HW 7 Key 2

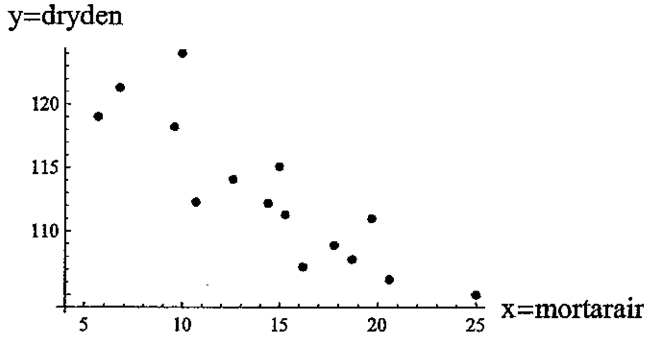
```
mortarair =
  {5.7, 6.8, 9.6, 10.0, 10.7, 12.6, 14.4, 15.0, 15.3, 16.2, 17.8, 18.7, 19.7, 20.6, 25.0 };
```

```
dryden = {119, 121.3, 118.2, 124.0, 112.3, 114.1,
   112.2, 115.1, 111.3, 107.2, 108.9, 107.8, 111.0, 106.2, 105.0};
```

```
Length[mortarair]
```

```
15
```

```
ListPlot[Table[{mortarair[[i]], dryden[[i]]}, {i, 1, 15}], AxesOrigin -> {4, 104},
 AxesLabel → {"x=mortarair", "y=dryden"}, PlotStyle -> PointSize[0.02]]
```



```
xmortarair = Table[{1, mortarair[[i]]}, {i, 1, 15}];
```

```
MatrixForm[xmortarair]
```

$$\begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \\ 1 & 10. \\ 1 & 10.7 \\ 1 & 12.6 \\ 1 & 14.4 \\ 1 & 15. \\ 1 & 15.3 \\ 1 & 16.2 \\ 1 & 17.8 \\ 1 & 18.7 \\ 1 & 19.7 \\ 1 & 20.6 \\ 1 & 25. \end{pmatrix}$$

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

$$\begin{pmatrix} 119.0 \\ 121.3 \\ 118.2 \\ 124.0 \\ 112.3 \\ 114.1 \\ 112.2 \\ 115.1 \\ 111.3 \\ 107.2 \\ 108.9 \\ 107.8 \\ 111.0 \\ 106.2 \\ 105.0 \end{pmatrix} \sim \begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \\ 1 & 10.0 \\ 1 & 10.7 \\ 1 & 12.6 \\ 1 & 14.4 \\ 1 & 15.0 \\ 1 & 15.3 \\ 1 & 16.2 \\ 1 & 17.8 \\ 1 & 18.7 \\ 1 & 19.7 \\ 1 & 20.6 \\ 1 & 25.0 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \\ 1 & 10.0 \\ 1 & 10.7 \\ 1 & 12.6 \\ 1 & 14.4 \\ 1 & 15.0 \\ 1 & 15.3 \\ 1 & 16.2 \\ 1 & 17.8 \\ 1 & 18.7 \\ 1 & 19.7 \\ 1 & 20.6 \\ 1 & 25.0 \end{pmatrix}^{-1} \begin{pmatrix} 119.0 \\ 121.3 \\ 118.2 \\ 124.0 \\ 112.3 \\ 114.1 \\ 112.2 \\ 115.1 \\ 111.3 \\ 107.2 \\ 108.9 \\ 107.8 \\ 111.0 \\ 106.2 \\ 105.0 \end{pmatrix} \sim \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \text{ (Careful! It is not a genuine inverse, only LS.)}$$

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

```
betahatmortar = PseudoInverse[xmortarair].dryden

{126.249, -0.917622}
```

$\hat{\beta}0$ = 126.249
$\hat{\beta}1$ = -0.917622
The above results matched the book's

From the line above, the LS estimated slope and intercept are 126.249 and - 0.917622 respectively. Next, find the fitted values $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

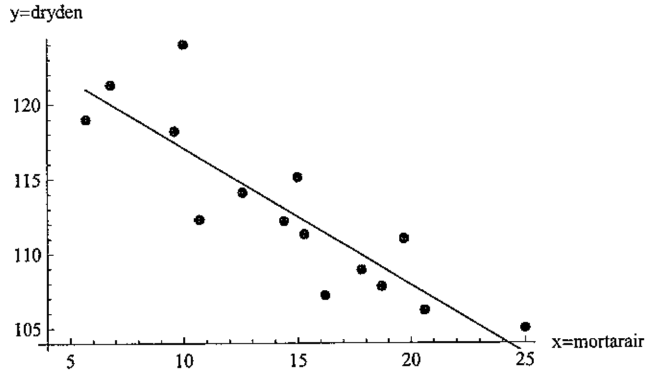
```
drydenhat = xmortarair.betahatmortar

{121.018, 120.009, 117.44, 117.073, 116.43, 114.687, 113.035,
 112.485, 112.209, 111.383, 109.915, 109.089, 108.172, 107.346, 103.308}
```

The above calculated fitted values matched those in the book.

