Regression with Polynomials and Interactions

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics
University of Minnesota (Twin Cities)

Updated 04-Jan-2017
Copyright © 2017 by Nathaniel E. Helwig
Outline of Notes

1) Polynomial Regression:
   - Polynomials review
   - Model form
   - Model assumptions
   - Ordinary least squares
   - Orthogonal polynomials
   - Example: MPG vs HP

2) Interactions in Regression:
   - Overview
   - Nominal*Continuous
   - Example #1: Real Estate
   - Example #2: Depression
   - Continuous*Continuous
   - Example #3: Oceanography
Polynomial Regression
Polynomial Function: Definition

Reminder: a polynomial function has the form

\[ f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n \]
\[ = \sum_{j=0}^{n} a_j x^j \]

where \( a_j \in \mathbb{R} \) are the coefficients and \( x \) is the indeterminate (variable).

Note: \( x^j \) is the \( j \)-th order polynomial term

- \( x \) is first order term, \( x^2 \) is second order term, etc.
- The degree of a polynomial is the highest order term
Polynomial Function: Simple Regression

```r
> x = seq(-1, 1, length=50)
> y = 2 + 2 * (x^2)
> plot(x, y, main="Quadratic")
> qmod = lm(y ~ x)
> abline(qmod)
```

```r
> x = seq(-1, 1, length=50)
> y = 2 + 2 * (x^3)
> plot(x, y, main="Cubic")
> cmod = lm(y ~ x)
> abline(cmod)
```
The polynomial regression model has the form

\[ y_i = b_0 + \sum_{j=1}^{p} b_j x_i^j + e_i \]

for \( i \in \{1, \ldots, n\} \) where

- \( y_i \in \mathbb{R} \) is the real-valued response for the \( i \)-th observation
- \( b_0 \in \mathbb{R} \) is the regression intercept
- \( b_j \in \mathbb{R} \) is the regression slope for the \( j \)-th degree polynomial
- \( x_i \in \mathbb{R} \) is the predictor for the \( i \)-th observation
- \( e_i \overset{iid}{\sim} N(0, \sigma^2) \) is a Gaussian error term
The polynomial regression model has the form

$$y = Xb + e$$

or

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^p \\ 1 & x_2 & x_2^2 & \cdots & x_2^p \\ 1 & x_3 & x_3^2 & \cdots & x_3^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^p \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{pmatrix}$$

Note that this is still a linear model, even though we have polynomial terms in the design matrix.
PR Model Assumptions (scalar)

The fundamental assumptions of the PR model are:

1. Relationship between \( X \) and \( Y \) is polynomial
2. \( x_i \) and \( y_i \) are observed random variables (known constants)
3. \( e_i \overset{iid}{\sim} N(0, \sigma^2) \) is an unobserved random variable
4. \( b_0, b_1, \ldots, b_p \) are unknown constants
5. \( (y_i|x_i) \overset{ind}{\sim} N(b_0 + \sum_{j=1}^p b_j x_i^j, \sigma^2) \)

Note: focus is estimation of the polynomial curve.
In matrix terms, the error vector is multivariate normal:

\[ e \sim N(0_n, \sigma^2 I_n) \]

In matrix terms, the response vector is multivariate normal given \( X \):

\[ (y|X) \sim N(Xb, \sigma^2 I_n) \]
Some important properties of the PR model include:

1. Need $n > p$ to fit the polynomial regression model
2. Setting $p = 1$ produces simple linear regression
3. Setting $p = 2$ is quadratic polynomial regression
4. Setting $p = 3$ is cubic polynomial regression
5. Rarely set $p > 3$; use cubic spline instead
The ordinary least squares (OLS) problem is

$$\min_{\mathbf{b} \in \mathbb{R}^{p+1}} \| \mathbf{y} - \mathbf{Xb} \|^2$$

where $\| \cdot \|$ denotes the Frobenius norm.

