

# Linear Regression (part 2)

Predictive Modeling & Statistical Learning

Gaston Sanchez

CC BY-SA 4.0

# Multiple Linear Regression by Ordinary Least Squares

# Regression with various predictors

- ▶ Multiple Linear Regression
- ▶  $p$  predictors  $x_1, x_2, \dots, x_p$
- ▶ one response variable  $y$
- ▶ Do not confuse *Multiple* with *Multivariate*
- ▶ Multivariate Regression implies several responses (i.e.  $y_1, \dots, y_q$ )

# Introduction

Suppose we observe a quantitative response  $Y$  and  $p$  different predictors,  $X_1, X_2, \dots, X_p$ .

We assume a linear relationship of the form:

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + \varepsilon$$

# Advertising Data

```
# file in folder data/ of github repo  
Advertising <- read.csv("data/Advertising.csv", row.names = 1)
```

```
Warning in file(file, "rt"): cannot open file  
'../..data/Advertising.csv': No such file or directory  
Error in file(file, "rt"): cannot open the connection  
Error in head(Advertising, n = 8): object 'Advertising'  
not found
```

(first 8 rows)

# Data set Advertising

Response:

- ▶  $Y$ : Sales

Predictors:

- ▶  $X_1$ : TV
- ▶  $X_2$ : Radio
- ▶  $X_3$ : Newspaper

Linear model:

$$\text{Sales} = \beta_0 + \beta_1 \text{TV} + \beta_2 \text{Radio} + \beta_3 \text{Newspaper} + \epsilon$$

## Some vector-matrix notation

Given the actual data values, we may write the model as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

for  $i = 1, \dots, n$

## Some vector-matrix notation

Given the actual data values, we may write the model as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

for  $i = 1, \dots, n$

It will be more convenient to use vector-matrix notation:

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



## Some vector-matrix notation

If we consider an intercept term  $\beta_0$ , then we have:

$$\underset{n \times 1}{\mathbf{y}} = \underset{n \times (p+1)}{\mathbf{X}} \times \underset{(p+1) \times 1}{\boldsymbol{\beta}} + \underset{n \times 1}{\boldsymbol{\epsilon}}$$

which can also be represented by:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ 1 & x_{31} & \cdots & x_{3p} \\ \vdots & \ddots & \vdots & \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

## Some vector-matrix notation

If the data is **mean-centered** (i.e.  $\bar{X}_1 = \dots = \bar{X}_p = 0$ )

$$\underset{n \times 1}{\mathbf{y}} = \underset{n \times p}{\mathbf{X}} \times \underset{p \times 1}{\boldsymbol{\beta}} + \underset{n \times 1}{\boldsymbol{\varepsilon}}$$

which can also be represented by:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ x_{31} & \cdots & x_{3p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

# OLS Estimation

# OLS Estimation

Assuming a linear model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

the challenge involves finding parameter estimates denoted by  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  that provide the “best” approximation for  $Y$ :

$$Y \approx \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p$$

or more commonly

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p$$

# Matrix Notation

Model

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

Estimation: fitted (or predicted) values

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}\mathbf{b}$$

Residuals: observed - predicted

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$$

# Matrix Notation

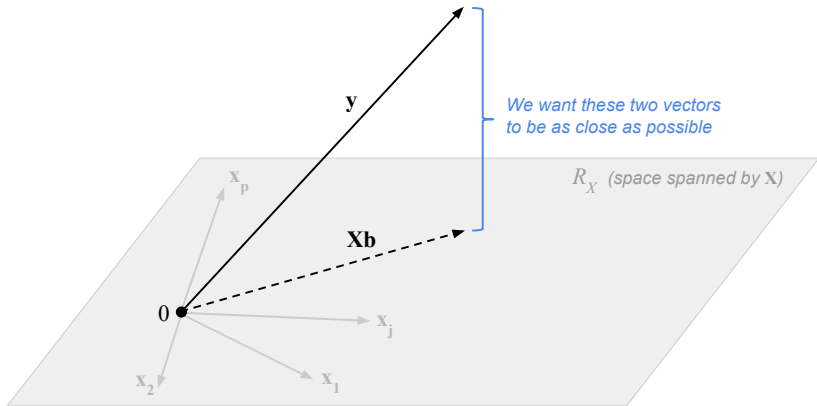
We want to calculate  $\mathbf{b} = \hat{\boldsymbol{\beta}}$  such that  $\hat{\mathbf{y}}$  is a good approximation of  $\mathbf{y}$ .

