

STAC67: Regression Analysis

Lecture 10

Thibault Randrianarisoa

October 9, 2026

University of Toronto Scarborough

Announcement: the project brief is out

On Quercus: the **brief**, the **rubric** and the **nine datasets**, with a description of every variable.

- Optional, worth 20% of the grade, and counted only if it raises it.
- Groups of two to four students, working on one dataset of the list.
- **To find partners**: answer the Piazza poll (not anonymous) with the datasets you are ready to work on.
- **To declare your group**: sign up together in the Quercus group of your dataset (People, then Groups).

Groups: Fri 6 Nov. Checkpoint: Fri 13 Nov. Report: Fri 27 Nov. Presentations: week of 30 Nov.

Review of Lecture 9

The rules we use, for constant matrices A and B , are:

$$E(AY) = A E(Y), \quad \sigma^2\{AY\} = A \sigma^2\{Y\} A',$$

$$\sigma\{AY, BY\} = A \sigma^2\{Y\} B'.$$

Review of Lecture 9

The rules we use, for constant matrices A and B , are:

$$E(\mathbf{A}\mathbf{Y}) = \mathbf{A} E(\mathbf{Y}), \quad \sigma^2\{\mathbf{A}\mathbf{Y}\} = \mathbf{A} \sigma^2\{\mathbf{Y}\} \mathbf{A}',$$

$$\sigma\{AY, BY\} = A \sigma^2\{Y\} B'$$

in matrix form:

$$= \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

Simple linear regression in matrix form:

$$\text{Exp}^n \quad \begin{cases} Y = X\beta + \epsilon, & \sigma^2\{\epsilon\} = \sigma^2 I, & \hat{\beta} = (X'X)^{-1}X'Y, \\ \hat{Y} = HY & \text{with } H = X(X'X)^{-1}X', & e = (I - H)Y. \end{cases}$$

Review of Lecture 9

The rules we use, for constant matrices A and B , are:

$$E(AY) = A E(Y), \quad \sigma^2\{AY\} = A \sigma^2\{Y\} A',$$

$$\sigma\{AY, BY\} = A \sigma^2\{Y\} B'.$$

Simple linear regression in matrix form:

$$Y = X\beta + \epsilon, \quad \sigma^2\{\epsilon\} = \sigma^2 I, \quad \hat{\beta} = (X'X)^{-1} X'Y,$$

$$\hat{Y} = HY \text{ with } H = X(X'X)^{-1}X', \quad e = (I - H)Y.$$

We stopped at the mean and variance of $\hat{\beta}$: we pick up there.

The mean and variance of $\hat{\beta}$

$\hat{\beta} = \mathbf{A}\mathbf{Y}$ with the constant matrix $\mathbf{A} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Apply the rules:

The mean and variance of $\hat{\beta}$

$\hat{\beta} = \mathbf{A}\mathbf{Y}$ with the constant matrix $\mathbf{A} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Apply the rules:

$$E(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' E(\mathbf{Y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} \beta = \beta : \text{ unbiased.}$$

$= \mathbf{X}\beta + \varepsilon$

The mean and variance of $\hat{\beta}$

$\hat{\beta} = \mathbf{A}\mathbf{Y}$ with the constant matrix $\mathbf{A} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Apply the rules:

$$E(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' E(\mathbf{Y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} \beta = \beta : \quad \text{unbiased.}$$

$$\sigma^2\{\hat{\beta}\} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \sigma^2 \mathbf{I} \mathbf{X} (\mathbf{X}'\mathbf{X})^{-1}$$

The mean and variance of $\hat{\beta}$

$\hat{\beta} = \mathbf{A}\mathbf{Y}$ with the constant matrix $\mathbf{A} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Apply the rules:

$$E(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' E(\mathbf{Y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} \beta = \beta : \quad \text{unbiased.}$$

$$\begin{aligned}\sigma^2\{\hat{\beta}\} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \sigma^2 \mathbf{I} \mathbf{X} (\mathbf{X}'\mathbf{X})^{-1} \\ &= \sigma^2 (\mathbf{X}'\mathbf{X})^{-1} \underbrace{\mathbf{X}'\mathbf{X} (\mathbf{X}'\mathbf{X})^{-1}}_{\mathbf{I}} = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1}.\end{aligned}$$

The mean and variance of $\hat{\beta}$

$\hat{\beta} = \mathbf{A}\mathbf{Y}$ with the constant matrix $\mathbf{A} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Apply the rules:

$$E(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' E(\mathbf{Y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} \beta = \beta : \quad \text{unbiased.}$$

$$\begin{aligned}\sigma^2\{\hat{\beta}\} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \sigma^2 \mathbf{I} \mathbf{X} (\mathbf{X}'\mathbf{X})^{-1} \\ &= \sigma^2 (\mathbf{X}'\mathbf{X})^{-1} \underbrace{\mathbf{X}'\mathbf{X} (\mathbf{X}'\mathbf{X})^{-1}}_{\mathbf{I}} = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1}.\end{aligned}$$

We used $((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')' = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$, since $(\mathbf{X}'\mathbf{X})^{-1}$ is symmetric.

$$\sigma^2\{\hat{\beta}\} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} \sigma^2\left(\frac{1}{n} + \frac{\bar{X}^2}{S_{xx}}\right) & -\frac{\bar{X}\sigma^2}{S_{xx}} \\ -\frac{\bar{X}\sigma^2}{S_{xx}} & \frac{\sigma^2}{S_{xx}} \end{bmatrix} \begin{matrix} \leftarrow \text{Var}(\hat{\beta}_0), \text{Cov}(\hat{\beta}_0, \hat{\beta}_1) \\ \leftarrow \text{Cov}(\hat{\beta}_0, \hat{\beta}_1), \text{Var}(\hat{\beta}_1) \end{matrix}$$

$$\sigma^2\{\hat{\beta}\} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} \sigma^2\left(\frac{1}{n} + \frac{\bar{X}^2}{S_{xx}}\right) & -\frac{\bar{X}\sigma^2}{S_{xx}} \\ -\frac{\bar{X}\sigma^2}{S_{xx}} & \frac{\sigma^2}{S_{xx}} \end{bmatrix} \begin{matrix} \leftarrow \text{Var}(\hat{\beta}_0), \text{Cov}(\hat{\beta}_0, \hat{\beta}_1) \\ \leftarrow \text{Cov}(\hat{\beta}_0, \hat{\beta}_1), \text{Var}(\hat{\beta}_1) \end{matrix}$$

All three formulas of Lecture 4 are collected in this matrix.

