

You will have to use a computer in a university lab (e.g. Wells Hall B-Wing).

This *Mathematica* notebook contains a number of useful functions briefly described below. The first time you attempt to use one of these functions a panel will pop up asking "Do you want to evaluate all the initialization cells?" to which you must answer yes.

To enter a given command line you click on the screen whereupon a horizontal line should appear at the cursor. When right brackets are in view on the *Mathematica* panel you want to click at a place where a horizontal line will extend between two such brackets if you desire a new line. If you attempt to type multiple commands into a single bracketed location *Mathematica* will become confused.

Type the command you wish to execute then PRESS THE ENTER KEY ON THE NUMERIC KEYPAD. This is required because *Mathematica* wants to use the return or other enter key to move to the next line. You do not want to move to a new line. You want to enter a command. That is why you must use the ENTER key on the numeric keypad. You may also use SHIFT and ENTER together instead of using ENTER on the numeric keypad.

To save your work select save from the pull down file menu, which saves it as a *Mathematica* .nb (notebook) file. If you wish to print your work at home, select print, then select the option of saving as a PDF. You will be unable to work with the .nb *Mathematica* file itself unless you have *Mathematica* installed (unlikely) but you can transport and print the .pdf file virtually anywhere.

Click the line below and press ENTER on the numeric keypad.

```
size[{4.5, 7.1, 7.8, 9.1}]
```

4

Just above, I clicked to open a new line then typed

```
size[{4.5, 7.1, 7.8, 9.1}]
```

followed by a press of the numeric keypad ENTER key. Notice that off to the right of the entry there are nested brackets joining the command line and its output 4 = the number of data items in {4.5, 7.1, 7.8, 9.1}.

■ **A complete list of the commands in this notebook and what they do.**

`size`{4.5, 7.1, 7.8, 9.1} returns 4

`mean`{4.5, 7.1, 7.8, 9.1} returns the mean 7.125

`median`{4.5, 7.1, 7.8, 9.1} returns the median of the list {4.5, 7.1, 7.8, 9.1}

`s`{4.5, 7.1, 7.8, 9.1} returns the sample standard deviation $s=1.93628$

`sd`{4.5, 7.1, 7.8, 9.1} returns the n-divisor version of standard deviation $s=1.67686$

`r`[**x**, **y**] returns the sample correlation $r = \frac{\overline{xy} - \bar{x}\bar{y}}{\sqrt{\overline{x^2} - \bar{x}^2} \sqrt{\overline{y^2} - \bar{y}^2}}$ for paired data.

`sample`{4.5, 7.1, 7.8, 9.1}, 10] returns 10 samples from {4.5, 7.1, 7.8, 9.1}

`ci`{4.5, 7.1, 7.8, 9.1}, 1.96] returns a 1.96 coefficient CI for the mean from given data

`bootci`[mean, {4.5, 7.1, 7.8, 9.1}, 10000, 0.95] returns 0.95 bootstrap ci for pop mean

`smooth`{4.5, 7.1, 7.8, 9.1}, 0.2] returns the density for data at bandwidth 0.2

`smooth2`{4.5, 7.1, 7.8, 9.1}, 0.2] returns the density for data at bandwidth 0.2

overlaid with normal densities having $sd = 0.2$ around each data value

`smoothdistribution`{1, 700}, {4, 300}, 0.2] returns the density at bandwidth 0.2

for a list consisting of 700 ones and 300 fours.

`popSALES` is a file of 4000 sales amounts used for examples

entering `popSALES` will spill 4000 numbers onto the screen. To prevent

that enter `popSALES;` instead (the appended semi-colon suppresses output).

`betahat`[**matrix x**, **data y**] returns the least squares coefficients $\hat{\beta}$ for a fit of the model $y = x\beta + \epsilon$.

`resid`[**matrix x**, **data y**] returns the estimated errors $\hat{\epsilon} = y - x\hat{\beta}$ (see `betahat` above).

`R`[**matrix x**, **data y**] returns the **multiple correlation** between the fitted values $x\hat{\beta}$ and data y .

`xquad`[**matrix x**] returns the full quadratic extension of a design matrix with constant term

`xcross`[**matrix x**] returns the extension of x to include all products of differing columns.

`betahatCOV`[**x matrix**, **data y**] returns the estimated covariance matrix of the vector `betahat` $\hat{\beta}$.

`normalprobabilityplot`[**data**, **dotsize**] returns a normal probability plot for data (e.g. with `dotsize .01`).

`t`[**df**, **conf**] returns the t-score used in lieu of z-score in a CI for confidence `conf` (`t`[Infinity, .95] ~ 1.96).