The OLS solution has the form

$$\hat{\mathbf{b}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

which is the same formula from SLR and MLR!
Fitted Values and Residuals

SCALAR FORM:

Fitted values are given by

\[ \hat{y}_i = \hat{b}_0 + \sum_{j=1}^{p} \hat{b}_j x_i^j \]

and residuals are given by

\[ \hat{e}_i = y_i - \hat{y}_i \]

MATRIX FORM:

Fitted values are given by

\[ \hat{y} = X\hat{b} = Hy \]

and residuals are given by

\[ \hat{e} = y - \hat{y} = (I_n - H)y \]
The estimated error variance is

\[ \hat{\sigma}^2 = \frac{SSE}{(n - p - 1)} \]

\[ = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / (n - p - 1) \]

\[ = \| (I_n - H) y \|^2 / (n - p - 1) \]

which is an unbiased estimate of error variance \( \sigma^2 \).

The estimate \( \hat{\sigma}^2 \) is the **mean squared error (MSE)** of the model.
Distribution of Estimator, Fitted Values, and Residuals

Just like in SLR and MLR, the PR assumptions imply that

\[ \hat{b} \sim N(b, \sigma^2 (X'X)^{-1}) \]

\[ \hat{y} \sim N(Xb, \sigma^2 H) \]

\[ \hat{e} \sim N(0, \sigma^2 (I_n - H)) \]

Typically \( \sigma^2 \) is unknown, so we use the MSE \( \hat{\sigma}^2 \) in practice.
Multicollinearity: Problem

Note that $x_i$, $x_i^2$, $x_i^3$, etc. can be highly correlated with one another, which introduces multicollinearity problem.

```r
> set.seed(123)
> x = runif(100)*2
> X = cbind(x, xsq=x^2, xcu=x^3)
> cor(X)

          x     xsq      xcu
x 1.000000 0.970308 0.9210726
xsq 0.970308 1.000000 0.9866033
xcu 0.921072 0.986603 1.0000000
```
Multicollinearity: Partial Solution

You could mean-center the $x_i$ terms to reduce multicollinearity.

```r
> set.seed(123)
> x = runif(100) * 2
> x = x - mean(x)
> X = cbind(x, xsq=x^2, xcu=x^3)
> cor(X)
```

```
          x  xsq  xcu
   x 1.00000000 0.03854803 0.91479660
xsq 0.03854803 1.00000000 0.04400704
xcu 0.91479660 0.04400704 1.00000000
```

But this doesn’t fully solve our problem...
Orthogonal Polynomials: Definition

To deal with multicollinearity, define the set of variables

\[ z_0 = a_0 \]
\[ z_1 = a_1 + b_1 x \]
\[ z_2 = a_2 + b_2 x + c_2 x^2 \]
\[ z_3 = a_3 + b_3 x + c_3 x^2 + d_3 x^3 \]

where the coefficients are chosen so that \( z_j'z_k = 0 \) for all \( j \neq k \).

The transformed \( z_j \) variables are called orthogonal polynomials.
The orthogonal projection of a vector $\mathbf{v} = \{v_i\}_{n \times 1}$ on to the line spanned by the vector $\mathbf{u} = \{u_i\}_{n \times 1}$ is

$$\text{proj}_u(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u}$$

where $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}'\mathbf{y}$ and $\langle \mathbf{u}, \mathbf{u} \rangle = \mathbf{u}'\mathbf{u}$ denote the inner products.

http://thejuniverse.org/PUBLIC/LinearAlgebra/LOLA/dotProd/proj.html
Orthogonal Polynomials: Gram-Schmidt

Can use the **Gram-Schmidt process** to form orthogonal polynomials.

Start with a linearly independent design matrix $X = [x_0, x_1, x_2, x_3]$ where $x_j = (x^j_1, \ldots, x^j_n)'$ is the $j$-th order polynomial vector.

Gram-Schmidt algorithm to form columnwise orthogonal matrix $Z$ that spans the same column space as $X$:

- $z_0 = x_0$
- $z_1 = x_1 - \text{proj}_{z_0}(x_1)$
- $z_2 = x_2 - \text{proj}_{z_0}(x_2) - \text{proj}_{z_1}(x_2)$
- $z_3 = x_3 - \text{proj}_{z_0}(x_3) - \text{proj}_{z_1}(x_3) - \text{proj}_{z_2}(x_3)$
Orthogonal Polynomials: R Functions

