How I Learned Loving Parameters
or
How Many Faces Are There in the World?

by

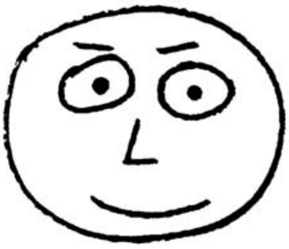
Josef Böhm
T^3 Austria
Würmla, Austria
nojoboehm@pgv.at

(Note: This activity was loosely translated and the text slightly edited by Carl Leinbach who can be contacted at leinbach@cs.gettysburg.edu)

In Clifford Pickover's very excellent book, *Computers, Pattern, Chaos and Beauty – Graphics from an Unseen World* [1], there is a short chapter called "Autocorrelation Cartoon-Faces for Speech". In this chapter several different forms of faces were generated. This struck me as a subject for a lesson on parameters. In fact it struck me that this idea is an excellent motivational tool for learning about parameters.

I have done this project first in DERIVE and then copied it almost step for step for the TI-92.

Beginning with a stylized face, similar to a children's rhyme belonging to a stick figure, called "Punkti-Punkti, Strichi-Strichi" in Austria and Germany. Isn´t it a real "Moon Face" (= "Mondgesichti")?

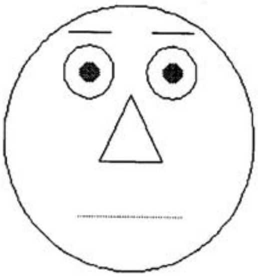


OK, now what characteristics of this face can we vary? The items in this list are not the only possibilities, but to make the task not too easy, this are my choices.

1. The shape of the face (f_f)
2. The shape of the eyes (e_f)
3. The size of the eyes (e_s)
4. The space between the eyes (e_sp)
5. The size of the pupils (p_s)
6. The length of the eyebrows (b_l)
7. The inclination of the left eyebrow (b_sl)
8. The inclination of the right eyebrow (b_sr)
9. The size of the nose (n_s)
10. The distance between the ends of the mouth (m_l)
11. The shape of the lower lip (u_f)
12. The shape of the upper lip (o_f)

This list can be extended (Ears, Hair, etc.). I leave the size of the face fixed, this will give us the prettiest ones to look at. In the list I have included within the parentheses the names of the parameters that will be responsible for the variation of the attributes of the face. We will choose values of our parameters in the interval ($-5 \leq$ parameter value ≤ 5). It is not necessary to use integers only, but we will start with integer parameters.

The "Typical Value" for all characteristic parameters is 0. From this choice we obtain the "Typical Face" from which all others will follow:



We begin with the shape of the face as a circle with radius 5. Then via the parameter form of an ellipse we receive various face forms from longish to broad depending on f_f:

[[(5+0.4*f_f)*cos(t), 5 + sin(t)] with $0 \leq t \leq 2\pi$ and $-5 \leq$ f_f ≤ 5.

In *DERIVE* one can do without the trigonometric functions using the implicit form of the ellipse. But the important fact, the use of parameter f_f - having there another meaning – remains the same (radius is 6 and `m, n` control the position on the PC-screen):

```
fa:=(x-m)^2/(6+0.25*f_f)^2+(y-n)^2/36=1
```

Three parameters are required to describe the eyes. They are the distance between the eyes, the size of the eyes, and the shape of the eyes.

The "Typical Eyes" have their centers at ($\pm$1.7, 2) and are circular shaped with a radius of 1.2. Then the parameters e_s, e_f, and e_sp have their effect to yield the parametric representation

[(1.2 + 0.1*e_s + 0.1*e_f)*cos(t) + 1.7 + 0.1*e_sp, (1.2 +0.1*e_s)*sin(t) + 2]

As a teacher you could also present the expression eyr_f and let the students investigate and describe the effect of the parameters and their interplay.

So step by step we can create our face.

The display of the nose is not a big problem. One defines the nose as a matrix, represented as three different points to make a triangle: [[0,1.7][-1.5,-1.3][1.5,-1.3]]. The order is important. The points are stretched or compressed from the origin by the factor 1 + 0.1*n_s. The triangle is drawn on the screen by using the `line` command.

I want the pupils to stand out, so I apply the `shade` command. Here we use the explicit representation of the upper and lower halves of the pupils. The pupils are "only" circles of variable size, their center coincides with the center of the eye. Two random numbers influence direction and density of shading.