`Tprob`[**t**, **df**] returns $P(|T| < t)$ for `df` (e.g. `T`[1.96, Infinity] ~ 0.95).

Some changes to the grading scale are announced below.

A. HW7 (below) is important. A portion of your GRADE on HW7 will also apply to boost your exam 2 GRADE as follows:

$$\text{NEW EXAM 2 GRADE} = \text{Exam 2 GRADE} + .15 \text{ HW7 GRADE}$$

B. Your homework GRADE is 25% of your course grade whereas the average of your exam 1 and 2 GRADES is $0.5 (2/3) = 33.3\%$ of your course GRADE (see the syllabus). That should give you a ready way to determine your overall grade to date.

C. A **modified grading scale** is now in place for the course. Your overall course GRADE will be determined as the LARGER of

(a) 0.5 (three exam average grade) + 0.25 (homework grade) + 0.25 (final exam grade)

(b) 0.25 (exams 1 and 2 average grade) + 0.2 (exam 3 grade)

+ 0.25 (homework grade) + 0.3 (final exam grade)

You do not have to elect a choice. It will be done automatically and will not give you a course grade lower than your grade under the original plan (a) found in the syllabus. As you can see, plan (b) places more empha-

sis on the work from here on out.

HW 7 is due at the end of class Wednesday, November 5. It will count towards your regular homework in the usual way.

Commensurate with its importance, HW7 will be graded rather strictly and to a higher standard as regards the accuracy, completeness, and professionalism of your write-up. Be very organized and clearly describe what you are doing. You may submit annotated computer printout, even with hand written comments, but do not just submit raw computer output or your grade will be severely reduced.

If I have to guess at what you have done, or whether you understand what you are doing, it is wrong. If it is messy or disorganized your grade will be reduced. If you make a mistake affecting your conclusion, it is wrong. Pretend that I have hired you to do this. I need to understand what you are doing for me, need to have clarity about your methods, confidence that you have done things correctly and have understood what you are doing.

To give you an idea of how this works in professional contacts I will relate an event of some years past. A department of MSU contacted me with a request that I help them understand a hot new area of statistics then taking over applied research in their field. They were about to hire a new statistician to replace one they were losing to retirement. They needed to get up to speed on what type of individual could serve their needs in the future, especially vis-a-vis the new methods they were hearing about, that were to play a big role in their future research and grants prospects. It impacted the competitiveness of their students in the job market. I undertook several lectures attended by nine of their faculty. A key point was their need to understand one particular paper they'd had trouble reading. One of my graduate students helped me by pulling out the raw data from the paper, implementing concise *Mathematica* code to step through the analysis, made all the comparisons with the results published in the paper (confirming them), and wrote everything up in very nice fashion. After a few lectures I dove into his analysis, handing it around. It was the talking point for getting their heads into the subject. That was very successful and they made a good hire.

Here is another example. I attended a meeting of physicists and statisticians where the objective was to understand attempts to develop statistical methods with which to identify (possible) high energy events masked by noise. Several teams had been laboring, over several years, to understand these difficult problems. They had carefully developed physics models, computer simulators of such models, and real apparatus from which relatively small numbers of very expensive measurements had been obtained. Could they say, with some confidence, that the high energy events were really occurring, or was it all just noise? In the meeting, an idea surfaced: perhaps off the shelf statistical methods might do about as well for this purpose as the specialized labors of the teams. Could it be? Around six of the younger participants spent that very night trying it out. Indeed, it looked to be so. Their undertaking, and its presentation, used the kinds of skills I am asking you to bring to this assignment.

Try statistical methods, compare with what the book says, explain the work as best you can.

1. Re-do example 12.11 page 471 in Little Software. I've partially done this below so you can see how it is accomplished. You must

a. Enter the x data and the y data making sure to maintain the proper 1:1 matchup of x-scores with y-scores.

b. Transform the data into matrix form as required by the *Mathematica* routine. The matrix setup is used in multiple linear regression and I am giving you a head start on this way of looking at straight line regression.

c. Obtain the least squares solution which includes
plot the data

form the LS estimate $\hat{\beta}_0$ of the y-intercept (see also formula on pg. 469)

form the LS estimate $\hat{\beta}_1$ of the y to x slope (see also formula on pg. 469)

plot the regression line together with the data

form the estimate $\hat{\sigma}^2$ of variance of errors (see also formula on pg. 470)

estimate standard error $\sqrt{\text{Var } \hat{\beta}_1}$ by $s_{\hat{\beta}_1}$ (see also formula on pg. 470)

determine the 95% CI for β_1 given by $\hat{\beta}_1 \pm 1.96 s_{\hat{\beta}_1}$

determine the fitted values $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$

pull off the residuals $y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$ left by least squares

prepare a normal probability plot of the residuals

calculate the coefficient of determination r^2

Where possible, compare your answers above with the output of SAS shown in table 12.15 and other commentary on this example found in your textbook.