Simple R function to orthogonalize an input matrix:

```R
orthog <- function(X, normalize=FALSE){
  np = dim(X)
  Z = matrix(0, np[1], np[2])
  Z[,1] = X[,1]
  for(k in 2:np[2]){
    Z[,k] = X[,k]
    for(j in 1:(k-1)){
      Z[,k] = Z[,k] - Z[,j]*sum(Z[,k]*Z[,j]) / sum(Z[,j]^2)
    }
  }
  if(normalize){ Z = Z %*% diag(colSums(Z^2)^-0.5) }
  Z
}
```

Nathaniel E. Helwig (U of Minnesota)
> set.seed(123)
> X = cbind(1, runif(10), runif(10))
> crossprod(X)

<table>
<thead>
<tr>
<th>[,1]</th>
<th>[,2]</th>
<th>[,3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,] 10.000000</td>
<td>5.782475</td>
<td>5.233693</td>
</tr>
<tr>
<td>[2,] 5.782475</td>
<td>4.125547</td>
<td>2.337238</td>
</tr>
<tr>
<td>[3,] 5.233693</td>
<td>2.337238</td>
<td>3.809269</td>
</tr>
</tbody>
</table>

> Z = orthog(X)
> crossprod(Z)

<table>
<thead>
<tr>
<th>[,1]</th>
<th>[,2]</th>
<th>[,3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,] 1.000000e+00</td>
<td>-4.440892e-16</td>
<td>-4.440892e-16</td>
</tr>
<tr>
<td>[2,] -4.440892e-16</td>
<td>7.818448e-01</td>
<td>-1.387779e-17</td>
</tr>
<tr>
<td>[3,] -4.440892e-16</td>
<td>-1.387779e-17</td>
<td>4.627017e-01</td>
</tr>
</tbody>
</table>

> Z = orthog(X, norm=TRUE)
> crossprod(Z)

<table>
<thead>
<tr>
<th>[,1]</th>
<th>[,2]</th>
<th>[,3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,] 1.000000e+00</td>
<td>-1.942890e-16</td>
<td>-2.220446e-16</td>
</tr>
<tr>
<td>[2,] -1.942890e-16</td>
<td>1.000000e+00</td>
<td>1.387779e-17</td>
</tr>
<tr>
<td>[3,] -2.220446e-16</td>
<td>1.387779e-17</td>
<td>1.000000e+00</td>
</tr>
</tbody>
</table>
Can also use the default `poly` function in R.

```r
> set.seed(123)
> x = runif(10)
> X = cbind(1, x, xsq=x^2, xcu=x^3)
> Z = orthog(X, norm=TRUE)
> z = poly(x, degree=3)
> Z[,2:4] = Z[,2:4] %*% diag(colSums(z^2)^0.5)
> Z[1:3,]
[1,] 0.3162278 -0.3287304 -0.07537277 0.5363745
[2,] 0.3162278  0.2375627 -0.06651752 -0.5097714
[3,] 0.3162278 -0.1914349 -0.26206273 0.2473705
> cbind(Z[1:3,1],z[1:3,])
          1          2          3
[1,] 0.3162278 -0.3287304 -0.07537277 0.5363745
[2,] 0.3162278  0.2375627 -0.06651752 -0.5097714
[3,] 0.3162278 -0.1914349 -0.26206273 0.2473705
```
Real Polynomial Data

Auto-MPG data from the UCI Machine Learning repository: http://archive.ics.uci.edu/ml/datasets/Auto+MPG

Have variables collected from $n = 398$ cars from years 1970–1982.

- mpg: miles per gallon
- cylinder: number of cylinders
- disp: displacement
- hp: horsepower
- weight: weight
- accel: acceleration
- year: model year
- origin: origin
- name: make and model
Best Linear Relationship

Best linear relationship predicting \( mpg \) from \( hp \).

```r
> plot(hp, mpg)
> linmod = lm(mpg ~ hp)
> abline(linmod)
```
Best Quadratic Relationship

Best quadratic relationship predicting \( \text{mpg} \) from \( \text{hp} \).

```r
> quadmod = lm(mpg ~ hp + I(hp^2))
> hpseq = seq(50, 250, by = 5)
> Xmat = cbind(1, hpseq, hpseq^2)
> hphat = Xmat %*% quadmod$coef
> plot(hp, mpg)
> lines(hpseq, hphat)
```
Best Cubic Relationship