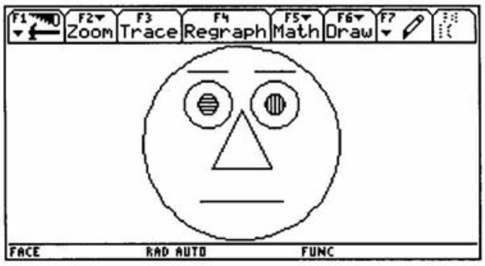
Especially interesting and profitable for understanding parameters is considering the eyebrows and the mouth. The middle of the eyebrow is associated with the distance between the eyes and the size of the eyes (`brx` and `bry`). In order to control the length of the eybrows one has to normalize their direction vector. The slope of the eyebrow will be determined by tan(b_sr*π/20) (respectively `b_zl` since each brow is "unique"). The extent of the brow will again be drawn from a specific parameter's value using an appropriate parameter representation.

Thus far we have used line segments, ellipses, and matrices. We are still missing quadratic functions. Lower and upper lip give us an opportunity to introduce parabolas. Coming out from the two corners of the mouth (±(2.1+0.1m_l) ,-3) decides the mouth shape as two parabolas – one for the bottom, the other for the top. The space between the lips is determined by the two parameters u_f and o_f. The parabolas will be drawn only in the area between the two corners of the mouth, -mptr and +mptr. If you will accept a longer time for plotting then shade the mouth, too to have a nice effect.

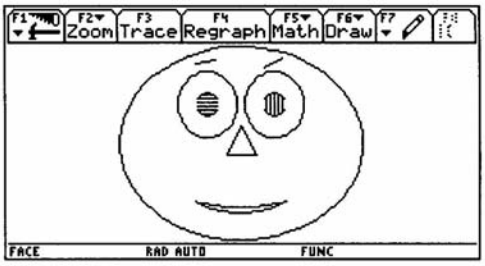
The parabola has the form $y = a\,x^2 + b$ and is defined by two points, (mptr,-3) and (0,-3±0.3*l_f) where l_f stands for u_f and o_f, respectively.

```
faces(f_f,e_f,e_s,e_sp,p_s,b_l,br_sr,br_sr,br_sl,n_s,m_l,u_f,o_f)
Prgm
Local nose,bry,brx,bl,mptr
ClrDraw:ClrGraph
(1+0.1*n_s)*[[0,1.7][-1.5,-1.3][1.5,-1.3]]→nose
©Face
DrawParm(5+0.4*f_f)*cos(t),5*sin(t),0,6.4,0.1
©Right and Left Eye
DrawParm(1.2+0.1e_s+0.1*e_f)*cos(t)+1.7+0.1*e_sp,
       (1.2+0.1*e_s)*sin(t)+2,0,6.4,0.1
DrawParm(1.2+0.1e_s+0.1*e_f)*cos(t)-1.7-0.1*e_sp,
       (1.2+0.1*e_s)*sin(t)+2,0,6.4,0.1
Shade -√((0.6+0.08*p_s)^2-(x+1.7+0.1*e_sp)^2+2,√((0.6+0.08*p_s)^2-
      (x-1.7+0.1*e_sp)^2)+2,,,rand(4),rand(3)
Shade -√((0.6+0.08*p_s)^2-(x-1.7-0.1*e_sp)^2+2,√((0.6+0.08*p_s)^2-
      (x-1.7-0.1*e_sp)^2)+2,,,rand(4),rand(3)
©Right and left eye brows
1.7+0.1*e_sp→brx: 3.7+0.1*e_s→bry
norm([[1,tan(br_sr*π/20)]] →bl
DrawParm brx+t/bl,bry+t*tan(br_sr*π/20)/bl,-1-0.06*bl,1+0.06*b_l,0.05
norm([[1,tan(-br_sl*π/20)]] →bl
DrawParm -brx+t/bl,bry+t*tan(-br_sl*π/20)/bl,-1-0.06*bl,1+0.06*b_l,0.05
©Nose
Line nose[1,1],nose[1,2],nose[2,1],nose[2,2]
Line nose[2,1],nose[2,2],nose[3,1],nose[3,2]
Line nose[3,1],nose[3,2],nose[1,1],nose[1,2]
©Mouth
2.1+0.1*m_l→mptr
DrawFunc when(x≥-mptr and x≤mptr,-0.3*u_f/mptr^2*x^2-3+u_f*0.3,undef)
DrawFunc when(x≥-mptr and x≤mptr,0.3*o_f/mptr^2*x^2-3-o_f*0.3,undef)
EndPrgm
```

Now the program is ready to be tested. We begin with the "Typical Face" and another common face.



faces(0,0,0,0,0,0,0,0,0,0,0,0)



faces(3,-2,5,0,1,-4,3,-1,-5,3,-2,1)

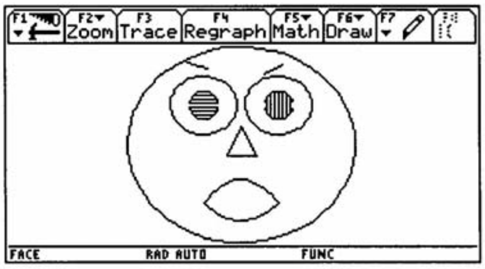
This goes in the right direction, but we would also like to leave things to chance. Not only do we want to fix the parameters, but also to let them be chosen randomly. At first we automate what we have done thus far and choose integer parameters from –5 to +5.