The idea is to choose  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  that minimize the *size* of the residuals:

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\mathbf{b}$$

What criteria should be used to minimize the *size* of the residuals?

# Geometric illustration



# Matrix Notation

Our wish is to minimize the residuals for all  $i = 1, 2, \dots, n$ :

$$e_i = y_i - \hat{y}_i$$

Among the the possible criteria to minimize we have:

- ▶  $\min \{\sum_{i=1}^n e_i^2\}$   $L_2$ -norm
- ▶  $\min \{\sum_{i=1}^n |e_i|\}$   $L_1$ -norm
- ▶  $\min \{\max(e_i)\}$   $L_\infty$ -norm
- ▶ *etc*



# Matrix Notation

Least Squares involves minimizing the sum of squares ( $L_2$ -norm):

$$\min \left\{ \sum_{i=1}^n e_i^2 \right\}$$

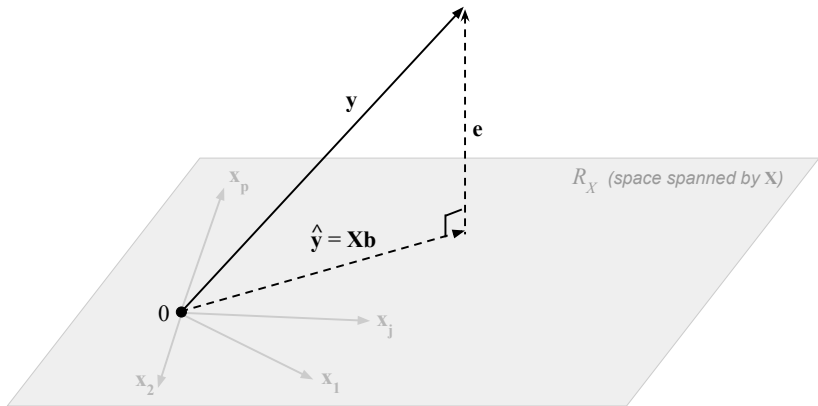
This sum is better known as the Residual Sum of Squares ( $RSS$ )

$$RSS = \sum_{i=1}^n e_i^2$$

In vector-matrix notation:

$$RSS = \mathbf{e}^T \mathbf{e} = \|\mathbf{e}\|^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2$$

# Least Squares Geometry



# OLS Geometric Idea

## Geometrically speaking

- ▶ the response lies in an  $n$ -dimensional space:  $\mathbf{y} \in \mathbb{R}^n$
- ▶ the vector of parameters lies in a  $p$ -dimensional space:  
 $\boldsymbol{\beta} \in \mathbb{R}^p$
- ▶ in OLS, the response is projected orthogonally onto the model space spanned by  $\mathbf{X}$
- ▶ the fit is represented by projection  $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$
- ▶ the difference between the fit and the data is the residual vector  $\mathbf{e}$
- ▶ the residual vector lies in an  $(n - p)$ -dimensional space:  
 $\mathbf{b} \in \mathbb{R}^{(n-p)}$

# Least Squares Minimization

# OLS Minimization

OLS Criterion:

$$\min \left\{ \sum_{i=1}^n e_i^2 \right\} = \min \{ \|\mathbf{e}\|^2 \}$$

This means that the “best”  $\mathbf{b}$  is the one which minimizes the *RSS*:

$$RSS(\mathbf{b}) = \sum_{i=1}^n e_i^2 = \mathbf{e}^T \mathbf{e} = (\mathbf{y} - \mathbf{X}\mathbf{b})^T (\mathbf{y} - \mathbf{X}\mathbf{b})$$

# OLS Minimization

Differentiating  $RSS(\mathbf{b})$  with respect to  $\mathbf{b}$  yields:

$$\frac{RSS(\mathbf{b})}{\partial \mathbf{b}} = -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \mathbf{b}$$

# OLS Minimization

Differentiating  $RSS(\mathbf{b})$  with respect to  $\mathbf{b}$  yields:

$$\frac{RSS(\mathbf{b})}{\partial \mathbf{b}} = -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \mathbf{b}$$