Check for possible cubic relationship:

```r
> cubmod = lm(mpg ~ hp + I(hp^2) + I(hp^3))
> summary(cubmod)
```

Call:
`lm(formula = mpg ~ hp + I(hp^2) + I(hp^3))`

Residuals:
```
  Min     1Q Median     3Q    Max
-14.704 -2.449  0.152  2.204 15.816
```

Coefficients:
```
                Estimate Std. Error t value  Pr(>|t|)
(Intercept)  6.068e+01  4.563e+00 13.298 < 2e-16 ***
hp          -5.689e-01  1.179e-01  -4.824 2.03e-06 ***
I(hp^2)     2.079e-03  9.479e-04   2.193   0.0289 *
I(hp^3)     -2.147e-06  2.378e-06  -0.903   0.3673
```

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.375 on 388 degrees of freedom
(6 observations deleted due to missingness)
Multiple R-squared: 0.6882, Adjusted R-squared: 0.6858
F-statistic: 285.5 on 3 and 388 DF, p-value: < 2.2e-16
Orthogonal versus Raw Polynomials

Compare orthogonal and raw polynomials:

```r
> quadomod = lm(mpg ~ poly(hp, degree=2))
> summary(quadomod)$coef

Call:
  lm(formula = mpg ~ poly(hp, degree = 2))

Residuals:
   Min     1Q   Median     3Q    Max
-12.159  -2.964  -0.796  -0.035  12.687

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             23.446     0.221   106.1  < 2e-16 ***
poly(hp, degree = 2)1  -120.14     4.374   -27.5   4.2e-93 ***
poly(hp, degree = 2)2   44.09      4.374    10.1    2.2e-21 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.922 on 26 degrees of freedom
Multiple R-squared:  0.894, Adjusted R-squared:  0.889
F-statistic:  485 on 2 and 26 DF,  p-value: < 2.2e-16

> summary(quadomod)$coef

Call:
  lm(formula = mpg ~ poly(hp, degree = 2))

Residuals:
   Min     1Q   Median     3Q    Max
-12.159  -2.964  -0.796  -0.035  12.687

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             56.900      1.800   31.6   1.7e-109 ***
hp                      -0.466      0.031  -14.9    2.3e-40 ***
I(hp^2)                 0.0012      0.0001   10.1    2.2e-21 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.922 on 26 degrees of freedom
Multiple R-squared:  0.894, Adjusted R-squared:  0.889
F-statistic:  485 on 2 and 26 DF,  p-value: < 2.2e-16
```
Orthogonal Polynomials from Scratch

We can reproduce the same significance test results using `orthog`:

```r
> widx = which(is.na(hp)==FALSE)
> hp = hp[widx]
> mpg = mpg[widx]
> X = orthog(cbind(1, hp, hp^2))
> quadomod = lm(mpg ~ X[,2] + X[,3])
> summary(quadomod)$coef
```

|                | Estimate   | Std. Error   | t value | Pr(>|t|)      |
|----------------|------------|--------------|---------|---------------|
| (Intercept)    | 23.445918367 | 0.2209163488 | 106.13030 | 2.752212e-289 |
| X[, 2]         | -0.157844733 | 0.0057467395 | -27.46683 | 4.169400e-93  |
| X[, 3]         | 0.001230536  | 0.0001220759 | 10.08009  | 2.196340e-21  |

Nathaniel E. Helwig (U of Minnesota)
Regression with Polynomials and Interactions
Updated 04-Jan-2017 : Slide 29
Interactions in Regression
**Interaction Term: Definition**

MLR model with two predictors and an interaction

\[ y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i1} x_{i2} + e_i \]

where

- \( y_i \in \mathbb{R} \) is the real-valued **response** for the \( i \)-th observation
- \( b_0 \in \mathbb{R} \) is the regression **intercept**
- \( b_1 \in \mathbb{R} \) is the main effect of the first predictor
- \( b_2 \in \mathbb{R} \) is the main effect of the second predictor
- \( b_3 \in \mathbb{R} \) is the **interaction effect**
- \( e_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \) is a Gaussian **error term**
An interaction between $X_1$ and $X_2$ means that the relationship between $X_1$ and $Y$ differs depending on the value of $X_2$ (and vice versa).

