Scientific Computing for Biologists

Linear Algebra Review II & Multiple Regression

Instructor: Paul M. Magwene

18 September 2012

Overview of Lecture

- More Linear Algebra
  - Linear combinations and Spanning Spaces
  - Subspaces
  - Basis vectors
  - Dimension
  - Rank
- More on Regression
  - Multiple regression
  - Curvilinear regression
  - Major axis regression

Hands-on Session

- Literate Programming with R and knitr
- Multiple regression

Space Spanned by a List of Vectors

Definition

Let X be a finite list of n-vectors. The **space spanned** by X is the set of all vectors that can be written as linear combinations of the vectors in X.

A space spanned includes the zero vector and is closed under addition and multiplication by a scalar.

Remember that a *linear combination* of vectors is an equation of the form $z = b_1x_1 + b_2x_2 + \cdots + b_px_p$

Subspaces

$\mathbb{R}^n$ denotes the seat of real n-vectors - the set of all $n \times 1$ matrices with entries from the set $\mathbb{R}$ of real numbers.

Definition

A **subspace** of $\mathbb{R}^n$ is a subset S of $\mathbb{R}^n$ with the following properties:

1. $0 \in S$
2. If $u \in S$ then $ku \in S$ for all real numbers k
3. If $u \in S$ and $v \in S$ then $u + v \in S$

Examples of subspaces of $\mathbb{R}^n$:

- any space spanned by a list of vectors in $\mathbb{R}^n$
- the set of all solution to an equation $Ax = 0$ where A is a $p \times n$ matrix, for any number p.

Basis

Let S be a subspace of $\mathbb{R}^n$. Then there is a finite list, X of vectors from S such that S is the space spanned by X.

Let S be a subspace of $\mathbb{R}^n$ spanned by the list $(u_1, u_2, \ldots, u_n)$. Then there is a linearly independent sublist of $(u_1, u_2, \ldots, u_n)$ that also spans S.

Definition

A list X is a **basis** for S if:

- X is linearly independent
- S is the subspace spanned by X

Dimension

Let S be a subspace of $\mathbb{R}^n$.

Definition

The **dimension** of S is the number of elements in a basis for S.

Rank of a Matrix

Let A by an $n \times p$ matrix.

Definition

The **rank** of A is equal to the dimension of the row space of A which is equal to the dimension of the column space of A.

Where the row space of A is the space spanned by the list of rows of A and the column space of A is defined similarly.

Equivalence Theorem

Let A by an $p \times p$ matrix. The following are equivalent

- A is singular
- the rank of A is less than p
- the columns of A form a LD list in $\mathbb{R}^n$.
- the rows of A form a LD list in $\mathbb{R}^n$
- the equation $Ax = 0$ has non-trivial solutions
- the determinant of A is zero