2. Repeat (1) for the data of Example 12.4 pg. 457.

Below, I show how to do these things using only a portion of the data from Example 12.11. You will use *all* of the data of 12.11.

```
mortarair = {5.7, 6.8, 9.6};
```

```
dryden = {119, 121.3, 118.2};
```

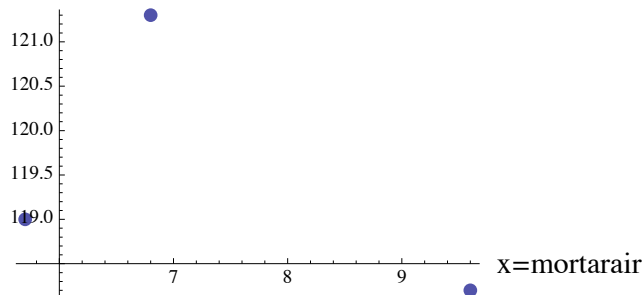
```
Length[mortarair]
```

```
3
```

You may wish to use a different point size in your plot for best legibility.

```
ListPlot[Table[{mortarair[[i]], dryden[[i]]}, {i, 1, 3}],
  AxesLabel -> {"x=mortarair", "y=dryden"}, PlotStyle -> PointSize[0.03]
```

y=dryden



```
xmortarair = Table[{1, mortarair[[i]]}, {i, 1, 3}];
```

```
MatrixForm[xmortarair]
```

$$\begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \end{pmatrix}$$

PseudoInverse is a least squares solver applicable to systems of linear equations. It produces the unique solution of simultaneous linear equations in several variables (such as the normal equations of Least Squares) if there is one. If not, it produces a particular choice of a least squares solution known as the Moore - Penrose Inverse. In the present example, the matrix formulation of the equations of our linear model is

$$\begin{pmatrix} 119.0 \\ 121.3 \\ 118.2 \end{pmatrix} \sim \begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

Because the points (x, y) do not fall on a line (see the plot above) there is no exact solution of these 3 equations in only 2 unknowns. The least squares solver uses a Pseudo-Inverse to find a Least Squares solution

$$\begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \end{pmatrix}^{-1} \begin{pmatrix} 119.0 \\ 121.3 \\ 118.2 \end{pmatrix} \sim \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \text{ (Careful! It is not a genuine inverse, only LS.)}$$

Observe the little dot in the code below. It denotes matrix product and is very important!

```
betahatmortar = PseudoInverse[xmortarair].dryden
```

```
{122.315, -0.38211}
```

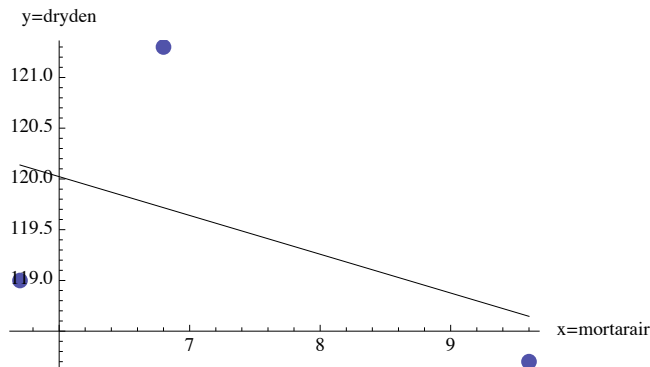
From the line above, the LS estimated slope and intercept are 122.315 and -0.38211 respectively. Next, find the fitted values $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

```
drydenhat = xmortarair.betahatmortar
```

```
{120.137, 119.717, 118.647}
```

Next, overlay the LS line on the plot of (x, y). Notice that the code below asks that *Mathematica* join the line of LS fitted values, i.e. the line joins points (x_i, \hat{y}_i) . See that LS produces fitted values falling perfectly on a line (otherwise *Mathematica* would have plotted a broken zig-zag line).

```
Show[ListPlot[Table[{mortarair[[i]], dryden[[i]]}, {i, 1, 3}],
  AxesLabel -> {"x=mortarair", "y=dryden"}, PlotStyle -> PointSize[0.03],
  Graphics[Line[Table[{mortarair[[i]], drydenhat[[i]]}, {i, 1, 3}]]]]
```



Examine the plot above. Check visually that the intercept is indeed $\hat{\beta}_0$ (calculated above) and the slope is indeed $\hat{\beta}_1$. Check visually that the heights of the regression line at the mortarair values {5.7, 6.8, 9.6} are indeed your previously calculated fitted values {120.137, 119.717, 118.647}. Also by eye, see that your residuals, calculated next, are indeed the signed vertical gaps between the points of the plot and the regression line.

