Linear Regression Models
P8111

Lecture 18

Jeff Goldsmith
March 31, 2015

THE DEPARTMENT OF BIOSTATISTICS
Columbia University
MAILMAN SCHOOL OF PUBLIC HEALTH

Today's Lecture

- Additive models
- Case study

Recall the goals of regression

- Estimation of $E(y|x) = f(x)$
- Prediction of future observations y given predictors x

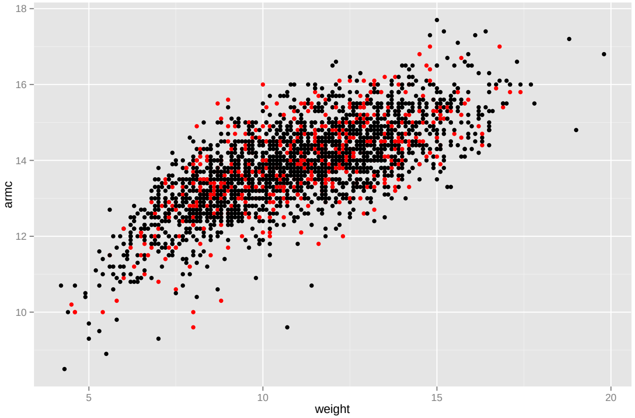
Some methods we've seen

- Simple linear regression
- Polynomial regression
- Spline models
- Penalized spline regression
- Non-parametric models

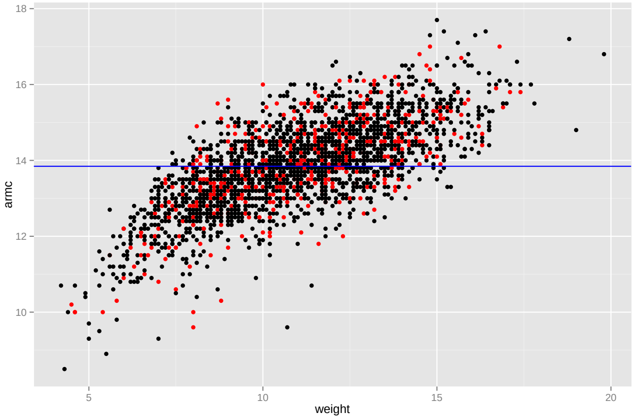
Example

- Arm circumference vs. weight
- How should we / can we estimate this?
    - Any of the above methods is possible
    - Which is "best" is a combination of inference, prediction, and model goals