Regression Models

Variable space view of multiple regression

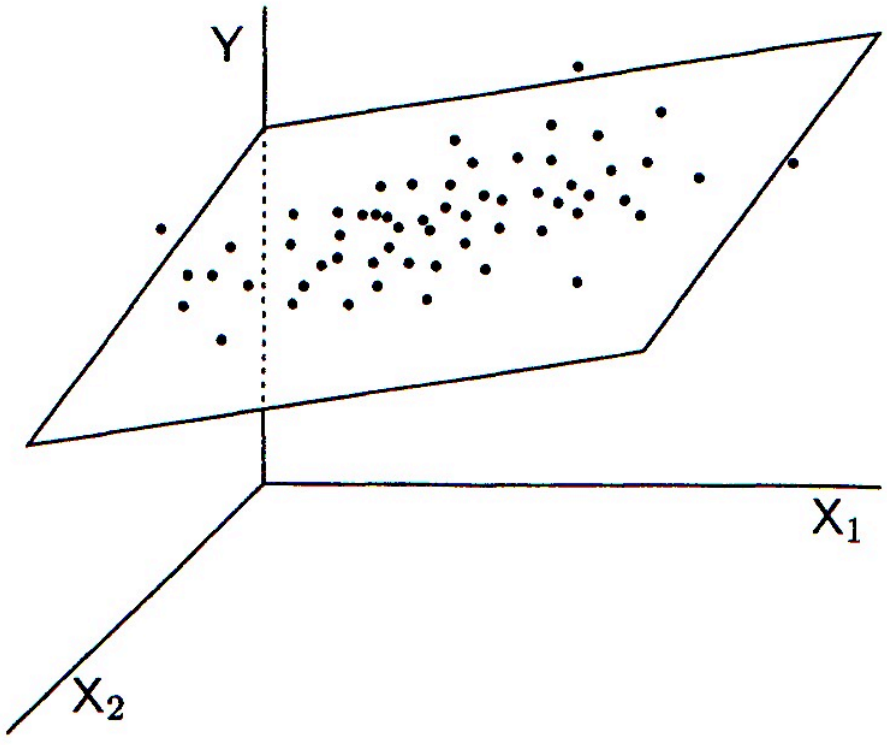
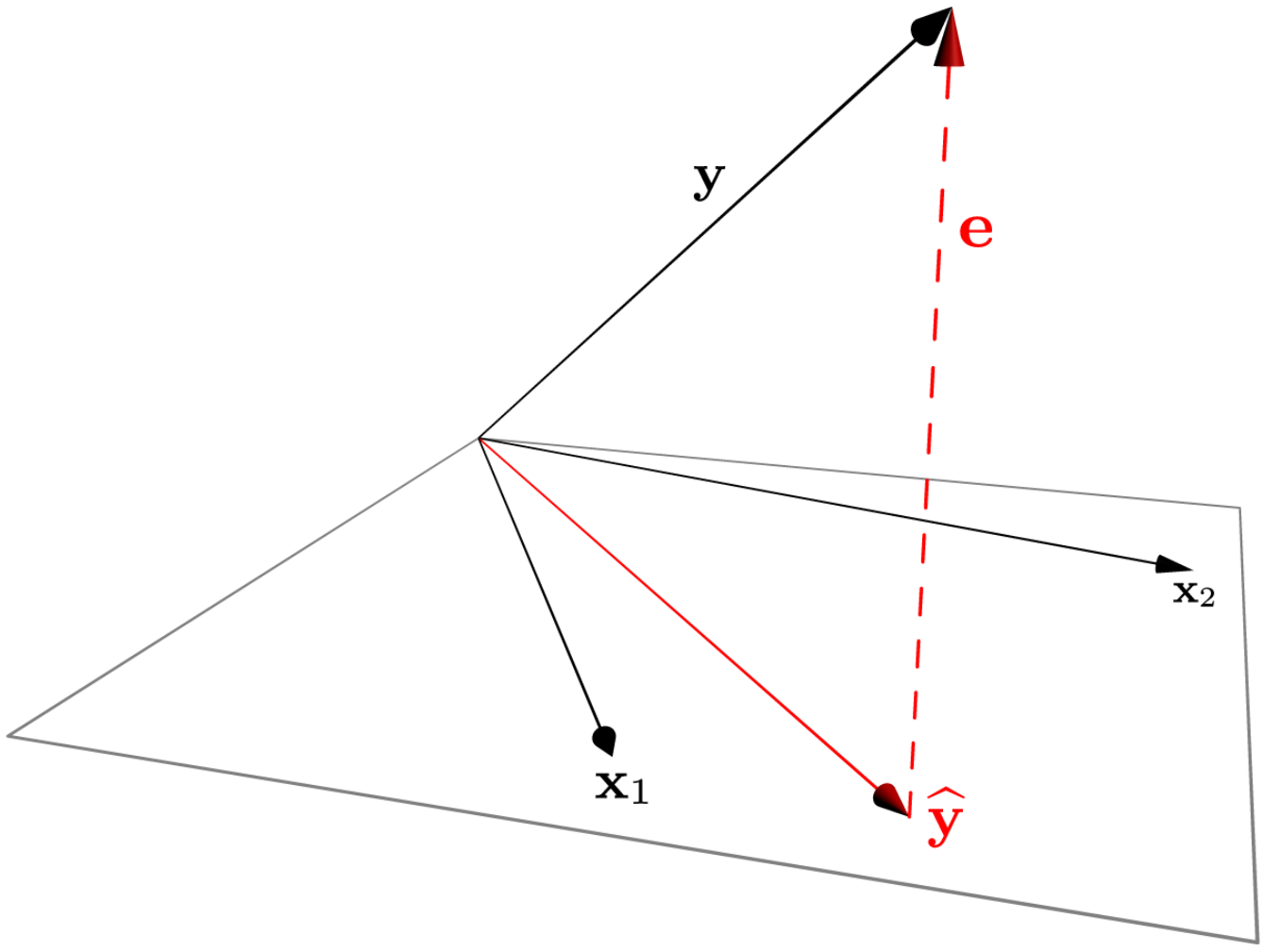


