

Linear Regression Models

P8111

Lecture 08

Jeff Goldsmith
February 16, 2016



THE DEPARTMENT OF
BIostatISTICS



Columbia University
MAILMAN SCHOOL
OF PUBLIC HEALTH

Today's Lecture

- LSE properties
- Identifiability in MLR
- Collinearity and near-collinearity
- MLR Example

Key points so far

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

- Our model is $\underline{y = X\beta + \epsilon}$ with $\epsilon \sim (0, \sigma^2 I)$
- The design matrix \underline{X} contains the terms included in the model
- We've derived least squares solutions under some conditions

$$\hat{\beta}_{\text{LSE}} = (X^T X)^{-1} X^T y \quad \checkmark \checkmark \checkmark$$

Mean, variance and covariance of a random vector

- Let $\mathbf{y}^T = [y_1, \dots, y_n]$ be an n -component random vector. Then its mean and variance are defined as

$$E(\mathbf{y})^T = [E(y_1), \dots, E(y_n)]$$

$$Var(\mathbf{y}) = E[(\mathbf{y} - E\mathbf{y})(\mathbf{y} - E\mathbf{y})^T] = E(\mathbf{y}\mathbf{y}^T) - (E\mathbf{y})(E\mathbf{y})^T$$

- Let \mathbf{y} and \mathbf{z} be an n -component and an m -component random vector respectively. Then their covariance is an $n \times m$ matrix defined by

$$Cov(\mathbf{y}, \mathbf{z}) = E[(\mathbf{y} - E\mathbf{y})(\mathbf{z} - E\mathbf{z})^T]$$

Basics on random vectors

Let A be a $t \times n$ non-random matrix and B be a $p \times m$ non-random matrix. Then

$$\begin{aligned} E(\mathbf{A}\mathbf{y}) &= \mathbf{A}E\mathbf{y} \\ \text{Var}(\mathbf{A}\mathbf{y}) &= \mathbf{A}\text{Var}(\mathbf{y})\mathbf{A}^T \\ \text{Cov}(\mathbf{A}\mathbf{y}, \mathbf{B}\mathbf{z}) &= \mathbf{A}\text{Cov}(\mathbf{y}, \mathbf{z})\mathbf{B}^T \end{aligned}$$

Unbiasedness of LSEs

$$E(\hat{\beta}) =$$

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

$$\epsilon \sim (0, \sigma^2 I) \checkmark$$

$$\checkmark y \sim (\underline{x\beta}, \underline{\sigma^2 I}) \checkmark$$

$$E(\underbrace{(x^T x)^{-1} x^T}_{} y)$$

$$= (x^T x)^{-1} x^T E(y)$$

$$\sim \cancel{(x^T x)^{-1} x^T} \beta$$

$$= \beta$$

Variance of LSEs

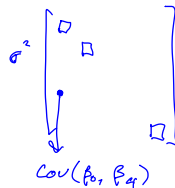
$$\text{Var}(\hat{\beta}) =$$

$$\text{Var}(\underline{x^T x}^{-1} x^T y)$$

$$= (x^T x)^{-1} x^T \text{var}(y) x (x^T x)^{-1}$$

$$= \sigma^2 (x^T x)^{-1} \underbrace{(x^T x x^T x)^{-1}}_{\sigma^2 I}$$

$$= \sigma^2 (x^T x)^{-1}$$



$$\widehat{\text{Var}}(\hat{\beta}) = \hat{\sigma}^2 (x^T x)^{-1}$$

$$\text{Var}(c\hat{\beta}) =$$

$$c \text{Var}(\hat{\beta}) c^T$$

$$\sigma^2 c (x^T x)^{-1} c^T$$

$$\left[\begin{array}{c|ccc} 1 & x_{i1} & x_{i2} & x_{i3} \\ \hline c^T & & & \end{array} \right] = y_i$$

Sampling distribution of $\hat{\beta}$

$$\hat{\beta} \sim (\beta, \sigma^2(X^T X)^{-1})$$

If our usual assumptions are satisfied and $\epsilon \sim \underline{N} [0, \sigma^2 I]$ then

$$\hat{\beta} \sim N [\beta, \sigma^2 (X^T X)^{-1}]. \quad y \sim N(X\beta, \sigma^2 \Sigma)$$

- This will be used later for inference.
- Even without Normal errors, asymptotic Normality of LSEs is possible under reasonable assumptions.