Pro: model is more flexible (i.e., we’ve added a parameter)

Con: model is (sometimes) more difficult to interpret.
Nominal Variables

Suppose that \( X \in \{ x_1, \ldots, x_g \} \) is a nominal variable with \( g \) levels.

- Nominal variables are also called categorical variables
- Example: \( \text{sex} \in \{ \text{female}, \text{male} \} \) has two levels
- Example: \( \text{drug} \in \{ A, B, C \} \) has three levels

To code a nominal variable (with \( g \) levels) in a regression model, we need to include \( g - 1 \) different variables in the model.

- Use dummy coding to absorb \( g \)-th level into intercept

\[
    x_{ij} = \begin{cases} 
    1 & \text{if } i\text{-th observation is in } j\text{-th level} \\
    0 & \text{otherwise} 
    \end{cases} 
\]

for \( j \in \{ 1, \ldots, g - 1 \} \)
Nominal Interaction with Two Levels

Revisit the MLR model with two predictors and an interaction

\[ y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i1} x_{i2} + e_i \]

and suppose that \( x_{i2} \in \{0, 1\} \) is a nominal predictor.

If \( x_{i2} = 0 \), the model is:

\[ y_i = b_0 + b_1 x_{i1} + e_i \]

- \( b_0 \) is expected value of \( Y \) when \( x_{i1} = x_{i2} = 0 \)
- \( b_1 \) is expected change in \( Y \) for 1-unit change in \( x_{i1} \) (if \( x_{i2} = 0 \))

If \( x_{i2} = 1 \), the model is:

\[ y_i = (b_0 + b_2) + (b_1 + b_3) x_{i1} + e_i \]

- \( b_0 + b_2 \) is expected value of \( Y \) when \( x_{i1} = 0 \) and \( x_{i2} = 1 \)
- \( b_1 + b_3 \) is expected change in \( Y \) for 1-unit change in \( x_{i1} \) (if \( x_{i2} = 1 \))
Real Estate Data Description

Using house price data from Kutner, Nachtsheim, Neter, and Li (2005).

Have three variables in the data set:

- **price**: price house sells for (thousands of dollars)
- **value**: house value before selling (thousands of dollars)
- **corner**: indicator variable (=1 if house is on corner of block)

Total of $n = 64$ independent observations (16 corners).

Want to predict selling **price** from appraisal **value**, and determine if the relationship depends on the **corner** status of the house.
Real Estate Data Visualization

- Scatter plot showing the relationship between Appraisal Value and Selling Price, with data points labeled as 'Corner' and 'Non-Corner'.
- The plot includes a regression line indicating the trend between the two variables.
Real Estate Data Visualization (R Code)

```r
> house=read.table("~/Desktop/notes/data/houseprice.txt",header=TRUE)
> house[1:3,]
   price value corner
  1  78.8  76.4  0
  2  73.8  74.3  0
  3  64.6  69.6  0
  4  76.2  73.6  0
  5  87.2  76.8  0
  6  70.6  72.7  1
  7  86.0  79.2  0
  8  83.1  75.6  0
> plot(house$value,house$price,pch=ifelse(house$corner==1,0,16),
+ xlab="Appraisal Value",ylab="Selling Price")
> legend("topleft",c("Corner","Non-Corner"),pch=c(0,16),bty="n")
```
Fit model with interaction between \textit{value} and \textit{corner}:

```r
> hmod = lm(price ~ value*corner, data=house)
> hmod
```