Figure 4.1: *The regression of* Y *onto* X_1 *and* X_2 *as a scatterplot in variable space.*

Subject Space Geometry of Multiple Regression



Multiple Regression

Let Y be a vector of values for the outcome variable. Let X_i be explanatory variables and let x_i be the mean-centered explanatory variables.

$$Y = \hat{Y} + e$$

where –

Uncentered version:

$$\hat{Y} = a1 + b_1X_1 + b_2X_2 + \cdots + b_pX_p$$

Centered version:

$$\hat{y} = b_1x_1 + b_2x_2 + \cdots + b_px_p$$

Statistical Model for Multiple Regression

In matrix form:

$$y = Xb + e$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}; X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix};$$

$$b = \begin{bmatrix} a \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix}; e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

Estimating the Coefficients for Multiple Regression

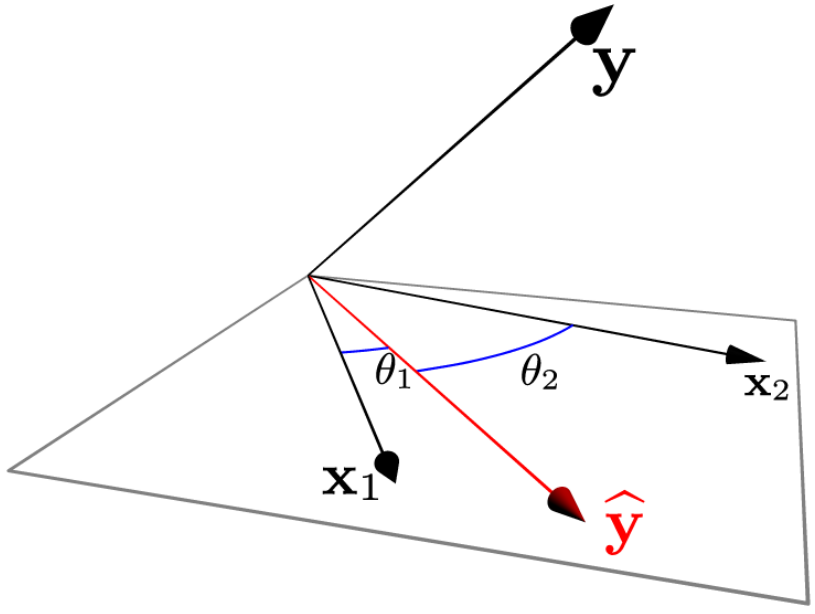
$$y = Xb + e$$

Estimate b as:

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

Multiple Regression Loadings

The regression **loadings** should be examined as well as the regression coefficients.



