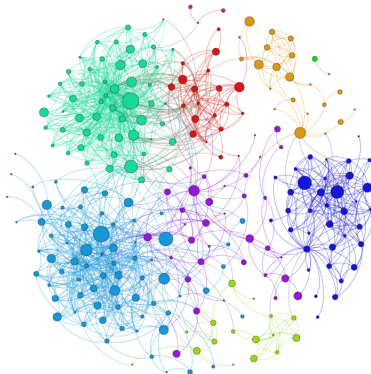


Regression

Jiaming Mao

Xiamen University



Copyright © 2017–2019, by Jiaming Mao

This version: Fall 2019

Contact: jmao@xmu.edu.cn

Course homepage: jiamingmao.github.io/data-analysis



All materials are licensed under the **Creative Commons Attribution-NonCommercial 4.0 International License**.

Linear Regression

The linear regression model¹ is a discriminative model with $f(x) = E[y|x]$ ² as the target function and $\mathcal{H} = \{h(x)\}$ consisting of linear functions³:

$$h(x) = x'\beta$$

, where $x = (1, x_1, \dots, x_p)'$ and $\beta = (\beta_0, \beta_1, \dots, \beta_p)'$.

The goal is to find $g \in \mathcal{H}$ that best approximates f .

¹Note on terminology: linear regression can refer broadly to the use of any linear models for regression purposes. Historically, however, it often refers more narrowly to least squares linear regression.

²The **conditional expectation function (CEF)**, $E[y|x]$, is also known as the **regression function**.

³Since each $h(x)$ is associated with a unique β , $h(x)$ is said to be **parametrized** by β . In this case, choosing a hypothesis h is equivalent to choosing a parameter β .

Linear Regression

- Error measures:

$$E_{out}(h) = E \left[(y - h(x))^2 \right] \quad (1)$$

$$E_{in}(h) = \frac{1}{N} \sum_{i=1}^N (y_i - h(x_i))^2 \quad (2)$$

- The VC dimension of a linear model is $p + 1$ ⁴. For $N \gg p$, the linear model generalizes well from E_{in} to E_{out} .

⁴ p is the dimension of the input space.

Linear Regression

Let

$$\begin{aligned}\beta^* &= \arg \min_{\beta} E \left[(y - x' \beta)^2 \right] \\ &= \underbrace{E (xx')^{-1}}_{(p+1) \times (p+1)} \underbrace{E (xy)}_{(p+1) \times 1}\end{aligned}\tag{3}$$

β^* is the *population* regression coefficient.

$x' \beta^*$ is the best⁵ linear predictor of y given x in the underlying population.

⁵in the sense of minimizing the L2 loss function.

Linear Regression

Recall that the CEF $f(x) = E[y|x]$ is the best⁵ predictor of y given x in the class of **all** functions of x .

The function $x'\beta^*$ provides the best⁵ **linear** approximation to the CEF⁶:

$$\beta^* = \arg \min_{\beta} E \left[(E[y|x] - x'\beta)^2 \right]$$

⁶Generally,

$$\begin{aligned} \arg \min_h E \left[(y - h(x))^2 \right] &= \arg \min_h E \left[(y - E[y|x] + E[y|x] - h(x))^2 \right] \\ &= \arg \min_h E \left[(y - E[y|x])^2 + (E[y|x] - h(x))^2 \right. \\ &\quad \left. + 2(y - E[y|x])(E[y|x] - h(x)) \right] \\ &= \arg \min_h E \left[(E[y|x] - h(x))^2 \right] \end{aligned}$$

Linear Regression

Let $e^* \equiv y - x'\beta^*$. By construction,

$$\underbrace{E(xe^*)}_{(p+1) \times 1} = 0 \quad (4)$$

In particular, if x contains a constant term, then (4) $\Rightarrow E(e^*) = 0$. In this case e^* and x are uncorrelated.

Linear Regression

We can separate the constant term and write the linear model as

$$y = \beta_0 + \tilde{x}'\tilde{\beta} + e$$

, where $\tilde{x} = (x_1, \dots, x_p)'$ and $\tilde{\beta} = (\beta_1, \dots, \beta_p)'$.

Then (3) \Rightarrow

$$\begin{aligned}\tilde{\beta}^* &= \text{Var}(\tilde{x})^{-1} \text{Cov}(\tilde{x}, y) \\ \beta_0^* &= E(y) - E(\tilde{x})' \tilde{\beta}^*\end{aligned}\tag{5}$$

Linear Regression

When $p = 1$,

$$y = \beta_0 + \beta_1 x_1 + e$$

(5) \Rightarrow

$$\beta_1^* = \frac{\text{Cov}(x_1, y)}{\text{Var}(x_1)} \quad (6)$$

When $p > 1$, (5) \Rightarrow for any $j \in \{1, \dots, p\}$,

$$\beta_j^* = \frac{\text{Cov}(u_j^*, y)}{\text{Var}(u_j^*)} \quad (7)$$

, where u_j^* is the residual from a regression of x_j on all the other inputs.

Linear Regression

- $\beta^* = E(xx')^{-1} E(xy)$ is the $(p+1) \times 1$ vector with the j^{th} ($j > 1$) element being $\beta_j^* = \frac{Cov(u_j^*, y)}{Var(u_j^*)}$.
- Each β_j^* is the slope coefficient on a scatter plot with y on the y -axis and u_j^* on the x -axis.

The OLS Estimator

Given observed data $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\} \sim^{i.i.d.} p(x, y)$, we have, for $i = 1, \dots, N$,

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + e_i$$

, which can be written as

$$Y = X\beta + e$$

, where $Y = [y_1, \dots, y_N]'$, $e = [e_1, \dots, e_N]'$, and

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Np} \end{bmatrix} = \begin{bmatrix} x_1' \\ \vdots \\ x_N' \end{bmatrix}$$

, where $x_i = [1, x_{i1}, \dots, x_{ip}]'$.

The OLS Estimator

Minimizing the in-sample error (2) \Rightarrow

$$\begin{aligned}\hat{\beta} &= \left[\sum_{i=1}^N x_i x_i' \right]^{-1} \sum_{i=1}^N x_i y_i \\ &= (X'X)^{-1} X'Y\end{aligned}\tag{8}$$

$\hat{\beta}$ is the *least squares* regression coefficient – the sample estimate of β^* .

The OLS Estimator

When $p = 1$,

$$\begin{aligned}\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}_{i1} \\ \hat{\beta}_1 &= \frac{\sum_{i=1}^N (x_{i1} - \bar{x}_{i1}) y_i}{\sum_{i=1}^N (x_{i1} - \bar{x}_{i1})^2}\end{aligned}$$

, where $\bar{x}_{i1} = \frac{1}{N} \sum_{i=1}^N x_{i1}$.

When $p > 1$, for any $j \in \{1, \dots, p\}$,

$$\hat{\beta}_j = \frac{\sum_{i=1}^N \hat{u}_{ij} y_i}{\sum_{i=1}^N \hat{u}_{ij}^2} = \frac{\hat{u}'_j Y}{\hat{u}'_j \hat{u}_j} \quad (9)$$

, where $\hat{u}_j = (\hat{u}_{1j}, \dots, \hat{u}_{Nj})'$, and \hat{u}_{ij} is the estimated residual from a regression of x_{ij} on $(1, \{x_{ik}\}_{k \neq j})$.

The OLS Estimator

Generate some data:

$$x_1 \sim U(0, 1)$$

$$x_2 = 0.5x_1 + 0.5r, \quad r \sim U(0, 1)$$

$$y = 1 - 2.5x_1 + 5x_2 + e, \quad e \sim \mathcal{N}(0, 1)$$

```
n <- 500
e <- rnorm(n)
x1 <- runif(n)
x2 <- 0.5*x1 + 0.5*runif(n)
y <- 1 - 2.5*x1 + 5*x2 + e
```

The OLS Estimator

```
require(AER)
reg <- lm(y ~ x1 + x2)
coeftest(reg)

##
## t test of coefficients:
##
##              Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)  1.01013    0.11884   8.4997 2.233e-16 ***
## x1          -2.59166    0.22529 -11.5039 < 2.2e-16 ***
## x2           5.06250    0.31213  16.2193 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The OLS Estimator

```
X <- cbind(rep(1,n),x1,x2)
beta <- solve(t(X)%*%X)%*%t(X)%*%y
t(beta)

##                x1        x2
## [1,] 1.010133 -2.591657 5.062497

x1reg <- lm(x1~x2)
u1 <- residuals(x1reg)
b1 <- cov(u1,y)/var(u1)

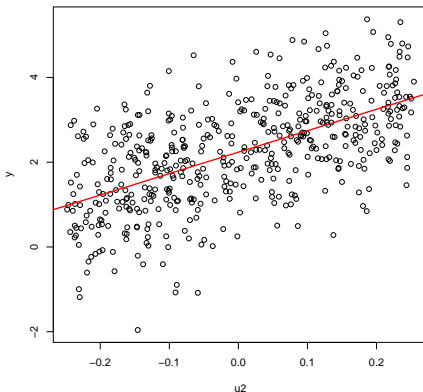
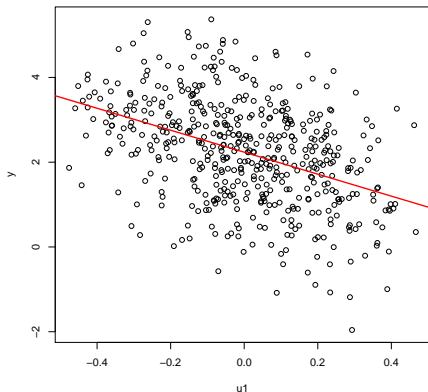
x2reg <- lm(x2~x1)
u2 <- residuals(x2reg)
b2 <- cov(u2,y)/var(u2)

b0 <- mean(y) - b1*mean(x1) - b2*mean(x2)
cbind(b0,b1,b2)

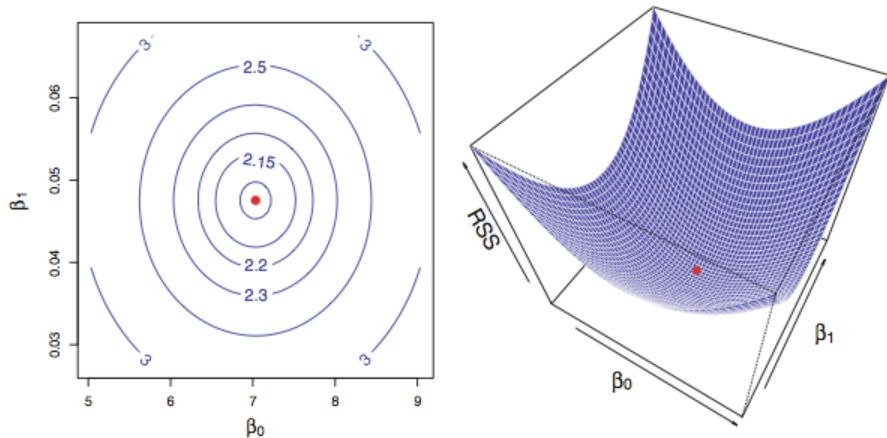
##                b0        b1        b2
## [1,] 1.010133 -2.591657 5.062497
```


The OLS Estimator

