Assignment 3 **2.086/2.090 Spring 2013**

Released: *Wednesday, 20 March, at 5 PM.*

Due: *Monday, 8 April, at 5 PM.*

Upload your solution as a zip file "`YOURNAME_ASSIGNMENT_3`" which includes the script for each question *as well as* all MATLAB functions (of your own creation) called by your scripts; both scripts and functions must conform to the formats described in **Instructions** and **Questions** below. You should also include in your folder all the `grade_o_matic .p` files for Assignment 3.

Instructions

Before embarking on this assignment you should

(1) Complete the Textbook reading for Unit III

(2) Execute ("cell-by-cell") the MATLAB Tutorial on MATLAB Ma-trix/Vector Operations and Least Squares. (Note Chapter 18 of the textbook also addresses relevant MATLAB issues.)

(3) Download the `Assignment_3_Materials` folder. This folder contains a template for the script associated with each question (`A3Qy_Template` for Question `y`), as well as a template for each function which we ask you to create (`func_Template` for a function `func`). The `Assignment_3_Materials` folder also contains the `grade_o_matic` codes needed for Assignment 3. (Please see Assignment 1 for a description of `grade_o_matic`.)

We indicate here several general format and performance requirements:

(a.) Your script for Question `y` of Assignment `x` *must* be a proper MATLAB ".m" script file and *must* be named `AxQy.m`. In some cases the script will be trivial and you may submit the template "as is" — just remove the `_Template` — in your "`YOURNAME_ASSIGNMENT_3` folder. But note that you still must submit a proper `AxQy.m` script or `grade_o_matic_A3` will not perform correctly.

(b.) In this assignment, for each question `y`, we will specify inputs and outputs both for the script `A3Qy` and (as is more traditional) any requested MATLAB functions; we shall denote the former as script inputs and script outputs and the latter as function inputs and function outputs. For each question and hence each script, and also each function, we will identify *allowable instances* for the inputs — the parameter values or "parameter domains" for which the codes must work.

(c.) Recall that for scripts, input variables must be assigned *outside* your script (of course before the script is executed) — *not* inside your script — in the workspace; all other variables required by the script must be defined *inside* the script. Hence you should test your scripts in the following fashion: `clear` the workspace; assign the input variables in the workspace; run your script. Note for MATLAB functions you need not take such precautions: all inputs and outputs are passed through the input and output argument lists; a function enjoys a private workspace.

(d.) We ask that in the submitted version of your scripts and functions you suppress all display by placing a ";" at the end of each line of code. (Of course during debugging you will often choose to display many intermediate and final results.) We also require that **before** you upload your solution you should run `grade_o_matic_A3` (from your `YOURNAME_ASSIGNMENT_3` folder) for final confirmation that all is in order.

Questions

1. (10 points) *Preamble: You are not to use* MATLAB *for this question (except of course for the multiple-choice script file* `AxQy.m` *for* `grade_o_matic_A3`*) either to identify or confirm the correct result — you should develop your answer without recourse to a computer or even a calculator. The point of this question is to make sure that you understand the basic matrix operations.*

   You are given a matrix A of size 2×3,

   $$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \end{pmatrix},$$

   in MATLAB `A = [1, 2, 3; 0, 1, 1]`, and a second matrix B of size 3×2,

   $$B = \begin{pmatrix} 2 & -1 \\ 0 & 2 \\ 0 & 1 \end{pmatrix},$$

   in MATLAB `B = [2, -1; 0, 2; 0, 1]`. Here the size of a matrix, $m \times n$, refers to the number of rows (m) and number of columns (n) in the matrix.