Equating to zero we have the so-called *normal equations*:

$$\mathbf{X}^T \mathbf{X} \mathbf{b} = \mathbf{X}^T \mathbf{y}$$

# OLS Minimization

Assuming that the matrix  $\mathbf{X}^T\mathbf{X}$  is nonsingular (invertible), the unique ordinary least squares (OLS) estimator of  $\beta$  is given by:

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



# OLS Minimization

Assuming that the matrix  $\mathbf{X}^T\mathbf{X}$  is nonsingular (invertible), the unique ordinary least squares (OLS) estimator of  $\beta$  is given by:

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

The fitted values are

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

What conditions are needed for  $\mathbf{X}^T\mathbf{X}$  to be invertible?

# Example: Advertising Data

```
# number of observations
n <- nrow(Advertising)

## Error in nrow(Advertising): object 'Advertising' not
found

# model matrix
X <- as.matrix(Advertising[, c('TV', 'Radio', 'Newspaper')])

## Error in as.matrix(Advertising[, c("TV", "Radio",
"Newspaper")]): object 'Advertising' not found

X <- cbind(Intercept = rep(1, n), X)

## Error in cbind(Intercept = rep(1, n), X): object 'n'
not found

# response variable
y <- Advertising$Sales

## Error in eval(expr, envir, enclos): object
```

# Example: Advertising Data

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

```
# coefficients
b <- solve(t(X) %*% X) %*% t(X) %*% y

## Error in t(X): object 'X' not found

b

## Error in eval(expr, envir, enclos): object 'b' not
found
```

# Example: Advertising Data

Predicted (fitted) values:

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

```
# fitted  
fitted <- X %*% b  
  
## Error in eval(expr, envir, enclos): object 'X' not  
found
```

# Observed -vs- Predicted (fitted) values

```
## Error in plot(y, fitted, las = 1, type = "n", xlab = "y  
observed", ylab = "y fitted"): object 'y' not found  
## Error in int_abline(a = a, b = b, h = h, v = v, untf =  
untf, ...): plot.new has not been called yet  
## Error in points(y, fitted, pch = 19, col = "#55555555",  
cex = 1.5): object 'y' not found  
## Error in text.default(20, 24, label = "identity line",  
col = "#88888855"): plot.new has not been called yet
```

# OLS Minimization

The fitted values are

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

Let  $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ , then:

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$$

where  $\mathbf{H}$  is commonly known as the **hat matrix**.

What's so special about  $\mathbf{H}$ ?

# Example: Advertising Data

$$\hat{y} = Hy$$

```
# equivalent with the Hat matrix
H <- X %*% solve(t(X) %*% X) %*% t(X)

## Error in eval(expr, envir, enclos): object 'X' not
found

y_hat <- H %*% y

## Error in eval(expr, envir, enclos): object 'H' not
found
```

# Review: projection Matrices

Let  $L \subseteq \mathbb{R}^n$  be a **linear subspace**, i.e.  $L = \text{span}\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$  for some  $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbb{R}^n$ .

If  $V \in \mathbb{R}^{n \times k}$  contains  $\mathbf{v}_1, \dots, \mathbf{v}_k$  on its columns, then

$$\text{span}\{\mathbf{v}_1, \dots, \mathbf{v}_k\} = \{a_1\mathbf{v}_1 + \dots + a_k\mathbf{v}_k : a_1, \dots, a_k \in \mathbb{R}\} = \text{col}(V)$$

The function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  that projects points onto  $L$  is called the **projection map** onto  $L$ . This is actually a linear function,  $F(\mathbf{x}) = P_L \mathbf{x}$ , where  $P_L \in \mathbb{R}^{n \times n}$  is the **projection matrix** onto  $L$ .



