

# Linear Regression Models

## P8111

Lecture 06

Jeff Goldsmith  
February 9, 2016



THE DEPARTMENT OF  
**BIostatISTICS**



Columbia University  
MAILMAN SCHOOL  
OF PUBLIC HEALTH

# Today's lecture

- Multiple Linear Regression

- Assumptions

- Interpretation

- Some models

*categorical  
interactions*

# Motivation

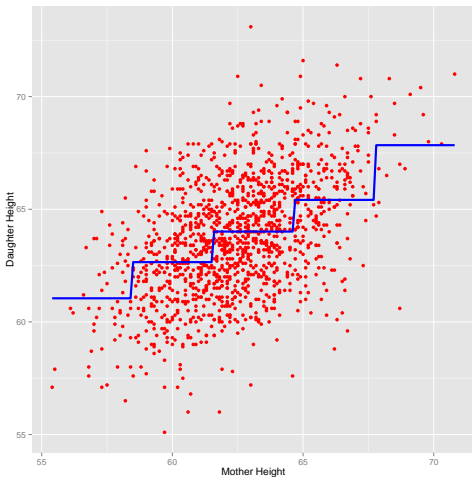
Most applications involve more than one covariate – if more than one thing can influence an outcome, you need multiple linear regression.

- ✓ ■ Improved description of  $y|x$   $E(g(x))$
- ✓ ■ More accurate estimates and predictions
  - Allow testing of multiple effects
  - Includes multiple predictor types

# Why not bin all predictors?

- Divide  $x_i$  into  $k_i$  bins
- Stratify data based on inclusion in bins across  $x$ 's
- Find mean of the  $y_i$  in each category
- Possibly a reasonable non-parametric model

# Why not bin all predictors?



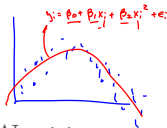
# Why not bin all predictors?

- More predictors = more bins
- If each  $x$  has 5 bins, you have  $5^p$  overall categories
- May not have enough data to estimate distribution in each category
- Curse of dimensionality is a problem in a lot of non-parametric statistics

# Multiple linear regression model

$$E(y|x) = f(x; \beta)$$

$$y = x\beta + \epsilon$$



- Observe data  $(y_i, x_{i1}, \dots, x_{ip})$  for subjects  $1, \dots, n$ . Want to estimate  $\beta_0, \beta_1, \dots, \beta_p$  in the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \epsilon_i; \epsilon_i \stackrel{iid}{\sim} (0, \sigma^2)$$

- Assumptions (residuals have mean zero, constant variance, are independent) are as in SLR
- ★ ■ Impose linearity which (as in the SLR) is a big assumption
- Our primary interest will be  $E(y|x)$
- Eventually estimate model parameters using least squares

$$\hat{\beta}??$$

# Predictor types

- Continuous
- Categorical
- Ordinal





# Interpretation of coefficients

$$\beta_0 = E(y|x_1 = 0, \dots, x_p = 0)$$

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

- Centering some of the  $x$ 's may make this more interpretable

# Interpretation of coefficients

$$\beta_1 = \underbrace{(\beta_0 + \beta_1)}_{\text{at } x_1=1} - \underbrace{(\beta_0)}_{\text{at } x_1=0}$$

$$= (\beta_0 + \beta_1 + \underset{\uparrow}{17}\beta_2 + \underset{\uparrow}{43}\beta_3 \dots) - (\beta_0 + \beta_2 \underset{\uparrow}{17} + \beta_3 \dots \underset{\uparrow}{43})$$

$$E(y | x_1=1, x_2=\underset{\uparrow}{17} \dots) - E(y | x_1=0, x_2=\underset{\uparrow}{17} \dots)$$

$\beta_1$  = the diff in  $E(y)$  for a 1-unit  $\Delta x_1$ ,

keeping everything else fixed!

## Example with two predictors

$$E(y | x_1 = 10, x_2 = 0) \\ x_2 = 1)$$

Suppose we want to regress weight on age and sex.

- Model is  $y_i = \beta_0 + \beta_1 x_{i,age} + \beta_2 x_{i,sex} + \epsilon_i$
- Age is continuous starting with age 0; sex is binary, coded so that  $x_{i,sex} = 0$  for men and  $x_{i,sex} = 1$  for women
  - ▶ In your dataset, sex should be a factor variable ...