   (i) (2.5 points) The product $C = AB$

   (a) yields $C = \begin{pmatrix} 2 & 0 & 0 \\ 3 & 2 & 1 \\ 5 & 2 & 1 \end{pmatrix}$

   (b) yields $C = \begin{pmatrix} 2 & 3 & 5 \\ 0 & 2 & 2 \\ 0 & 1 & 1 \end{pmatrix}$

   (c) yields $C = \begin{pmatrix} 2 & 6 \\ 0 & 3 \end{pmatrix}$

   (d) yields $C = \begin{pmatrix} 2 & 0 \\ 6 & 3 \end{pmatrix}$

   (e) can not be performed

   where "can not be performed" means that the operation is not allowed by the rules of matrix algebra.

(ii) (2.5 points) The product $C = BA$

(a) yields $C = \begin{pmatrix} 2 & 0 & 0 \\ 3 & 2 & 1 \\ 5 & 2 & 1 \end{pmatrix}$

(b) yields $C = \begin{pmatrix} 2 & 3 & 5 \\ 0 & 2 & 2 \\ 0 & 1 & 1 \end{pmatrix}$

(c) yields $C = \begin{pmatrix} 2 & 6 \\ 0 & 3 \end{pmatrix}$

(d) yields $C = \begin{pmatrix} 2 & 0 \\ 6 & 3 \end{pmatrix}$

(e) can not be performed

where "can not be performed" means that the operation is not allowed by the rules of matrix algebra.

(iii) (2.5 points) The product $C = (B^{\mathrm{T}} A^{\mathrm{T}})^{\mathrm{T}}$

(a) yields $C = \begin{pmatrix} 2 & 0 & 0 \\ 3 & 2 & 1 \\ 5 & 2 & 1 \end{pmatrix}$

(b) yields $C = \begin{pmatrix} 2 & 3 & 5 \\ 0 & 2 & 2 \\ 0 & 1 & 1 \end{pmatrix}$

(c) yields $C = \begin{pmatrix} 2 & 6 \\ 0 & 3 \end{pmatrix}$

(d) yields $C = \begin{pmatrix} 2 & 0 \\ 6 & 3 \end{pmatrix}$

(e) can not be performed

where "can not be performed" means that the operation is not allowed by the rules of matrix algebra. Recall that $^{\mathrm{T}}$ denotes the transpose.

(*iv*) (2.5 points) The sum $C = A + B$

(*a*) yields $C = \begin{pmatrix} 3 & 2 & 3 \\ -1 & 3 & 2 \end{pmatrix}$

(*b*) yields $C = \begin{pmatrix} 3 & -1 \\ 2 & 3 \\ 3 & 2 \end{pmatrix}$

(*c*) yields $C = \begin{pmatrix} 3 & 1 \\ 0 & 3 \end{pmatrix}$

(*d*) yields $C = \begin{pmatrix} 3 & 0 \\ 1 & 3 \end{pmatrix}$

(*e*) can not be performed

where "can not be performed" means that the operation is not allowed by the rules of matrix algebra.

The template `A3Q1_Template.m` contains the multiple-choice format required by `grade_o_matic_A3`.

2. (*10 points*) *Preamble: You are not to use* MATLAB *for this question (except of course for the multiple-choice script file* `AxQy.m` *for* `grade_o_matic_A3`*) either to identify or confirm the correct result — you should develop your answer without recourse to a computer or even a calculator. The point of this question is to make sure that you understand the basic matrix operations.*

We run the script

```
% begin script

clear
% note we "clear" the workspace

A = zeros(1000,1000);
for i = 1:1000
    A(i,1) = 1.0;
end
A(426,12) = 4.0;
A(12,426) = 3.0;
A(426,426) = -1.0;
A(999,1000) = 5.0;

w = 2*ones(1000,1); % note w is a column vector of all two's

v = A*w;

M = max(v);
```

```
% end script
```

where we recall that `max` is the MATLAB built-in function which returns the maximum of a vector. The questions below refer to the values of the variables after execution of this script.

(*i*) (2.5 points) The value of `v(12)` is

(*a*) 2

(*b*) 4

(*c*) 8

(*d*) 10

(*e*) 12

Hint: Consider the "row interpretation" of the matrix-vector product.