# Review: projection Matrices

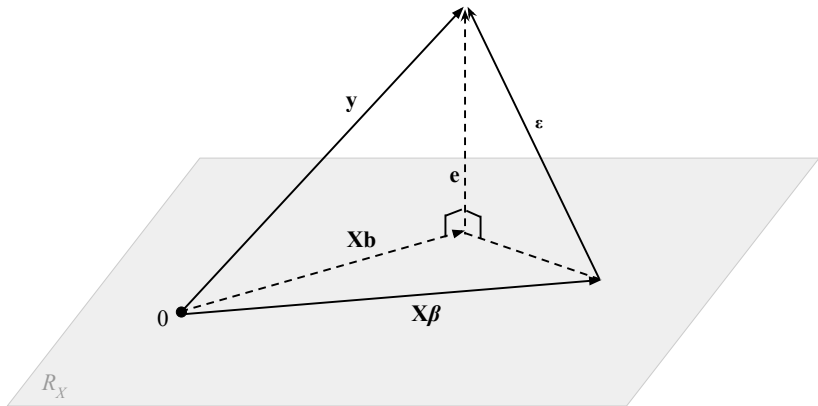
A projection matrix  $P_L \in \mathbb{R}^{n \times n}$

- ▶ is a linear transformation
- ▶ is symmetric:  $P_L = P_L^T$
- ▶ is idempotent:  $P_L^2 = P_L$
- ▶ Furthermore:
  - $P_L \mathbf{x} = \mathbf{x}$  for all  $\mathbf{x} \in L$
  - $P_L \mathbf{x} = \mathbf{0}$  for all  $\mathbf{x} \perp L$

# The Hat matrix

- ▶  $\mathbf{H}$  is a linear transformation
- ▶  $\mathbf{H}$  is symmetric:  $\mathbf{H} = \mathbf{H}^T$
- ▶  $\mathbf{H}$  is idempotent:  $\mathbf{H} = \mathbf{H}^2$
- ▶ The hat matrix is an **orthogonal projector** or *projection matrix*
- ▶  $\mathbf{Q} = \mathbf{I} - \mathbf{H}$  is the orthogonal complement or “counterpart” of  $\mathbf{H}$

# Least Squares Geometry

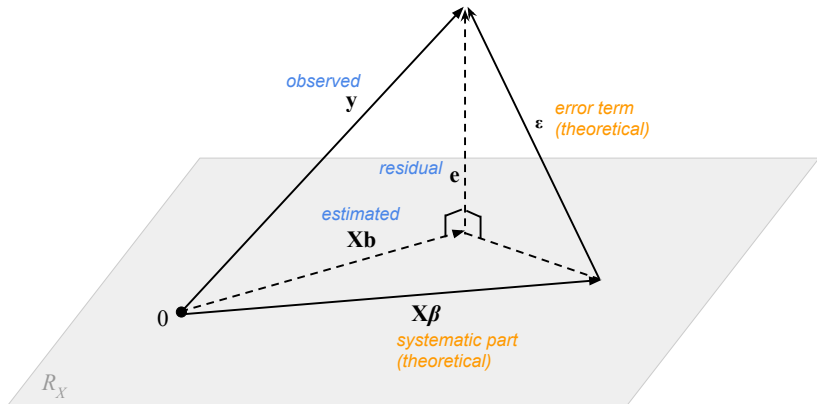


# About the Hat matrix $\mathbf{H}$

The theoretical model  $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$  defines a decomposition of  $\mathbf{y}$  in two unknown terms:

- ▶  $\mathbf{X}\boldsymbol{\beta} \in \mathbb{R}_X$
- ▶  $\boldsymbol{\varepsilon} \in \mathbb{R}^n$

# Least Squares Geometry



# About the Hat matrix $\mathbf{H}$

The theoretical model  $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$  defines a decomposition of  $\mathbf{y}$  in two unknown terms:

- ▶  $\mathbf{X}\boldsymbol{\beta} \in \mathbb{R}_X$
- ▶  $\boldsymbol{\varepsilon} \in \mathbb{R}^n$

# About the Hat matrix $\mathbf{H}$

The theoretical model  $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$  defines a decomposition of  $\mathbf{y}$  in two unknown terms:

- ▶  $\mathbf{X}\boldsymbol{\beta} \in \mathbb{R}_X$
- ▶  $\boldsymbol{\varepsilon} \in \mathbb{R}^n$

The OLS method proposes a solution  $\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$  that minimizes the “length” of the residual vector  $\mathbf{e}$  by orthogonally projecting  $\mathbf{y}$  as  $\mathbf{X}\mathbf{b}$  in the spanned space of  $\mathbf{X}$ , and by projecting  $\boldsymbol{\varepsilon}$  as  $\mathbf{e}$  in the subspace to  $\mathbb{R}_X$ .