## Example with two predictors

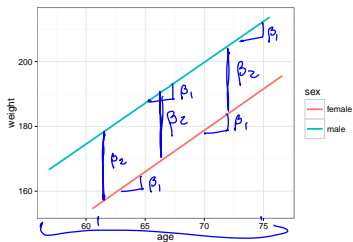
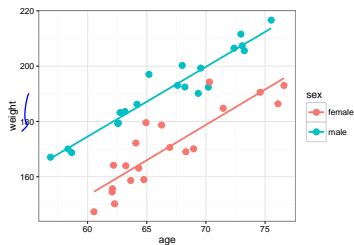
$\beta_1 =$  change in  $E(y)$  for a 1 unit  $\Delta$  age,  
keeping sex fixed.

$\beta_2 =$  " " "  
comparing females to males.  
keeping age fixed

# Example with two predictors

$$E(y|x_2) = \beta_0 + \beta_1 x_{age} + \beta_2 x_{sex}$$

$$\beta_0 = E(y | \text{age} = 0, \text{sex} = \text{male})$$



# Example: MLR

```
> summary(data.mlr)  
  age      sex      weight  
Min.   :56.86  female:20  Min.   :147.3  
1st Qu.:62.71  male  :20  1st Qu.:168.3  
Median :65.72                Median :181.4  
Mean   :66.70                Mean   :180.9  
3rd Qu.:70.23                3rd Qu.:193.0  
Max.   :76.60                Max.   :216.6
```

# Example: MLR

```
> linmod = lm(weight ~ age + sex, data = data.mlr)
> tidy(linmod)
```

	term	estimate	std.error	statistic	p.value
1	(Intercept)	1.060460	12.2597341	0.08650011	9.315353e-01
2	age	2.537768	0.1828033	13.88250888	3.022310e-16
3	sexmale	21.116049	1.8470731	11.43216717	1.056992e-13

Handwritten annotations in blue:

- An arrow points down to `sex` in the model formula.
- An arrow points up to `weight` in the model formula.
- An arrow points up to the `estimate` column header.
- An arrow points up to the `std.error` column header.
- An arrow points up to the `sexmale` term.
- An arrow points up to the `sexmale` estimate value (21.116049).
- A bracket underlines the `sexmale` row.
- A bracket underlines the `std.error` column.
- A bracket underlines the `statistic` column.
- A bracket underlines the `p.value` column.
- Labels  $\hat{\beta}_0$ ,  $\hat{\beta}_1$ , and  $\hat{\beta}_2$  are placed near the intercept, age, and sexmale estimates respectively.

# Example: MLR

```
> summary(linmod)
```

```
Call:
```

```
lm(formula = weight ~ age + sex, data = data.mlr)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-8.8987 -3.2152 -0.2969  2.3688 14.8074
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.0605     12.2596   0.087   0.932
age            2.5378      0.1828  13.883 3.02e-16 ***
sexmale       21.1160      1.8471  11.432 1.06e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 5.841 on 37 degrees of freedom
Multiple R-squared:  0.8977, Adjusted R-squared:  0.8921
F-statistic: 162.3 on 2 and 37 DF,  p-value: < 2.2e-16
```

*n = 40*



# Example: MLR

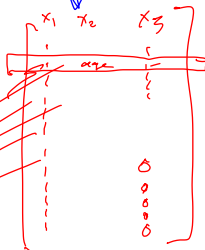
$$y_i = \beta_0^{x_0=1} + \beta_1 x_{i,age} + \beta_2 x_{i,sex} + \epsilon_i$$

```
> head(data.mlr)
Source: local data frame [6 x 3]
```

```
  ↓      ↓      ↓
  age    sex    weight
  (dbl) (fctr)  (dbl)
1 62.58799 male 179.4342
2 65.18893 male 197.0306
3 73.06852 male 207.3838
4 56.85860 male 167.0692
5 69.56368 male 199.3080
6 67.99770 male 200.2703
```

```
> model.matrix(linmod) %>% head
  (Intercept)  age sexmale
1           1 62.58799      1
2           1 65.18893      1
3           1 73.06852      1
4           1 56.85860      1
5           1 69.56368      1
6           1 67.99770      1
```

$$y = X\beta + \epsilon$$

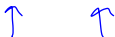


# Example: MLR

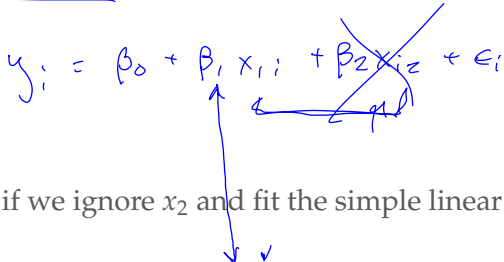
```
> tail(data.mlr)
Source: local data frame [6 x 3]

   age      sex  weight
  (dbl) (fctr)  (dbl)
1 64.75572 female 158.9645
2 63.64315 female 158.6567
3 64.08004 female 172.2003
4 64.32532 female 163.0857
5 68.96513 female 170.1063
6 64.93602 female 179.5558

> model.matrix(linmod) %>% tail
  (Intercept)    age sexmale
35           1 64.75572      0
36           1 63.64315      0
37           1 64.08004      0
38           1 64.32532      0
39           1 68.96513      0
40           1 64.93602      0
```



# Omitted variable bias

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \epsilon_i$$


What happens if we ignore  $x_2$  and fit the simple linear regression:

$$y_i = \beta_0^* + \beta_1^* x_{1,i} + \epsilon_i^*$$

Does  $\beta_1^* = \beta_1$ ? Does “total” association equal “partial” association?