```
plot(u1,y)
abline(lm(y~u1),col="red",lwd=2)
plot(u2,y)
abline(lm(y~u2),col="red",lwd=2)
```

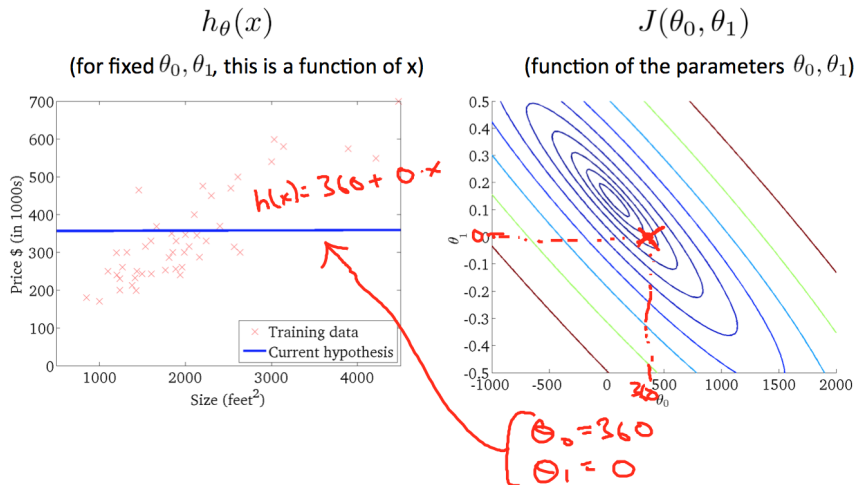


Searching for the best hypothesis



Contour and three-dimensional plots of $RSS = \sum_{i=1}^N (y_i - x_i' \beta)^2$

Searching for the best hypothesis



$$\mathcal{H} = \{h_{\theta}(x) = \theta_0 + \theta_1 x\}$$

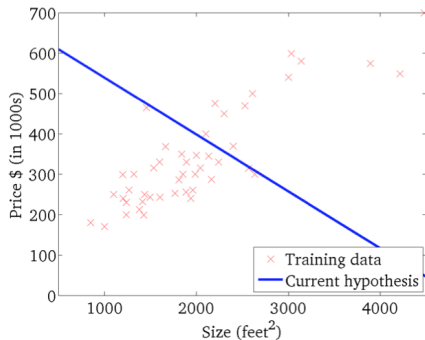
Left: training data and $h_{\theta}(x)$ for a particular $\theta = (\theta_0, \theta_1)$

Right: RSS: $J(\theta_0, \theta_1) = \sum_i (y_i - \theta_0 - \theta_1 x_i)^2$

Searching for the best hypothesis

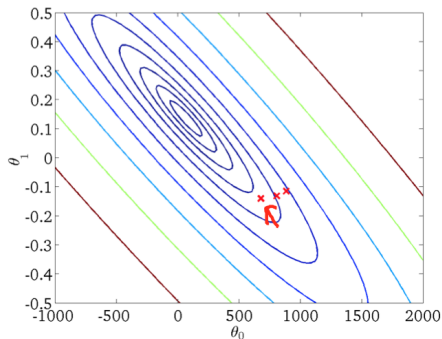
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

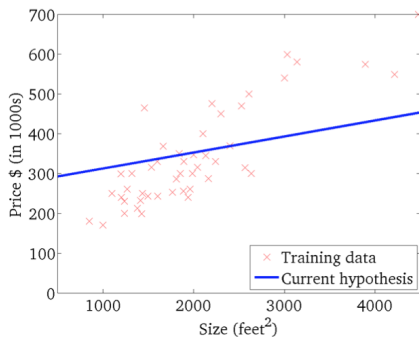
(function of the parameters θ_0, θ_1)



Searching for the best hypothesis

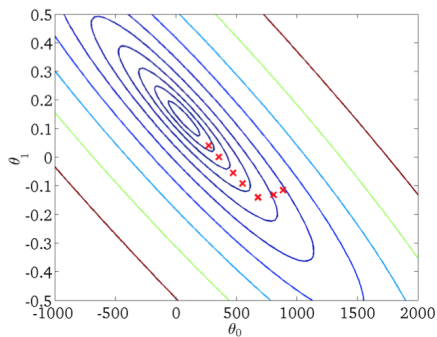
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

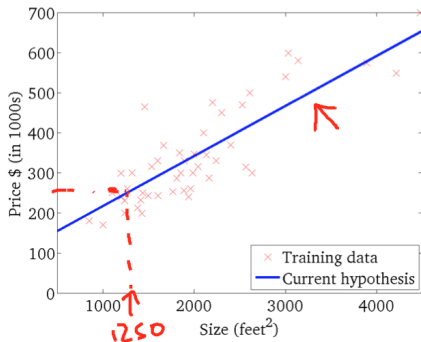
(function of the parameters θ_0, θ_1)



Searching for the best hypothesis

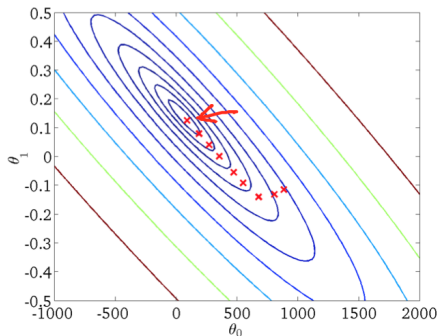
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



Geometric Interpretation

Consider two n -dimensional vectors: $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$. The **Euclidean distance** between a and b is:

$$\|a - b\| = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} = \sqrt{(a - b) \cdot (a - b)}$$

The cosine of the angle between a and b is:

$$\cos \theta = \frac{a \cdot b}{\|a\| \|b\|}$$

, where $\|a\| = \|a - 0\|$ is the length of a .

When $a \cdot b = 0$, a and b are **orthogonal**, denoted by $a \perp b$.

Geometric Interpretation

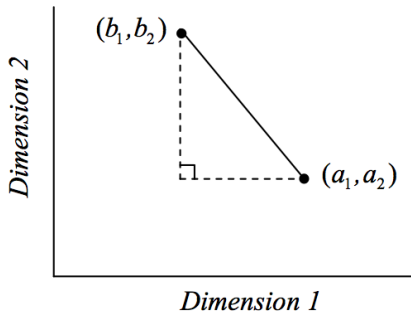
The linear space spanned by a , denoted by $\mathcal{R}(a)$, is the collection of points $\beta a = (\beta a_1, \dots, \beta a_n)$ for any real number β .

The **projection** of b onto $\mathcal{R}(a)$ is the point b^* in $\mathcal{R}(a)$ that is closest to b in terms of Euclidean distance:

$$b^* = \left(\frac{a \cdot b}{\|a\|^2} \right) a$$

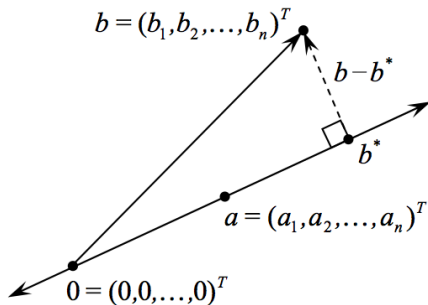
- $(b - b^*) \perp a$

Geometric Interpretation



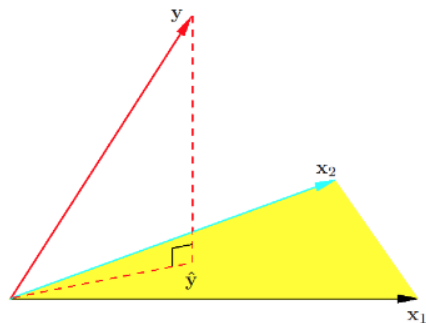
Euclidean Distance in two Dimensions

Geometric Interpretation



Geometric Interpretation

The linear regression fit \hat{Y} is the projection of Y onto the linear space spanned by $\{\mathbf{1}, X_1, \dots, X_p\}$ ⁷.



⁷ $X_j = (x_{1j}, \dots, x_{Nj})'$ for $j = 1, \dots, p$.

Geometric Interpretation

- **Projection matrix** $\mathbb{H} = X(X'X)^{-1}X'$

$$\mathbb{H}Y = \hat{Y}$$

- ▶ \mathbb{H} is also called the **hat matrix**^{8,9}.

- $\hat{e} = Y - X\hat{\beta} = (\mathbb{I} - \mathbb{H})Y \perp \mathcal{R}(\mathbf{1}, X_1, \dots, X_p)$.

- ▶ $\hat{e} \perp X_j \forall j$.

- ▶ $\hat{e} \perp \mathbf{1} \Rightarrow \sum_i \hat{e}_i = 0$.

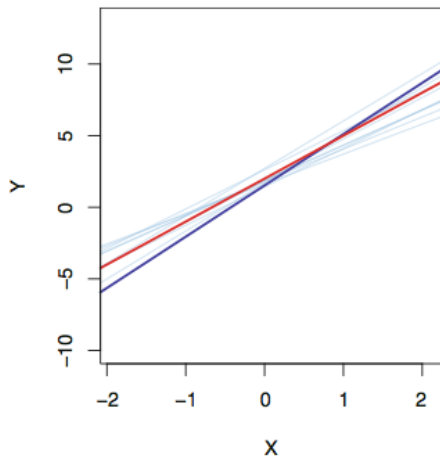
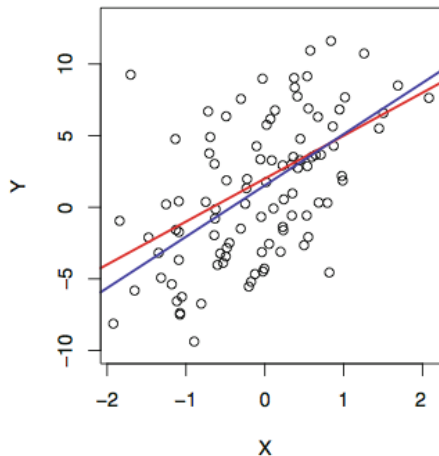
⁸Since it “puts a hat” on Y .

⁹The hat matrix has many special properties such as: $\mathbb{H}^2 = \mathbb{H}$, $(\mathbb{I} - \mathbb{H})^2 = (\mathbb{I} - \mathbb{H})$, and $\text{trace}(\mathbb{H}) = 1 + p$.

Asymptotic Properties

- $\hat{\beta}$ is unbiased: $E(\hat{\beta}) = \beta^*$.
- But how much does $\hat{\beta}$ vary around β^* ?

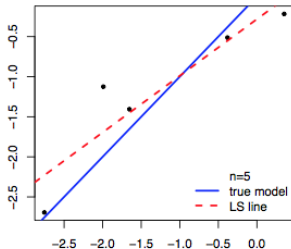
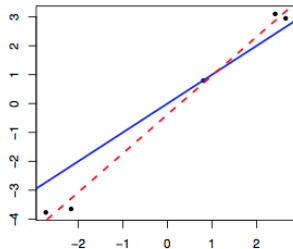
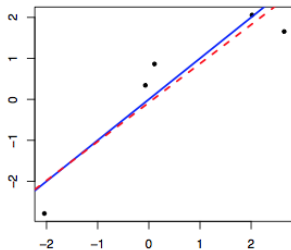
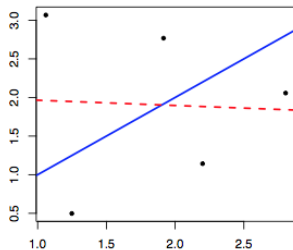
Asymptotic Properties



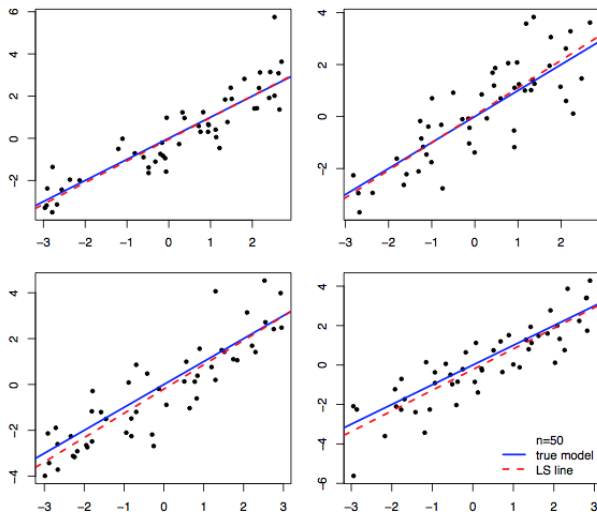
red: $x'\beta^*$. blue: $x'\hat{\beta}$

Right: $x'\hat{\beta}$ based on 10 random set of observations.

Asymptotic Properties



Asymptotic Properties



Asymptotic Properties

By the central limit theorem,

$$\sqrt{N} \left(\hat{\beta} - \beta^* \right) \rightarrow^d \mathcal{N} \left(0, E \left(x x' \right)^{-1} E \left[x x' \left(e^* \right)^2 \right] E \left(x x' \right)^{-1} \right)$$

- $V \left(\hat{\beta} \right) = \underbrace{N^{-1} E \left(x x' \right)^{-1} E \left[x x' \left(e^* \right)^2 \right] E \left(x x' \right)^{-1}}_{(p+1) \times (p+1)}$ is the **asymptotic variance** of $\hat{\beta}$ conditional on x .
- $V \left(\hat{\beta} \right)$ quantifies the uncertainty of $\hat{\beta}$ due to random sampling.

Asymptotic Properties

$$\begin{aligned}\widehat{V}(\widehat{\beta}) &= \left[\sum_{i=1}^N x_i x_i' \right]^{-1} \left(\sum_{i=1}^N x_i x_i' \widehat{e}_i^2 \right) \left[\sum_{i=1}^N x_i x_i' \right]^{-1} \\ &= (X'X)^{-1} (X'\Omega X) (X'X)^{-1} \\ &\rightarrow^p V(\widehat{\beta})\end{aligned}\tag{10}$$

$$, \text{ where } \Omega = \text{diag}(\widehat{e}_1^2, \dots, \widehat{e}_N^2) = \begin{bmatrix} \widehat{e}_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \widehat{e}_N^2 \end{bmatrix}.$$

Asymptotic Properties

Homoskedasticity: $E \left[(e^*)^2 \middle| x \right] = \sigma^2$

Heteroskedasticity: $E \left[(e^*)^2 \middle| x \right] = \sigma^2(x)$

Under homoskedasticity,

$$\sqrt{N} \left(\hat{\beta} - \beta^* \right) \rightarrow^d \mathcal{N} \left(0, E \left(x x' \right)^{-1} \sigma^2 \right)$$

$$\hat{V} \left(\hat{\beta} \right) = \left(X' X \right)^{-1} \hat{\sigma}^2 \tag{11}$$

Asymptotic Properties

From (9), we can also derive the homoskedastic asymptotic variance of $\hat{\beta}_j$ – the $(j+1)^{th}$ diagonal element of $V(\hat{\beta})$ – as:

For $j = 1, \dots, p$,

$$\sqrt{N}(\hat{\beta}_j - \beta_j^*) \rightarrow^d \mathcal{N}\left(0, \frac{\sigma^2}{\text{Var}(u_j)}\right)$$
$$\hat{V}(\hat{\beta}_j) = \frac{\hat{\sigma}^2}{\hat{u}_j' \hat{u}_j} \quad (12)$$

Asymptotic Properties

- **t-statistic**

$$t_j = \frac{\hat{\beta}_j - \beta_j^*}{\widehat{\text{se}}(\hat{\beta}_j)} \rightarrow^d \mathcal{N}(0, 1)$$

, where $\widehat{\text{se}}(\hat{\beta}_j) = \sqrt{\widehat{V}(\hat{\beta}_j)}$.

- **95% confidence interval** for β_j^* :

$$\left[\hat{\beta}_j - 1.96 \times \widehat{\text{se}}(\hat{\beta}_j), \hat{\beta}_j + 1.96 \times \widehat{\text{se}}(\hat{\beta}_j) \right]$$

- ▶ The interval represents a **set estimate** of β_j^* .

Hypothesis Testing

$$\mathbb{H}_0 : \beta_j^* = 0 \text{ vs. } \mathbb{H}_1 : \beta_j^* \neq 0$$

Under H_0 ,

$$t_j = \frac{\hat{\beta}_j}{\widehat{se}(\hat{\beta}_j)} \rightarrow^d \mathcal{N}(0, 1) \quad (13)$$

P-value: probability of observing any value more extreme than $|t_j|$ under \mathbb{H}_0 . (13) \Rightarrow in large sample,

$$p\text{-value} \approx 2(1 - \Phi(|t_j|)) \quad (14)$$

, where Φ is the CDF of $\mathcal{N}(0, 1)$.

Hypothesis Testing

For significance level α , reject \mathbb{H}_0 if $|t_j| > c_\alpha = \Phi^{-1}(1 - \alpha/2)$, or equivalently, if p -value $< \alpha$ ¹⁰.

- c_α is called the **asymptotic critical value**.
- Common practice: $\alpha = 5\%$ ($c_{.05} \approx 1.96$), $\alpha = 10\%$ ($c_{.10} \approx 1.64$), $\alpha = 1\%$ ($c_{.01} \approx 2.58$).

¹⁰It is worth emphasizing that (14) is only valid *in large samples*, since it is based on the asymptotic distribution of t_j . Any p -values calculated using (14) on small samples should *not* be trusted. In general, hypothesis tests based on the asymptotic properties of test statistics are only valid for large samples.

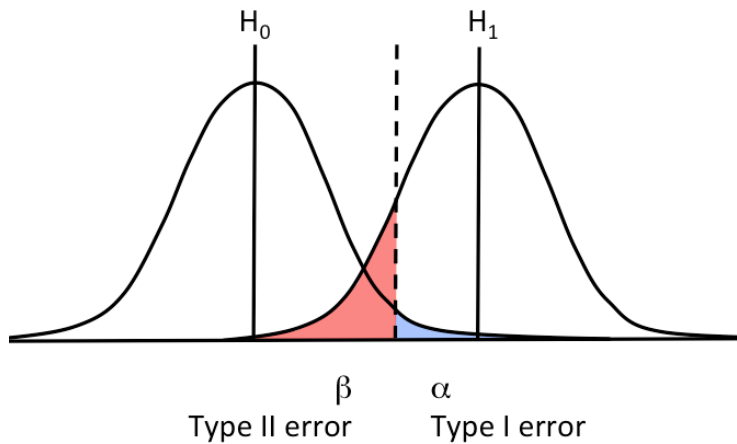
Hypothesis Testing

Hypothesis Testing Decisions

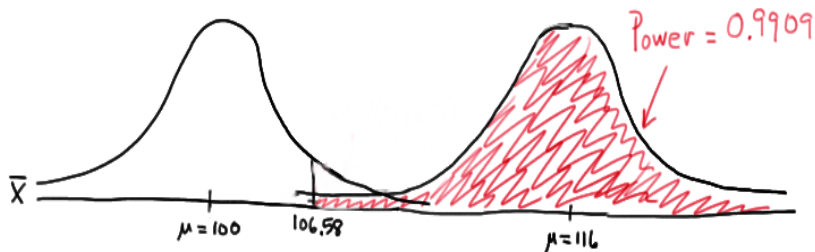
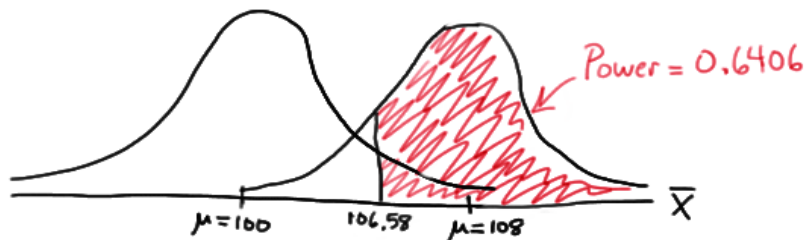
	Accept \mathbb{H}_0	Reject \mathbb{H}_0
\mathbb{H}_0 true	Correct Decision	Type I Error
\mathbb{H}_1 true	Type II Error	Correct Decision

- α is the **size** of the test – the probability of making a Type I error: $\Pr(\text{reject } \mathbb{H}_0 | \mathbb{H}_0 \text{ is true})$.
- The **power** or **sensitivity** of a test, is the probability of rejecting \mathbb{H}_0 when \mathbb{H}_1 is true. Thus $(1 - \text{power})$, denoted by β , is the probability of making a Type II error: $\Pr(\text{fail to reject } \mathbb{H}_0 | \mathbb{H}_1 \text{ is true})$.
 - ▶ Power \uparrow as $\alpha \uparrow$, or sample size $N \uparrow$, or the true (population) parameter value is further away from its hypothesized value under \mathbb{H}_0 .

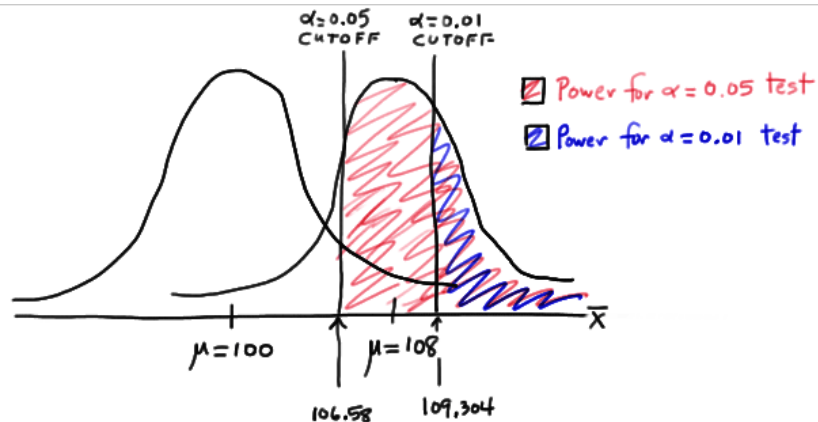
Hypothesis Testing



Hypothesis Testing



Hypothesis Testing



$$R^2 = \frac{\sum_{i=1}^N (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^N (y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^N \hat{e}_i^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

- measures the amount of variation in y_i accounted for by the model:
1 = perfect, 0 = perfect misfit.
- cannot go down when you add regressors.
 - ▶ Intuition: adding more regressors always allow us to fit the training data more accurately (i.e., reduce E_{in} , but not necessary E_{out})¹¹.

¹¹Technically, $\hat{\beta}$ is chosen to minimize $\sum_i \hat{e}^2$. if you add a regressor, you can always set the coefficient of that regressor equal to zero to get the same $\sum_i \hat{e}^2$. Therefore R^2 cannot go down.

Robust Standard Errors

(10) is known as **heteroskedasticity-consistent (HC) standard error**, **robust standard error**, or **White standard error**.

Let's generate some data:

$$x = U(0, 100)$$

$$y = 5x + e, e \sim \mathcal{N}(0, \exp(x))$$

