STT 351 **11-16-07**

In[13]:=

```
z[r_] :=
 Apply[Plus, Table[Random[], {i, 1, 12}]] - 6
```

z[1] returns an approximately **standard normal** distributed sample as does z[r] for any real number r. We need the r to force *Mathematica* to generate indpendent copies in certain situations.

In[15]:=

```
{z[1], z[2], z[3], z[4], z[5], z[6], z[7], z[8]}
```

Out[15]=

```
{0.772312, -0.69582, -0.225729, 0.179642,
 -0.93014, -1.54908, -0.853784, -0.0496516}
```

1. Sketch the standard normal probability density identifying the mean and sd as recognizable elements of your sketch and locate the above sample values by means of short vertical slashes placed at points on the z-axis.

2. In your sketch above identify the segment of the z-axis that is approximately produced by a plot of tiny dots placed at 100000 scaled independent samples

$$\frac{z[\,]}{\sqrt{2\,\text{Log}[100000]}}.$$

This is an example of "normal patterning" in which independent normal samples, even in higher dimensions, take on the shapes of ellipses. In one dimension the ellipse is a line segment.

3. The joint normal density for two **independent** z-scores $\{Z_1, Z_2\}$ (called by {z[],z[]}) is for each possible values (z_1, z_2) given by the **product** of their marginal densities and

is therefore

$$f(z_1,\ z_2) = \frac{1}{\sqrt{2\pi}}\ e^{-\frac{z_1{}^2}{2}}\ \frac{1}{\sqrt{2\pi}}\ e^{-\frac{z_2{}^2}{2}}$$

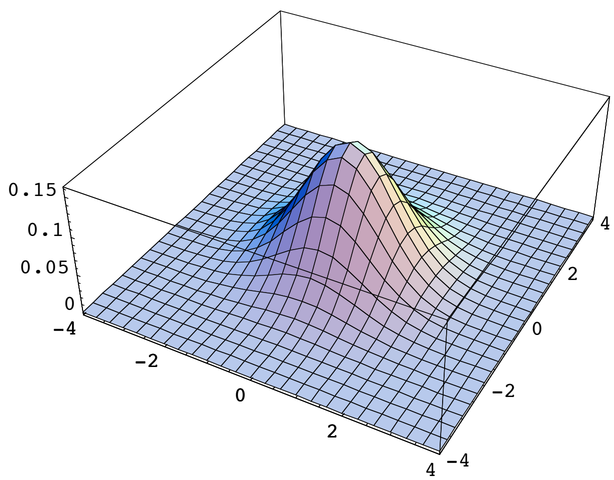
If we plot this joint density of two independent standard normal r.v. it is seen to have the **isotropic** property. What does that mean? Refer to the pictures below (due to default values in digitization the contour plot is jagged, but it should be perfectly smooth).