Reading $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$

$$\sigma^2\{\hat{\beta}\} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} \sigma^2\left(\frac{1}{n} + \frac{\bar{X}^2}{S_{xx}}\right) & -\frac{\bar{X}\sigma^2}{S_{xx}} \\ -\frac{\bar{X}\sigma^2}{S_{xx}} & \frac{\sigma^2}{S_{xx}} \end{bmatrix} \begin{matrix} \leftarrow \text{Var}(\hat{\beta}_0), \text{Cov}(\hat{\beta}_0, \hat{\beta}_1) \\ \leftarrow \text{Cov}(\hat{\beta}_0, \hat{\beta}_1), \text{Var}(\hat{\beta}_1) \end{matrix}$$

All three formulas of Lecture 4 are collected in this matrix.

Replacing σ^2 by MSE gives the **estimated** variance-covariance matrix

$$s^2\{\hat{\beta}\} = MSE (\mathbf{X}'\mathbf{X})^{-1},$$

and the standard errors $s\{\hat{\beta}_0\}$, $s\{\hat{\beta}_1\}$ are the square roots of its diagonal.

In R: $\hat{\beta}$ from the matrices

We consider the crime rate data of Lectures 4 to 6, with `fit <- lm(Y ~ X)`.

- `model.matrix(fit)` builds the design matrix X : a column of 1s, then the column of X_i ;
- `t(A)` is the transpose A' , `A %*% B` the matrix product AB , `solve(A)` the inverse A^{-1} .

In R: $\hat{\beta}$ from the matrices

We consider the crime rate data of Lectures 4 to 6, with `fit <- lm(Y ~ X)`.

- `model.matrix(fit)` builds the design matrix X : a column of 1s, then the column of X_i ;
- `t(A)` is the transpose A' , `A %*% B` the matrix product AB , `solve(A)` the inverse A^{-1} .

```
X <- model.matrix(fit)
XtX <- t(X) %*% X
beta_hat <- solve(XtX) %*% t(X) %*% Crime$Y
cbind(manual_fit = drop(beta_hat), lm = coef(fit))
```

```
##                manual_fit                lm
## (Intercept) 20517.5999 20517.5999
## X           -170.5752  -170.5752
```

In R: $\hat{\beta}$ from the matrices

We consider the crime rate data of Lectures 4 to 6, with `fit <- lm(Y ~ X)`.

- `model.matrix(fit)` builds the design matrix X : a column of 1s, then the column of X_i ;
- `t(A)` is the transpose A' , `A %*% B` the matrix product AB , `solve(A)` the inverse A^{-1} .

```
X <- model.matrix(fit)
XtX <- t(X) %*% X
beta_hat <- solve(XtX) %*% t(X) %*% Crime$Y
cbind(manual_fit = drop(beta_hat), lm = coef(fit))
```

```
##                manual_fit                lm
## (Intercept) 20517.5999 20517.5999
## X           -170.5752  -170.5752
```

We get the same estimates as `lm`, but computed with the formula

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

In R: the estimated variance-covariance matrix

$MSE = SSE/(n - 2)$, then $s^2\{\hat{\beta}\} = MSE (\mathbf{X}'\mathbf{X})^{-1}$:

```
MSE <- sum(resid(fit)^2) / (nrow(X) - 2)
s2_beta <- MSE * solve(XtX)
s2_beta
```

```
##                (Intercept)                X
## (Intercept)    10742942 -135845.953
## X              -135846   1728.425
```

```
vcov(fit) # computed by lm
```

```
##                (Intercept)                X
## (Intercept)    10742942 -135845.953
## X              -135846   1728.425
```

In R: the estimated variance-covariance matrix

$MSE = SSE/(n - 2)$, then $s^2\{\hat{\beta}\} = MSE (\mathbf{X}'\mathbf{X})^{-1}$:

```
MSE <- sum(resid(fit)^2) / (nrow(X) - 2)
s2_beta <- MSE * solve(XtX)
s2_beta
```

```
##                (Intercept)                X
## (Intercept)    10742942 -135845.953
## X              -135846   1728.425
```

```
vcov(fit)                # computed by lm
```

```
##                (Intercept)                X
## (Intercept)    10742942 -135845.953
## X              -135846   1728.425
```

The diagonal holds the estimated variances: $s\{\hat{\beta}_1\} = \sqrt{1728.4} = 41.57$, the standard error behind the crime rate t test of Lectures 5 and 6.

Several predictors

Models with several predictors

Most problems have **more than one** potential predictor. The aim is to find the effect of each one, **controlling for the others**.

Models with several predictors

Most problems have **more than one** potential predictor. The aim is to find the effect of each one, **controlling for the others**.

The **first-order model** with two numeric predictors:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i, \quad E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2.$$

The mean response is a **plane** above the (X_1, X_2) plane.

Models with several predictors

Most problems have **more than one** potential predictor. The aim is to find the effect of each one, **controlling for the others**.

The **first-order model** with two numeric predictors:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i, \quad E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2.$$

The mean response is a **plane** above the (X_1, X_2) plane.

The same framework will allow us to handle:

- polynomial terms X_i^p , for curved relations;
- products of predictors $X_{i1}X_{i2}$, when the effect of one depends on the level of another;
- discrete variables, for categorical predictors.