```
n <- 1e3  
x <- 100*runif(n)  
y <- rnorm(n, mean=5*x, sd=exp(x))
```

Robust Standard Errors

```
require(AER)
coeftest(lm(y~x)) # homoskedastic standard error

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.0116e+41  9.0736e+40   1.1148   0.26519
## x            -3.0822e+39  1.5634e+39  -1.9715   0.04895 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(lm(y~x),vcov=vcovHC) # robust standard error

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.0116e+41  8.6253e+40   1.1728   0.2412
## x            -3.0822e+39  2.6314e+39  -1.1713   0.2417
```

The Bootstrap

- The bootstrap is a statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical method.
 - ▶ For example, it can provide an estimate of the standard error of a coefficient.
- The term is believed to derive from “The Surprising Adventures of Baron Munchausen” by Rudolph Erich Raspe¹²:

The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.

¹²We also have the Munchausen number – a number that is equal to the sum of each digit raised to the power of itself. E.g., $3435 = 3^3 + 4^4 + 3^3 + 5^5$.

The Bootstrap



Munchhausen

O. Herrfurth pinx

Baron Munchausen
pulls himself out of
a mire by his own
hair (illustration by
Oskar Herrfurth)

The Bootstrap

Suppose we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y . Suppose our goal is to minimize the total risk, or variance, of our investment. Then the problem is to choose α such that

$$\alpha = \arg \min_{\gamma} \text{Var} [\gamma X + (1 - \gamma) Y] \quad (15)$$

(15) \Rightarrow

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}} \quad (16)$$

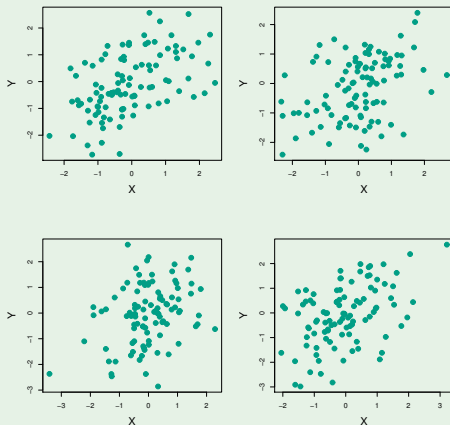
The Bootstrap

Suppose we do not know $\sigma_X^2, \sigma_Y^2, \sigma_{XY}$ but have access to a random sample \mathcal{D} that is drawn from $p(X, Y)$. Then we can compute $\hat{\sigma}_X^2, \hat{\sigma}_Y^2, \hat{\sigma}_{XY}$ from \mathcal{D} and calculate $\hat{\alpha}$.

Simulation:

- $\sigma_X^2 = 1, \sigma_Y^2 = 1.25, \sigma_{XY} = 0.5$ ($\Rightarrow \alpha = 0.6$)
- Draw random samples $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ from $p(X, Y)$.

The Bootstrap



Each panel displays 100 simulated returns for investments X and Y .
The resulting estimates for α are 0.576, 0.532, 0.657, and 0.651 clockwise.

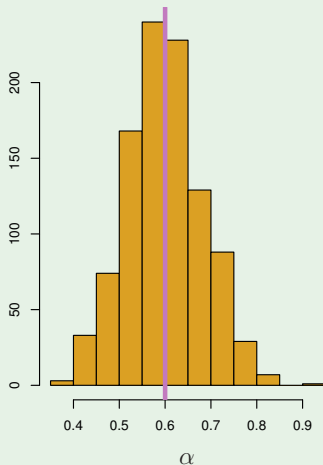
The Bootstrap

To estimate the standard deviation of α , we simulate R random samples \mathcal{D} from $p(X, Y)$ and estimate α R times \Rightarrow

$$\hat{\alpha} = \frac{1}{R} \sum_{r=1}^R \hat{\alpha}_r$$
$$\widehat{se}(\hat{\alpha}) = \sqrt{\frac{1}{R-1} \sum_{r=1}^R (\hat{\alpha}_r - \hat{\alpha})^2}$$

- Let $n = 100$ and $R = 1000$. One run of this simulation $\Rightarrow \hat{\alpha} = 0.5996$ and $\widehat{se}(\hat{\alpha}) = 0.083$.

The Bootstrap

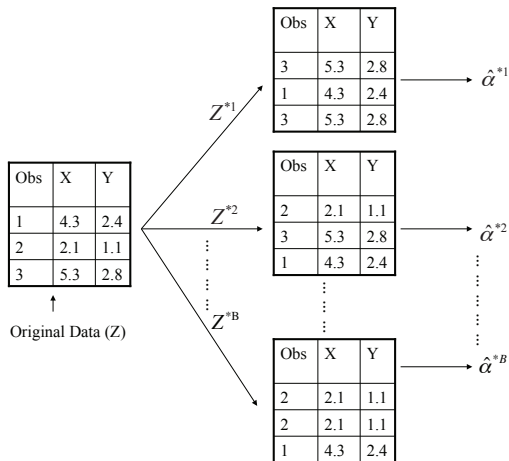


A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.

The Bootstrap

- In practice, we cannot generate new samples from the true population.
- Instead, The bootstrap approach generates new samples from the observed sample itself, by repeatedly drawing observations from the observed sample **with replacement**.
- Each generated bootstrap sample contains the same number of observations as the original observed sample. As a result, some observations may appear more than once in a given bootstrap sample and some not at all.

The Bootstrap



The bootstrap approach on a sample containing 3 observations.

The Bootstrap

```
# Function to calculate alpha
alpha <- function(data,index){
  X <- data$X[index]
  Y <- data$Y[index]
  return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))
}
#
# 'Portfolio' is a simulated data set containing the returns of X and Y
require(ISLR) # contains 'Portfolio'
n <- nrow(Portfolio)
bootsample <- sample(n,n,replace=T) # generate one bootstrap sample
alpha(Portfolio,bootsample) # calculate alpha based on the bootstrap sample

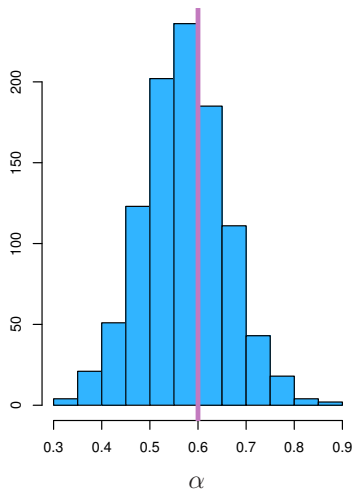
## [1] 0.4896806
```


The Bootstrap

```
# Calculate alpha based on 1000 bootstrap samples
require(boot)
boot(Portfolio,alpha,R=1000)