In[5]:=

```
f[u_, v_] := (Exp[-u^2 / 2] / √(2 π)) (Exp[-v^2 / 2] / √(2 π))
```

In[8]:=

```
Plot3D[f[u, v], {u, -4, 4}, {v, -4, 4}, PlotRange → All]
```

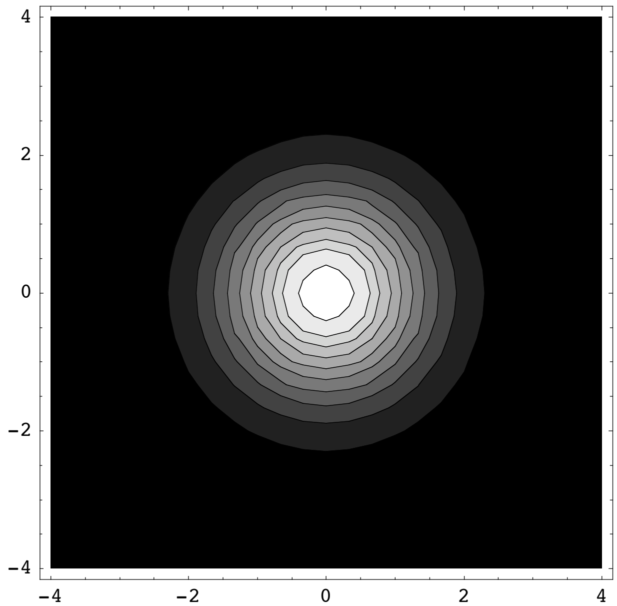


Out[8]=

```
- SurfaceGraphics -
```

In[49]:=

```
ContourPlot[f[u, v], {u, -4, 4},
 {v, -4, 4}, PlotRange → All]
```



Out[49]=

```
- ContourGraphics -
```

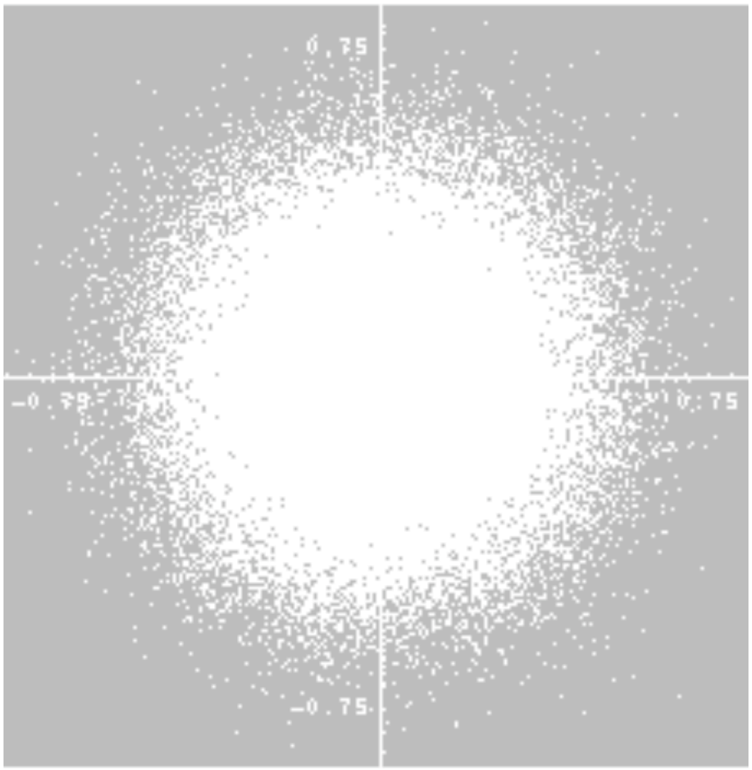
4. Normal patterning occurs in every dimension provided the random variables are jointly normal. Below we have a plot of 100000 independent samples scaled back towards the mean {0,0}:

$$\frac{\{z[\,], z[\,]\}}{\sqrt{2\,\mathrm{Log}[100000]}}$$

This should reveal the contour shape above. Keep in mind these are entirely independent samples but under proper scaling back to their mean they reveal the proper shape of the contours of any given normal density generating the samples.

In[44]:=

```
ListPlot[Table[{z[1], z[2]}, {i, 1, 100000}] /
   Sqrt[2 Log[100000]], AspectRatio → Automatic,
  Background → GrayLevel[0.7],
  DefaultColor → RGBColor[1, 1, 1]]
```



5. In the plot just above locate the following **scaled** samples{z[1], z[2]} / $\sqrt{2\,\mathbf{Log[8]}}$. Use a small circle to identify each of these **scaled** points.

```
Table[{z[1], z[2]}, {i, 1, 8}] / Sqrt[2 Log[8]]
{{-0.257824, 0.195439}, {-0.290177, 0.215324},
 {0.136383, -0.22303}, {-0.974893, -0.656856},
 {-0.302932, -0.820734}, {0.424575, 0.667169},
 {-0.193937, 0.433257}, {0.0442738, 0.015511}}
```

The thing to remember is that all **multivariate normal** samples x_1, ..., x_n, regardless of dimension, obey this phenonenon. Depending upon the context each x_i may be a normally distributed random number, vector or even a normally distributed random curve. In the case of random vectors or curves the coordinates may be mutually correlated (dependent). Regardless, the independent scaled (values, vectors or curves)

$$\left\{ \frac{x_1}{\sqrt{2\,\mathrm{Log}[n]}}, \ldots, \frac{x_n}{\sqrt{2\,\mathrm{Log}[n]}} \right\}$$

will for large n plot in a shape revealing the countours of the normal density generating those normal sample objects. In simple vector plots these will appear as ellipses but may assume other shapes when plotting curves or other complicated normally distributed objects.