Example: acceleration of cars

We observe 390 cars in the 'autompg' dataset from the UCI Machine Learning Repository: acc, the time to accelerate from 0 to 60 mph (in seconds), against the weight wt (in lb) and the horsepower hp.

```
fit2 <- lm(acc ~ wt + hp, data = autompg)
round(coef(summary(fit2)), 5)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	18.46959	0.32651	56.56593	0
## wt	0.00229	0.00021	11.10182	0
## hp	-0.09326	0.00455	-20.47774	0

Example: acceleration of cars

We observe 390 cars in the 'autmpg' dataset from the UCI Machine Learning Repository: acc, the time to accelerate from 0 to 60 mph (in seconds), against the weight wt (in lb) and the horsepower hp.

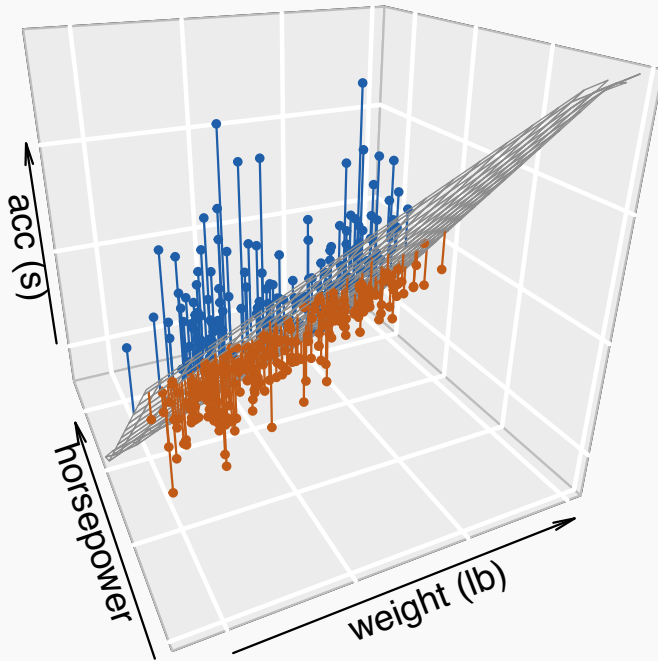
```
fit2 <- lm(acc ~ wt + hp, data = autmpg)
round(coef(summary(fit2)), 5)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	18.46959	0.32651	56.56593	0
## wt	0.00229	0.00021	11.10182	0
## hp	-0.09326	0.00455	-20.47774	0

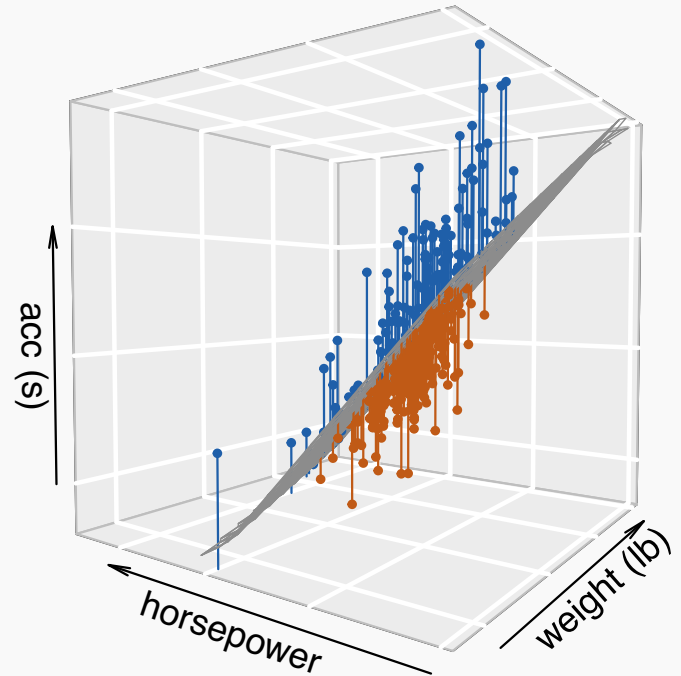
Reading $\hat{\beta}_1$: among cars with the same horsepower, each extra 1000 lb adds about 2.3 seconds to the mean time to 60 mph.

The fitted plane, from two angles

The plane seen from above



The plane seen edge on



Each vertical segment is a residual.

Why not two simple regressions?

Regressing acc on wt alone gives a slope of -1.36 seconds per 1000 lb: **heavier cars seem faster.**

Why not two simple regressions?

Regressing acc on wt alone gives a slope of -1.36 seconds per 1000 lb: **heavier cars seem faster.**

With hp in the model, the slope is +2.29 seconds per 1000 lb: **heavier cars are slower**, at a given horsepower.

Why not two simple regressions?

Regressing acc on wt alone gives a slope of -1.36 seconds per 1000 lb: **heavier cars seem faster.**

With hp in the model, the slope is +2.29 seconds per 1000 lb: **heavier cars are slower**, at a given horsepower.

The two predictors are strongly related,

$$r(\text{wt}, \text{hp}) = 0.86$$

Heavy cars tend to have big engines. Alone, the weight carries the effect of the engine with it.

Why not two simple regressions?

Regressing acc on wt alone gives a slope of -1.36 seconds per 1000 lb: **heavier cars seem faster.**

With hp in the model, the slope is +2.29 seconds per 1000 lb: **heavier cars are slower**, at a given horsepower.

The two predictors are strongly related,

$$r(\text{wt}, \text{hp}) = 0.86$$

Heavy cars tend to have big engines. Alone, the weight carries the effect of the engine with it.

A coefficient in a multiple regression is the effect of its predictor with the others held fixed. It can differ from the simple regression slope, even in sign.