**Call:**

```
lm(formula = price ~ value * corner, data = house)
```

**Coefficients:**

```
(Intercept)    value   corner   value:corner
  -126.905     2.776   76.022    -1.107
```
Real Estate Regression: Significance of Terms

```r
> summary(hmod)

Call:
  lm(formula = price ~ value * corner, data = house)

Residuals:
     Min      1Q  Median      3Q     Max
-10.8470 -2.1639  0.0913  1.9348  9.9836

Coefficients:
                       Estimate  Std. Error   t value     Pr(>|t|)
(Intercept)             -126.9052   14.7225   -8.620 4.33e-12 ***
value                   2.7759     0.1963    14.142  < 2e-16 ***
corner                  76.0215    30.1314     2.523  0.01430 *
value:corner            -1.1075     0.4055    -2.731  0.00828 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.893 on 60 degrees of freedom
Multiple R-squared: 0.8233, Adjusted R-squared: 0.8145
F-statistic: 93.21 on 3 and 60 DF,  p-value: < 2.2e-16
```
Real Estate Regression: Interpreting Results

\[
\hat{b}_0 = -126.90 \text{ is expected selling price (in thousands of dollars) for non-corner houses that were valued at } 0.
\]

\[
\hat{b}_0 + \hat{b}_2 = -126.90 + 76.022 = -50.878 \text{ is expected selling price (in thousands of dollars) for corner houses that were valued at } 0.
\]

\[
\hat{b}_1 = 2.776 \text{ is the expected increase in selling price (in thousands of dollars) corresponding to a 1-unit ($1,000) increase in appraisal value for non-corner houses}
\]

\[
\hat{b}_1 + \hat{b}_3 = 2.775 - 1.107 = 1.668 \text{ is the expected increase in selling price (in thousands of dollars) corresponding to a 1-unit ($1,000) increase in appraisal value for corner houses}
\]
Real Estate Regression: Plotting Results
Real Estate Regression: Plotting Results (R Code)

```r
> plot(house$value, house$price, pch=ifelse(house$corner==1,0,16),
+       xlab="Appraisal Value", ylab="Selling Price")
> abline(hmod$coef[1],hmod$coef[2])
> legend("topleft",c("Corner","Non-Corner"),lty=2:1,pch=c(0,16),bty="n")
```

Note that if you input the (appropriate) coefficients, you can still use the `abline` function to draw the regression lines.
Depression Data Description


Total of $n = 36$ subjects participated in a depression study.

Have three variables in the data set:

- **effect**: effectiveness of depression treatment (high=effective)
- **age**: age of the participant (in years)
- **method**: method of treatment (3 levels: A, B, C)

Predict effectiveness from participant’s age and treatment method.
Depression Data Visualization

- Age
- Effect
- A
- B
- C

Data points are labeled with letters A, B, and C. The graph shows a trend where the effect increases with age, and the interaction between age and the effect appears to be significant, especially at higher ages.
Depression Data Visualization (R Code)

```r
> depression = read.table("~/Desktop/notes/data/depression.txt", header=TRUE)
> depression[1:8,]
  effect age method
1      56  21     A
2      41  23     B
3      40  30     B
4      28  19     C
5      55  28     A
6      25  23     C
7      46  33     B
8      71  67     C
> plot(depression$age, depression$effect, xlab="Age", ylab="Effect", type="n")
> text(depression$age, depression$effect, depression$method)
```
Fit model with interaction between \texttt{age} and \texttt{method}

\begin{verbatim}
> dmod = lm(effect ~ age*method, data=depression)
> dmod$coef

(Intercept)      age methodB methodC age:methodB age:methodC
47.5155913 0.3305073 -18.5973852 -41.3042101 0.1931769 0.7028836
\end{verbatim}

Note that R creates two indicator variables for \texttt{method}:

- \( x_{iB} = \begin{cases} 
  1 & \text{if } i\text{-th observation is in treatment method B} \\
  0 & \text{otherwise} 
\end{cases} \)

- \( x_{iC} = \begin{cases} 
  1 & \text{if } i\text{-th observation is in treatment method C} \\
  0 & \text{otherwise} 
\end{cases} \)
Depression Regression: Significance of Terms

```r
> summary(dmod)

Call:
  lm(formula = effect ~ age * method, data = depression)

Residuals:
   Min     1Q  Median     3Q    Max
-6.4366 -2.7637  0.1887  2.9075  6.5634

Coefficients:
                    Estimate  Std. Error   t value  Pr(>|t|)
(Intercept)     47.51559     3.82523  12.4223     2.34e-13 ***
age              0.33051     0.08149   4.0563     0.000328 ***
methodB        -18.59739     5.41573  -3.4340     0.001759 **
methodC        -41.30421     5.08453  -8.1245     4.56e-09 ***
age:methodB     0.19318     0.11660   1.6570     0.108001
age:methodC     0.70288     0.10896   6.4513     3.98e-07 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.925 on 30 degrees of freedom
Multiple R-squared:  0.9143, Adjusted R-squared:  0.9001
F-statistic: 64.04 on 5 and 30 DF,  p-value: 4.264e-15
```
Depression Regression: Interpreting Results

\[ \hat{b}_0 = 47.516 \] is expected treatment effectiveness for subjects in method A who are 0 y/o.