Definitions

- Fitted values: $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{H}\mathbf{y}$
"Hat matrix" (with arrow pointing to \mathbf{H})
- Residuals / estimated errors: $\hat{\boldsymbol{\epsilon}} = \mathbf{y} - \hat{\mathbf{y}}$
- Residual sum of squares: $\sum_{i=1}^n \hat{\epsilon}_i^2 = \hat{\boldsymbol{\epsilon}}^T \hat{\boldsymbol{\epsilon}}$
- Residual variance: $\hat{\sigma}^2 = \frac{\text{RSS}}{n-p-1}$
- Degrees of freedom: $n - p - 1$
 $y_i = \beta_0 + \beta_1 x_i + \dots + \beta_p x_i^p + \epsilon_i$
pred other than intercept

R^2 and sums of squares

- Regression sum of squares $SS_{reg} = \sum(\hat{y}_i - \bar{y})^2$
- Residual sum of squares $SS_{res} = \sum(y_i - \hat{y}_i)^2$
- Total sum of squares $SS_{tot} = \sum(y_i - \bar{y})^2$
- Coefficient of determination

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

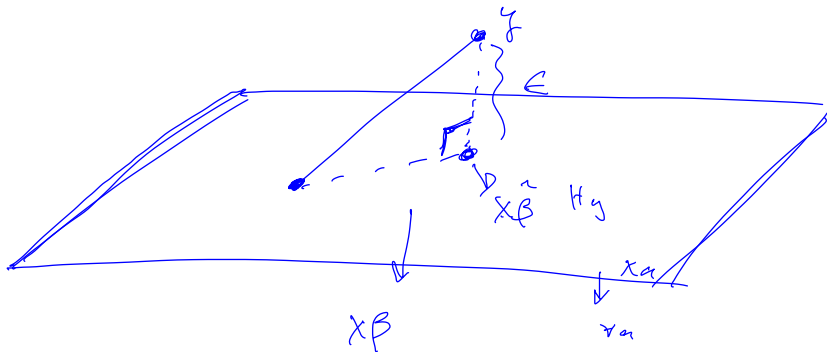
Hat matrix

Some properties of the hat matrix:

- It is a projection matrix: $\mathbf{H}\mathbf{H} = \mathbf{H}$
- It is symmetric: $\mathbf{H}^T = \mathbf{H}$
- The residuals are $\hat{\mathbf{e}} = (\mathbf{I} - \mathbf{H})\mathbf{y}$ $y - Hy = y - \hat{y}$
- The inner product of $(\mathbf{I} - \mathbf{H})\mathbf{y}$ and $\mathbf{H}\mathbf{y}$ is zero (predicted values and residuals are uncorrelated).

Projection space interpretation

The hat matrix projects \mathbf{y} onto the column space of \mathbf{X} . Alternatively, minimizing the $RSS(\beta)$ is equivalent to minimizing the Euclidean distance between \mathbf{y} and the column space of \mathbf{X} .

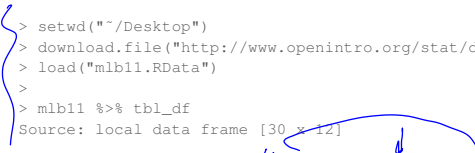


MLR Example: Moneyball

- “Moneyball” used statistics to help identify key player features that contributed to winning baseball games
- We’ll look at association between runs scored and team-level covariates
- First, load the data in R workspace and understand the variables:

MLR Example: Moneyball

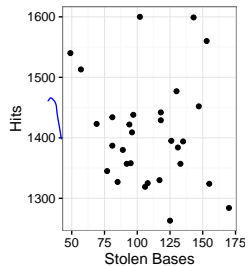
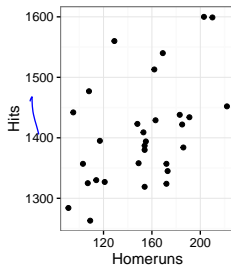
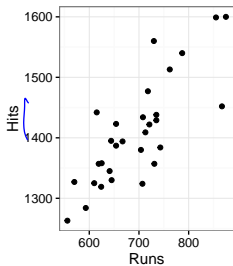
```
> setwd("~/Desktop")
> download.file("http://www.openintro.org/stat/data/mlb11.RData", destfile = "mlb11.RData")
> load("mlb11.RData")
>
> mlb11 %>% tbl_df
Source: local data frame [30 x 12]
```