What “holding horsepower fixed” means

The slope of wt in the multiple regression can be obtained in three simple regressions:

1. regress acc on hp: the residuals are the part of acc that hp does not explain;
2. regress wt on hp: the residuals are the part of wt that hp does not explain;
3. regress the first residuals on the second.

$$\begin{array}{l} 1. \text{ acc} = \hat{\beta}_0 + \hat{\beta}_1 \text{ hp} + \hat{\varepsilon} \\ 2. \text{ wt} = \hat{\beta}_0 + \hat{\beta}_1 \text{ hp} + \hat{\varepsilon} \\ 3. \hat{\varepsilon} = \bar{\beta}_0 + \bar{\beta}_1 \varepsilon + \dots \end{array} \rightarrow \begin{array}{l} \text{acc} = \hat{\beta}_0 + \hat{\beta}_1 \text{ hp} \\ + \bar{\beta}_0 + \bar{\beta}_1 \varepsilon + \dots \end{array}$$

What “holding horsepower fixed” means

The slope of wt in the multiple regression can be obtained in three simple regressions:

1. regress acc on hp: the residuals are the part of acc that hp does not explain;
2. regress wt on hp: the residuals are the part of wt that hp does not explain;
3. regress the first residuals on the second.

```
e_acc <- resid(lm(acc ~ hp, data = autmpg))  
e_wt <- resid(lm(wt ~ hp, data = autmpg))  
c(three_steps = coef(lm(e_acc ~ e_wt))[[2]],  
  multiple = coef(fit2)[["wt"]])
```

```
## three_steps      multiple  
## 0.002291864 0.002291864
```

What “holding horsepower fixed” means

The slope of wt in the multiple regression can be obtained in three simple regressions:

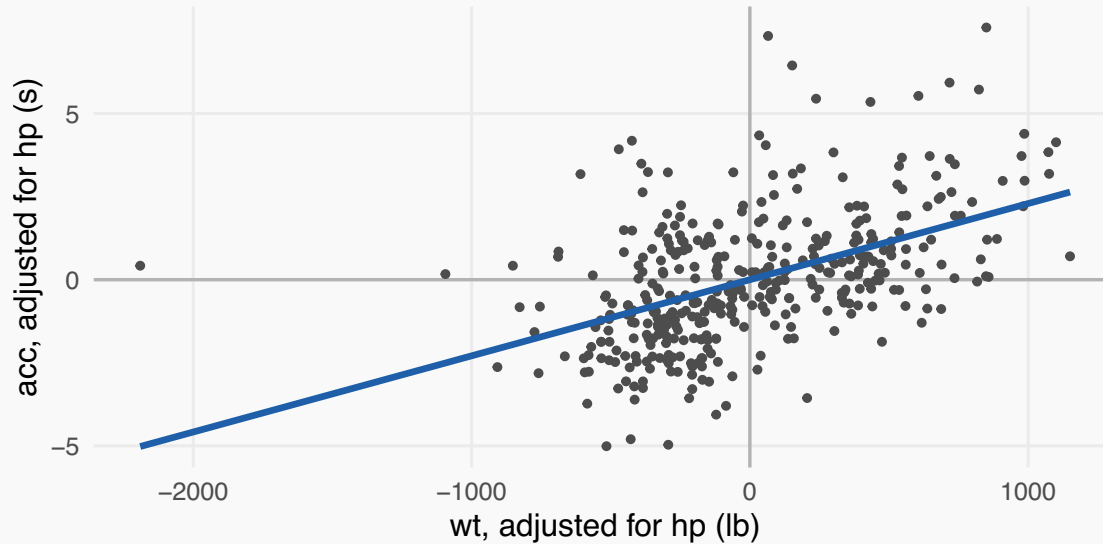
1. regress acc on hp: the residuals are the part of acc that hp does not explain;
2. regress wt on hp: the residuals are the part of wt that hp does not explain;
3. regress the first residuals on the second.

```
e_acc <- resid(lm(acc ~ hp, data = autmpg))
e_wt <- resid(lm(wt ~ hp, data = autmpg))
c(three_steps = coef(lm(e_acc ~ e_wt))[[2]],
  multiple = coef(fit2)[["wt"]])
```

```
## three_steps      multiple
## 0.002291864 0.002291864
```

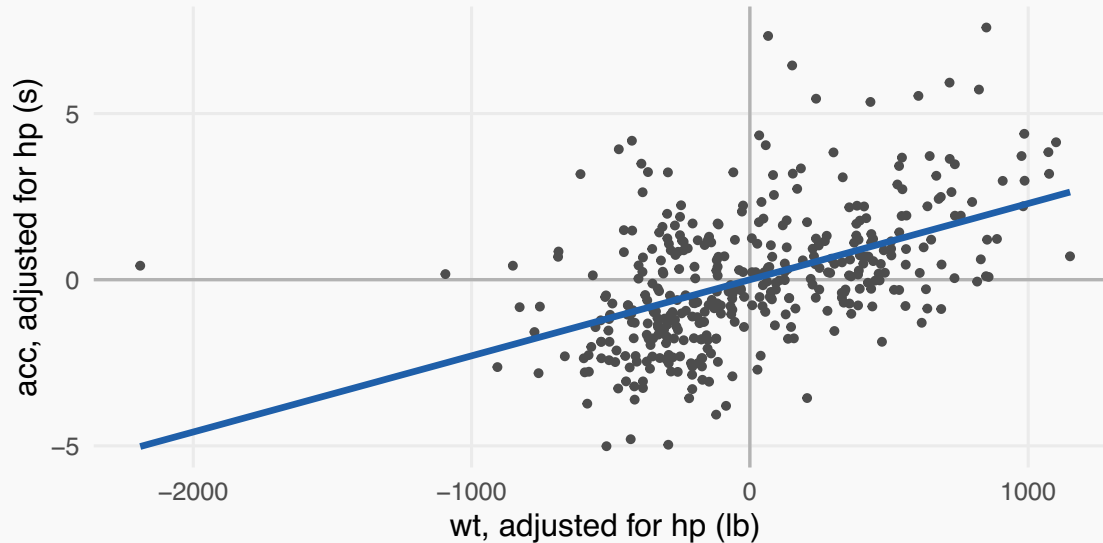
$\hat{\beta}_{wt}$ is the effect of the part of weight that horsepower does not explain: the effect of weight *adjusted for* horsepower.

The same idea, in a picture



The slope of the blue line is $\hat{\beta}_{\text{wt}} = 0.00229$ s per lb.

The same idea, in a picture



The slope of the blue line is $\hat{\beta}_{wt} = 0.00229$ s per lb.

If the two predictors were uncorrelated, adjusting would change nothing: the multiple and simple regression slopes would coincide.

The multiple regression model

The model with $p - 1$ predictors

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{p-1} X_{i,p-1} + \epsilon_i, \quad i = 1, \dots, n,$$

with errors that are uncorrelated, with mean 0 and common variance σ^2 (and normal, for inference).