(*ii*) (2.5 points) The value of `v(426)` is

(*a*) 2

(*b*) 4

(*c*) 8

(*d*) 10

(*e*) 12

Hint: Consider the "row interpretation" of the matrix-vector product.

(*iii*) (2.5 points) The value of `v(1000)` is

(*a*) 2

(*b*) 4

(*c*) 8

(*d*) 10

(*e*) 12

Hint: Consider the "row interpretation" of the matrix-vector product.

(*iv*) (2.5 points) The value of `M` is

(*a*) 2

(*b*) 4

(*c*) 8

(*d*) 10

(*e*) 12

Hint: Consider the "column interpretation" of the matrix-vector product.

The template `A3Q2_Template.m` contains the multiple-choice format required by `grade_o_matic_A3`.

3. (*20 points*) We define, for a given integer m, $h = 1/(m-1)$; $x_i = (i-1)h$, $1 \le i \le m$; the $m \times 2$ matrix X,

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix}; \tag{1}$$

and the $m \times 1$ vector Y,

$$Y = 0.1 \begin{pmatrix} \sin \pi x_1 \\ \sin \pi x_2 \\ \vdots \\ \sin \pi x_m \end{pmatrix}. \tag{2}$$

Note $X_{i1} = 1$, $1 \le i \le m$, $X_{i2} = x_i$, $1 \le i \le m$, and $Y_i = 0.1 \sin(\pi x_i)$, $1 \le i \le m$.

We denote by $\hat{\beta} \equiv (\hat{\beta}_0 \ \hat{\beta}_1)^{\mathrm{T}} \in \mathbb{R}^2$ the least-squares solution to the overdetermined system defined by X and Y: $\hat{\beta}$ is the minimizer of $\|r(\beta)\|^2$ over all $\beta \in \mathbb{R}^2$; here $r(\beta) \equiv Y - X\beta$ is the residual. We would like you to write a script which computes $\hat{\beta}$ and also compares $r(\hat{\beta})$ to $r(\beta^{\text{cand}})$ for some given $\beta^{\text{cand}} \in \mathbb{R}^2$ (here the superscript "cand" indicates some "candidate" β other than the least-squares solution).

The script takes two script inputs. The first input is a scalar m which must correspond in your script to MATLAB variable `m`; the set of allowable instances for `m`, or input parameter domain, is the set of integers $1 \le$ `m` ≤ 10000. The second input is a 2×1 (column vector) β^{cand} which must correspond in your script to MATLAB variable `beta_cand`; the set of allowable instances for `beta_cand`, or input parameter domain, is the space of 2×1 vectors. The script yields two script outputs. The first output is the 2×1 vector $\hat{\beta}$ (the least squares solution defined above) which must correspond in your script to MATLAB variable `beta_hat`. The second output is $\delta \equiv \|r(\beta^{\text{cand}})\| - \|r(\hat{\beta})\|$ and must correspond in your script to MATLAB variable `delta`. Finally, note that X and Y should be defined inside your script according to (1) and (2), respectively. The template is provided in `A3Q3_Template`.

Two points: First, we ask that you use the MATLAB built-in function `pi` for π so as not to introduce any unnecessary truncation. Second, you might wish to take advantage of the output `delta` as a first check that your code is performing correctly — in particular, what should be the sign of `delta`?

4. (*20 points*) The stress-strain relation for an isotropic material in the elastic regime is given by Hooke's Law

$$\sigma^{\text{stress}} = E\varepsilon^{\text{strain}} \tag{3}$$

where σ^{stress} is the uniaxial stress (in units of $\mathrm{N/m^2}$), E is the Young's modulus of the material (in $\mathrm{N/m^2}$), and $\varepsilon^{\text{strain}}$ is the strain (dimensionless). We can write this more generally as a model for the stress (our dependent variable) in terms of the strain (our independent variable) and a regression coefficient vector $\beta = (\beta_0 \;\; \beta_1)^{\mathrm{T}}$:

$$\sigma^{\text{stress}}_{\text{model}}(\varepsilon^{\text{strain}}; \beta) = \beta_0 + \beta_1 \varepsilon^{\text{strain}} \; . \tag{4}$$

We assume that this model is *bias-free*: there exists a unique β^{true} such that equation (4) exactly predicts the stress σ^{stress} for any given strain $\varepsilon^{\text{strain}}$; we identify from (3) that β_0^{true} will equal zero and that β_1^{true} will equal the Young's modulus E.