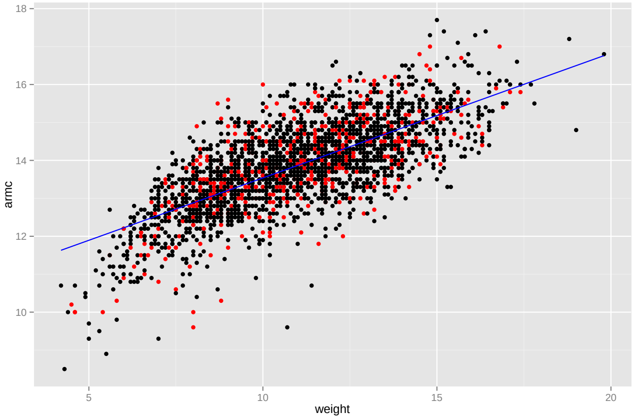
Example



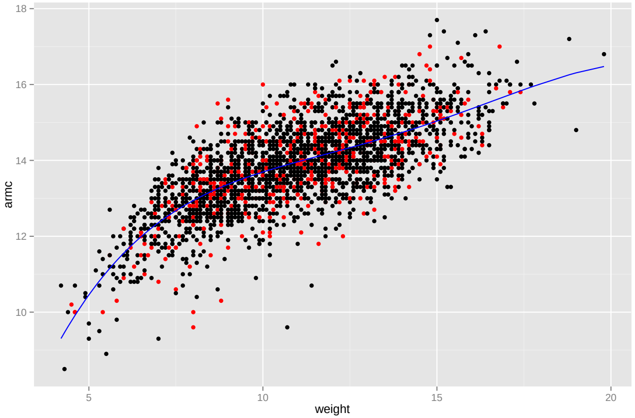
Example



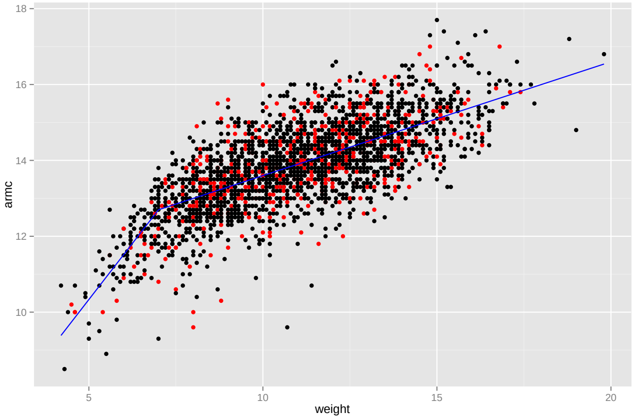
Example



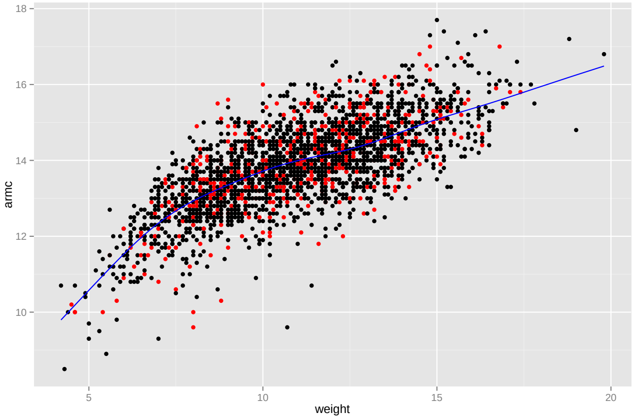
Example



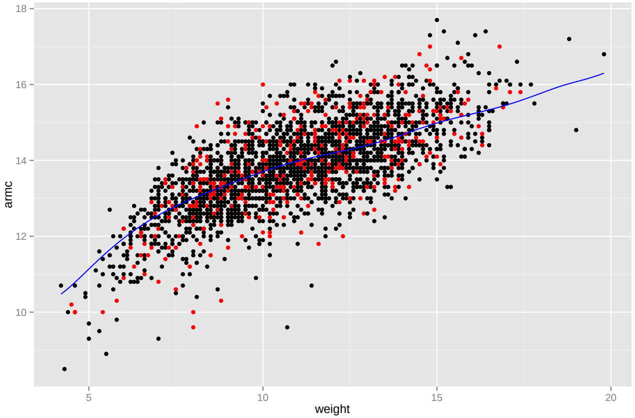
Example



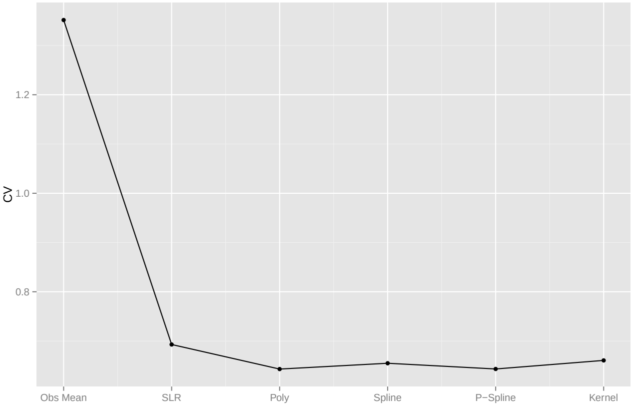
Example



Example



Example



Example

So which one is best?