- There are p parameters $\beta_0, \dots, \beta_{p-1}$, and we need $n > p$ observations.
- Simple linear regression is the case $p = 2$.

The model with $p - 1$ predictors

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{p-1} X_{i,p-1} + \epsilon_i, \quad i = 1, \dots, n,$$

with errors that are uncorrelated, with mean 0 and common variance σ^2 (and normal, for inference).

- There are p parameters $\beta_0, \dots, \beta_{p-1}$, and we need $n > p$ observations.
- Simple linear regression is the case $p = 2$.

Interpretation.

- β_k is the change in the mean of Y per unit increase of X_k , **all the other predictors held fixed**.
- β_0 is the mean of Y when every predictor is 0: meaningful only if that is within the scope of the data.

The model in matrix form

$$\underset{n \times 1}{\mathbf{Y}} = \underset{n \times p}{\mathbf{X}} \underset{p \times 1}{\boldsymbol{\beta}} + \underset{n \times 1}{\boldsymbol{\epsilon}}, \quad \mathbf{X} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1,p-1} \\ 1 & X_{21} & \cdots & X_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n1} & \cdots & X_{n,p-1} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}.$$

The model in matrix form

$$\underset{n \times 1}{\mathbf{Y}} = \underset{n \times p}{\mathbf{X}} \underset{p \times 1}{\boldsymbol{\beta}} + \underset{n \times 1}{\boldsymbol{\epsilon}}, \quad \mathbf{X} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1,p-1} \\ 1 & X_{21} & \cdots & X_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n1} & \cdots & X_{n,p-1} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}.$$

Each column of $\mathbf{X}$ after the first holds the values of one predictor.

The model in matrix form

$$\underset{n \times 1}{\mathbf{Y}} = \underset{n \times p}{\mathbf{X}} \underset{p \times 1}{\boldsymbol{\beta}} + \underset{n \times 1}{\boldsymbol{\epsilon}}, \quad \mathbf{X} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1,p-1} \\ 1 & X_{21} & \cdots & X_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n1} & \cdots & X_{n,p-1} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}.$$

Each column of $\mathbf{X}$ after the first holds the values of one predictor.

As before, $E(\boldsymbol{\epsilon}) = \mathbf{0}$ and $\sigma^2\{\boldsymbol{\epsilon}\} = \sigma^2\mathbf{I}$, so that

$$E(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}, \quad \sigma^2\{\mathbf{Y}\} = \sigma^2\mathbf{I},$$

and with normal errors, $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$.

Least squares

Least squares minimizes

$$Q(\beta) = \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1})^2 = (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta).$$

Least squares

Least squares minimizes

$$Q(\beta) = \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1})^2 = (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta).$$

Expanding the product,

$$Q(\beta) = \mathbf{Y}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}\beta + \beta'\mathbf{X}'\mathbf{X}\beta$$

Least squares

Least squares minimizes

$$Q(\beta) = \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_{p-1} X_{i,p-1})^2 = (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta).$$

Expanding the product,

$$Q(\beta) = \mathbf{Y}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}\beta + \beta'\mathbf{X}'\mathbf{X}\beta = \mathbf{Y}'\mathbf{Y} - 2\beta'\mathbf{X}'\mathbf{Y} + \beta'\mathbf{X}'\mathbf{X}\beta,$$

since $\mathbf{Y}'\mathbf{X}\beta$ is a number, equal to its own transpose $\beta'\mathbf{X}'\mathbf{Y}$.

$$\frac{\partial Q(\beta)}{\partial \beta} = 0 - 2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\beta = 0$$

The least squares estimator

We use, without proof, two rules for derivatives with respect to the vector β :

$$\frac{\partial}{\partial \beta} \beta' \mathbf{a} = \mathbf{a}, \quad \frac{\partial}{\partial \beta} \beta' \mathbf{A} \beta = 2\mathbf{A} \beta \quad \text{for a symmetric } \mathbf{A}.$$

The least squares estimator

We use, without proof, two rules for derivatives with respect to the vector β :

$$\frac{\partial}{\partial \beta} \beta' \mathbf{a} = \mathbf{a}, \quad \frac{\partial}{\partial \beta} \beta' \mathbf{A} \beta = 2\mathbf{A}\beta \quad \text{for a symmetric } \mathbf{A}.$$

Setting the derivative of Q to zero:

$$\frac{\partial Q}{\partial \beta} = -2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\beta = \mathbf{0} \implies \mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{X}'\mathbf{Y},$$

the **normal equations**, one for each parameter.

The least squares estimator

We use, without proof, two rules for derivatives with respect to the vector β :

$$\frac{\partial}{\partial \beta} \beta' a = a, \quad \frac{\partial}{\partial \beta} \beta' A \beta = 2A\beta \quad \text{for a symmetric } A.$$

Setting the derivative of Q to zero:

$$\frac{\partial Q}{\partial \beta} = -2X'Y + 2X'X\beta = 0 \implies X'X\hat{\beta} = X'Y,$$

the **normal equations**, one for each parameter.

If $X'X$ is invertible,

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

exactly as in simple linear regression. That this is a minimum, and not only a critical point, we take for granted.

When is $X'X$ invertible?

When X has **full column rank**: no column of X is a linear combination of the others. In words, no predictor is an exact linear function of the other predictors.

When is $X'X$ invertible?

When X has **full column rank**: no column of X is a linear combination of the others. In words, no predictor is an exact linear function of the other predictors.

Example. Add the weight in kilograms to a model that already has it in pounds:

```
autompg$wt_kg <- 0.4536 * autompg$wt
coef(lm(acc ~ wt + wt_kg + hp, data = autompg))
```

```
## (Intercept)          wt          wt_kg          hp
## 18.469592413  0.002291864          NA -0.093258670
```

When is $X'X$ invertible?

When X has **full column rank**: no column of X is a linear combination of the others. In words, no predictor is an exact linear function of the other predictors.