To determine the regression coefficients (and hence the Young's modulus) we provide data: $m \times 1$ arrays `strains_meas` (the values of the strain at which we take stress measurements) and `stresses_meas` (the corresponding measured values of the stress). We assume that the measurements are given by

$$\texttt{stresses\_meas(i)} = \sigma^{\text{stress}}_{\text{model}}(\texttt{strains\_meas(i)}; \beta^{\text{true}}) + \epsilon(\texttt{i}), \quad 1 \le i \le m \; ,$$

where the noise $\epsilon(\texttt{i})$ satisfies our assumptions normal zero-mean (N1), homoscedastic (N2), and uncorrelated (N3). In our example here, $m = 8$.

We shall estimate β^{true} by the least-squares solution $\hat{\beta}$. We recall that $\hat{\beta}$ minimizes

$$\left(\sum_{i=1}^{m} \left(\texttt{stresses\_meas(i)} - \sigma^{\text{stress}}_{\text{model}}(\texttt{strains\_meas(i)}; \beta) \right)^2 \right)^{1/2} , \tag{5}$$

over all possible 2-vectors β.

To obtain the least-squares solution $\hat{\beta}$ (`beta_hat` in MATLAB) we run the script below.[1]

```
% begin script

m = 8;

X = [ones(m,1), EXPR1]; % EXPR1 to be specified in part (i)

beta_hat = X \ EXPR2;    % EXPR2 to be specified in part (ii)

% end script
```

[1]Note that prior to running the script the MATLAB workspace contains *only* `strains_meas` and `stresses_meas`.

(i) (5 points) In the script above, `EXPR1` should be taken as

(a) `stresses_meas`

(b) `strains_meas`

(c) `ones(m,1)`

(d) none of the above

(ii) (5 points) In the script above, `EXPR2` should be taken as

(a) `stresses_meas`

(b) `strains_meas`

(c) `ones(m,1)`

(d) none of the above

Note in the remainder of this question we assume that you have chosen the correct options above such that `beta_hat` of the script is equal to $\hat{\beta}$ which minimizes (5).

(iii) (5 points) Is it possible that

$$\sigma^{\text{stress}}_{\text{model}}(\texttt{strains\_meas(i)};\hat{\beta}) < \texttt{stresses\_meas(i)}$$

at *all* the data points, $1 \leq \texttt{i} \leq m$?

(a) Yes

(b) No

(iv) (5 points) Is it possible that

$$\left(\mathop{}\limits_{i=1}^{m} \left(\texttt{stresses\_meas(i)} - \sigma^{\text{stress}}_{\text{model}}(\texttt{strains\_meas(i)};\hat{\beta}) \right)^2 \right)^{1/2}$$
$$< \left(\sum_{i=1}^{m} \left(\texttt{stresses\_meas(i)} - \sigma^{\text{stress}}_{\text{model}}(\texttt{strains\_meas(i)};\beta^{\text{true}}) \quad {}^2 \right)^{1/2} \quad ?$$

(a) Yes

(b) No

We recall that $\beta^{\text{true}} = \begin{pmatrix} 0 \\ E \end{pmatrix}$ and E is the true Young's modulus of the material.

The template `A3Q4_Template.m` contains the multiple-choice format required by `grade_o_matic_A3`.