Additive models

- For scatterplot smoothing, we've focused on a single predictor
- Most real examples have multiple predictors
- Non-linearity can arise in any variable, or in multiple variables
- Previously we have addressed this using polynomials in MLRs

Additive models

Additive models are a very general framework for addressing non-linearity

- Given p predictors, the additive model is

$$\mathrm{E}y|x_1,\ldots,x_p = f(x_1,\ldots,x_p) = \beta_0 + \sum_{k=1}^{p} f(x_k) \tag{1}$$

- Each $f(\cdot)$ is a smooth function (can be a line)

Additive models

Additive models are a very general framework for addressing non-linearity

- In theory, each smooth function can be estimated in a variety of ways
  - Polynomials, splines, penalized splines, kernel smoothers, etc
- In practice, penalized splines is a pretty unified framework for fitting additive models
- Quick note – the intercept is not identifiable ...

How estimation might go ...

Backfitting

Backfitting is a more algorithmic method for estimating model parameters

- Start out by setting $f(x_k) = 0$ for all k
- Initialize $\hat{\beta}_0 = \bar{y}$
- Iterate the follow steps until convergence:
    - For each $f(x_k)$ in turn, estimate

$$f(x_k) = \text{smooth}(y - \hat{\beta}_0 - \sum_{k' \neq k} f(x_{k'})) \tag{2}$$

    - Center each $f(x_k)$

Additive models vs MLR

- Additive models generalize the idea of including polynomial terms in an MLR
- Of course, there are tradeoffs ...
  - On the plus side:
  - On the minus side:

Additive models example

Continue with Nepalese children

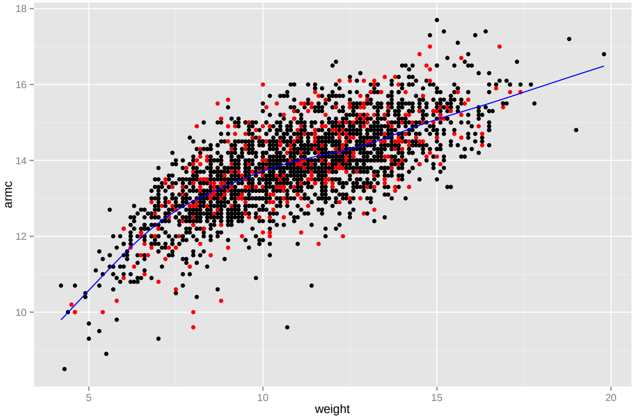
- Looked at arm circumference vs weight
- Other variables include sex, age, height
- How can we include these other variables?

Some code notes

Fitting additive models in R:

```
library(mgcv)
fx = gam(armc ~ s(weight), data = data.train)
> summary(fx)

Family: gaussian
Link function: identity

Formula:
armc ~ s(weight)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.82341    0.01472   939.3   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
            edf Ref.df     F p-value
s(weight) 5.437  6.575 501.3  <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.553   Deviance explained = 55.4%
GCV score = 0.57969  Scale est. = 0.57829   n = 2670
```

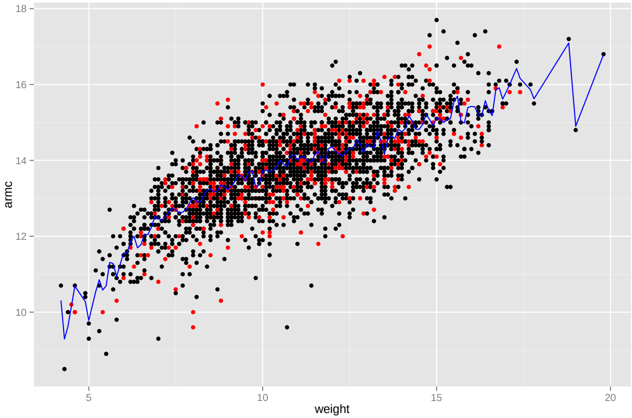
Plot



Some code notes

```
fx = gam(armc ~ s(weight, k = 100),
        data = data.train, sp = (.0001))
```