# Residuals and Theoretical Errors

The *residuals*  $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$  are the OLS estimates of the unobservable errors  $\boldsymbol{\varepsilon}$ .

The residual vector can also be written as:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{b} = (\mathbf{I} - \mathbf{H})\boldsymbol{\varepsilon} = \mathbf{Q}\boldsymbol{\varepsilon}$$



## Example: Advertising Data

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{b} = (\mathbf{I} - \mathbf{H})\boldsymbol{\varepsilon} = \mathbf{Q}\boldsymbol{\varepsilon}$$

```
# residuals  
residuals <- y - y_hat  
  
## Error in eval(expr, envir, enclos): object 'y' not  
found
```

# Computation

# OLS Solution

The vector of OLS estimates  $\mathbf{b}$  is given by  $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

In R, you may calculate  $\mathbf{b}$  with something like this:

```
# beta coefficients  
XtXi <- solve(t(X) %*% X)  
b <- XtXi %*% t(X) %*% y
```

# OLS Solution

Although this works, computationally it is not the best way to compute  $\mathbf{b}$ .

Most computer programs don't compute  $(\mathbf{X}^T \mathbf{X})^{-1}$  directly. Instead, they typically use the QR decomposition.

# QR Decomposition

Any matrix  $\mathbf{X}$  can be written as:

$$\mathbf{X} = \mathbf{Q}\mathbf{R}$$

where:

- ▶  $\mathbf{Q}$  is an  $n \times p$  orthogonal matrix:  
 $\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}$
- ▶  $\mathbf{R}$  is a  $p \times p$  upper triangular matrix

# OLS solution via QR Decomposition

$$\begin{aligned}\mathbf{b} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= ((\mathbf{QR})^T \mathbf{QR})^{-1} (\mathbf{QR})^T \mathbf{y} \\ &= (\mathbf{R}^T \mathbf{Q}^T \mathbf{QR})^{-1} \mathbf{R}^T \mathbf{Q}^T \mathbf{y} \\ &= (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T \mathbf{Q}^T \mathbf{y} \\ &= \mathbf{R}^{-1} (\mathbf{R}^{-T} \mathbf{R}^T) \mathbf{Q}^T \mathbf{y} \\ &= \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{y}\end{aligned}$$

# OLS solution via QR Decomposition

$$\begin{aligned}\mathbf{b} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{y}\end{aligned}$$

- ▶ we don't really want to invert  $\mathbf{R}$
- ▶ we just want to recognize that we have a new system:

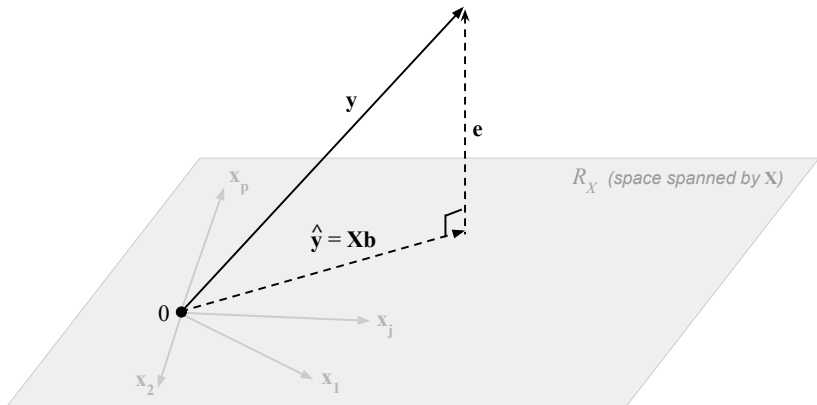
$$\mathbf{R}\mathbf{b} = \mathbf{Q}^T \mathbf{y}$$

- ▶ In practice you apply some backsubstitution routine to solve such system (you'll do that in the lab)

# Assessing the Quality of the Fit



# Assessing the quality of the fit



Assuming that the data is mean-centered, then the lengths of the vectors in  $\mathbb{R}^n$  can be interpreted in term of variances.