| | <u>team</u> | <u>runs</u> | <u>at_bats</u> | <u>hits</u> | <u>homeruns</u> | <u>bat_avg</u> | <u>strikeouts</u> | <u>stolen_bases</u> | <u>wins</u> | <u>new</u> |
|----|---------------------|-------------|----------------|-------------|-----------------|----------------|-------------------|---------------------|-------------|------------|
| | (fctr) | (int) | (int) | (int) | (int) | (dbl) | (int) | (int) | (int) | (int) |
| 1 | Texas Rangers | 855 | 5659 | 1599 | 210 | 0.283 | 930 | 143 | 96 | |
| 2 | Boston Red Sox | 875 | 5710 | 1600 | 203 | 0.280 | 1108 | 102 | 90 | |
| 3 | Detroit Tigers | 787 | 5563 | 1540 | 169 | 0.277 | 1143 | 49 | 95 | |
| 4 | Kansas City Royals | 730 | 5672 | 1560 | 129 | 0.275 | 1006 | 153 | 71 | |
| 5 | St. Louis Cardinals | 762 | 5532 | 1513 | 162 | 0.273 | 978 | 57 | 90 | |
| 6 | New York Mets | 718 | 5600 | 1477 | 108 | 0.264 | 1085 | 130 | 77 | |
| 7 | New York Yankees | 867 | 5518 | 1452 | 222 | 0.263 | 1138 | 147 | 97 | |
| 8 | Milwaukee Brewers | 721 | 5447 | 1422 | 185 | 0.261 | 1083 | 94 | 96 | |
| 9 | Colorado Rockies | 735 | 5544 | 1429 | 163 | 0.258 | 1201 | 118 | 73 | |
| 10 | Houston Astros | 615 | 5598 | 1442 | 95 | 0.258 | 1164 | 118 | 56 | |
| .. | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

```
Variables not shown: new_slug (dbl), new_obs (dbl)
```

Exploratory plots



MLB data

- team
- runs
- at_bats
- hits
- homeruns
- bat_avg
- strikeouts
- wins
- new_onbase
- new_slug
- new_obs

Multiple Linear Regression

```
> linmod = lm(runs ~ at_bats + hits + homeruns + stolen_bases, data = mlb11)
```

```
> tidy(linmod)
```

| | term | estimate | std.error | statistic | p.value |
|---|--------------|-------------|-------------|-----------|--------------|
| 1 | (Intercept) | 581.2109940 | 526.4062575 | 1.104111 | 2.800591e-01 |
| 2 | at_bats | -0.2023278 | 0.1173616 | -1.723970 | 9.705991e-02 |
| 3 | hits | 0.6974143 | 0.1131428 | 6.164017 | 1.911117e-06 |
| 4 | homeruns | 1.2535062 | 0.1593185 | 7.867926 | 3.178626e-08 |
| 5 | stolen_bases | 0.5229741 | 0.1686315 | 3.101284 | 4.727771e-03 |

R does what we expect

```
> X = cbind(1, mlb11$at_bats, mlb11$hits, mlb11$homeruns, mlb11$stolen_bases)
> y = (mlb11$runs)
>
> betaHat = solve(t(X) %*% X) %*% t(X) %*% y
> betaHat
      [,1]
[1,] 581.2109940
[2,] -0.2023278
[3,]  0.6974143
[4,]  1.2535062
[5,]  0.5229741
```

R does what we expect

```
> fitted = X %*% betaHat
> sigmaHat = sqrt(t(y - fitted) %*% (y - fitted) / (30-4-1))
> sigmaHat
      [,1]
[1,] 26.84777
```

R does what we expect

```
> VarBeta = as.numeric(sigmaHat^2) * (solve(t(X) %**% X))
> VarBeta
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 277103.547951 -6.092180e+01 42.6907437568 -1.1735722100 -5.003475e+00
[2,]   -60.921800  1.377374e-02 -0.0108596986  0.0008804808  8.390983e-05
[3,]   42.690744 -1.085970e-02  0.0128013011 -0.0054905039  8.247119e-04
[4,]   -1.173572  8.804808e-04 -0.0054905039  0.0253823891  1.779074e-03
[5,]   -5.003475  8.390983e-05  0.0008247119  0.0017790736  2.843658e-02
> sqrt(diag(VarBeta))
[1] 526.4062575  0.1173616  0.1131428  0.1593185  0.1686315
```

Least squares estimates

- $\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$
- A condition on $(\mathbf{X}^T \mathbf{X})$:
 - If $(\mathbf{X}^T \mathbf{X})$ is singular, there are infinitely many least squares solutions, making $\hat{\beta}$ non-identifiable (can't choose between different solutions)

Non-identifiability

- Can happen if X is not of full rank, i.e. the columns of X are linearly dependent (for example, including weight in Kg and lb as predictors)
- Can happen if there are fewer data points than terms in X : $n < p$ (having 100 predictors and only 50 observations)
- Generally, the $p \times p$ matrix $(X^T X)$ is invertible if and only if it has rank p .