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, 15}], AxesOrigin -> {4, 104},
  AxesLabel → {"x=mortarair", "y=dryden"}, PlotStyle -> PointSize[0.02]],
 Graphics[Line[Table[{mortarair[[i]], drydenhat[[i]]}, {i, 1, 15}]]]]
```



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, 10.0, 10.7, 12.6, 14.4, 15.0, 15.3, 16.2, 17.8, 18.7, 19.7, 20.6, 25.0} are indeed your previously calculated fitted values {121.018, 120.009, 117.44, 117.073, 116.43, 114.687, 113.035, 112.485, 112.209, 111.383, 109.915, 109.089, 108.172, 107.346, 103.308}. 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.

```
drydenresid = dryden - drydenhat
```

```
{-2.01844, 1.29094, 0.760281, 6.92733, -4.13033, -0.586853, -0.835134,
 2.61544, -0.909274, -4.18341, -1.01522, -1.28936, 2.82826, -1.14588, 1.69166}
```

Above calculated Residual values matched those in the book.

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, 10.0, 10.7, 12.6, 14.4, 15.0, 15.3, 16.2, 17.8, 18.7, 19.7, 20.6, 25.0}.

$$x = \begin{pmatrix} 1 & 5.7 \\ 1 & 6.8 \\ 1 & 9.6 \\ 1 & 10.0 \\ 1 & 10.7 \\ 1 & 12.6 \\ 1 & 14.4 \\ 1 & 15.0 \\ 1 & 15.3 \\ 1 & 16.2 \\ 1 & 17.8 \\ 1 & 18.7 \\ 1 & 19.7 \\ 1 & 20.6 \\ 1 & 25.0 \end{pmatrix}$$

This is called the ***design*** matrix. The probability model, stated in matrix form, is simply written $y = x.\ \beta + \epsilon$ 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 $\sigma > 0$.

$$\begin{pmatrix}119.0\\121.3\\118.2\\124.0\\112.3\\114.1\\112.2\\115.1\\111.3\\107.2\\108.9\\107.8\\111.0\\106.2\\105.0\end{pmatrix} = \begin{pmatrix}1 & 5.7\\1 & 6.8\\1 & 9.6\\1 & 10.0\\1 & 10.7\\1 & 12.6\\1 & 14.4\\1 & 15.0\\1 & 15.3\\1 & 16.2\\1 & 17.8\\1 & 18.7\\1 & 19.7\\1 & 20.6\\1 & 25.0\end{pmatrix} \begin{pmatrix}\beta_0\\\beta_1\end{pmatrix} + \begin{pmatrix}\epsilon_1\\\epsilon_2\\\epsilon_3\\\epsilon_4\\\epsilon_5\\\epsilon_6\\\epsilon_7\\\epsilon_8\\\epsilon_9\\\epsilon_{10}\\\epsilon_{11}\\\epsilon_{12}\\\epsilon_{13}\\\epsilon_{14}\\\epsilon_{15}\end{pmatrix}$$

$$\mathrm{y} \quad = \quad \mathrm{x} \quad \beta \;+\; \epsilon$$

When writing this matrix model we drop the dot and just write y = xβ + ϵ. ***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 $(x^{\mathrm{tr}}\, x)^{-1}\sigma^2$, provided it exists. This is always the case if the columns of x are not linearly dependent.

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

The preferred estimator of σ^2 is $\hat{\sigma}^2 = \frac{n-1}{n-d} s^2_{\text{residuals}}$, 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:

```
                                            15 - 1
Inverse[Transpose[xmortarair].xmortarair] -------- (s[drydenresid])²
                                            15 - 2