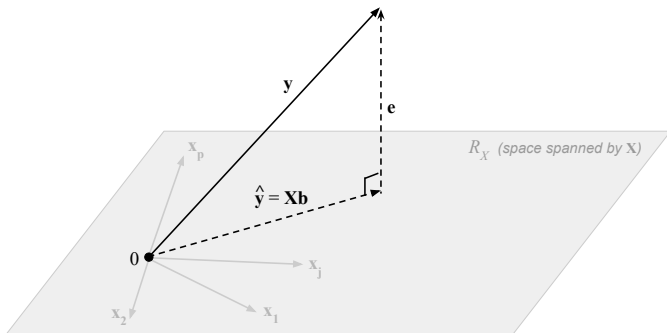
The Pythagoras theorem applied to the square triangle can be written as:

$$\mathbf{y}^T \mathbf{y} = \mathbf{e}^T \mathbf{e} + \mathbf{b}^T \mathbf{X}^T \mathbf{X} \mathbf{b}$$

equivalently:

$$\|\mathbf{y}\|^2 = \|\mathbf{X}\mathbf{b}\|^2 + \|\mathbf{e}\|^2$$

# Assessing the quality of the fit



$$\|\mathbf{y}\|^2 = \|\mathbf{X}\mathbf{b}\|^2 + \|\mathbf{e}\|^2$$

The Pythagoras theorem:

$$\mathbf{y}^T \mathbf{y} = \mathbf{e}^T \mathbf{e} + \mathbf{b}^T \mathbf{X}^T \mathbf{X} \mathbf{b}$$

can be reexpressed as:

$$\sum (y_i)^2 = \sum (y_i - \hat{y}_i)^2 + \sum (\hat{y}_i)^2$$

Dividing by  $n$ , we put things in terms of variances:

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

# Variance Decomposition

$$\underbrace{\frac{1}{n} \sum (y_i)^2}_{\text{total variance}} = \underbrace{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}_{\text{residual variance}} + \underbrace{\frac{1}{n} \sum (\hat{y}_i)^2}_{\text{explained variance}}$$

# Multiple Correlation Coefficient

We define the **coefficient of multiple correlation** as

$$R^2 = \text{cor}^2(\mathbf{y}, \hat{\mathbf{y}}) = \text{cor}^2(\mathbf{y}, \mathbf{X}\mathbf{b})$$

$R^2$  can be expressed in various forms:

$$R^2 = \frac{\text{cov}^2(\mathbf{y}, \hat{\mathbf{y}})}{\text{var}(\mathbf{y})\text{var}(\hat{\mathbf{y}})} = \frac{\text{var}(\hat{\mathbf{y}})}{\text{var}(\mathbf{y})} = \frac{\text{explained variance}}{\text{total variance}}$$

# Multiple Correlation

$$R^2 = \text{cor}^2(\mathbf{y}, \hat{\mathbf{y}}) = \text{cor}^2(\mathbf{y}, \mathbf{X}\mathbf{b})$$

```
# coefficient of multiple correlation
R2 <- cor(y, y_hat)

## Error in is.data.frame(y):  object 'y_hat' not found

R2

## Error in eval(expr, envir, enclos):  object 'R2' not
found
```

$R^2$  is the proportion of the variability in  $y$  explained by the model

# Multiple Correlation Coefficient

$R^2$  describes the fraction of the total variance of  $\mathbf{y}$  that is explained by  $\hat{\mathbf{y}}$

By minimizing  $\sum_{i=1}^n e_i^2$ , we actually maximize  $R^2$ .

What does this mean?



# Multiple Correlation Coefficient

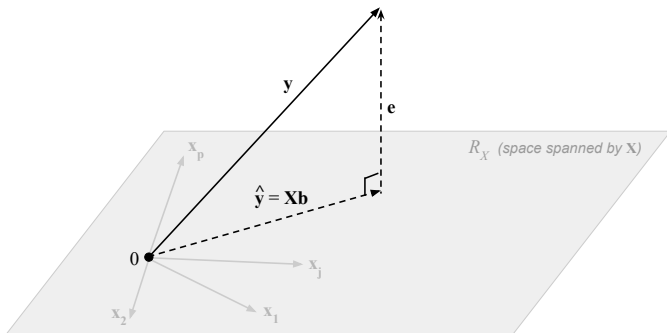
$R^2$  describes the fraction of the total variance of  $\mathbf{y}$  that is explained by  $\hat{\mathbf{y}}$

By minimizing  $\sum_{i=1}^n e_i^2$ , we actually maximize  $R^2$ .