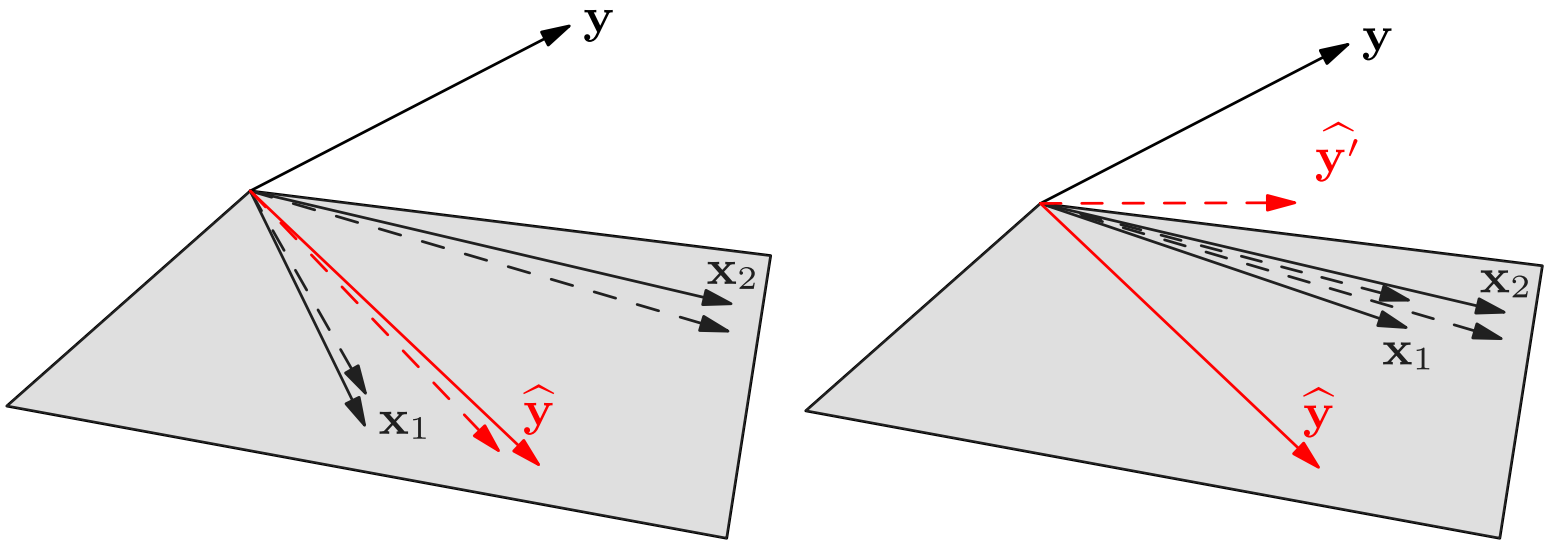
Loadings are given by:

$$\cos\theta_{\vec{x_j},\vec{\hat{y}}} = \frac{\vec{x_j}\cdot\vec{\hat{y}}}{|\vec{x_j}||\vec{\hat{y}}|}$$

Multiple regression: Cautions and Tips

- Comparing the size of regression coefficients only makes sense if all the predictor variables have the same scale
- The predictor variables (columns of X) must be linearly independent; when they're not the variables are **multicollinear**
- Predictor variables that are **nearly multicollinear** are, perhaps, even more difficult to deal with

Why is near multicollinearity of the predictors a problem?



(a) Non-collinear predictors (b) Nearly collinear predictors

Figure: When predictors are nearly collinear, small differences in the vectors can result in large differences in the estimated regression.

What can I do if my predictors are (nearly) collinear?

- Drop some of the linearly dependent sets of predictors.
- Replace the linearly dependent predictors with a combined variable.
- Define orthogonal predictors, via linear combinations of the original variables (PC regression approach)
- 'Tweak' the predictor variables so that they're no longer multicollinear (Ridge regression).