# Omitted variable bias

HW2

# Omitted variable bias

There are two conditions under which  $E(\hat{\beta}_1^*) = \beta_1$ :

- The omitted variable is unrelated to the outcome

- The omitted variable is uncorrelated with the retained variable

## Still only two predictors

Suppose we think that the effect of age on weight is different for men and women. How might we approach this problem?

- Separate models?
- Interactions?

# Interpretation of coefficients

# Example: Interactions

```
> linmod = lm(weight ~ age * sex, data = data.mlr)
> tidy(linmod)
```

	term	estimate	std.error	statistic	p.value
1	(Intercept)	-33.529162	16.3317392	-2.053006	4.739377e-02
2	age	2.714362	0.2468607	10.995522	4.636445e-13
3	sexmale	-63.357026	22.7687183	-2.782635	8.532471e-03
4	age:sexmale	2.049536	0.3412012	6.006825	6.805174e-07



# Example: Interactions

```
> head(data.mlr)
Source: local data frame [6 x 3]

   age      sex  weight
  (dbl) (fctr)  (dbl)
1 62.24128 male 199.0986
2 67.10186 male 220.4382
3 60.98623 male 198.6198
4 75.57168 male 263.4126
5 67.97705 male 221.7642
6 61.07719 male 190.6024

> model.matrix(linmod) %>% head
  (Intercept)      age sexmale age:sexmale
1           1 62.24128         1 62.24128
2           1 67.10186         1 67.10186
3           1 60.98623         1 60.98623
4           1 75.57168         1 75.57168
5           1 67.97705         1 67.97705
6           1 61.07719         1 61.07719
```

# Example: Interactions

```
> tail(data.mlr)
Source: local data frame [6 x 3]

   age      sex  weight
  (dbl) (fctr)  (dbl)
1 57.73764 female 116.8223
2 63.51003 female 140.5238
3 63.63426 female 136.4259
4 65.64412 female 144.1169
5 72.60015 female 161.9464
6 70.57905 female 152.9105
> model.matrix(linmod) %>% tail
  (Intercept)      age sexmale age:sexmale
35           1 57.73764         0         0
36           1 63.51003         0         0
37           1 63.63426         0         0
38           1 65.64412         0         0
39           1 72.60015         0         0
40           1 70.57905         0         0
```

# Categorical predictors

- Assume  $X$  is a categorical / nominal / factor variable with  $k$  levels
- With only one categorical  $X$ , we have the classic one-way ANOVA design
- Can't use a single predictor with levels  $1, 2, \dots, K$  – this has the wrong interpretation
- Need to create *indicator* or *dummy* variables

# Indicator variables

- Let  $x$  be a categorical variable with  $k$  levels (e.g. with  $k = 3$  “low”, “med”, “high”).
- Choose one group as the baseline (e.g. “low”)
- Create  $(k - 1)$  binary terms to include in the model:

$$x_{\text{med},i} = I(x_i = \text{“med”})$$

$$x_{\text{high},i} = I(x_i = \text{“high”})$$

- For a model with no additional predictors, pose the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_{k-1} x_{i,k-1} + \epsilon_i$$

and estimate parameters using least squares

- Note distinction between *predictors* and *terms*

# Categorical predictor design matrix

# ANOVA model interpretation

Using the model  $y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_{k-1} x_{i,k-1} + \epsilon_i$ , interpret

$\beta_0 =$

$\beta_1 =$

# Equivalent model

Define the model  $y_i = \beta_1 x_{i1} + \dots + \beta_k x_{i,k} + \epsilon_i$  where there are indicators for each possible group

$$\beta_1 =$$

$$\beta_2 =$$

## Example for categorical predictor

Suppose you want to compare the effect of placebo, exercise and a drug on blood pressure. You set up a trial to do this and gather data  $y_i, treatment_i$  on  $n$  subjects.

- Analyze results using the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$$

where  $x_{i1}$  indicates that subject  $i$  exercised and  $x_{i2}$  indicates that subject  $i$  received medication.