Again, you will be working with *all* the data of 12.11.

```
drydenresid = dryden - drydenhat
```

```
{-1.13685, 1.58347, -0.44662}
```

Matrix Setup gives covariances of the estimates. Let x denote the matrix whose first column is all ones and whose second column holds the x -values {5.7, 6.8, 9.6}.

$$x = \begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \end{pmatrix}$$

This is called the *design* matrix. The probability model, stated in matrix form, is simply written $y = x \cdot \beta + \epsilon$ where the errors ϵ_i are assumed to be statistically independent random variables having the same normal distribution with mean 0 and some unknown sd $\sigma > 0$.

$$\begin{pmatrix} 119.0 \\ 121.3 \\ 118.2 \end{pmatrix} = \begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}$$

$$y = x \beta + \epsilon$$

When writing this matrix model we drop the dot and just write $y = x\beta + \epsilon$. **Mathematica needs the dot however.**

A very nice thing happens. The variances and covariances of the estimators $\hat{\beta}_0, \hat{\beta}_1$ are just the entries of the matrix $(x^{\text{tr}} x)^{-1} \sigma^2$, provided it exists. This is always the case if the columns of x are not linearly dependent.

In *Mathematica*, $(x^{\text{tr}} x)^{-1}$ is coded `Inverse[Transpose[x].x]`.

The preferred estimator of σ^2 is $\hat{\sigma}^2 = \frac{n-1}{n-d} s_{\text{residuals}}^2$, where d is the number of columns of the design matrix x (for straight line regression above $d = 2$).

So the least squares estimators of intercept and slope have variances and covariance that are *estimated* by the entries of the following matrix:

```
Inverse[Transpose[xmortarair].xmortarair]  $\frac{3-1}{3-2}$  (s[drydenresid])2
{{28.1713, -3.6432}, {-3.6432, 0.494552}}

MatrixForm[%]

$$\begin{pmatrix} 28.1713 & -3.6432 \\ -3.6432 & 0.494552 \end{pmatrix}$$

```

From the above, we estimate the variance of $\hat{\beta}_0$ to be 28.1713, the variance of $\hat{\beta}_1$ to be 0.494552, and the covariance of $\hat{\beta}_0$ with $\hat{\beta}_1$ (same as cov of $\hat{\beta}_1$ with $\hat{\beta}_0$) to be -3.6432.

So a 95% CI for β_0 (the true intercept absent errors of observation) is

$$122.315 \pm t \sqrt{28.1713}$$

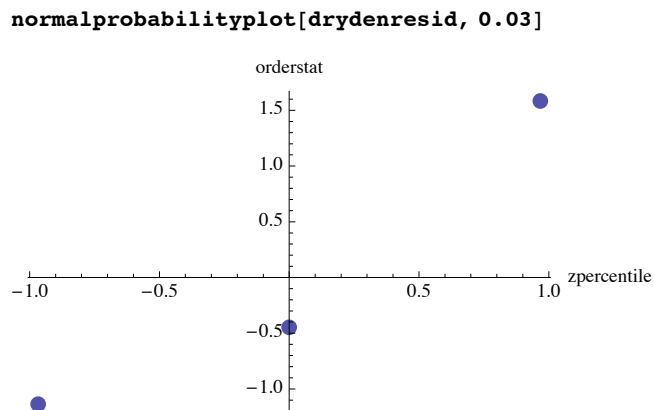
and a 95% CI for β_1 (the true slope absent errors of observation) is

$$-0.38211 \pm t \sqrt{0.494552}$$

where t is for degrees of freedom $d-2$ and $\alpha = 0.025$ (for 95% confidence). This use of t results in an exact ci provided the measurements (x, y) are from a process under statistical control.

As part of this assignment, when you work with the full data of Example 12.11, you will want to check that this CI for slope agrees with the one reported on page 472.

For a *partial* check on the *normal errors assumption* of the probability model it is customary to perform a *normal probability plot for the residuals* to see if it departs very much from a straight line. Since n is only 3, not much can be learned from the plot. But when you do the same for the full data of Example 12.11 you can address the issue more confidently.



Here is the correlation between the independent variable mortarair and the dependent variable

dryden. Squaring it gives the coefficient of determination which is "the fraction of var y accounted for by regression on x." It is not very large in this tiny example.

```
r[mortarair, dryden]
```

```
-0.477429
```

```
%^2
```

```
0.227938
```