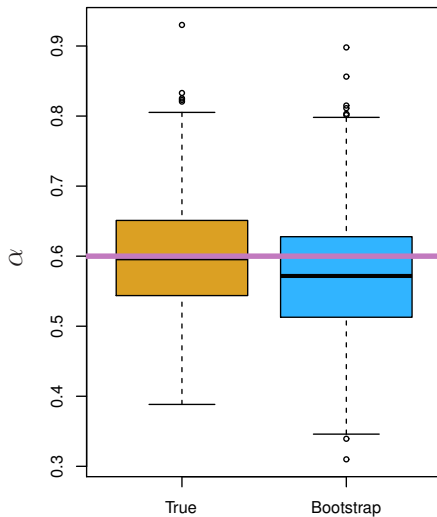
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Portfolio, statistic = alpha, R = 1000)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 0.5758321 0.002353412 0.08752433
```

The Bootstrap

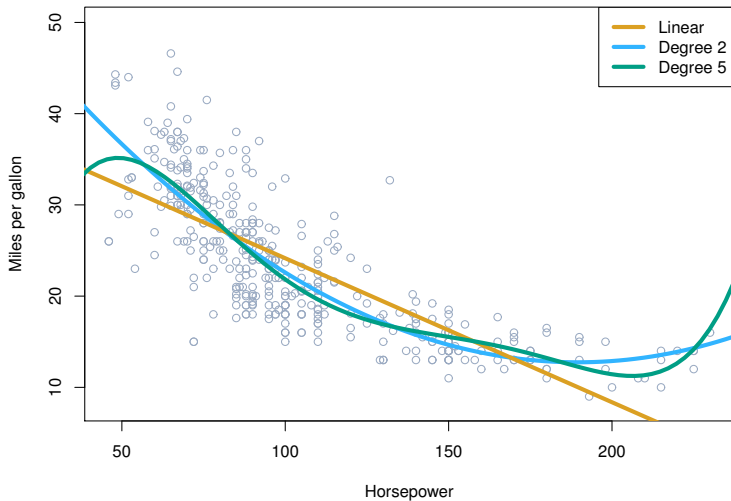


A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set

The Bootstrap



MPG and Horsepower



MPG and Horsepower

```
require(ISLR) # contains the data set 'Auto'
require(boot)
beta <- function(data,index){
  coef(lm(mpg~horsepower,data=data,subset=index))
}
boot(Auto,beta,R=1000)

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Auto, statistic = beta, R = 1000)
##
##
## Bootstrap Statistics :
##      original      bias      std. error
## t1* 39.9358610  0.0269563085 0.859851825
## t2* -0.1578447 -0.0002906457 0.007402954
```

MPG and Horsepower

```
require(AER)
coeftest(lm(mpg ~ horsepower, data=Auto)) # homoskedastic std err

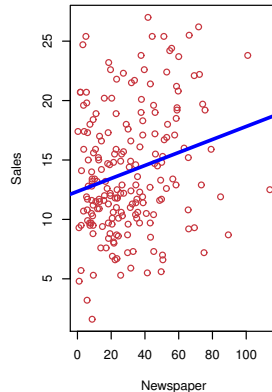
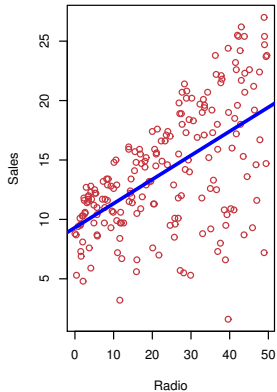
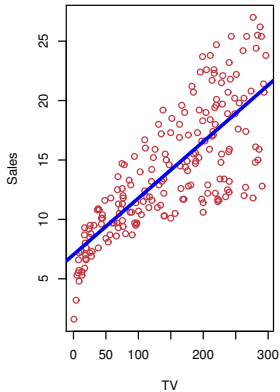
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.9358610   0.7174987   55.660 < 2.2e-16 ***
## horsepower  -0.1578447   0.0064455  -24.489 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

MPG and Horsepower

```
coeftest(lm(mpg ~ horsepower, data=Auto),vcov=vcovHC) # robust std err

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.9358610  0.8644903  46.196 < 2.2e-16 ***
## horsepower  -0.1578447  0.0074943 -21.062 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Advertising and Sales



Advertising and Sales

	Coefficient	Std. error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	9.312	0.563	16.54	< 0.0001
radio	0.203	0.020	9.92	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	12.351	0.621	19.88	< 0.0001
newspaper	0.055	0.017	3.30	< 0.0001

Simple regression of **sales** on **TV**, **radio** and **newspaper** respectively.

Advertising and Sales

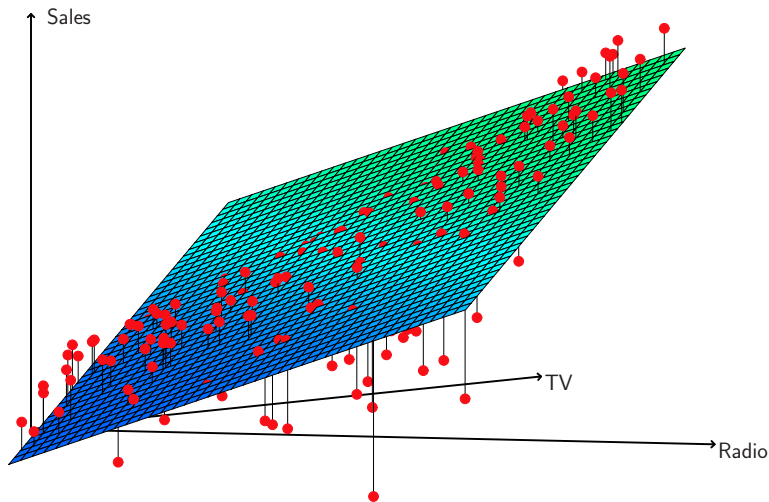
	Coefficient	Std. error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

$$sales = \beta_0 + \beta_1 TV + \beta_2 radio + \beta_3 newspaper + e$$

Advertising and Sales

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

Advertising and Sales



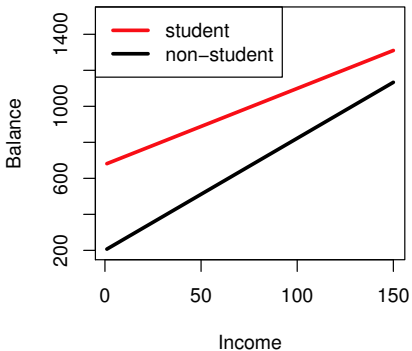
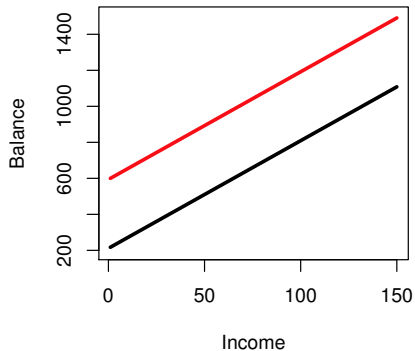
$$sales = \beta_0 + \beta_1 TV + \beta_2 radio + e$$

Advertising and Sales

	Coefficient	Std. error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

$$sales = \beta_0 + \beta_1 TV + \beta_2 radio + \beta_3 (TV \times radio) + e$$

Credit Card Balance



Credit Card Balance

Let $student \in \{0, 1\}$ indicate student status. Two models:

$$\begin{aligned} balance &= \beta_0 + \beta_1 income + \beta_2 student + e \\ &= \begin{cases} \beta_0 + \beta_1 income + e & \text{if not student} \\ (\beta_0 + \beta_2) + \beta_1 income + e & \text{if student} \end{cases} \end{aligned}$$

$$\begin{aligned} balance &= \beta_0 + \beta_1 income + \beta_2 student + \beta_3 income \times student + e \\ &= \begin{cases} \beta_0 + \beta_1 income + e & \text{if not student} \\ (\beta_0 + \beta_2) + (\beta_1 + \beta_3) income + e & \text{if student} \end{cases} \end{aligned}$$

Interaction Terms and the Hierarchy Principle

- Sometimes an interaction term has a very small p-value, but the associated main effects do not.
- **The hierarchy principle:** If we include an interaction in a model, we should also include the main effects, even if the p-values associated with their coefficients are not significant.

Log-Linear Regression

When y changes on a multiplicative or percentage scale, it is often appropriate to use $\log(y)$ as the dependent variable¹³:

$$y = Ae^{\beta x + e} \Rightarrow \log(y) = \log(A) + \beta x + e$$

e.g.,

$$\log(\text{GDP}) = \alpha + g \times t + e$$

, where t = years, $\alpha = \log(\text{base year GDP})$, and g = annual growth rate.

¹³ Suppose y grows at a rate i . If i is continuously compounded, then $y_t = y_0 \lim_{n \rightarrow \infty} \left(1 + \frac{i}{n}\right)^{nt} = y_0 e^{it} \Rightarrow \log(y_t) = \log(y_0) + i \times t$. If i is not continuously compounded, then $y_t = y_0 (1 + i)^t \Rightarrow \log(y_t) = \log(y_0) + t \log(1 + i) \approx \log(y_0) + i \times t$.

Elasticity and Log-Log Regression

In a log-log model:

$$\log(y) = \beta_0 + \beta_1 \log(x) + e$$

β_1 can often be interpreted as an elasticity measure:

$$\beta_1 = \frac{\partial \log(y)}{\partial \log(x)} = \frac{\partial y / y}{\partial x / x} \approx \frac{\% \Delta y}{\% \Delta x}$$

e.g.,

$$\log(\text{sales}) = \beta_0 + \beta_1 \log(\text{price}) + e$$

Target Transform

14

¹⁴Note: in general, $E[f(y)] \neq f(E[y])$. In particular, by the Jensen's inequality, $E[\log(y)] < \log(E[y])$. Therefore, if $E[\log(y)|x] = \alpha + \beta x$, then $E[y|x] > \exp(\alpha + \beta x)$.

If we are willing to assume

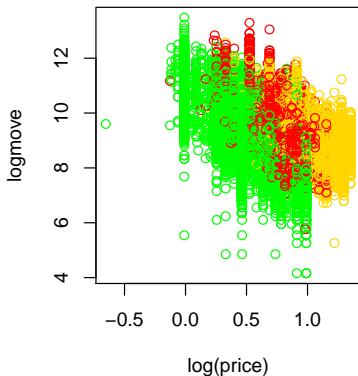
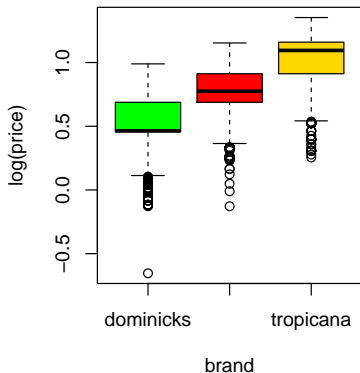
$$\log(y) = \alpha + \beta x + e, \quad e \sim \mathcal{N}(0, \sigma^2)$$

, then we have: $E[y|x] = \exp(\alpha + \beta x + \frac{1}{2}\sigma^2) = \exp(E[\log(y)|x] + \frac{1}{2}\sigma^2)$.

Orange Juice

Three brands: Tropicana, Minute Maid, Dominick's

Data from 83 stores on price, sales (units moved), and whether featured in the store



Orange Juice

$$\log(\text{sales}) = \alpha + \beta \log(\text{price}) + e$$

```
require(AER)
oj <- read.csv('oj.csv')
reg1 <- lm(logmove ~ log(price), data=oj)
coeftest(reg1)

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.4234      0.0154   679.0   <2e-16 ***
## log(price)   -1.6013      0.0184  -87.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Orange Juice

$\log(\text{sales}) = \alpha_b + \beta_b \log(\text{price}) + e$, where b denotes brand

```
reg2 <- lm(logmove ~ log(price)*brand, data=oj)
coeftest(reg2)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.9547    0.0207   529.14 <2e-16 ***
## log(price)       -3.3775    0.0362  -93.32 <2e-16 ***
## brandminute.maid    0.8883    0.0416   21.38 <2e-16 ***
## brandtropicana     0.9624    0.0464   20.72 <2e-16 ***
## log(price):brandminute.maid  0.0568    0.0573    0.99    0.32
## log(price):brandtropicana    0.6658    0.0535   12.44 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Orange Juice

$$\log(\text{sales}) = (\alpha_{0b} + \text{feature} \times a_{1b}) + (\beta_{0b} + \text{feature} \times \beta_{1b}) \times \log(\text{price}) + e$$

```
reg3 <- lm(logmove ~ log(price)*brand*feat, data=oj)
coeftest(reg3)
```

```
##
## t test of coefficients:
##
##
```

	Estimate	Std. Error	t value	Pr(> t)	
## (Intercept)	10.4066	0.0234	445.67	< 2e-16	***
## log(price)	-2.7742	0.0388	-71.45	< 2e-16	***
## brandminute.maid	0.0472	0.0466	1.01	0.31	
## brandtropicana	0.7079	0.0508	13.94	< 2e-16	***
## feat	1.0944	0.0381	28.72	< 2e-16	***
## log(price):brandminute.maid	0.7829	0.0614	12.75	< 2e-16	***
## log(price):brandtropicana	0.7358	0.0568	12.95	< 2e-16	***
## log(price):feat	-0.4706	0.0741	-6.35	2.2e-10	***
## brandminute.maid:feat	1.1729	0.0820	14.31	< 2e-16	***
## brandtropicana:feat	0.7853	0.0987	7.95	1.9e-15	***
## log(price):brandminute.maid:feat	-1.1092	0.1222	-9.07	< 2e-16	***
## log(price):brandtropicana:feat	-0.9861	0.1241	-7.95	2.0e-15	***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Orange Juice

- Elasticity¹⁵: -1.6
- Brand-specific elasticities:
Dominick's: -3.4 , Minute Maid: -3.4 , Tropicana: -2.7
- How does featuring a product affect its elasticity?

	Dominick's	Minute Maid	Tropicana
not featured	-2.8	-2.0	-2.0
featured	-3.2	-3.6	-3.5

¹⁵What economic assumptions need to be satisfied in order for the coefficients to be interpreted as price elasticities of demand?

CAPM

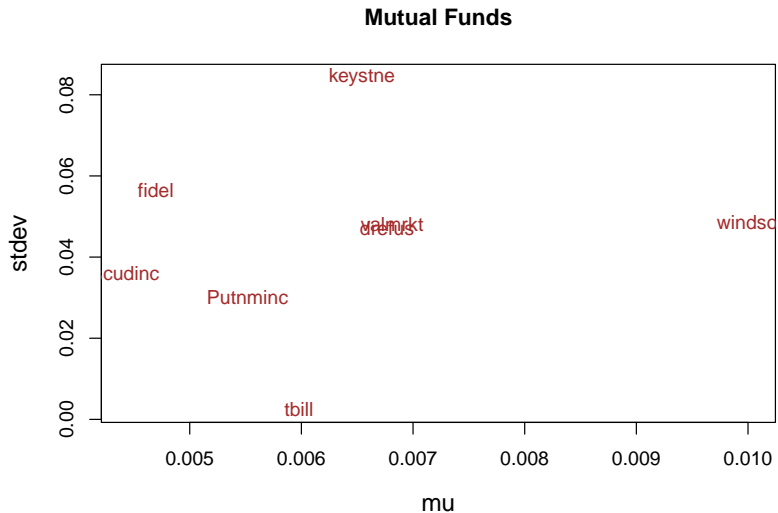
The Capital Asset Pricing Model (CAPM) for asset A relates return $R_{A,t}$ to the market return, $R_{M,t}$:

$$R_{A,t} = \alpha + \beta R_{M,t} + e$$

When asset A is a mutual fund, this CAPM regression can be used as a performance benchmark for fund managers.