Infinite solutions

Suppose I fit a model $y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i$.

- I have estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 2$
- I put in a new variable $x_2 = x_1$
- My new model is $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$
- Possible least squares estimates that are equivalent to my first model:
 - ▶ $\hat{\beta}_0 = 1, \hat{\beta}_1 = 2, \hat{\beta}_2 = 0$
 - ▶ $\hat{\beta}_0 = 1, \hat{\beta}_1 = 0, \hat{\beta}_2 = 2$
 - ▶ $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1002, \hat{\beta}_2 = -1000$
 - ▶ ...

Non-identifiability

- Often due to data coding errors (variable duplication, scale changes)
- Pretty easy to detect and resolve
- Certain kinds can be addressed using *penalties* (later topic)
- A bigger problem is near-unidentifiability (collinearity)

Causes of collinearity

- Arises when variables are highly correlated, but not exact duplicates
- Commonly arises in data (perfect correlation is usually there by mistake)
- Might exist between several variables, i.e. a linear combination of several variables exists in the data
- A variety of tools exist (correlation analyses, multiple R^2 , eigen decompositions)

Effects of collinearity

Suppose I fit a model $y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i$.

- I have estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 2$
- I put in a new variable $x_2 = x_1 + \text{error}$, where *error* is pretty small
- My new model is $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$
- Possible least squares estimates that are nearly equivalent to my first model:
 - ▶ $\hat{\beta}_0 = 1, \hat{\beta}_1 = 2, \hat{\beta}_2 = 0$
 - ▶ $\hat{\beta}_0 = 1, \hat{\beta}_1 = 0, \hat{\beta}_2 = 2$
 - ▶ $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1002, \hat{\beta}_2 = -1000$
 - ▶ ...
- A unique solution exists, but it is hard to find

Effects of collinearity

- Collinearity results in a “flat” RSS
- Makes identifying a unique solution difficult
- Dramatically inflates the variance of LSEs

Example: mother and daughter heights

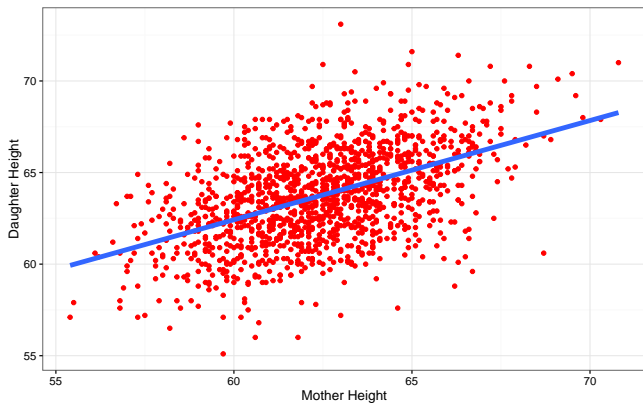
- 1375 mother/daughter pairs (see Lecture 1)
- Want to predict daughter height based on mother height
- Data originally comes in inches

Example: mother and daughter heights

Simple linear regression analysis

```
> linmod = lm(Dheight~Mheight, data = heights)
> tidy(linmod)
  term      estimate  std.error statistic    p.value
1 (Intercept) 29.917437 1.62246940  18.43945 5.211879e-68
2      Mheight  0.541747 0.02596069  20.86797 3.216915e-84
```

Example: mother and daughter heights



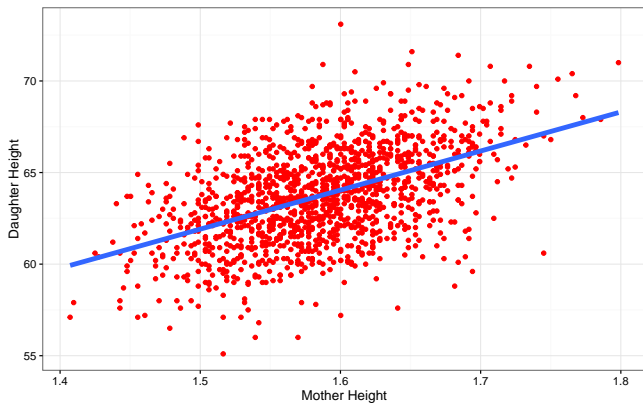
Example: change of variables

What happens if mother's height is expressed in meters?

```
> heights = mutate(heights, Mheight_m = Mheight * .0254)
>
> linmod = lm(Dheight~Mheight_m, data = heights)
> tidy(linmod)
```

| | term | estimate | std.error | statistic | p.value |
|---|-------------|----------|-----------|-----------|--------------|
| 1 | (Intercept) | 29.91744 | 1.622469 | 18.43945 | 5.211879e-68 |
| 2 | Mheight_m | 21.32862 | 1.022074 | 20.86797 | 3.216915e-84 |