5. (*30 points*) As described and demonstrated in class, the friction coefficient between a robot wheel and the ground plays a crucial role in robot navigation and performance. Amontons' "law" states that the maximum static friction force (tangential to the ground) will be given by

$$F_{\text{f, static}}^{\text{max}} = \mu_{\text{s}} \, F_{\text{normal, applied}} \ , \tag{6}$$

where μ_{s} is the static coefficient of friction and $F_{\text{normal, applied}}$ is the normal force exerted by the wheel on the ground (related to robot weight). The coefficient of friction will of course depend on the pair of participating materials. We would like to verify Amontons' law and in particular confirm — or more precisely, not reject — the hypothesis that $F_{\text{f, static}}^{\text{max}}$ does not depend on surface area. We shall return to this hypothesis in Question 6(*ii*); here in Question 5 we focus on the necessary regression codes.

Towards that end, we must next postulate a dependence (or "model")

$$F_{\text{f, static}}^{\text{max}}(F_{\text{normal, applied}}, A_{\text{surface}}; \beta) = \beta_0 + \beta_1 \, F_{\text{normal, applied}} + \beta_2 \, A_{\text{surface}} \ , \tag{7}$$

where A_{surface} is the surface area of the contact and $\beta = (\beta_0, \beta_1, \beta_2)^{\text{T}}$. We expect (and we shall hypothesize) — but we do not *a priori* assume — from Messieurs Amontons and Coulomb that $\beta_0 = 0$ and $\beta_2 = 0$. You may assume that the experimental measurements are of the form

$$F_{\text{f, static}}^{\text{max, meas}} = F_{\text{f, static}}^{\text{max}}(F_{\text{normal, applied}}, A_{\text{surface}}; \beta^{\text{true}}) + \epsilon \ , \tag{8}$$

where β^{true} is the true value of β in the absence of noise, and ϵ is the noise; you may further assume that the noise is normal zero-mean, homoscedastic, and uncorrelated (at different $F_{\text{normal, applied}}$ and A_{surface}) per our assumptions N1, N2, and N3, respectively.

In this question we would like you to write a script which, for some given set of data, performs a regression analysis[2] to determine estimates ($\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$, respectively) and associated 95% confidence-level joint confidence intervals ($I_1^{\text{joint}}, I_2^{\text{joint}}$, and I_3^{joint}, respectively) for the coefficients β_0^{true}, β_1^{true}, and β_2^{true}. (You will also need to calculate the estimate $\hat{\sigma}_m$ for the experimental noise, however, $\hat{\sigma}_m$ is not a "deliverable" but rather an internal matter between you and your script.) We emphasize that your script should perform correctly for any set of (real or synthetic) data — and indeed `grade_o_matic_A3` will test several different instances; you might yourself devise several test cases for which you can anticipate the correct answers and hence test your script. (In Question 6(*ii*) we shall consider real data for a particular pair of materials.)

The script takes three script inputs: $F_{\text{f, static}}^{\text{max, meas}}(i)$, $F_{\text{normal, applied}}\,(i)$, and $A_{\text{surface}}(i)$, for $1 \leq i \leq m$, where $F_{\text{f, static}}^{\text{max, meas}}(i)$ is the maximum measured static friction force corresponding to the normal load $F_{\text{normal, applied}}\,(i)$ and the surface area $A_{\text{surface}}(i)$. These inputs must correspond in your script to MATLAB $m \times 1$ vectors `F_fstaticmaxmeas`, `F_normalload`, and `A_surface`, respectively. There is no restriction on allowable instances except that $1 \leq m \leq 10000$; note that m is not an input and should instead be deduced from (say) `F_fstaticmaxmeas`. We emphasize that entry i of `F_fstaticmaxmeas`, `F_normalload`, and `A_surface` contains, respectively, the measured friction force $F_{\text{f, static}}^{\text{max, meas}}$ (in Newtons), the prescribed normal load $F_{\text{normal, applied}}$ (in Newtons), and the prescribed surface area A_{surface} (in cm^2), for the i^{th}

