

```
% A line starting with a % is a comment

% clear will erase all the variables that Matlab currently hold in memory
clear;

% This would close everything but Matlab itself: opened files, graphic windows, ...
close all;

% We can create variables
x = 4;

% Contrary to SAS, a command does not need to be terminated by a ;
% If we do not put a ; at the end, the results are displayed in
% the output window
y = 5

% We can create vectors. The [ and ] signs are telling Matlab that what's
% in between are elements of a vector (or matrix).
% A space is treated as a comma: it tells Matlab to stack the numbers in a row
% A ; tells Matlab to stack the number in a column
a = [1 2 3 4]
b = [5, 6, 7, 8]
c = [9; 10; 11; 12]

% We can stack vectors to create a matrix
d = [a; b]

% We can create a matrix directly
e = [1, 2, 3, 4; 5, 6, 7, 8]

% you can create matrices of zeros, ones or an identity matrix
e = zeros(4,5)
e = ones(3,2)
e = eye(3)

% We can add, subtract, multiply and divide numbers
f = 1+2
f = 1-2
f = 1*2
f = 1/2

% We can multiply matrices and vectors
f = d*c

% We can add and subtract matrices and vectors
f = a-b

% We can invert a matrix
g = [10, 3; 3, 5]
h = inv(g)
g*h
```

```
% We can use ... to put one statement over multiple lines
i = [1, 2, 3, ...
     4, 5, 6]

% We can transpose vectors and matrices with '
i = i'

% On vectors and matrices we can do element-by-element operations
% (multiplications and divisions) using the dot operators
f = a .* b
f = a ./ b

% We can read a data file in text format (columns separated by spaces) using
% the load function
j = load('baseball.csv')

% I can select the subset of a matrix. For example, rows 4 to 6 and columns
% 2 and 3
k = j(4:6,2:3)

% Now I want to regress the number of wins on an intercept and the payroll
% (measured in millions of dollars)
y = j(:,2);

N = length(y); % Matlab is case-sensitive. You can have a variable named 'N'
               % and another variable named 'n'
x = zeros(N,2);
x(:,1) = ones(N,1);
x(:,2) = j(:,1)/1000000;

betaHat = inv(x'*x)*x'*y

% I can compute the residuals
uHat = y - x*betaHat;

% I can estimate the variance of the error term
sigma2Hat = sum(uHat.^2)/(N-2);

% I can compute the variance matrix of betaHat
varBetaHat = sigma2Hat*inv(x'*x)

% The standard errors are
stdBetaHat = sqrt(diag(varBetaHat))

% I can compute the p-value for the hypothesis that the payroll coefficient
% is equal to zero (2-sided)
t = (betaHat(2)-0)/stdBetaHat(2)
pValue = 2*(1-normcdf(abs(t)))

% We can do loops using the "for" statement. Quick example: adding the
```

```
% numbers from 10 to 20
```

```
total = 0;
```

```
for i=10:1:20
```

```
    i
```

```
    total = total+i
```

```
end
```

```
total
```