Now our pupils might raise the question – if they don't´the the teacher could - with this way of doing things is it possible to assign an individual face to every person in the world? How many different faces could be designed in this way?

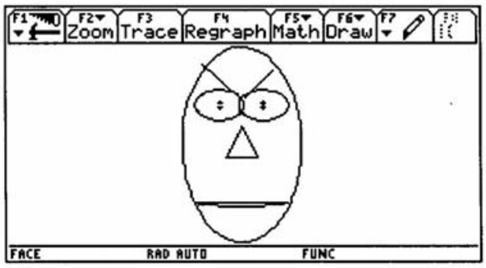
Then we let go the – unnecessary – demand for integer parameter values and allow all random values between –6 and +6.

```
faceint()
Prgm
Local v_
seq(6-rand(11),k,1,12)→v_
faces(v_[1],v_[2],v_[3],v_[4],v_[5],v_[6],v_[7],v_[8],
       v_[9],v_[10],v_[11],v_[12])
EndPrgm

facernd()
Prgm
Local v_
seq(6-12*rand(),k,1,12)→v
faces(v_[1],v_[2],v_[3],v_[4],v_[5],v_[6],v_[7],v_[8],
       v_[9],v_[10],v_[11],v_[12])
EndPrgm
```



faceint( )



facernd( )

This is all very nice, but when nature deals the cards, the characteristics are not equally distributed within an interval, but they are normally distributed. And here we have a beautiful chance to introduce a demonstration and a visualization of the Mean and Variance. To simplify matters we give the same mean and standard deviation to all parameters, and with these values generate faces of a certain type, which might vary more or less.

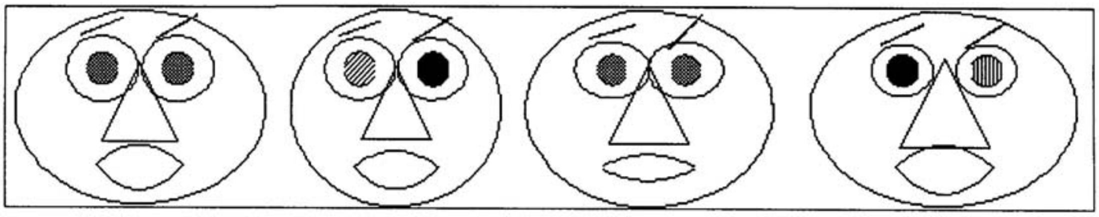
```
facenorm(mu,sigma)
Prgm
Local v_
seq(randNorm(mu,sigma),k,1,12) →v_
faces(v_[1],v_[2],v_[3],v_[4],v_[5],v_[6],v_[7],v_[8],
       v_[9],v_[10],v_[11],v_[12])
EndPrgm
```

When statistics enters the game, then one single event is not sufficient at all. I will create a whole "Ancestor´s Gallery" of my forebearers on the screen.

```
ancestors(mu,sigma,anz)
Prgm
Local i_
ClrDraw
For i_,1,anz
facenorm(mu,sigma)
StoPic #("fa"&string(i_))
ClrDraw
EndFor
EndPrgm
```

```
showfam(anz)
Prgm
Local i_
For i_,1,anz
ClrDraw
RclPic #("fa"&string(i_))
Pause
EndFor
EndPrgm
```