[2] The statistical formulation is provided in the Unit III Lecture Notes; note $\rho_{\gamma,n,m} = \rho_{\gamma,k,q}$ corresponds to $s_{\gamma,n,m-n} = s_{\gamma,k,q}$ in the textbook and can be computed in MATLAB as $\rho_{\gamma,n,m} =$ `sqrt(n * finv(GAMMA, n, m-n))`.

measurement. The script yields six script outputs: scalar $\hat{\beta}_0$, scalar $\hat{\beta}_1$, scalar $\hat{\beta}_2$, 1×2 array I_0^{joint}, 1×2 array I_1^{joint}, and 1×2 array I_2^{joint}, which must correspond in your script to MATLAB variables `beta_hat_0`, `beta_hat_1`, `beta_hat_2`, `I_joint_0`, `I_joint_1`, and `I_joint_2`, respectively. A template is provided in `A3Q5_Template`.

6. (*10 points*)

(*i*) (*5 points*) This question relates to Question 3. In the limit $m \to \infty$, $\hat{\beta}_0$ tends to

(*a*) 0.06362

(*b*) 0.06364

(*c*) 0.06366

(*d*) 0.06368

We would prefer that you deduce the answer from theoretical considerations but of course you may also corroborate your result with empirical results from your script of Question 3. (Note you may neglect here any finite-precision or round-off effects. For convenience we display the result rounded to five digits after the decimal.)

(*ii*) (*5 points*) This question relates to Question 5. We earlier conducted experiments, for a particular pair of robot wheel and ground materials, in which we obtained friction force measurements $F_{\text{f, static}}^{\text{max, meas}}$ (in Newtons) as a function of normal load $F_{\text{normal, applied}}$ (in Newtons) and (nominal) surface area of contact A_{surface} (in cm^2). The turntable apparatus and experimental protocol are described in the Lecture Notes for Unit III as well as in the textbook.

The experimental data comprises 50 measurements: 2 (distinct) measurements at each of 25 points on a 5×5 "grid" in $(F_{\text{normal, applied}}, A_{\text{surface}})$ space. The data is provided to you in the `.mat` file `friction_data` (available in the `Assignment_3_Materials` folder) as 50×1 arrays `F_fstaticmaxmeas_r`, `F_normalload_r`, and `A_surface_r` (where the `_r` indicates *real* data): entry i of `F_fstaticmaxmeas_r`, `F_normalload_r`, and `A_surface_r` provides, respectively, the measured friction force $F_{\text{f, static}}^{\text{max, meas}}$ (in Newtons), the prescribed normal load $F_{\text{normal, applied}}$ (in Newtons), and the prescribed surface area A_{surface} (in cm^2), for the i^{th} measurement; for example, in the first measurement, $i = 1$, the measured friction force is 0.1080 Newtons, the imposed normal load is 0.9810 Newtons, and the nominal surface area is 1.2903 cm^2. This data will now serve to test our hypotheses.

In particular, we ask that you run your script of Question 5 with script inputs specified as `F_fstaticmaxmeas` = `F_fstaticmaxmeas_r`, `F_normalload` = `F_normalload_r`, and `A_surface` = `A_surface_r`. From the script outputs you can conclude with 95% confidence level that, for the particular pair of materials tested,

(*a*) $F_{\text{f, static}}^{\text{max}}$ does *not* depend on A_{surface};

(*b*) $F_{\text{f, static}}^{\text{max}}$ *does* depend on A_{surface};

(c) the statement "$F_{\text{f, static}}^{\max}$ does *not* depend on A_{surface}" is consistent with the data;

(d) the statement "$F_{\text{f, static}}^{\max}$ does *not* depend on A_{surface}" is inconsistent with the data.

Note you may assume in answering this question that the noise in the real experiment does indeed honor our assumptions N1, N2, and N3 such that the confidence intervals are valid and may thus serve to quantify the effect of noise on the regression estimates for β_0^{true}, β_1^{true}, and β_2^{true}.

The template `A3Q6_Template.m` contains the multiple-choice format required by `grade_o_matic_A3`.