Curvilinear Regression

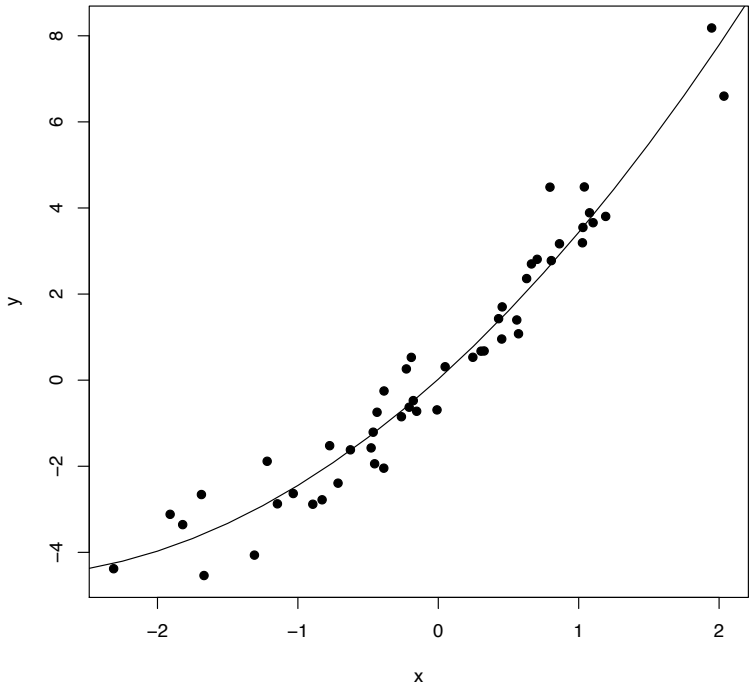
Curvilinear regression using **polynomial models** is simply multiple regression with the x_i replace by powers of x.

$$\hat{y} = b_1 x + b_2 x^2 + \cdots + b_p x^n$$

Note:

- this is still a *linear* regression (linear in the coefficients)
- best applied when a specific hypothesis justifies there use
- generally not higher than quadratic or cubic