Remarks. We have studied the multiple linear regression model:

$$\boxed{y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_d x_{id} + \epsilon_i \text{ for } i = 1, \ldots, n.}$$

This may also be written in matrix form:

$$y = x\,\beta + \epsilon, \quad y \text{ is } n\times 1,\ x \text{ is } n\times d,\ \beta \text{ is } d\times 1,\ \epsilon \text{ is } n\times 1$$

with

$$y = \begin{matrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{matrix} \qquad x = \begin{matrix} x_{11} & \ldots & x_{1d} \\ x_{21} & \ldots & x_{2d} \\ . & . & . \\ . & . & . \\ . & . & . \\ x_{n1} & \ldots & x_{nd} \end{matrix} \qquad \beta = \begin{matrix} \beta_1 \\ \beta_2 \\ . \\ \beta_d \end{matrix} \qquad \epsilon = \begin{matrix} \epsilon_1 \\ \epsilon_2 \\ . \\ . \\ . \\ \epsilon_n \end{matrix}$$

Solving the normal equations of least squares obtained from differentiating the sum of squares of discrepancies, left vertically by any proposed fit of the form x $\hat{\beta}$ to y, we found that if the columns of x are **linearly independent** the unique **coefficients of least squares fit** are provided by:

$$\hat{\beta} = (x^{tr}\, x)^{-1}\, x^{tr}\, y = \beta + \boxed{(x^{tr}\, x)^{-1}\, x^{tr}\, \epsilon}$$

The term in the box is **least squares performed on errors**. If one uses this least squares fit whose coefficients are $\hat{\beta}$ the resulting fitted values $\hat{y}$ = x $\hat{\beta}$ will ordinarily not fit the data y perfectly but will leave **residuals**:

$$\hat{\epsilon} = y - \hat{y} = y - x\,\hat{\beta}$$

If the regression model above is satisfied for independent N[0, σ^2] errors ϵ_i this induces a random distribution on $\hat{\beta}$, which after all depends (linearly in fact) upon these errors. That distribution is then **multivariate normal** with:

$$E\ \hat{\beta}_j = \beta_j \qquad \text{and} \qquad \text{Cov}(\hat{\beta}_{j1}, \hat{\beta}_{j2}) = \left[(x^{tr}\, x)^{-1}\right]_{j1\, j2} \sigma^2$$

for all j, j1, j2 from 1 to d. Unknown errors' standard deviation σ is estimated by a **modified sample standard deviation** of the list of residuals $\hat{\epsilon}$:

$$s_{\hat{\epsilon}} = \sqrt{\frac{n}{n-d}}\ \sqrt{\overline{\hat{\epsilon}^2} - \bar{\hat{\epsilon}}^2}$$

In the above we see that the modification is to **use divisor n—d instead of the customary n—1**. Turning to confidence intervals for the estimated coefficients $\hat{\beta}_j$ we have estimated margins of error:

$$(\text{t or z for 95\%}) \sqrt{[(x^{tr}\, x)^{-1}]_{j\,j}\, s_{\hat{\epsilon}}^{2}}$$

Remarks. For independent N[0, σ^2] errors t is applicable and exact for all n > d in the above. On the other hand z is an approximation valid for large n since we are estimating σ. If the constant term is included in the model, as is most often the case, there is the simplifiaction that the sample mean of the least squares residuals is zero. That is, **with constant term $\bar{\hat{\epsilon}} = 0$**.

