate)
 [1] "X"        "lcavol"   "lweight" "age"
 [5] "lbph"     "svi"      "lcp"     "gleason"
 [9] "pgg45"    "lpsa"     "train"
> prostate.train <- prostate[prostate$train==T,2:10]
> prostate.test <- prostate[prostate$train==F,2:10]
```

Example: Prostate Data

```
> prostate.lm <- lm(lpsa ~ lcavol + lweight + age + lbph
   + svi + lcp + gleason + pgg45, data=prostate.train)
> # Other way:
> # prostate.lm <- lm(lpsa ~., data=prostate.train)
> # Exclude intercept by:
> # prostate.lm <- lm(lpsa ~ lcavol + lweight + age + lbph
   + svi + lcp + gleason + pgg45 - 1, data=prostate.train)
> y.pred.lm <- predict(prostate.lm,prostate.test)
> mean((y.pred.lm-prostate.test$lpsa)^2)
[1] 0.521274
```

Note: the data in the book was scaled before use, so $\hat{\beta}$ differs