{{5.0824, -0.309887}, {-0.309887, 0.0213128}}
```

```
MatrixForm[%]
```

```
( 5.0824      -0.309887 )
( -0.309887   0.0213128 )
```

From the above, we estimate the variance of $\hat{\beta}_0$ to be 5.0824, the variance of $\hat{\beta}_1$ to be 0.0213128, and the covariance of $\hat{\beta}_0$ with $\hat{\beta}_1$ (same as cov of $\hat{\beta}_1$ with $\hat{\beta}_0$) to be -0.309887.

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

$126.249 \pm t\sqrt{5.0824}$ (df is 13 and α is 0.025 in this case)

```
126.249 + 2.16 √5.0824
131.119
```

```
126.249 - 2.16 √5.0824
121.379
```

The confidence interval of $\hat{\beta}0$ is (121.379, 131.119).

A 95 % CI for Subscript[β, 1] (the true slope absent errors of observation) is
-0.917622 ± t Sqrt[0.0213128]
where t is for degrees of freedom d - 2 = 13 and α = 0.025 (for 95 % confidence).This use of t results in an exact ci provided the measurements (x, y) are from a process under statistcal control.

```
-0.917622 + 2.16 √0.0213128
-0.602286
```

```
-0.917622 - 2.16 √0.0213128
-1.23296
```

```
The confidence interval of β1 is (-1.233, -0.603) which macthed the book's resu
```

```
As part of this assignment, when you work with the full data c
```

```
mean[(dryden - xmortarair.{126.249, -0.917622})^2]
7.49622
```

The residuals of our *Mathematica* solution have the following mean of squares

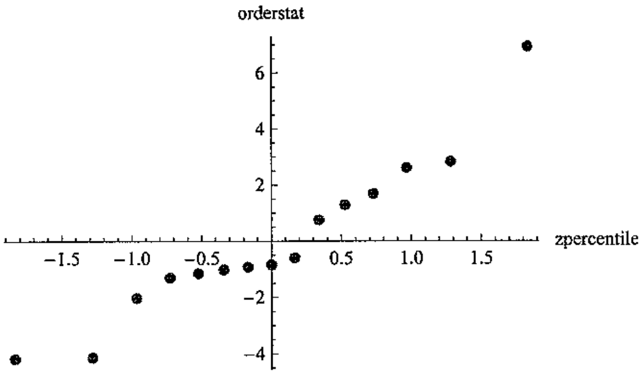
```
mean[(dryden - xmortarair.{126.249, -0.917622})^2]
```

```
7.49622
```

Our findings matched with results in the book.

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.

```
normalprobabilityplot [drydenresid, 0.02]
```



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]
```

```
-0.867421
```

```
%^2
```

```
0.75242
```

12.04

```
mortarair = {99.0, 101.1, 102.7, 103.0, 105.4, 107.0,
   108.7, 110.8, 112.1, 112.4, 113.6, 113.8, 115.1, 115.4, 120.0 };
```

```
In[33]:= dryden = {28.8, 27.9, 27.0, 25.2, 22.8,
            21.5, 20.9, 19.6, 17.1, 18.9, 16.0, 16.7, 13.0, 13.6, 10.8};
```

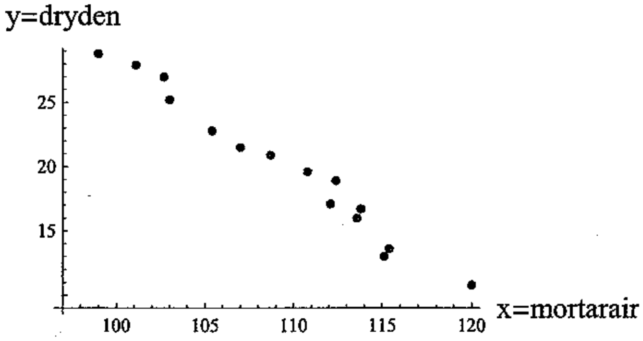
```
In[34]:= Length[mortarair]

Out[34]= 15
```

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

```
In[35]:= ListPlot[Table[{mortarair[[i]], dryden[[i]]}, {i, 1, 15}], AxesOrigin -> {97, 9},
          AxesLabel → {"x=mortarair", "y=dryden"}, PlotStyle -> PointSize[0.02]]
```

Out[35]=



```
In[36]:= xmortarair = Table[{1, mortarair[[i]]}, {i, 1, 15}];
```

```
In[37]:= MatrixForm[xmortarair]

Out[37]//MatrixForm=
 1  99.
 1  101.1
 1  102.7
 1  103.
 1  105.4
 1  107.
 1  108.7
 1  110.8
 1  112.1
 1  112.4
 1  113.6
 1  113.8
 1  115.1
 1  115.4
 1  120.
```