```
# 'mfunds.csv' contains data on the historical returns of  
# 6 mutual funds as well as the market return  
mfund <- read.csv('mfunds.csv')  
mu <- apply(mfund,2,mean)  
stdev <- apply(mfund,2,sd)
```

CAPM



CAPM

```
CAPM <- lm(as.matrix(mfund[,1:6]) ~ mfund$valmrkt)
```

```
CAPM
```

```
##
```

```
## Call:
```

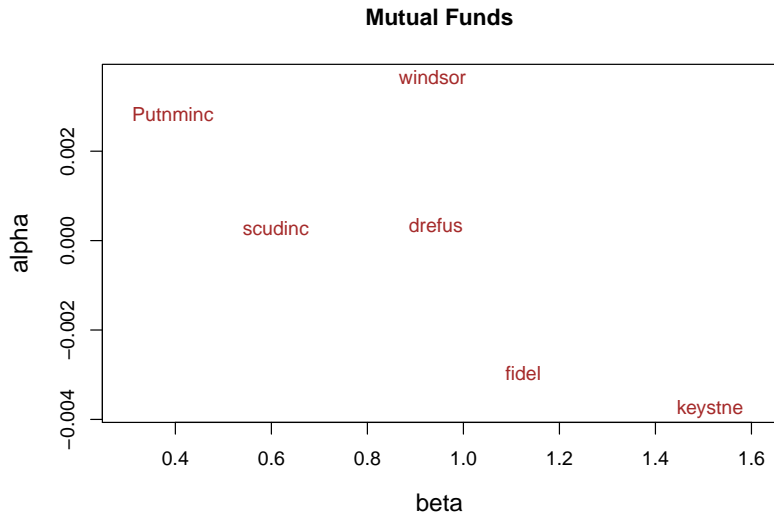
```
## lm(formula = as.matrix(mfund[, 1:6]) ~ mfund$valmrkt)
```

```
##
```

```
## Coefficients:
```

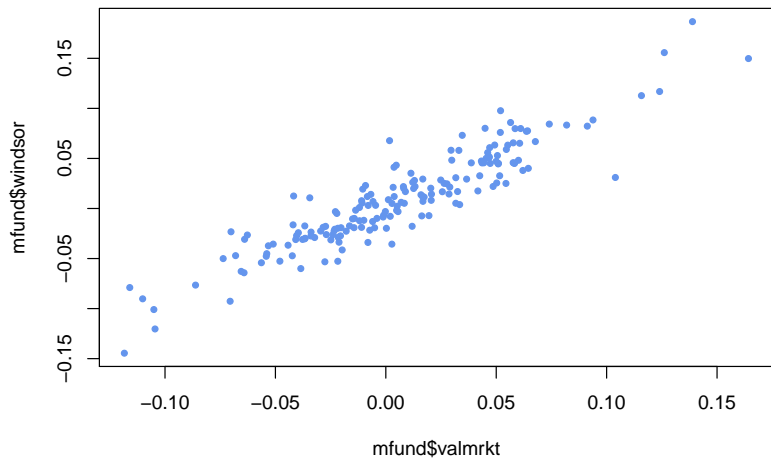
##	drefus	fidel	keystne	Putnminc	scudinc
## (Intercept)	0.0003462	-0.0029655	-0.0037704	0.0028271	0.000281
## mfund\$valmrkt	0.9424286	1.1246549	1.5137186	0.3948280	0.609202
##	windsor				
## (Intercept)	0.0036469				
## mfund\$valmrkt	0.9357170				

CAPM



CAPM

Look at windsor (which dominates the market):



CAPM

Does Windsor have an “alpha” over the market?

$H_0 : \alpha = 0$ vs. $H_1 : \alpha \neq 0$

```
require(AER)
reg <- lm(mfund$windsor ~ mfund$valmrkt)
coeftest(reg)

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0036469  0.0014094   2.5876  0.01046 *
## mfund$valmrkt 0.9357170  0.0291499  32.1002 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

CAPM

Now look at beta:

$H_0 : \beta = 1$, Windsor is just the market (+ alpha).

$H_1 : \beta \neq 1$, Windsor softens or exaggerates market moves.

```
linearHypothesis(reg, "mfund$valmrkt = 1")
```

```
## Linear hypothesis test
```

```
##
```

```
## Hypothesis:
```

```
## mfund$valmrkt = 1
```

```
##
```

```
## Model 1: restricted model
```

```
## Model 2: mfund$windsor ~ mfund$valmrkt
```

```
##
```

```
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
```

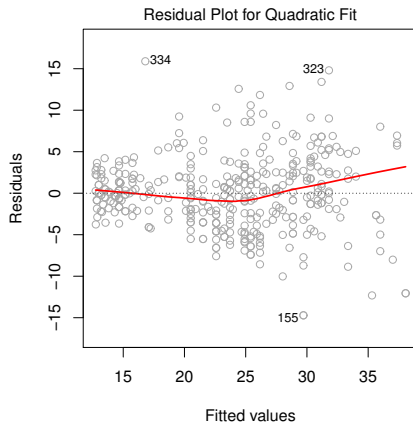
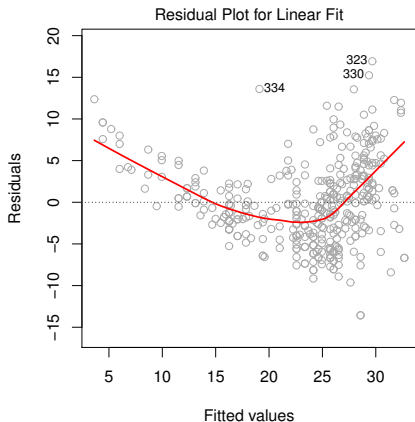
```
## 1      179 0.064082
```

```
## 2      178 0.062378   1 0.0017042 4.8632 0.02872 *
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Regression Diagnostics: Is the CEF linear?



Regression Diagnostics: Is the CEF linear?

Anscombe's quartet comprises four datasets that have similar statistical properties ...

```
anscombe <- read.csv('anscombe.csv')
attach(anscombe)
c(x.m1=mean(x1), x.m2=mean(x2), x.m3=mean(x3), x.m4=mean(x4))

## x.m1 x.m2 x.m3 x.m4
##    9    9    9    9

c(y.m1=mean(y1), y.m2=mean(y2), y.m3=mean(y3), y.m4=mean(y4))

##      y.m1      y.m2      y.m3      y.m4
## 7.500909 7.500909 7.500000 7.500909
```

Regression Diagnostics: Is the CEF linear?

```
c(x.sd1=sd(x1),x.sd2=sd(x2),x.sd3=sd(x3),x.sd3=sd(x4))
```

```
##      x.sd1      x.sd2      x.sd3      x.sd3  
## 3.316625 3.316625 3.316625 3.316625
```

```
c(y.sd1=sd(y1),y.sd2=sd(y2),y.sd4=sd(y3),y.sd4=sd(y4))
```

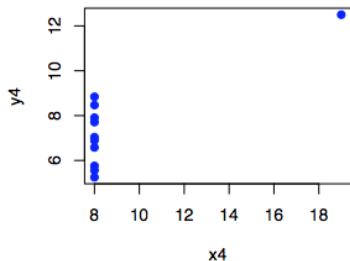
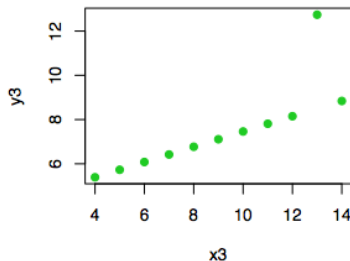
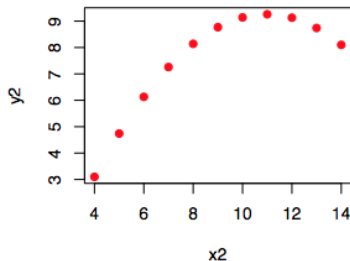
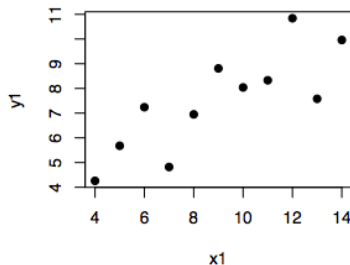
```
##      y.sd1      y.sd2      y.sd4      y.sd4  
## 2.031568 2.031657 2.030424 2.030579
```

```
c(cor1=cor(x1,y1),cor2=cor(x2,y2),cor3=cor(x3,y3),cor4=cor(x4,y4))
```

```
##      cor1      cor2      cor3      cor4  
## 0.8164205 0.8162365 0.8162867 0.8165214
```

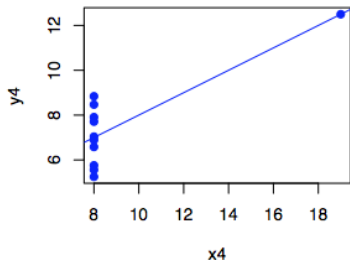
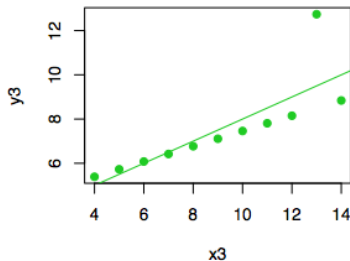
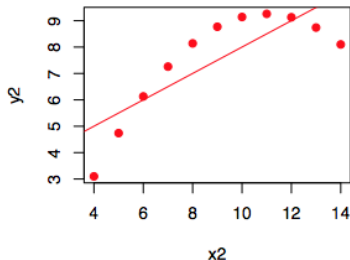
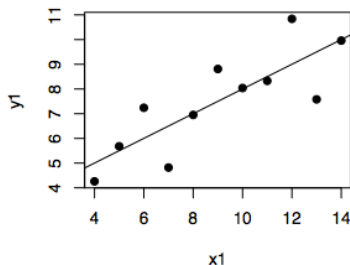
Regression Diagnostics: Is the CEF linear?

...but vary considerably when graphed:



Regression Diagnostics: Is the CEF linear?

Linear regression on each dataset:



Regression Diagnostics: Is the CEF linear?

The regression lines and R^2 values are the same...