\[ \hat{b}_0 + \hat{b}_2 = 47.516 - 18.598 = 28.918 \] is expected treatment effectiveness for subjects in method B who are 0 years old.

\[ \hat{b}_0 + \hat{b}_3 = 47.516 - 41.304 = 6.212 \] is expected treatment effectiveness for subjects in method C who are 0 years old.

\[ \hat{b}_1 = 0.331 \] is the expected increase in treatment effectiveness corresponding to a 1-unit (1 year) increase in age for treatment method A.

\[ \hat{b}_1 + \hat{b}_4 = 0.331 + 0.193 = 0.524 \] is the expected increase in treatment effectiveness corresponding to a 1-unit (1 year) increase in age for treatment method B.

\[ \hat{b}_1 + \hat{b}_5 = 0.331 + 0.703 = 1.033 \] is the expected increase in treatment effectiveness corresponding to a 1-unit (1 year) increase in age for treatment method C.
Depression Regression: Plotting Results

Interactions in Regression

Depression Example

Nathaniel E. Helwig (U of Minnesota)
Depression Regression: Plotting Results

```r
> plot(depression$age, depression$effect, xlab="Age", ylab="Effect", type="n")
> text(depression$age, depression$effect, depression$method)
> abline(dmod$coef[1], dmod$coef[2])
> abline(dmod$coef[1] + dmod$coef[3], dmod$coef[2] + dmod$coef[5], lty=2)
> legend("bottomright", c("A", "B", "C"), lty=1:3, bty="n")
```
Interactions between Continuous Variables

Revisit the MLR model with two predictors and an interaction

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i1} x_{i2} + e_i$$

and suppose that $x_{i1}, x_{i2} \in \mathbb{R}$ are both continuous predictors.

In this case, the model terms can be interpreted as:

- $b_0$ is expected value of $Y$ when $x_{i1} = x_{i2} = 0$
- $b_1 + b_3 x_{i2}$ is expected change in $Y$ corresponding to 1-unit change in $x_{i1}$ holding $x_{i2}$ fixed (i.e., conditioning on $x_{i2}$)
- $b_2 + b_3 x_{i1}$ is expected change in $Y$ corresponding to 1-unit change in $x_{i2}$ holding $x_{i1}$ fixed (i.e., conditioning on $x_{i1}$)
Visualizing Continuous*Continuous Interactions

Simple regression

Multiple regression (additive)

Multiple regression (interaction)
Oceanography Data Description

- Data originally from TAO project: [http://www.pmel.noaa.gov/tao/](http://www.pmel.noaa.gov/tao/)
- Note that I have preprocessed the data a bit before analysis.
Buoys collect lots of different data:

```r
elnino[1:4,]
```

<table>
<thead>
<tr>
<th>obs</th>
<th>year</th>
<th>month</th>
<th>day</th>
<th>date</th>
<th>latitude</th>
<th>longitude</th>
<th>zon.winds</th>
<th>mer.winds</th>
<th>humidity</th>
<th>air.temp</th>
<th>ss.temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>4297</td>
<td>94</td>
<td>1</td>
<td>1</td>
<td>940101</td>
<td>-0.01</td>
<td>250.01</td>
<td>-4.3</td>
<td>2.6</td>
<td>89.9</td>
<td>23.21</td>
<td>23.37</td>
</tr>
<tr>
<td>4298</td>
<td>94</td>
<td>1</td>
<td>2</td>
<td>940102</td>
<td>-0.01</td>
<td>250.01</td>
<td>-4.1</td>
<td>1</td>
<td>90</td>
<td>23.16</td>
<td>23.45</td>
</tr>
<tr>
<td>4299</td>
<td>94</td>
<td>1</td>
<td>3</td>
<td>940103</td>
<td>-0.01</td>
<td>250.01</td>
<td>-3</td>
<td>1.6</td>
<td>87.7</td>
<td>23.14</td>
<td>23.71</td>
</tr>
<tr>
<td>4300</td>
<td>94</td>
<td>1</td>
<td>4</td>
<td>940104</td>
<td>0.00</td>
<td>250.00</td>
<td>-3</td>
<td>2.9</td>
<td>85.8</td>
<td>23.39</td>
<td>24.29</td>
</tr>
</tbody>
</table>