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

$$\begin{pmatrix} 28.8 \\ 27.9 \\ 27.0 \\ 25.2 \\ 22.8 \\ 21.5 \\ 20.9 \\ 19.6 \\ 17.1 \\ 18.9 \\ 16.0 \\ 16.7 \\ 13.0 \\ 13.6 \\ 10.8 \end{pmatrix} \sim \begin{pmatrix} 1 & 99. \\ 1 & 101.1 \\ 1 & 102.7 \\ 1 & 103. \\ 1 & 105.4 \\ 1 & 107. \\ 1 & 108.7 \\ 1 & 110.8 \\ 1 & 112.1 \\ 1 & 112.4 \\ 1 & 113.6 \\ 1 & 113.8 \\ 1 & 115.1 \\ 1 & 115.4 \\ 1 & 120. \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & 99. \\ 1 & 101.1 \\ 1 & 102.7 \\ 1 & 103. \\ 1 & 105.4 \\ 1 & 107. \\ 1 & 108.7 \\ 1 & 110.8 \\ 1 & 112.1 \\ 1 & 112.4 \\ 1 & 113.6 \\ 1 & 113.8 \\ 1 & 115.1 \\ 1 & 115.4 \\ 1 & 120. \end{pmatrix}^{-1} \begin{pmatrix} 28.8 \\ 27.9 \\ 27.0 \\ 25.2 \\ 22.8 \\ 21.5 \\ 20.9 \\ 19.6 \\ 17.1 \\ 18.9 \\ 16.0 \\ 16.7 \\ 13.0 \\ 13.6 \\ 10.8 \end{pmatrix} \sim \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \text{ (Careful! It is not a genuine inverse, only LS.)}$$

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

```
In[38]:= betahatmortar = PseudoInverse[xmortarair].dryden

Out[38]= {118.91, -0.904731}
```

β1 = -0.904731
β0 = 118.91
The above calculation matched the results in the book.

From the line above, the LS estimated slope and intercept are 118.91 and - 0.904731 respectively. Next, find the fitted values $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

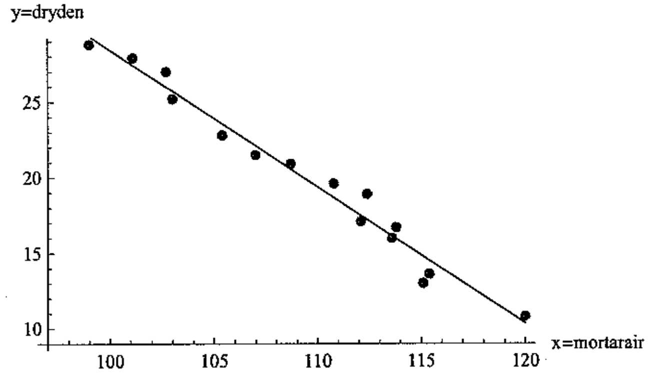
```
In[39]:= drydenhat = xmortarair.betahatmortar

Out[39]= {29.3416, 27.4416, 25.9941, 25.7227, 23.5513, 22.1037, 20.5657,
          18.6658, 17.4896, 17.2182, 16.1325, 15.9516, 14.7754, 14.504, 10.3422}
```

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).

```
In[40]:= Show[ListPlot[Table[{mortarair[[i]], dryden[[i]]}, {i, 1, 15}], AxesOrigin -> {97, 9},
           AxesLabel → {"x=mortarair", "y=dryden"}, PlotStyle -> PointSize[0.02]],
          Graphics[Line[Table[{mortarair[[i]], drydenhat[[i]]}, {i, 1, 15}]]]]
```

Out[40]=



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 {99.0, 101.1, 102.7, 103.0, 105.4, 107.0, 108.7, 110.8, 112.1, 112.4, 113.6, 113.8, 115.1, 115.4, 120.0} are indeed your previously calculated fitted values {29.3416, 27.4416, 25.9941, 25.7227, 23.5513, 22.1037, 20.5657, 18.6658, 17.4896, 17.2182, 16.1325, 15.9516, 14.7754, 14.504, 10.3422}. 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.

```
In[41]:= drydenresid = dryden - drydenhat

Out[41]= {-0.541582, 0.458353, 1.00592, -0.522659, -0.751305, -0.603736, 0.334306,
         0.93424, -0.38961, 1.68181, -0.132514, 0.748432, -1.77542, -0.903999, 0.457762}
```

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}.