# Example: categorical predictor

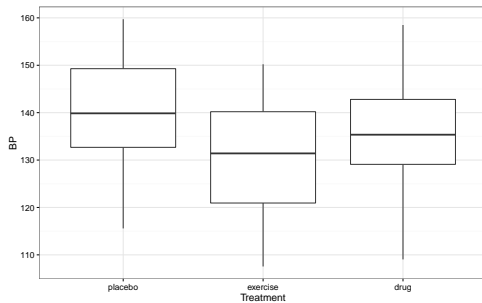
```
> ## load data
> load("BPDat.RDA")
>
> ## see what we've loaded
> head(BP)
  x1      x2
1  1 149.5939
2  1 155.5605
3  1 129.5920
4  1 149.3057
5  1 139.2455
6  1 120.3280
> summary(BP)
      x1      x2
Min.   :1   Min.   :107.5
1st Qu.:1   1st Qu.:128.0
Median :2   Median :136.8
Mean   :2   Mean    :169.9
3rd Qu.:3   3rd Qu.:144.8
Max.   :3   Max.    :999.0
```

# Example: categorical predictor

```
> ## tidy data
> BP = BP %>% rename(Treatment = x1, BP = x2) %>%
+   mutate(Treatment = factor(Treatment, levels = 1:3,
+                             labels = c("placebo", "exercise", "drug"))) %>%
+   filter(BP != 999)
>
> summary(BP)
  Treatment      BP
placebo :47  Min.   :107.5
exercise:47  1st Qu.:127.0
drug    :50  Median :136.7
                Mean   :135.3
                3rd Qu.:143.5
                Max.   :159.7
>
> BP %>% group_by(Treatment) %>% summarize(n = n(),
+                                         group_mean = mean(BP),
+                                         group_median = median(BP))
Source: local data frame [3 x 4]
```

	Treatment (fctr)	n (int)	group_mean (dbl)	group_median (dbl)
1	placebo	47	140.3368	139.8598
2	exercise	47	130.6135	131.4055
3	drug	50	135.0942	135.3504

# Example: categorical predictor



# Example: categorical predictor

$$bp_i = \beta_0 + \beta_1 tx_{\text{exer},i} + \beta_2 tx_{\text{drug},i} + \epsilon_i$$

```
> lm(BP ~ Treatment, data = BP) %>% tidy
      term      estimate std.error statistic    p.value
1 (Intercept) 140.336772  1.647753  85.168558 3.906601e-123
2 Treatmentexercise -9.723234  2.330275  -4.172569 5.240892e-05
3 Treatmentdrug -5.242587  2.295055  -2.284297 2.384739e-02
>
>
> lm(BP ~ Treatment, data = BP) %>% model.matrix %>% head
(Intercept) Treatmentexercise Treatmentdrug
1           1                0            0
2           1                0            0
3           1                0            0
4           1                0            0
5           1                0            0
6           1                0            0
```

# Example: releveling categorical predictor

$$bp_i = \beta_0 + \beta_1 tx_{\text{plac},i} + \beta_2 tx_{\text{drug},i} + \epsilon_i$$

```
> BP %>% mutate(Treatment = relevel(Treatment, ref = "exercise")) %>%  
+   lm(BP ~ Treatment, data = .) %>%  
+   tidy
```

	term	estimate	std.error	statistic	p.value
1	(Intercept)	130.613538	1.647753	79.267654	7.929548e-119
2	Treatmentplacebo	9.723234	2.330275	4.172569	5.240892e-05
3	Treatmentdrug	4.480647	2.295055	1.952305	5.288319e-02

# Example: no intercept

$$bp_i = \beta_1 tx_{\text{exer},i} + \beta_2 tx_{\text{plac},i} + \beta_3 tx_{\text{drug},i} + \epsilon_i$$

```
> lm(BP ~ 0 + Treatment, data = BP) %>% tidy
      term estimate std.error statistic      p.value
1 Treatmentplacebo 140.3368  1.647753   85.16856 3.906601e-123
2 Treatmentexercise 130.6135  1.647753   79.26765 7.929548e-119
3   Treatmentdrug 135.0942  1.597556   84.56303 1.048075e-122
>
>
> BP %>% group_by(Treatment) %>% summarize(n = n(),
+                                           group_mean = mean(BP),
+                                           group_median = median(BP))
Source: local data frame [3 x 4]
```

	Treatment (fctr)	n (int)	group_mean (dbl)	group_median (dbl)
1	placebo	47	140.3368	139.8598
2	exercise	47	130.6135	131.4055
3	drug	50	135.0942	135.3504

# Today's big ideas

- Multiple linear regression models, interpretation, interactions, categorical predictors

- 
- Suggested reading: Faraway Ch 2.2 - 2.3; ISLR 3.2