```
areg <- list(areg1=lm(y1~x1),areg2=lm(y2~x2),areg3=lm(y3~x3),areg4=lm(y4~x4))
attach(areg)
cbind(areg1$coef,areg2$coef,areg3$coef,areg4$coef)

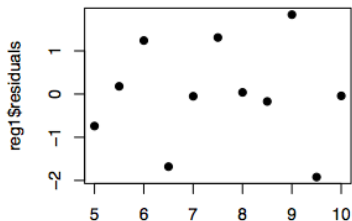
##           [,1]      [,2]      [,3]      [,4]
## (Intercept) 3.0000909 3.000909 3.0024545 3.0017273
## x1          0.5000909 0.500000 0.4997273 0.4999091

s <- lapply(areg,summary)
c(s$areg1$r.sq,s$areg2$r.sq,s$areg3$r.sq,s$areg4$r.sq)

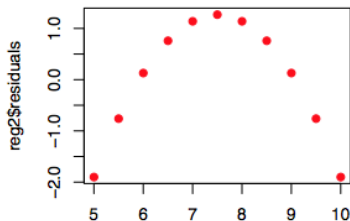
## [1] 0.6665425 0.6662420 0.6663240 0.6667073
```

Regression Diagnostics: Is the CEF linear?

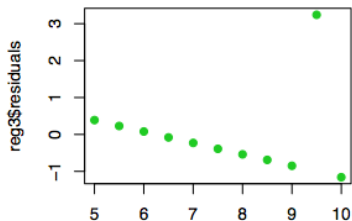
...but residual plots show the differences:



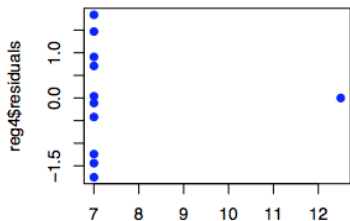
reg1\$fitted



reg2\$fitted

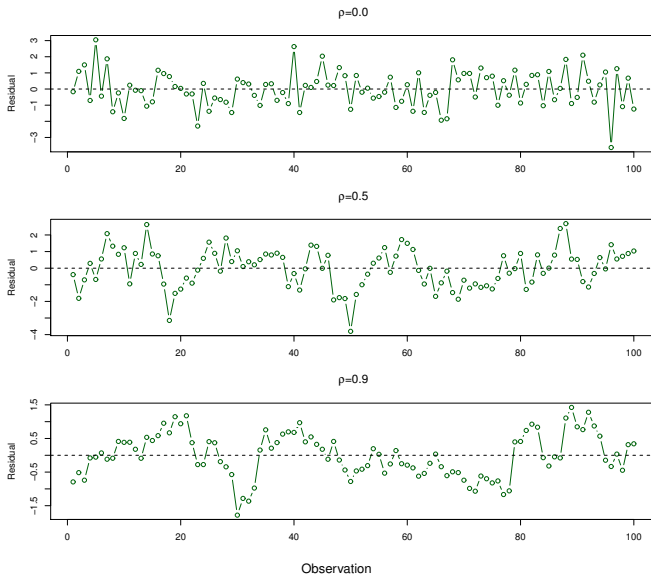


reg3\$fitted

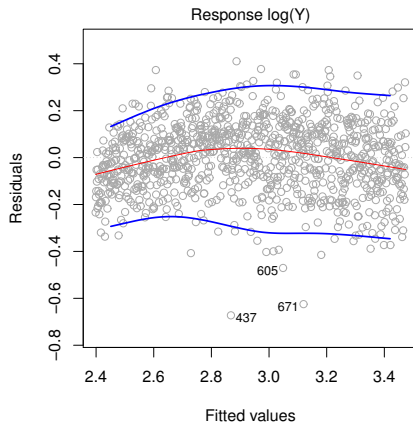
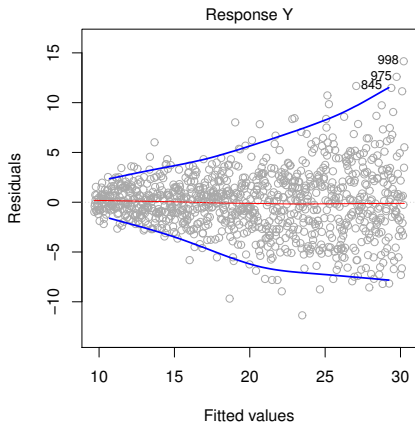


reg4\$fitted

Regression Diagnostics: Nonrandom Sampling



Regression Diagnostics: Heteroskedasticity



Regression Diagnostics: Collinearity

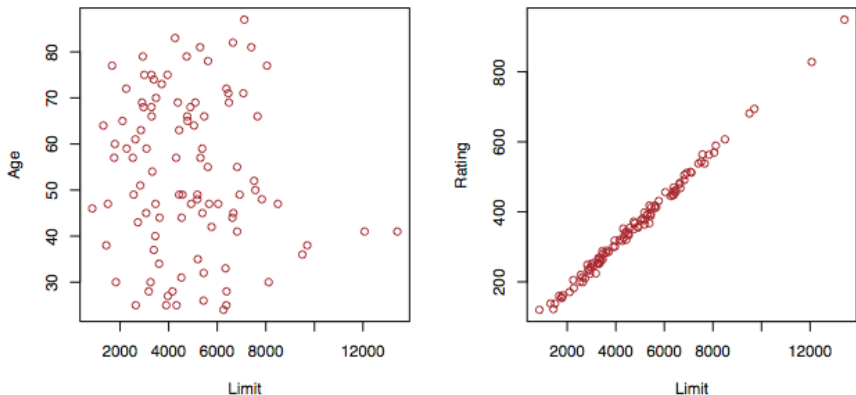


FIGURE 3.14. Scatterplots of the observations from the **Credit** data set. Left: A plot of **age** versus **limit**. These two variables are not collinear. Right: A plot of **rating** versus **limit**. There is high collinearity.

Regression Diagnostics: Collinearity

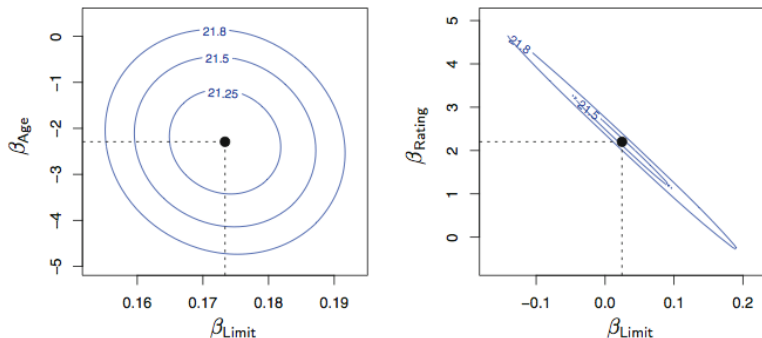


FIGURE 3.15. Contour plots for the RSS values as a function of the parameters β for various regressions involving the **Credit** data set. In each plot, the black dots represent the coefficient values corresponding to the minimum RSS. Left: A contour plot of RSS for the regression of **balance** onto **age** and **limit**. The minimum value is well defined. Right: A contour plot of RSS for the regression of **balance** onto **rating** and **limit**. Because of the collinearity, there are many pairs $(\beta_{\text{Limit}}, \beta_{\text{Rating}})$ with a similar value for RSS.

Regression Diagnostics: Collinearity

From (9), we can see that:

- If X_1, \dots, X_p are orthogonal, $\hat{\beta}_j$ is equal to the simple regression coefficient of y on $(\mathbf{1}, X_j)$.
 - ▶ $\hat{u}_j = X_j - \bar{X}_j$
- If X_1, \dots, X_p are correlated – in particular – if X_j is highly correlated with the other predictors, then \hat{u}_j will be close to 0. This makes $\hat{\beta}_j$ **unstable**, as both the denominator and the numerator are small.

From (12), we can see that:

- If X_j is highly correlated with the other predictors, the variance of $\hat{\beta}_j$ is **inflated**, making it less likely to be significant.

Regression Diagnostics: Collinearity

		Coefficient	Std. error	t-statistic	p-value
Model 1	Intercept	-173.411	43.828	-3.957	< 0.0001
	age	-2.292	0.672	-3.407	0.0007
	limit	0.173	0.005	34.496	< 0.0001
Model 2	Intercept	-377.537	45.254	-8.343	< 0.0001
	rating	2.202	0.952	2.312	0.0213
	limit	0.025	0.064	0.384	0.7012

Regression Diagnostics: Collinearity

- A simple way to detect collinearity is to look at the correlation matrix of the predictors.
- However, it is possible for collinearity to exist between three or more variables even if no pair of variables has a particularly high correlation. This is called **multicollinearity**.
- **Variance inflation factor (VIF):**

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R_{X_j|X_{-j}}^2}$$

, where $R_{X_j|X_{-j}}^2$ is the R^2 from a regression of X_j onto all of the other predictors.

- ▶ $VIF \geq 1$. Large VIF indicates a problematic amount of collinearity.

Regression Diagnostics: Collinearity

- When faced with the problem of collinearity, a simple solution is to drop one of the problematic variables.
- Suppose two variables both contribute in explaining y , but are highly correlated with each other.
 - ▶ Both will be insignificant if both are included in the regression model.
 - ▶ Dropping one will likely make the other significant.
- This is why we can't remove two (or more) supposedly insignificant predictors *at a time*: significance depends on what other predictors are in the model!

Maximum Likelihood Estimation

- While least squares regression learns a deterministic function $f(x)$ that directly maps each x into a prediction of y , an alternative approach is to learn the conditional distribution $p(y|x)$ and use the estimated $p(y|x)$ to form a prediction of y .
- To do so, let $\mathcal{H} = \{q_\theta(y|x) : \theta \in \Theta\}$, where the hypotheses $q_\theta(y|x)$ are conditional distributions parametrized by $\theta \in \Theta$.
- We select a $q_\theta(y|x) \in \mathcal{H}$, or equivalently, a $\theta \in \Theta$, by minimizing the empirical KL divergence, or equivalently, by maximizing the (log) likelihood function.

Maximum Likelihood Estimation

The log likelihood function¹⁶:

$$\log \mathcal{L}(\theta) = \sum_{i=1}^N \log q_{\theta}(y_i | x_i)$$

The maximum likelihood estimator chooses

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta)$$

¹⁶Also written as $\log \mathcal{L}(\theta | \mathcal{D})$ to emphasize its dependence on sample \mathcal{D} .

Normal Linear Model

The normal linear regression model is $\mathcal{H} = \{q_\theta(y|x)\}$, where

$$q_\theta(y|x) = \mathcal{N}(x'\beta, \sigma^2) \quad (17)$$

, where $\theta = (\beta, \sigma)$.

This is equivalent to assuming¹⁷:

$$y = x'\beta + e, \quad e \sim \mathcal{N}(0, \sigma^2) \quad (18)$$

¹⁷Notice the strong assumptions imposed by (17) and (18). In addition to assuming a linear regression function, we are now assuming that (1) at each x , the scatter of y around the regression function is Gaussian (**Gaussianity**); (2) the variance of this scatter is constant (**homoskedasticity**); and (3) there is no dependence between this scatter and anything else (**error independence**).

Normal Linear Model

Given sample \mathcal{D} and model (17),

$$\begin{aligned}\log \mathcal{L} &= \sum_{i=1}^N \log \left\{ \frac{1}{\sqrt{2\pi}\sigma} \exp \left(-\frac{1}{2\sigma^2} (y_i - x_i' \beta)^2 \right) \right\} \\ &= -\frac{N}{2} \log(2\pi) - N \log \sigma - \frac{1}{2\sigma^2} \underbrace{\sum_{i=1}^N (y_i - x_i' \beta)^2}_{\text{RSS}}\end{aligned}\tag{19}$$

Normal Linear Model

Maximizing (19) with respect to β and $\sigma \Rightarrow$

$$\frac{\partial \log \mathcal{L}}{\partial \beta} = 0 \Rightarrow \hat{\beta} = \left[\sum_{i=1}^N x_i x_i' \right]^{-1} \sum_{i=1}^N x_i y_i = (X'X)^{-1} X'Y$$

$$\frac{\partial \log \mathcal{L}}{\partial \sigma} = 0 \Rightarrow \hat{\sigma} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - x_i' \hat{\beta})^2}$$

Thus, maximum likelihood estimation of the normal linear model produces the same estimate of β as least squares regression.

Normal Linear Model

Let's fit the normal linear model (17) on the data we generated on page 14:

```
# Define the negative log likelihood function
nll <- function(theta){
  beta0 <- theta[1]
  beta1 <- theta[2]
  beta2 <- theta[3]
  sigma <- theta[4]
  N <- length(y)
  z <- (y - beta0 - beta1*x1 - beta2*x2)/sigma
  logL <- -1*N*log(sigma) - 0.5*sum(z^2)
  return(-logL)}

#
# Minimize the negative likelihood function
mlefit <- optim(c(0,0,0,1),nll) # initial value for theta: (0,0,0,1)
mlefit$par # parameter estimate

## [1] 1.010153 -2.591790 5.062709 1.004935
```

Normal Linear Model

```
# Alternatively, use the mle2 function from the bbmle package
require(bbmle)
parnames(nll) <- c("beta0", "beta1", "beta2", "sigma")
result <- mle2(nll, start=c(beta0=0, beta1=0, beta2=0, sigma=1))
summary(result)
```

```
## Maximum likelihood estimation
```

```
##
```

```
## Call:
```

```
## mle2(minuslogl = nll, start = c(beta0 = 0, beta1 = 0, beta2 = 0,
##      sigma = 1))
```

```
##
```

```
## Coefficients:
```

	Estimate	Std. Error	z value	Pr(z)
## beta0	1.010134	0.118487	8.5253	< 2.2e-16 ***
## beta1	-2.591654	0.224609	-11.5385	< 2.2e-16 ***
## beta2	5.062493	0.311189	16.2682	< 2.2e-16 ***
## sigma	1.004913	0.031778	31.6227	< 2.2e-16 ***

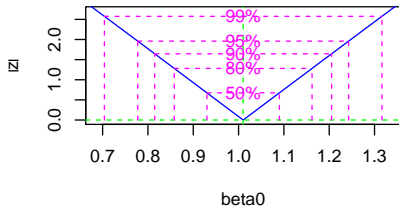
```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

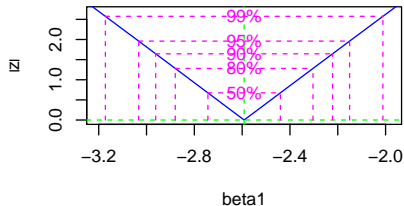
```
##
```

Normal Linear Model

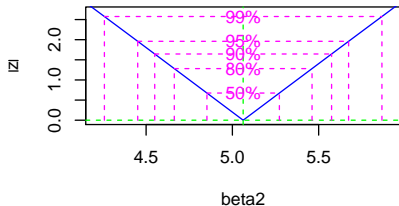
Likelihood profile: beta0



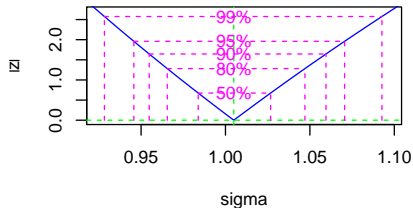
Likelihood profile: beta1



Likelihood profile: beta2



Likelihood profile: sigma

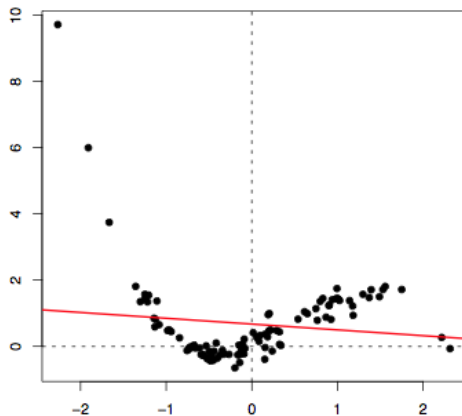


Moving Beyond Linearity

- The CEF $f(x) = E(y|x)$ is seldom linear. The least squares linear regression model, however, doesn't have to be linear in x either. We can move beyond linearity in inputs x as long as we retain linearity in parameters β ¹⁸.
- Polynomial regression is a standard way to extend linear regression to settings in which the relationship between x and y is nonlinear.