$$x = \begin{pmatrix} 1 & 99. \\ 1 & 101.1 \\ 1 & 102.7 \\ 1 & 103. \\ 1 & 105.4 \\ 1 & 107. \\ 1 & 108.7 \\ 1 & 110.8 \\ 1 & 112.1 \\ 1 & 112.4 \\ 1 & 113.6 \\ 1 & 113.8 \\ 1 & 115.1 \\ 1 & 115.4 \\ 1 & 120. \end{pmatrix}$$

This is called the ***design*** matrix. The probability model, stated in matrix form, is simply written $y = x.\ \beta + \epsilon$ 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 $\sigma > 0$.

$$\underset{y}{\begin{pmatrix} 28.8 \\ 27.9 \\ 27.0 \\ 25.2 \\ 22.8 \\ 21.5 \\ 20.9 \\ 19.6 \\ 17.1 \\ 18.9 \\ 16.0 \\ 16.7 \\ 13.0 \\ 13.6 \\ 10.8 \end{pmatrix}} \underset{=}{=} \underset{x}{\begin{pmatrix} 1 & 99. \\ 1 & 101.1 \\ 1 & 102.7 \\ 1 & 103. \\ 1 & 105.4 \\ 1 & 107. \\ 1 & 108.7 \\ 1 & 110.8 \\ 1 & 112.1 \\ 1 & 112.4 \\ 1 & 113.6 \\ 1 & 113.8 \\ 1 & 115.1 \\ 1 & 115.4 \\ 1 & 120. \end{pmatrix}} \underset{\beta}{\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}} \underset{+}{+} \underset{\epsilon}{\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \\ \epsilon_8 \\ \epsilon_9 \\ \epsilon_{10} \\ \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{14} \\ \epsilon_{15} \end{pmatrix}}$$

When writing this matrix model we drop the dot and just write y = xβ + ϵ. ***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,* $(\mathbf{x}^{\text{tr}}\,\mathbf{x})^{-1}$ is coded Inverse[Transpose[x].x].

The preferred estimator of σ^2 is $\hat{\sigma}^2 = \frac{n-1}{n-d} s^2_{\text{residuals}}$, 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:

```
                                                        15 - 1
In[50]:= Inverse[Transpose[xmortarair].xmortarair] ────── (s[drydenresid])²
                                                        15 - 2

Out[50]= {{20.2421, -0.184593}, {-0.184593, 0.00168825}}

In[51]:= MatrixForm[%]

Out[51]//MatrixForm=
          ( 20.2421     -0.184593  )
          ( -0.184593   0.00168825 )
```

From the above, we estimate the variance of $\hat{\beta}_0$ to be 20.2421, the variance of $\hat{\beta}_1$ to be 0.00168825, and the covariance of $\hat{\beta}_0$ with $\hat{\beta}_1$ (same as cov of $\hat{\beta}_1$ with $\hat{\beta}_0$) to be -0.184593.

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

$118.91 \pm t\sqrt{20.2421} = 118.91 \pm 2.16\sqrt{20.2421}$. So the interval is (109.1919,128.6281)

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

$-0.904731 \pm t\sqrt{0.00168825} = -0.904731 \pm 2.16\sqrt{0.00168828}$. So the interval is (-0.99349, -0.81598)

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 statistcal control.

The residuals have the following mean of squares:

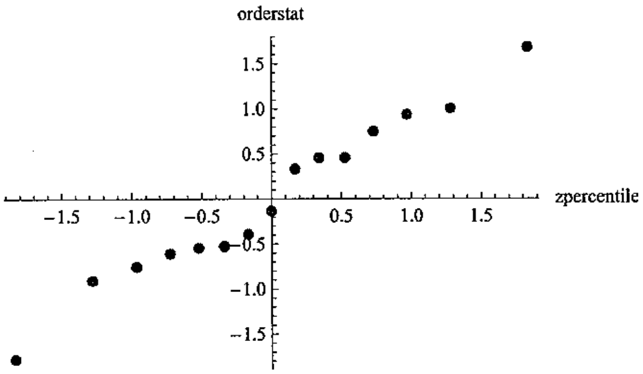
```
In[45]:= mean[(dryden - xmortarair.{118.909, -0.904730})^2]

Out[45]= 0.762589
```

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.

```
In[46]:= normalprobabilityplot [drydenresid, 0.02]
```

Out[46]=



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.

```
In[47]:= r[mortarair, dryden]

Out[47]= -0.986857

In[48]:= %^2

Out[48]= 0.973887
```