Plot

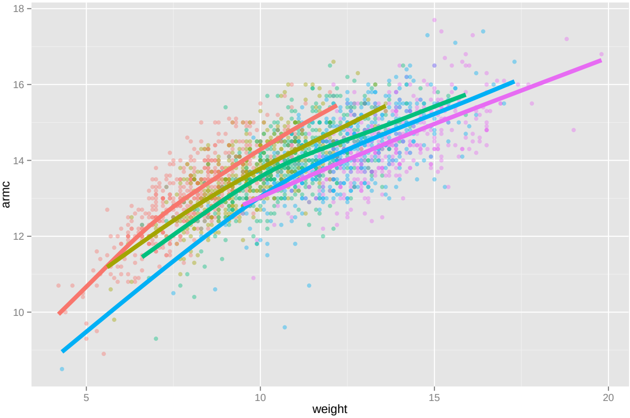


Separate boys and girls

Separate boys and girls

```
> fx = gam(armc ~ sex + sex * weight + s(weight), data = data.train)
> summary(fx)

Family: gaussian
Link function: identity

Formula:
armc ~ sex + sex * weight + s(weight)

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.497347   0.048504  10.254  < 2e-16 ***
sex         -0.374234   0.138216  -2.708  0.00682 **
weight       1.218742   0.005863 207.857  < 2e-16 ***
sex:weight   0.035782   0.012365   2.894  0.00384 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
            edf Ref.df     F p-value
s(weight) 5.297  6.434 441.4  <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.554   Deviance explained = 55.6%
GCV score = 0.57884  Scale est. = 0.57703   n = 2670
```

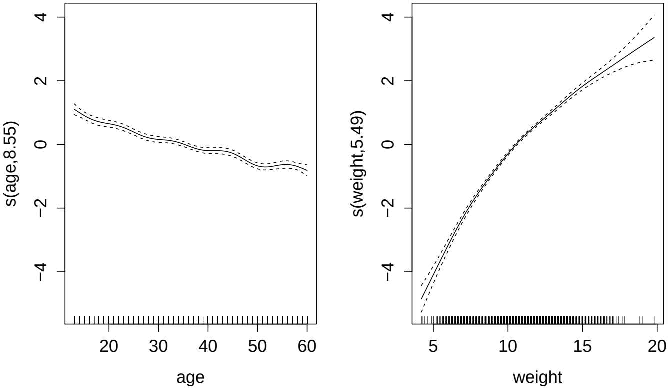
Separate by age



Separate by age

```
> fx = gam(armc ~ s(age) + s(weight), data = data.train)
> summary(fx)

Family: gaussian
Link function: identity

Formula:
armc ~ s(age) + s(weight)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.82341    0.01352    1022   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
            edf Ref.df      F p-value
s(age)    7.369  8.352  60.34  <2e-16 ***
s(weight) 4.916  6.054 487.88  <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.623   Deviance explained = 62.5%
GCV score = 0.49054  Scale est. = 0.4881    n = 2670
```

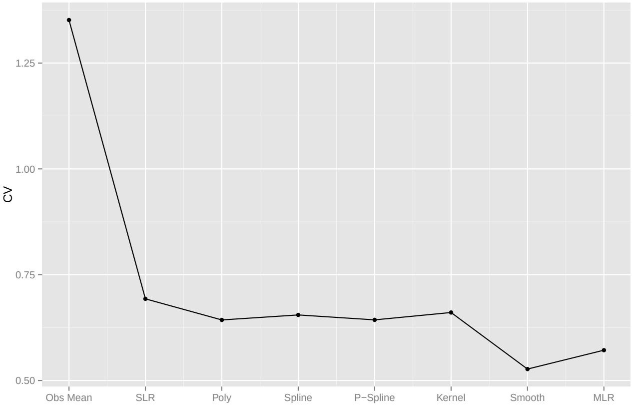
Separate by age



For comparison

```
> fx = lm(armc ~ age + weight, data = data.train)
> summary(fx)

Call:
lm(formula = armc ~ age + weight, data = data.train)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6662 -0.4746 -0.0039  0.4837  2.5447

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.711720   0.068146  142.51   <2e-16 ***
age         -0.037488   0.001770  -21.18   <2e-16 ***
weight       0.500365   0.009852   50.79   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.7308 on 2667 degrees of freedom
Multiple R-squared: 0.5879, Adjusted R-squared: 0.5876
F-statistic:  1903 on 2 and 2667 DF,  p-value: < 2.2e-16
```

Final CV comparison



Today's big ideas

- Additive models
- Case study