5. What is the key connection between the usual **linear model** and a **processes under statistical control**?

> **Ans. Variables $x_1, \ldots, x_d$, y of a process under statistical control necessarily satisfy a linear model y = x $\beta + \epsilon$ for some β and σ with $(x_1, \ldots, x_d).(\beta_1, \ldots, \beta_d)$ being the mean response of y when $x_1, \ldots, x_d$ are specified and σ^2 being the conditional variance of y about $(x_1, \ldots, x_d).(\beta_1, \ldots, \beta_d)$ when $(x_1, \ldots, x_d)$ are specified.**

Note that σ does not vary with $(x_1, \ldots, x_d)$, exactly as is the case with the usual linear model. Now, **draw the picture illustrating this phenomenon for the case of a two dimensional plot of (x,y) pairs that are jointly normally distributed (under joint statistical control).**

Variables $x_1, \ldots, x_d$, y of a process under **statistical control** are regarded as **jointly normally distributed**. The role of y as dependent variable is not particularly special as regards joint normality. We could just as well be speaking about x_1 as dependent variable and the rest, including y, as independent variables, at least so far as joint normality is concerned. They are all variables "under joint statistical control." But we've singled out y because we wish to control it (perhaps) through choice of the independent variables $x_1, \ldots, x_d$.

Remarks. We've used Little Software to solve for various fits and associated quantities as per the remarks above. I will not repeat the few software calls employed but ask that you have retained the ability to know their uses and interpretations if they appear in front of you. For example is you see

betahat[{{1,4.5},{1,3.2},{1,3.12}},{36.4, 44.7, 67.7}]

you know this is a linear regression set in matrix form and that its output will be the fitted y-intercept and slope. You know also that as usually presented the (x,y) data pairs are (4.5, 36.4), (3.2, 44.7), (3.12, 67.7).

7. **Fit of least squares line for 2-dim normal plots by eye. Reading off the means and standard deviations of x and y and the correlation by eye.** It is simple:

sample means of x, y are easily seen

block off an interval of ~68% of points around mean of x

block off an interval of ~68% of points around mean of y

Now you have an idea of the sample standard deviations of x and y.

lay off a line through the means joining the point one sdx right and one sdy up

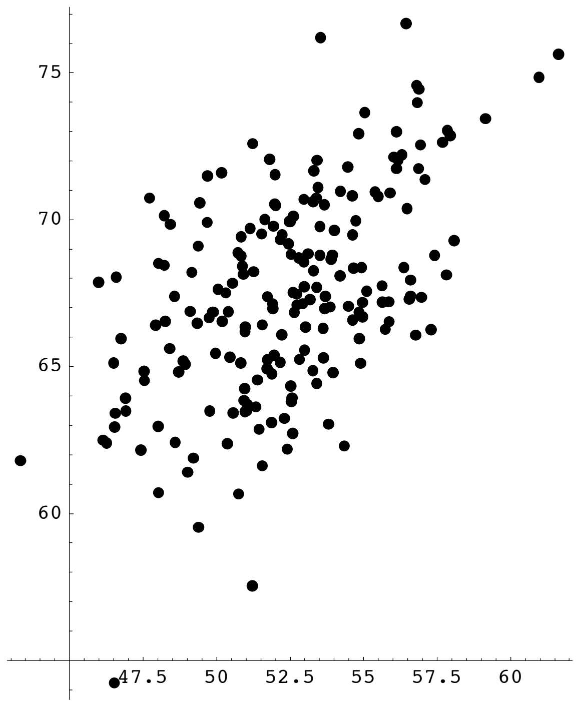
You now have the "naive" line, not the regression line.

draw the regression line by eye

The regression line plots through the vertical strip y-means.

estimate by eye the ratio of the slope of the regression line vs the naive line

That is the estimated correlation. Of course this is no substitute for calculation but it does help us to think about what is going on. Do all this for the example below which is a plot of 100 points (x,y) obtained from a correlated normal model.



Out[61]=
- Graphics -