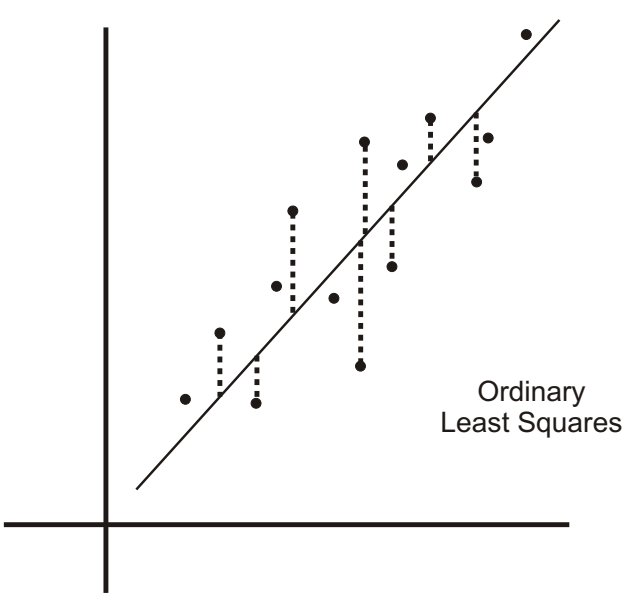
Example of Curvilinear Regression

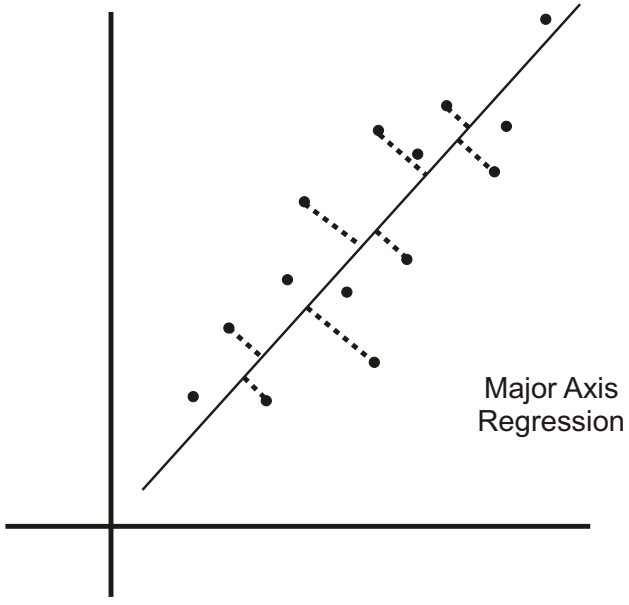


$$y = 3x + 0.5x^2 + e$$

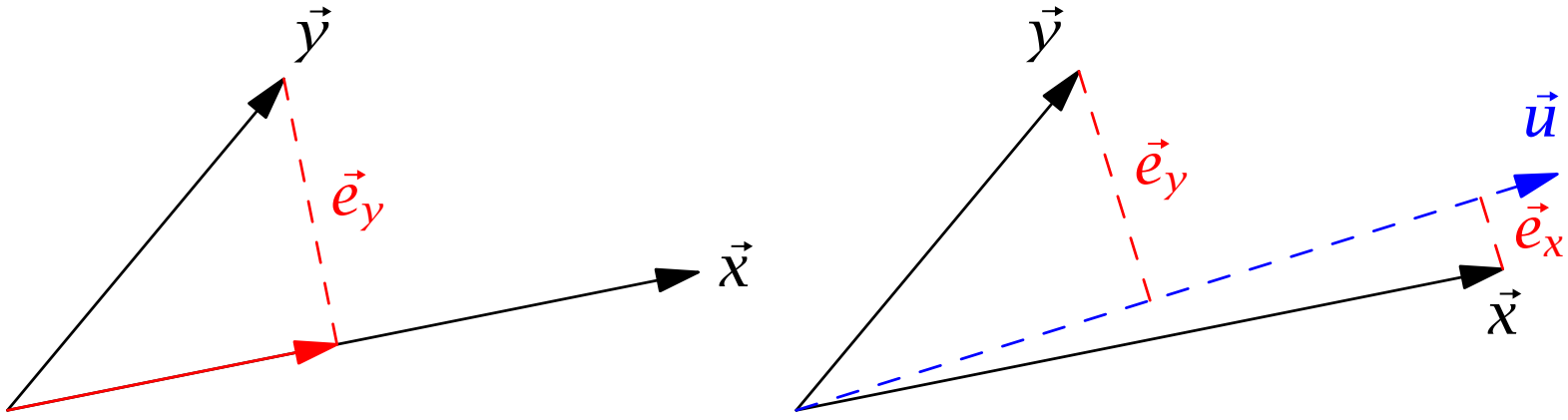
```
lm(formula = y ~ x + I(x^2))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.02229    0.11651   0.191    0.849
x            2.94001    0.09693  30.331  < 2e-16 ***
I(x^2)       0.47146    0.07685   6.135 1.68e-07 ***
```

Least Squares Regression vs. Major Axis Regression





Vector Geometry of Major Axis Regression



(a) OLS (b) Major Axis Regression

Figure: Vector geometry of ordinary least-squares and major axis regression.

Literate Programming

“Literate programming” is a concept coined by Donald Knuth, a preeminent computer scientist:

- Programs are useless with descriptions.
- Descriptions should be literate, not comments in code or typical reference manuals.
- The code in the descriptions should work.

Literate Programming and Reproducible Research

How literate programming can help to ensure your research is reproducible:

- The steps of your analyses are explicitly described, both as written text and the code and function calls used.
- Analyses can easily checked for correctness and reproduced from your literate code.
- Your literate code can serve as a template for future analyses, saving you time and the trouble of remembering all the gory details.

Tools for literate programming in R

- Sweave – works together with LaTeX to produce output.
- knitr – recent tool developed as a successor to Sweave. More flexible options for input and output; better formatting of figures, etc.