Example. Add the weight in kilograms to a model that already has it in pounds:

```
autompg$wt_kg <- 0.4536 * autompg$wt
coef(lm(acc ~ wt + wt_kg + hp, data = autompg))
```

```
## (Intercept)          wt          wt_kg          hp
## 18.469592413  0.002291864          NA -0.093258670
```

The column `wt_kg` is 0.4536 times the column `wt`: $X'X$ is singular, and R drops the redundant predictor (NA).

When is $X'X$ invertible?

When X has **full column rank**: no column of X is a linear combination of the others. In words, no predictor is an exact linear function of the other predictors.

Example. Add the weight in kilograms to a model that already has it in pounds:

```
autompg$wt_kg <- 0.4536 * autompg$wt
coef(lm(acc ~ wt + wt_kg + hp, data = autompg))
```

```
##      (Intercept)              wt              wt_kg              hp
## 18.469592413    0.002291864              NA -0.093258670
```

The column `wt_kg` is 0.4536 times the column `wt`: $X'X$ is singular, and R drops the redundant predictor (NA).

When predictors are *nearly* linear functions of each other, $X'X$ is nearly singular: this is multicollinearity, later in the course.

What $X'X$ and $X'Y$ contain

With two predictors ($p = 3$), every entry is a sum over the observations:

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} n & \sum X_{i1} & \sum X_{i2} \\ \sum X_{i1} & \sum X_{i1}^2 & \sum X_{i1}X_{i2} \\ \sum X_{i2} & \sum X_{i1}X_{i2} & \sum X_{i2}^2 \end{bmatrix}, \quad \mathbf{X}'\mathbf{Y} = \begin{bmatrix} \sum Y_i \\ \sum X_{i1}Y_i \\ \sum X_{i2}Y_i \end{bmatrix}.$$

What $X'X$ and $X'Y$ contain

With two predictors ($p = 3$), every entry is a sum over the observations:

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} n & \sum X_{i1} & \sum X_{i2} \\ \sum X_{i1} & \sum X_{i1}^2 & \sum X_{i1}X_{i2} \\ \sum X_{i2} & \sum X_{i1}X_{i2} & \sum X_{i2}^2 \end{bmatrix}, \quad \mathbf{X}'\mathbf{Y} = \begin{bmatrix} \sum Y_i \\ \sum X_{i1}Y_i \\ \sum X_{i2}Y_i \end{bmatrix}.$$

In general, $\mathbf{X}'\mathbf{X}$ is $p \times p$ and symmetric, with entry (j, k) equal to $\sum_i X_{ij}X_{ik}$, and $\mathbf{X}'\mathbf{Y}$ is $p \times 1$.

In R: the acceleration data

```
X <- model.matrix(fit2)           # 390 x 3
XtX <- t(X) %*% X
XtX

##               (Intercept)                wt                hp
## (Intercept)           390           1162338           40784
## wt                1162338  3745687164  132580385
## hp                40784   132580385   4843412

beta_hat <- solve(XtX) %*% t(X) %*% automp$acc
cbind(manual_fit = drop(beta_hat), lm = coef(fit2))

##               manual_fit                lm
## (Intercept) 18.469592413 18.469592413
## wt          0.002291864  0.002291864
## hp          -0.093258670 -0.093258670
```

Fitted values and residuals

Exactly as in simple linear regression:

$$\hat{Y} = X\hat{\beta} = HY, \quad H = X(X'X)^{-1}X', \quad e = Y - \hat{Y} = (I - H)Y.$$

Fitted values and residuals

Exactly as in simple linear regression:

$$\hat{Y} = X\hat{\beta} = HY, \quad H = X(X'X)^{-1}X', \quad e = Y - \hat{Y} = (I - H)Y.$$

The estimate of σ^2 divides the residual sum of squares by $n - p$:

$$SSE = e'e, \quad MSE = \frac{SSE}{n - p}.$$

We estimated p parameters, and each one costs a degree of freedom: $n - 2$ in simple linear regression.

Fitted values and residuals

Exactly as in simple linear regression:

$$\hat{Y} = X\hat{\beta} = HY, \quad H = X(X'X)^{-1}X', \quad e = Y - \hat{Y} = (I - H)Y.$$

The estimate of σ^2 divides the residual sum of squares by $n - p$:

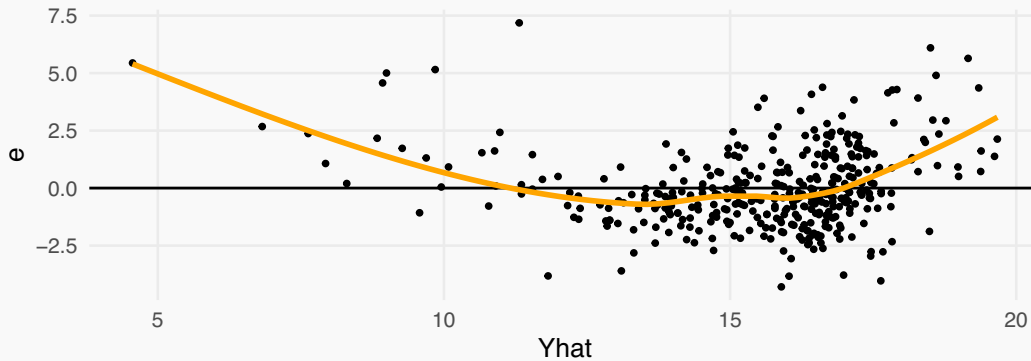
$$SSE = e'e, \quad MSE = \frac{SSE}{n - p}.$$

We estimated p parameters, and each one costs a degree of freedom: $n - 2$ in simple linear regression.

For the acceleration model, $MSE = 3.034$ on $390 - 3 = 387$ degrees of freedom, and $\sqrt{MSE} = 1.742$ s: the “Residual standard error” of `summary(fit2)`.