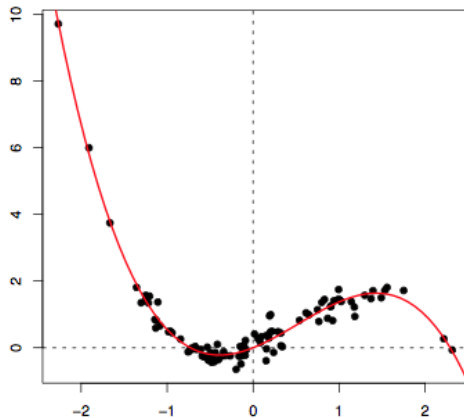
¹⁸We have already seen examples of including nonlinear terms in x such as $\log(x)$ and interaction effects (x_1x_2) in the regression model.

Polynomial Regression



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

Polynomial Regression



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

Wage Profile

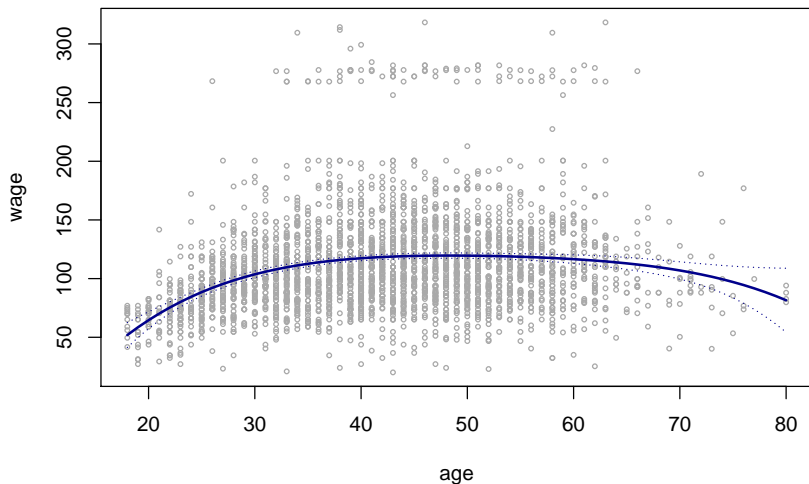
Data: income survey for men in central Atlantic region of USA

```
require(AER)
require(ISLR) # contains the data set 'Wage'
fit <- lm(wage ~ poly(age,4,raw=T), data=Wage) # degree-4 polynomial
coeftest(fit)

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.8415e+02  6.0040e+01 -3.0672 0.0021803 **
## poly(age, 4, raw = T)1  2.1246e+01  5.8867e+00  3.6090 0.0003124 ***
## poly(age, 4, raw = T)2 -5.6386e-01  2.0611e-01 -2.7357 0.0062606 **
## poly(age, 4, raw = T)3  6.8107e-03  3.0659e-03  2.2214 0.0263978 *
## poly(age, 4, raw = T)4 -3.2038e-05  1.6414e-05 -1.9519 0.0510386 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Wage Profile

Degree-4 Polynomial



Piecewise Constant Regression

- For the following analysis, consider modeling the relationship between y and a single input variable x .
- So far we have imposed a global structure on the relationship between x and y .
- Piecewise regression breaks the input space into distinct regions and fit a different relationship in each region.

Piecewise Constant Regression

How it works:

1. Divide the range of x into M regions by creating $M - 1$ cutpoints, or **knots**, ξ_1, \dots, ξ_{M-1} . Then construct the following dummy variables:

Region	$\phi(x)$
R_1	$\phi_1(x) = \mathcal{I}(x < \xi_1)$
R_2	$\phi_2(x) = \mathcal{I}(\xi_1 \leq x < \xi_2)$
\vdots	\vdots
R_M	$\phi_M(x) = \mathcal{I}(\xi_{M-1} \leq x)$

Piecewise Constant Regression

How it works:

2. Fit the following model:

$$y = \beta_1 \phi_1(x) + \beta_2 \phi_2(x) + \cdots + \beta_M \phi_M(x) + e \quad (20)$$

$\sum_{m=1}^M \beta_m \phi_m(x)$ is a **step function** or **piecewise constant function**, and (20) is called a **piecewise constant regression** model.

Piecewise Constant Regression

Solving (20) by least squares \Rightarrow

$$\hat{\beta}_m = \bar{y}_m$$

, where $\bar{y}_m \equiv \frac{1}{n_m} \sum_{x_i \in R_m} y_i$ ¹⁹.

i.e., for every $x \in R_m$, we make the same prediction, which is simply the mean of the response values for the training observations in R_m .

¹⁹ n_m is the number of observations in R_m .

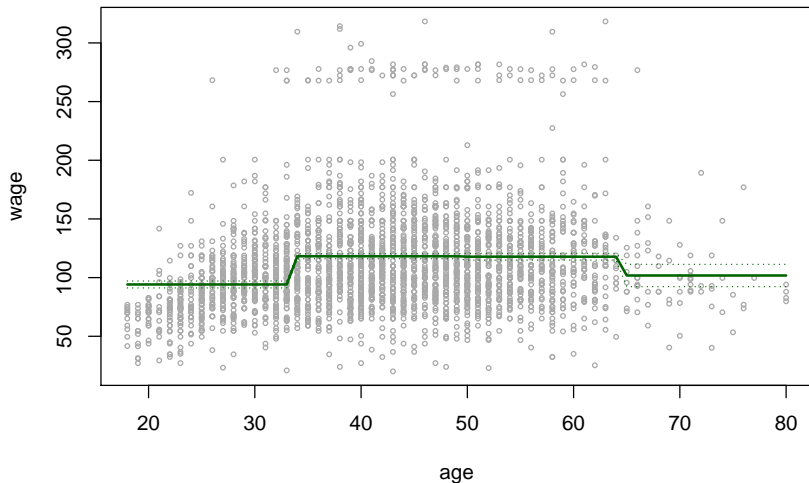
Wage Profile

```
# cut(x,M) divides x into M pieces of equal length
#           and generates the corresponding dummy variables
fit <- lm(wage ~ 0 + cut(age,4), data=Wage) # no intercept
coeftest(fit)

##
## t test of coefficients:
##
##               Estimate Std. Error t value Pr(>|t|)
## cut(age, 4)(17.9,33.5]   94.1584      1.4761  63.790 < 2.2e-16 ***
## cut(age, 4)(33.5,49]    118.2119      1.0808 109.379 < 2.2e-16 ***
## cut(age, 4)(49,64.5]    117.8230      1.4483  81.351 < 2.2e-16 ***
## cut(age, 4)(64.5,80.1]  101.7990      4.7640  21.368 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Wage Profile

Piecewise Constant



Basis Functions

In general, $\phi(x)$ are called **basis functions** and do not have to be dummy variables. They can be any functions of x .

A **linear basis function model** is defined as²⁰:

$$y = \beta_1 \phi_1(x) + \beta_2 \phi_2(x) + \cdots + \beta_M \phi_M(x) + e = \beta' \Phi(x) + e \quad (21)$$

, where $\beta = (\beta_1, \dots, \beta_M)'$ and $\Phi = (\phi_1, \dots, \phi_M)'$.

Solving (21) by least squares \Rightarrow

$$\hat{\beta} = (\Phi' \Phi)^{-1} \Phi' Y$$

, where $\Phi = \Phi(X)$.

²⁰Notice that (21) is the same as (20), except now $\phi(x)$ can be any function of x .

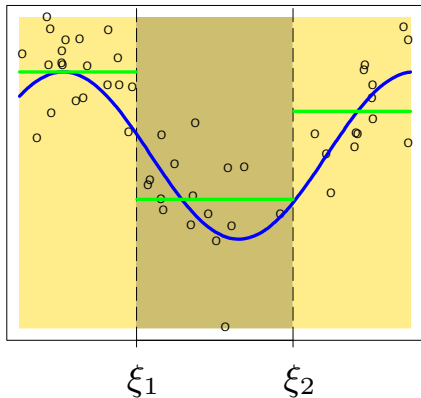
Regression Splines

- Polynomial and piecewise constant regression models are special cases of linear basis function models²¹.
- We can also do **piecewise polynomial regression**, which involves fitting different polynomials over different regions of x .

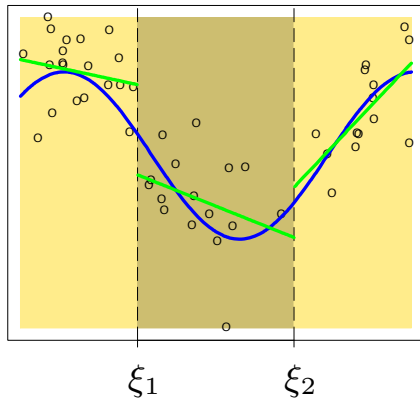
²¹For example, for K -degree polynomial regressions,
 $\phi_1(x) = 1, \phi_2(x) = x, \phi_3(x) = x^2, \dots, \phi_K(x) = x^K$.

Regression Splines

Piecewise Constant



Piecewise Linear



Regression Splines

Oftentimes it is desired that the fitted curve is **continuous** over the range of x , i.e. there should be no jump at the knots.

For piecewise linear regression with one knot (ξ), this means:

$$y = \begin{cases} \alpha_{10} + \alpha_{11}x + e & x < \xi \\ \alpha_{20} + \alpha_{21}(x - \xi) + e & x \geq \xi \end{cases} \quad (22)$$

under the constraint that

$$\alpha_{10} + \alpha_{11}\xi = \alpha_{20} \quad (23)$$

Regression Splines

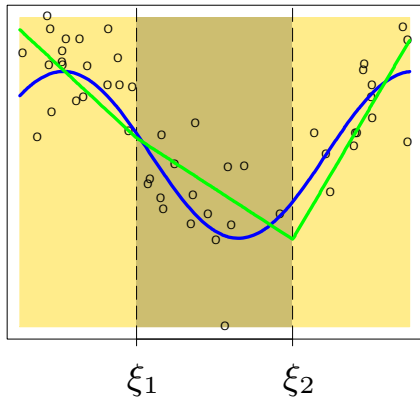
(22) and (23) \Rightarrow the continuous piecewise linear model can be parametrized as

$$y = \beta_0 + \beta_1 x + \beta_2 (x - \xi)_+ + e \quad (24)$$

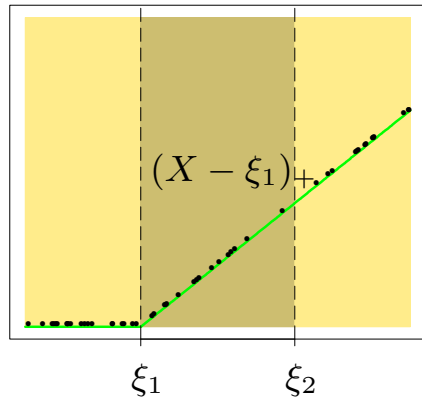
, where $\beta_0 = \alpha_{10}$, $\beta_1 = \alpha_{11}$, $\beta_2 = \alpha_{21} - \alpha_{11}$, and $(x - \xi)_+ \equiv (x - \xi) \mathcal{I}(x \geq \xi)$.

Regression Splines

Continuous Piecewise Linear



Piecewise-linear Basis Function



Regression Splines

For higher-order piecewise polynomial regression, in addition to the fitted curve being continuous, we may also want it to be **smooth** by requiring the derivatives of the piecewise polynomials to be also continuous at the knots.

For piecewise cubic polynomial regression with one knot (ξ), this means:

$$y = \begin{cases} \alpha_{10} + \alpha_{11}x + \alpha_{12}x^2 + \alpha_{13}x^3 + e & x < \xi \\ \alpha_{20} + \alpha_{21}(x - \xi) + \alpha_{22}(x - \xi)^2 + \alpha_{23}(x - \xi)^3 + e & x \geq \xi \end{cases} \quad (25)$$

subject to the constraints that the piecewise polynomials as well as their 1st and 2nd derivatives are continuous at ξ :

$$\begin{aligned} \alpha_{10} + \alpha_{11}\xi + \alpha_{12}\xi^2 + \alpha_{13}\xi^3 &= \alpha_{20} \\ \alpha_{11} + 2\alpha_{12}\xi + 3\alpha_{13}\xi^2 &= \alpha_{21} \\ \alpha_{12} + 3\alpha_{13}\xi &= \alpha_{22} \end{aligned} \quad (26)$$

Regression Splines

(25) and (26) \Rightarrow

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi)_+^3 + e \quad (27)$$

, where $\beta_0 = \alpha_{10}$, $\beta_1 = \alpha_{11}$, $\beta_2 = \alpha_{12}$, $\beta_3 = \alpha_{13}$, and $\beta_4 = \alpha_{23} - \alpha_{13}$.

Regression Splines

(24) and (27) are examples of **regression splines**. (24) is called a *linear spline* and (27) is called a *cubic spline*.