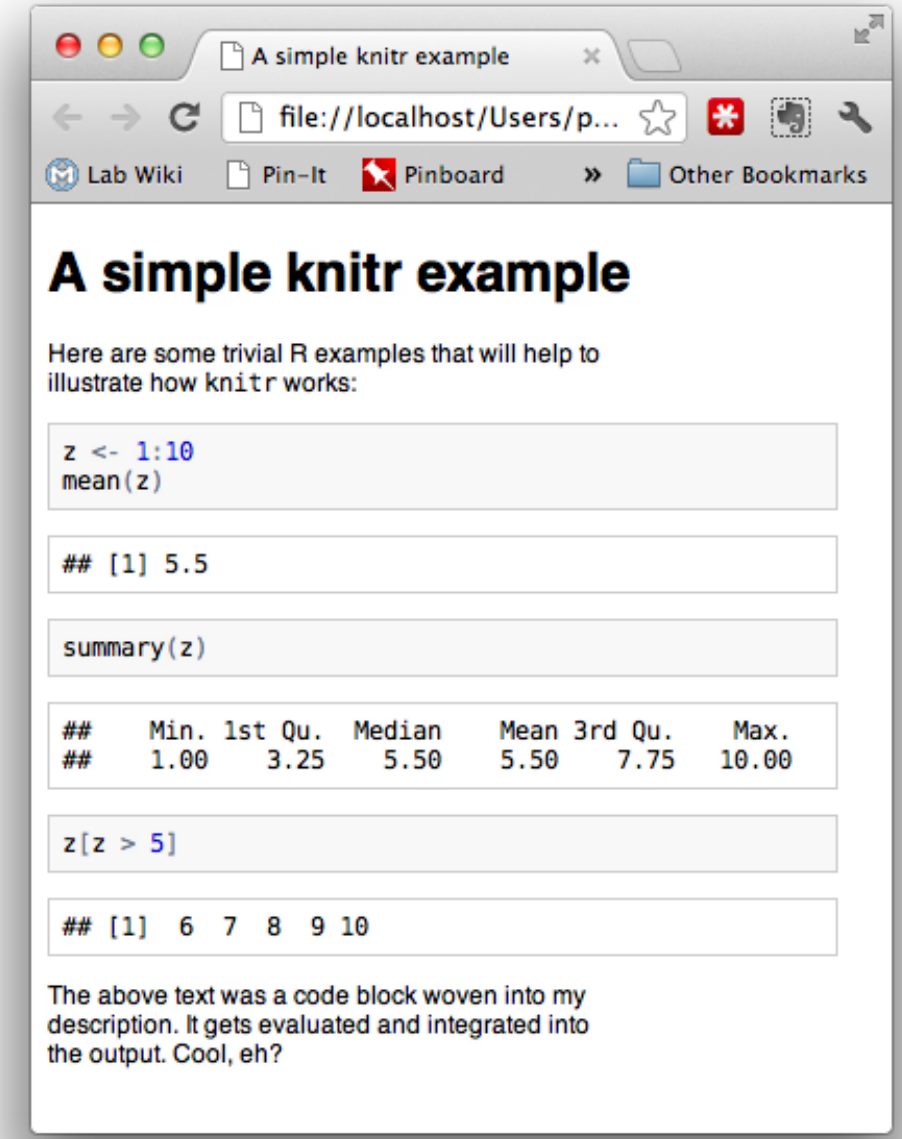
Literate Programming with R Markdown

knitr allows you to use Markdown, a simple markup syntax, to 'weave together' textual descriptions and R code. R code is delimited by so 'code chunks' setoff with three backticks, as illustrated below:

````
# A simple example

Here are some trivial R examples that will help to
illustrate how Sweave works:

```{r}
z <- 1:10
mean(z)
summary(z)
z[z > 5]
```

The above text was a code block woven into my
description. It gets evaluated and integrated into
the output. Cool, eh?
````

knitr output

HTML output produced by knitr for the code on the previous slide:



Fancier knitr output

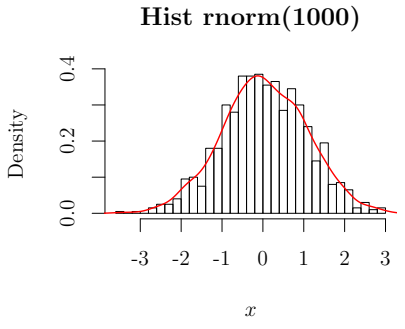
If you go to the trouble of learning to use LaTeX you can generate even nicer output with knitr (code highlighting, better figure formatting):

A knitr example that incorporates graphics is always nice. First, let's generate the data by drawing 1000 observations from the standard normal ($\mu = 0, \sigma = 1$).

```
data <- rnorm(1000)  # 1000 obs. drawn from standard normal
```

Next, we create a summary table:

```
summary(data)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -3.470  -0.656   0.004   0.035   0.755   2.930
```

Finally, we create a nice figure in which a density estimate is superimposed on a histogram:

```
hist(data, breaks = 40, freq = F, main = "Hist rnorm(1000)", xlab = "$x$")
lines(density(data), col = "red", lwd = 2)
```