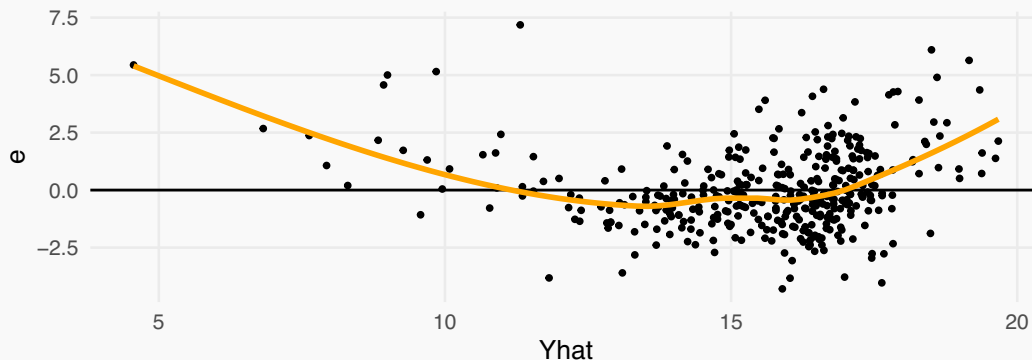
In R: fitted values and residuals

```
Yhat <- drop(X %*% beta_hat)
e <- autmpg$acc - Yhat
ggplot(data.frame(Yhat, e), aes(Yhat, e)) +
  geom_hline(yintercept = 0) + geom_point(size = 0.7) +
  geom_smooth(method = "loess", se = FALSE,
             colour = "orange")
```



In R: fitted values and residuals

```
Yhat <- drop(X %*% beta_hat)
e <- autmpg$acc - Yhat
ggplot(data.frame(Yhat, e), aes(Yhat, e)) +
  geom_hline(yintercept = 0) + geom_point(size = 0.7) +
  geom_smooth(method = "loess", se = FALSE,
             colour = "orange")
```



Positive residuals at both ends, negative in the middle: as in Lecture 8, a **curve** questions linearity. The plane misses some curvature; polynomial terms, later, will deal with it.

Properties of $\hat{\beta}$

The derivation of Lecture 9 never used the number of columns of $\mathbf{X}$, so it holds for any p :

$$E(\hat{\beta}) = \beta, \quad \sigma^2\{\hat{\beta}\} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \quad (p \times p), \quad s^2\{\hat{\beta}\} = MSE(\mathbf{X}'\mathbf{X})^{-1}.$$

Properties of $\hat{\beta}$

The derivation of Lecture 9 never used the number of columns of $\mathbf{X}$, so it holds for any p :

$$E(\hat{\beta}) = \beta, \quad \sigma^2\{\hat{\beta}\} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \quad (p \times p), \quad s^2\{\hat{\beta}\} = MSE(\mathbf{X}'\mathbf{X})^{-1}.$$

With normal errors, $\hat{\beta}$ is a linear function of a normal vector, so it is normal:

$$\hat{\beta} \sim N(\beta, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}).$$

Properties of $\hat{\beta}$

The derivation of Lecture 9 never used the number of columns of $\mathbf{X}$, so it holds for any p :

$$E(\hat{\beta}) = \beta, \quad \sigma^2\{\hat{\beta}\} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \quad (p \times p), \quad s^2\{\hat{\beta}\} = MSE(\mathbf{X}'\mathbf{X})^{-1}.$$

With normal errors, $\hat{\beta}$ is a linear function of a normal vector, so it is normal:

$$\hat{\beta} \sim N(\beta, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}).$$

Each line of the table of `summary(fit2)` gives $\hat{\beta}_k$, its standard error $s\{\hat{\beta}_k\}$, the square root of a diagonal entry of $s^2\{\hat{\beta}\}$, and $t^* = \hat{\beta}_k/s\{\hat{\beta}_k\}$, compared with a t distribution on $n - p$ degrees of freedom.

Properties of the fitted values

$\hat{Y} = X\hat{\beta}$ is a linear function of $\hat{\beta}$, with the constant matrix X :

Properties of the fitted values

$\hat{Y} = \mathbf{X}\hat{\beta}$ is a linear function of $\hat{\beta}$, with the constant matrix $\mathbf{X}$:

$$E(\hat{Y}) = \mathbf{X} E(\hat{\beta}) = \mathbf{X}\beta = E(Y),$$

Properties of the fitted values

$\hat{Y} = X\hat{\beta}$ is a linear function of $\hat{\beta}$, with the constant matrix X :

$$E(\hat{Y}) = X E(\hat{\beta}) = X\beta = E(Y),$$

$$\sigma^2\{\hat{Y}\} = X \sigma^2(X'X)^{-1} X' = \sigma^2 H.$$

Properties of the fitted values

$\hat{Y} = \mathbf{X}\hat{\beta}$ is a linear function of $\hat{\beta}$, with the constant matrix $\mathbf{X}$:

$$E(\hat{Y}) = \mathbf{X} E(\hat{\beta}) = \mathbf{X}\beta = E(Y),$$

$$\sigma^2\{\hat{Y}\} = \mathbf{X} \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' = \sigma^2 \mathbf{H}.$$

In particular $Var(\hat{Y}_i) = \sigma^2 h_{ii}$, with h_{ii} the i th diagonal entry of $\mathbf{H}$.

Properties of the fitted values

$\hat{Y} = X\hat{\beta}$ is a linear function of $\hat{\beta}$, with the constant matrix X :

$$E(\hat{Y}) = X E(\hat{\beta}) = X\beta = E(Y),$$

$$\sigma^2\{\hat{Y}\} = X \sigma^2(X'X)^{-1} X' = \sigma^2 H.$$

In particular $Var(\hat{Y}_i) = \sigma^2 h_{ii}$, with h_{ii} the i th diagonal entry of H .

The variance of the residuals, $\sigma^2\{e\}$, is yours to find in Assignment 2.