Regression Spline

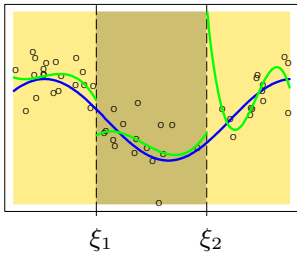
A degree- d spline is a piecewise degree- d polynomial, with continuity in derivatives up to degree $d - 1$ at each knot.

- In general, a degree- d spline with $M - 1$ knots has $d + M$ degrees of freedom²².

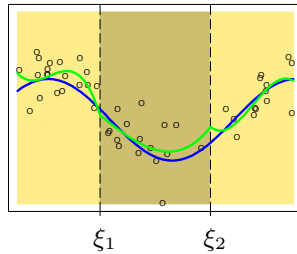
²²For example, a linear spline has $1 + M$ degrees of freedom (see (24)). A cubic spline has $3 + M$ degrees of freedom (see (27)). In comparison, a degree- d polynomial has $d + 1$ degrees of freedom.

Piecewise Cubic Polynomials

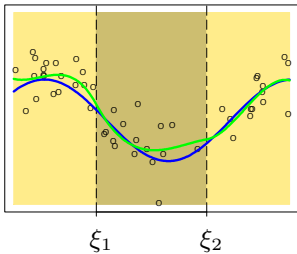
Discontinuous



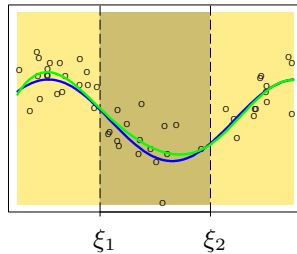
Continuous



Continuous First Derivative



Continuous Second Derivative



Natural Splines

- Splines tend to have high variance at the boundary ($x < \xi_1$ or $x \geq \xi_{M-1}$, where $M - 1$ is the total number of knots).
- A **natural spline** is a regression spline with additional boundary constraints: the function is required to be *linear* beyond the boundary knots, in order to produce more stable estimates.

Wage Profile

```
require(splines)

# Cubic Spline
# -----
# bs() generates B-spline basis functions with specified degrees
# of polynomials and knots
fit <- lm(wage ~ bs(age,knots=c(25,40,60),degree=3), data=Wage)
# knots at age 25,40,60

# Natural Cubic Spline
# -----
# ns() fits a natural cubic spline
fit2 <- lm(wage ~ ns(age,knots=c(25,40,60)))
```

Wage Profile

```
coeftest(fit)
```

```
##
```

```
## t test of coefficients:
```

```
##
```

```
##
```

```
## (Intercept)
```

```
Estimate Std. Error t value
```

```
60.4937 9.4604 6.3944
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)1 3.9805 12.5376 0.3175
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)2 44.6310 9.6263 4.6364
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)3 62.8388 10.7552 5.8426
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)4 55.9908 10.7063 5.2297
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)5 50.6881 14.4018 3.5196
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)6 16.6061 19.1264 0.8682
```

```
##
```

```
Pr(>|t|)
```

```
## (Intercept)
```

```
1.863e-10 ***
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)1 0.7508987
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)2 3.698e-06 ***
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)3 5.691e-09 ***
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)4 1.815e-07 ***
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)5 0.0004387 ***
```

```
## bs(age, knots = c(25, 40, 60), degree = 3)6 0.3853380
```

Wage Profile

```
coeftest(fit2)
```

```
##
```

```
## t test of coefficients:
```

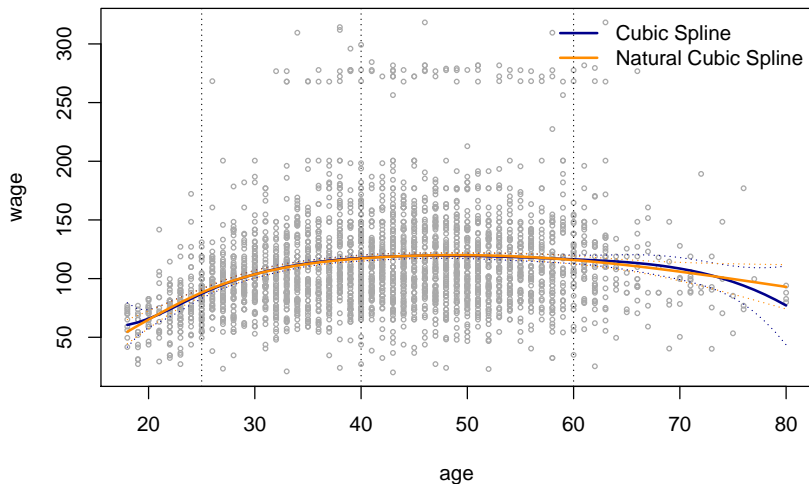
```
##
```

	Estimate	Std. Error	t value	Pr(> t)	
## (Intercept)	54.7595	5.1378	10.6581	< 2.2e-16	**
## ns(age, knots = c(25, 40, 60))1	67.4019	5.0134	13.4442	< 2.2e-16	**
## ns(age, knots = c(25, 40, 60))2	51.3828	5.7115	8.9964	< 2.2e-16	**
## ns(age, knots = c(25, 40, 60))3	88.5661	12.0156	7.3709	2.181e-13	**
## ns(age, knots = c(25, 40, 60))4	10.6369	9.8332	1.0817	0.2795	
## ---					

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Wage Profile

Cubic and Natural Cubic Spline



Generalized Additive Models

So far we have been dealing with a single input x in our discussion of polynomial regression and regression splines. A natural way to extend this discussion to multiple inputs is to assume the following model:

$$y = \omega_0 + \omega_1(x_1) + \omega_2(x_2) + \cdots \omega_p(x_p) + e \quad (28)$$

, where

$$\omega_j(x_j) = \sum_{m=1}^{M_j} \beta_{jm} \phi_{jm}(x_j)$$

(28) is called a **generalized additive model (GAM)**.

Generalized Additive Models

The GAM allows for flexible nonlinear relationships in each dimension of the input space while maintaining the additive structure of linear models.

- For example, we can fit a linear relationship in x_1 , a polynomial in x_2 , a cubic spline in x_3 , etc.
- The GAM remains a linear basis function model and therefore can be fit by least squares²³.

²³(28) is equivalent to

$$y = \beta' \Phi(x) + e$$

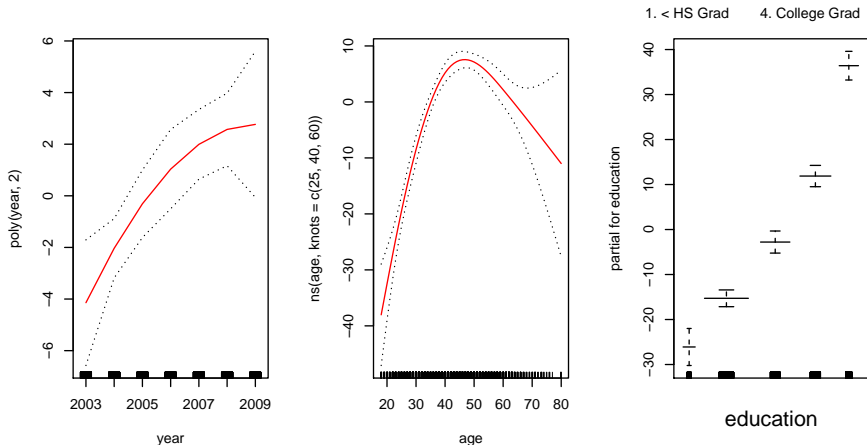
, where $\Phi = (\mathbf{1}, \phi_{11}, \dots, \phi_{1M_1}, \dots, \phi_{p1}, \dots, \phi_{pM_p})'$.

Wage Profile

```
fit <- lm(wage ~ poly(year,2) + ns(age,knots=c(25,40,60)) + education)
coeftest(fit)

##
## t test of coefficients:
##
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)      47.5751      4.8992  9.7108 < 2.2e-16 **
## poly(year, 2)1    130.4942     35.2930  3.6974 0.0002217 **
## poly(year, 2)2    -36.3005     35.2579 -1.0296 0.3032959
## ns(age, knots = c(25, 40, 60))1  51.1072      4.4572 11.4662 < 2.2e-16 **
## ns(age, knots = c(25, 40, 60))2  33.1989      5.0767  6.5394 7.237e-11 **
## ns(age, knots = c(25, 40, 60))3  53.5004     10.6621  5.0178 5.532e-07 **
## ns(age, knots = c(25, 40, 60))4  12.3733      8.6866  1.4244 0.1544320
## education2. HS Grad      10.8174      2.4305  4.4507 8.871e-06 **
## education3. Some College    23.3191      2.5626  9.0997 < 2.2e-16 **
## education4. College Grad    37.9867      2.5464 14.9176 < 2.2e-16 **
## education5. Advanced Degree  62.5184      2.7629 22.6275 < 2.2e-16 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Wage Profile



A GAM model of wage with a quadratic polynomial in year, a natural cubic spline in age, and a step function in education

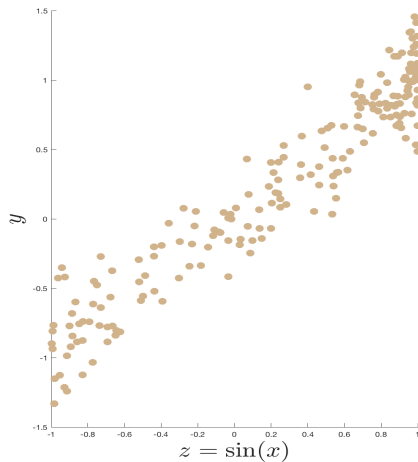
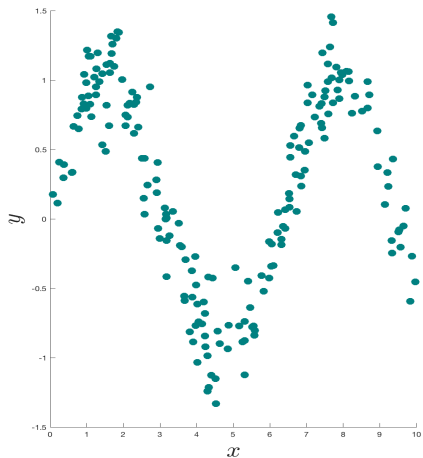
Generalization Issues

Fitting a linear basis function model (21) can be thought of as a two-step process:

- 1 Transform x into $\Phi(x)$ ²⁴.
 - Let $z = \Phi(x) \in \mathcal{Z}$. $\Phi: \mathcal{X} \rightarrow \mathcal{Z}$ is called a **feature transform**.
- 2 Fit the linear model: $\mathcal{H}_\Phi = \{h: h(z) = \beta'z\}$, where \mathcal{H}_Φ denotes the hypothesis set corresponding to the feature transform Φ .

²⁴ x can be multi-dimensional: $x = (x_1, \dots, x_p)$

Feature Transform



Left: data in \mathcal{X} -space; Right: data in \mathcal{Z} -space

Generalization Issues

If we decide on the feature transform Φ *before* seeing the data, then the VC generalization bound holds with $d_{VC}(\mathcal{H}_\Phi)$ as the VC dimension.

I.e., for any $g \in \mathcal{H}_\Phi$, with probability at least $1 - \delta$,

$$\begin{aligned} E_{out}(g) &\leq E_{in}(g) + \sqrt{\frac{8}{N} \ln \frac{4 \left((2N)^{d_{VC}} + 1 \right)}{\delta}} \\ &= E_{in}(g) + \mathcal{O} \left(\sqrt{\frac{d_{VC}}{N} \ln N} \right) \end{aligned} \tag{29}$$

, where $d_{VC} = d_{VC}(\mathcal{H}_\Phi)$.

Generalization Issues

Therefore, when choosing a high-order polynomial, or a spline with many degrees of freedom, or a GAM with complex nonlinearities in many dimensions, we cannot avoid the approximation-generalization tradeoff:

- More complex \mathcal{H}_Φ ($d_{VC}(\mathcal{H}_\Phi) \uparrow$) $\Rightarrow E_{in} \downarrow$
- Less complex \mathcal{H}_Φ ($d_{VC}(\mathcal{H}_\Phi) \downarrow$) $\Rightarrow |E_{out} - E_{in}| \downarrow$

Generalization Issues

What if we try a transformation Φ_1 first, and then, finding the results unsatisfactory, decide to use Φ_2 ? Then we are effectively using a model that contains both $\{\beta' \Phi_1(x)\}$ and $\{\beta' \Phi_2(x)\}$.

- For example, if we try a linear model first, then a quadratic polynomial, then a piecewise constant model, before settling on a cubic spline, then d_{VC} in (29) should be the VC dimension of a hypothesis set that contains not only the cubic spline model, but all of the aforementioned models.
- The process of trying a series of models until we get a satisfactory result is called **specification search** or **data snooping**. In general, the more models you try, the poorer your final result will generalize out of sample.

Acknowledgement I

Part of this lecture is based on the following sources:

- Gramacy, R. B. *Applied Regression Analysis*. Lecture at the University of Chicago Booth School of Business, retrieved on 2017.01.01. [[link](#)]
- Hastie, T., R. Tibshirani, and J. Friedmand. 2008. *The Elements of Statistical Learning* (2nd ed.). Springer.
- James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning: with Applications in R*. Springer.
- Ng, A. *Machine Learning*. Lecture at Stanford University, retrieved on 2017.01.01. [[link](#)]
- Penn State University. *Probability Theory and Mathematical Statistics*. Online course, retrieved on 2017.01.01. [[link](#)]
- Shalizi, C. R. 2019. *Advanced Data Analysis from an Elementary Point of View*. Manuscript.

Acknowledgement II

- Taddy, M. *Big Data*. Lecture at the University of Chicago Booth School of Business, retrieved on 2017.01.01. [[link](#)]