What does this mean?

In other words, the OLS fit provides a linear combination of the predictors that has maximum correlation with the response variable  $\mathbf{y}$ .

# Assessing the quality of the fit



$$\|\mathbf{y}\|^2 = \|\mathbf{X}\mathbf{b}\|^2 + \|\mathbf{e}\|^2$$

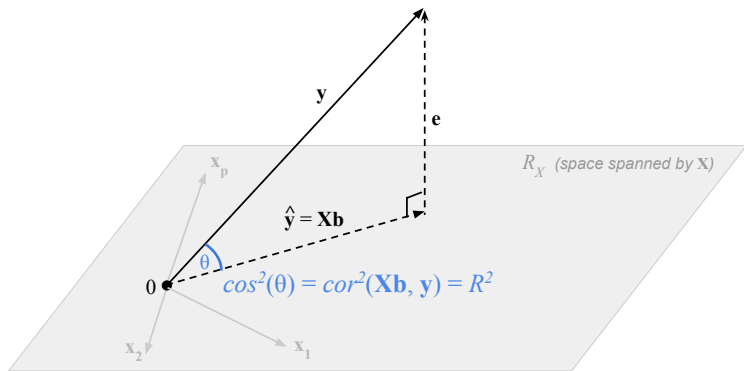
# Assessing the quality of the fit

$$\|\mathbf{y}\|^2 = \|\hat{\mathbf{y}}\|^2 + \|\mathbf{e}\|^2$$

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n e_i^2$$

$$R^2 = \frac{\|\hat{\mathbf{y}}\|^2}{\|\mathbf{y}\|^2} = \cos^2(\mathbf{y}, \hat{\mathbf{y}})$$

# Assessing the quality of the fit



## About $R^2$

- ▶  $R^2$  is one way to measure the quality of the fit.
- ▶ It doesn't tell you how accurate the coefficients are.
- ▶ It is a measure of *resubstitution error*.  
(not of generalization error)
- ▶ It depends on the number of predictors  $p$ .
- ▶ It is interesting from the theoretical-geometric point of view.
- ▶ But in practice it does not say much about the predictive performance of a model.

# Some Comments

- ▶ There is nothing in the Least Squares method that requires statistical inference: formal tests of null hypotheses or confidence intervals.
- ▶ In its simplest form, regression analysis can be performed without statistical inference.
- ▶ We will study the inferential framework in the next slides.

# References

- ▶ **An Introduction to Statistical Learning** by James et al (2013). Springer.
- ▶ **Statistical Regression and Classification** by Norman Matloff (2017). CRC Press.
- ▶ **Linear Models with R** by Julian J. Faraway (2015). CRC Press.
- ▶ **Modern Regression Methods** by Thomas Ryan (2009). Wiley.
- ▶ **A Modern Approach to Regression with R** by Simon Sheather (2009). Springer.
- ▶ **Modern Multivariate Statistical Techniques** by Julian J. Izenman (2008). Springer.
- ▶ **Data Mining and Statistics for Decision Making** by Stephane Tuffery (2011). *Chapter 11: Classification and prediction methods.* Wiley.

# References (French Literature)

- ▶ **Probabilites, analyse des donnees et statistique** by Gilbert Saporta (2011). *Chapter 17: La regression multiple et le modele lineaire general*. Editions Technip, Paris.
- ▶ **Statistique: Methodes pour decrire, expliquer et prevoir** by Michel Tenenhaus (2008). *Chapter 5: La Regression Multiple*. Dunod, Paris.
- ▶ **Regression avec R** by Cornillon and Matzner-Lober (2011). Springer.
- ▶ **Statistique Exploratoire Multidimensionnelle** by Lebart et al (2004). *Chapter 3, section 3.2: Regression multiple, modele lineaire*. Dunod, Paris.
- ▶ **Traitement des donnees statistiques** by Lebart et al. (1982). *Unit 3: Modele Lineaire*. Dunod, Paris.