Example: change in variables



Example: non-identifiability

What if we include mother's height in both inches and meters?

```
> linmod.col = lm(Dheight ~ Mheight + Mheight_m, data = heights)
> summary(linmod.col)
```

Call:

```
lm(formula = Dheight ~ Mheight + Mheight_m, data = heights)
```

Residuals:

| Min | 1Q | Median | 3Q | Max |
|--------|--------|--------|-------|-------|
| -7.397 | -1.529 | 0.036 | 1.492 | 9.053 |

Coefficients: (1 not defined because of singularities)

| | Estimate | Std. Error | t value | Pr(> t) |
|-------------|----------|------------|---------|------------|
| (Intercept) | 29.91744 | 1.62247 | 18.44 | <2e-16 *** |
| Mheight | 0.54175 | 0.02596 | 20.87 | <2e-16 *** |
| Mheight_m | NA | NA | NA | NA |

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

Residual standard error: 2.266 on 1373 degrees of freedom

Multiple R-squared: 0.2408, Adjusted R-squared: 0.2402

F-statistic: 435.5 on 1 and 1373 DF, p-value: < 2.2e-16

Example: non-identifiability

What if we include mother's height in both inches and meters?

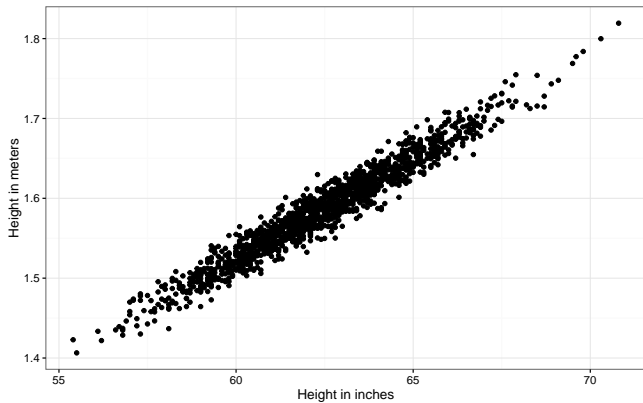
```
> X = as.matrix(cbind(1, select(heights, Mheight, Mheight_m)))
> solve(t(X) %*% X)
Error in solve.default(t(X) %*% X) :
  system is computationally singular: reciprocal condition number = 6.681e-18
```

Example: near-unidentifiability

Suppose height was measured twice: once in inches, once in meters. There's some measurement error comparing the two. What happens now?

```
> heights = mutate(heights, Mheight_m = Mheight_m + rnorm(1375, mean = 0, sd = .01))
> summarize(heights, cor = cor(Mheight, Mheight_m))
      cor
1 0.9716345
>
> X = as.matrix(cbind(1, select(heights, Mheight, Mheight_m)))
> solve(t(X) %*% X)
           1      Mheight  Mheight_m
1      0.512809433 -0.007399262 -0.03150439
Mheight -0.007399262  0.002346264 -0.08770449
Mheight_m -0.031504393 -0.087704488  3.47264911
```

Example: near-unidentifiability



Example: near-unidentifiability

What if we include mother's height in both inches and meters?

```
> linmod.me = lm(Dheight ~ Mheight + Mheight_m, data = heights)
> tidy(linmod.me)
```

| | term | estimate | std.error | statistic | p.value |
|---|-------------|------------|-----------|-----------|--------------|
| 1 | (Intercept) | 29.9594863 | 1.622801 | 18.461592 | 3.796458e-68 |
| 2 | Mheight | 0.6588078 | 0.109768 | 6.001822 | 2.493306e-09 |
| 3 | Mheight_m | -4.6350101 | 4.222967 | -1.097572 | 2.725840e-01 |

Some take away messages

- Collinearity can (and does) happen, so be careful
- Worst cases tend to be “pathological examples”, so don't lose hope
- Often contributes to the problem of variable selection, which we'll touch on later

Categorical predictor design matrix

Which of the following is a “correct” design matrix for a categorical predictor with 3 levels?

$$X_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad X_2 = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \end{bmatrix} \quad \text{or} \quad X_3 = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix}$$

Today's big ideas

- Identifiability, collinearity, categorical predictors
-

- Suggested reading: Faraway Ch 3.7 (pdf); ISLR 3.3.1