We will focus on predicting the sea surface temperatures \( \text{ss.temp} \) from the latitude and longitude locations of the buoys.
Oceanography Regression: Fit Additive Model

Fit additive model of latitude and longitude

> eladd = lm(ss.temp ~ latitude + longitude, data=elnino)
> summary(eladd)

Call:
  lm(formula = ss.temp ~ latitude + longitude, data = elnino)

Residuals:
  Min 1Q Median 3Q Max
-7.6055 -0.7229 0.1261 0.9039 5.0987

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.2636388 0.0305722 1153.45 <2e-16 ***
latitude 0.0257867 0.0010006 25.77 <2e-16 ***
longitude -0.0357496 0.0001445 -247.33 <2e-16 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.48 on 86498 degrees of freedom
Multiple R-squared:  0.4184, Adjusted R-squared:  0.4184
F-statistic: 3.112e+04 on 2 and 86498 DF,  p-value: < 2.2e-16
Oceanography Regression: Fit Interaction Model

Fit model with interaction between \texttt{latitude} and \texttt{longitude}

```r
> elint = lm(ss.temp ~ latitude*longitude, data=elnino)
> summary(elint)
```

Call:
\texttt{lm(formula = ss.temp ~ latitude * longitude, data = elnino)}

Residuals:
\begin{tabular}{rrrrr}
Min & 1Q & Median & 3Q & Max \\
-7.5867 & -0.6496 & 0.1000 & 0.8127 & 5.0922 \\
\end{tabular}

Coefficients:
\begin{tabular}{cccccc}
Estimate & Std. Error & t value & Pr(>|t|) \\
(Intercept) & 3.541e+01 & 2.913e-02 & 1215.61 & <2e-16 *** \\
latitude & -5.245e-01 & 5.862e-03 & -89.47 & <2e-16 *** \\
longitude & -3.638e-02 & 1.377e-04 & -264.22 & <2e-16 *** \\
latitude:longitude & 2.618e-03 & 2.752e-05 & 95.13 & <2e-16 *** \\
\end{tabular}

Signif. codes: \texttt{0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1}

Residual standard error: 1.408 on 86497 degrees of freedom
Multiple R-squared: 0.4735, Adjusted R-squared: 0.4735
F-statistic: 2.593e+04 on 3 and 86497 DF, p-value: < 2.2e-16
Oceanography Regression: Visualize Results

Additive Prediction

Interaction Prediction

Longitude

Latitude

Longitude

Latitude
Oceanography Regression: Visualize (R Code)

```r
newdata = expand.grid(longitude = seq(min(elnino$longitude), max(elnino$longitude), length=50),
                      latitude = seq(min(elnino$latitude), max(elnino$latitude), length=50))
yadd = predict(eladd, newdata)
image(seq(min(elnino$longitude), max(elnino$longitude), length=50),
     seq(min(elnino$latitude), max(elnino$latitude), length=50),
     matrix(yadd, 50, 50), col=rev(rainbow(100, end=3/4)),
     xlab="Longitude", ylab="Latitude", main="Additive Prediction")
yint = predict(elint, newdata)
image(seq(min(elnino$longitude), max(elnino$longitude), length=50),
     seq(min(elnino$latitude), max(elnino$latitude), length=50),
     matrix(yint, 50, 50), col=rev(rainbow(100, end=3/4)),
     xlab="Longitude", ylab="Latitude", main="Interaction Prediction")
```
Oceanography Regression: Smoothing Spline Solution

a) Buoy Locations

b) Main Effect: Temporal

c) Main Effect: Spatial