The mean response at a new point

At predictor values $X_{h1}, \dots, X_{h,p-1}$, write $\mathbf{X}_h = (1, X_{h1}, \dots, X_{h,p-1})'$. The mean response $E(Y_h) = \mathbf{X}_h' \boldsymbol{\beta}$ is estimated by

$$\hat{Y}_h = \mathbf{X}_h' \hat{\boldsymbol{\beta}}.$$

The mean response at a new point

At predictor values $X_{h1}, \dots, X_{h,p-1}$, write $\mathbf{X}_h = (1, X_{h1}, \dots, X_{h,p-1})'$. The mean response $E(Y_h) = \mathbf{X}_h' \boldsymbol{\beta}$ is estimated by

$$\hat{Y}_h = \mathbf{X}_h' \hat{\boldsymbol{\beta}}.$$

Again a linear function of $\hat{\boldsymbol{\beta}}$, with the $1 \times p$ constant matrix $\mathbf{X}_h'$:

$$E(\hat{Y}_h) = \mathbf{X}_h' \boldsymbol{\beta}, \quad \text{Var}(\hat{Y}_h) = \sigma^2 \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h,$$

estimated by

$$s^2\{\hat{Y}_h\} = \text{MSE} \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h.$$

The mean response at a new point

At predictor values $X_{h1}, \dots, X_{h,p-1}$, write $\mathbf{X}_h = (1, X_{h1}, \dots, X_{h,p-1})'$. The mean response $E(Y_h) = \mathbf{X}_h' \boldsymbol{\beta}$ is estimated by

$$\hat{Y}_h = \mathbf{X}_h' \hat{\boldsymbol{\beta}}.$$

Again a linear function of $\hat{\boldsymbol{\beta}}$, with the $1 \times p$ constant matrix $\mathbf{X}_h'$:

$$E(\hat{Y}_h) = \mathbf{X}_h' \boldsymbol{\beta}, \quad \text{Var}(\hat{Y}_h) = \sigma^2 \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h,$$

estimated by

$$s^2\{\hat{Y}_h\} = \text{MSE} \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h.$$

With one predictor, $\mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h = \frac{1}{n} + \frac{(X_h - \bar{X})^2}{S_{xx}}$: the formula of Lecture 6.

Predicting a new observation

A new observation at $\mathbf{X}_h$ is $Y_{h(\text{new})} = \mathbf{X}_h' \boldsymbol{\beta} + \epsilon_{h(\text{new})}$, with a new error uncorrelated with those of the sample.

Predicting a new observation

A new observation at $\mathbf{X}_h$ is $Y_{h(\text{new})} = \mathbf{X}_h' \boldsymbol{\beta} + \epsilon_{h(\text{new})}$, with a new error uncorrelated with those of the sample.

The prediction error has mean 0 and variance

$$\text{Var}(Y_{h(\text{new})} - \hat{Y}_h) = \sigma^2 + \sigma^2 \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h = \sigma^2 (1 + \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h),$$

estimated by $s^2\{\text{pred}\} = \text{MSE} (1 + \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h)$.

Predicting a new observation

A new observation at $\mathbf{X}_h$ is $Y_{h(\text{new})} = \mathbf{X}_h' \boldsymbol{\beta} + \epsilon_{h(\text{new})}$, with a new error uncorrelated with those of the sample.

The prediction error has mean 0 and variance

$$\text{Var}(Y_{h(\text{new})} - \hat{Y}_h) = \sigma^2 + \sigma^2 \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h = \sigma^2 (1 + \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h),$$

estimated by $s^2\{\text{pred}\} = \text{MSE} (1 + \mathbf{X}_h' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_h)$.

The first σ^2 is the new car's own error: it does not shrink as n grows.

In R: a car of 3000 lb and 100 hp

`vcov(fit2)` is $MSE(\mathbf{X}'\mathbf{X})^{-1}$ and `sigma(fit2)^2` is MSE .

```
Xh <- c(1, 3000, 100)
Yh <- sum(Xh * coef(fit2))
s2_Yh <- drop(t(Xh) %*% vcov(fit2) %*% Xh)
c(Yh = Yh, s_Yh = sqrt(s2_Yh),
  s_pred = sqrt(sigma(fit2)^2 + s2_Yh))

##           Yh           s_Yh           s_pred
## 16.01931725  0.09152615  1.74435944

new_car <- data.frame(wt = 3000, hp = 100)
predict(fit2, new_car, se.fit = TRUE)$se.fit

## [1] 0.09152615
```

In R: a car of 3000 lb and 100 hp

`vcov(fit2)` is $MSE(\mathbf{X}'\mathbf{X})^{-1}$ and `sigma(fit2)^2` is MSE .

```
Xh <- c(1, 3000, 100)
Yh <- sum(Xh * coef(fit2))
s2_Yh <- drop(t(Xh) %*% vcov(fit2) %*% Xh)
c(Yh = Yh, s_Yh = sqrt(s2_Yh),
  s_pred = sqrt(sigma(fit2)^2 + s2_Yh))

##           Yh           s_Yh           s_pred
## 16.01931725  0.09152615  1.74435944

new_car <- data.frame(wt = 3000, hp = 100)
predict(fit2, new_car, se.fit = TRUE)$se.fit

## [1] 0.09152615
```

The mean time to 60 mph of such cars is estimated very precisely ($s\{\hat{Y}_h\} = 0.092$ s), one car's much less ($s\{\text{pred}\} = 1.74$ s).

Summary: estimates and their precision

Estimates	Estimator	Estimated variance
β	$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$	$MSE (\mathbf{X}'\mathbf{X})^{-1}$
$E(Y_h) = \mathbf{X}'_h\beta$	$\hat{Y}_h = \mathbf{X}'_h\hat{\beta}$	$MSE \mathbf{X}'_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_h$
$Y_{h(\text{new})}$	$\hat{Y}_h$	$MSE (1 + \mathbf{X}'_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_h)$
σ^2	$MSE = SSE/(n - p)$	

Summary: estimates and their precision

Estimates	Estimator	Estimated variance
β	$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$	$MSE (\mathbf{X}'\mathbf{X})^{-1}$
$E(Y_h) = \mathbf{X}'_h\beta$	$\hat{Y}_h = \mathbf{X}'_h\hat{\beta}$	$MSE \mathbf{X}'_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_h$
$Y_{h(\text{new})}$	$\hat{Y}_h$	$MSE (1 + \mathbf{X}'_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_h)$
σ^2	$MSE = SSE/(n - p)$	

Next: confidence and prediction intervals, the analysis of variance and R^2 with several predictors.