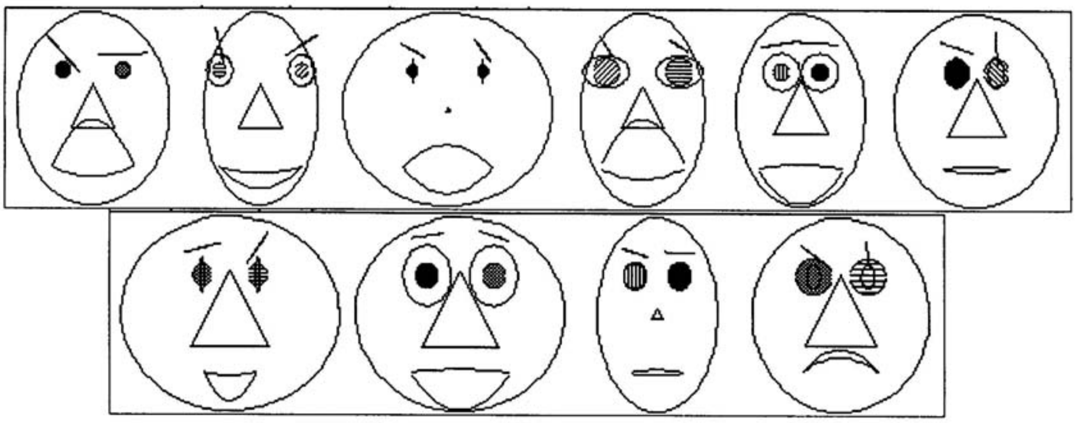
So I can perhaps reproduce my Forefathers on my father's side of the family – they all had faces such that the ratios of the sizes were essentially unchanged: `ancestors(3,1,4.`



Greatgrandfather Josef G.g.grandfather Josef G.g.g.grandfather Sepp G.g.g.uncle Rudolf

When we do the same investigation on my mother's side of the family we find a much greater variety in their physical characteristics, caused by their environment and history:

`ancestors(-2,3,10)` and then `showfam(10)`

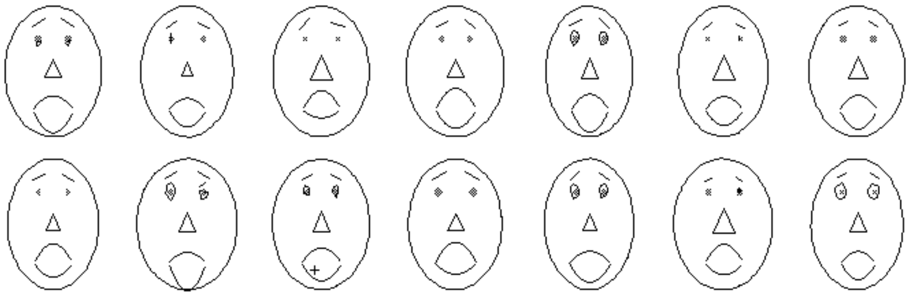


Here the greater standard deviation of 3 causes a substantially greater variability in the appearances of the faces.

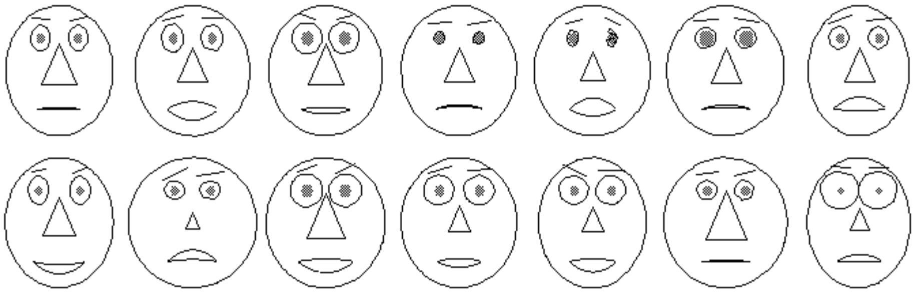
I have already done this project with DERIVE. The new programming language in DERIVE5 is challenging and at the same time fascinating. Here one can see on the PC-Screen the whole Ancestor Gallery at once like magic.

The Derive-Code will appear in the next *DERIVE & TI-92 Newsletter*, the official periodical published by the *International DERIVE & TI-92 User Group* (information at nojo.boehm@pgv.at)

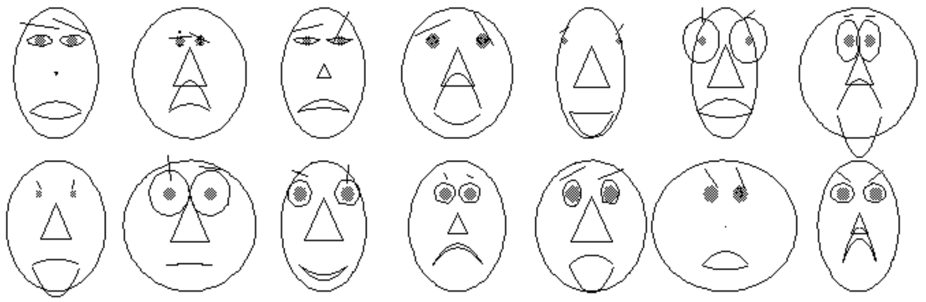
The family of Longheads: μ=-4, σ=1



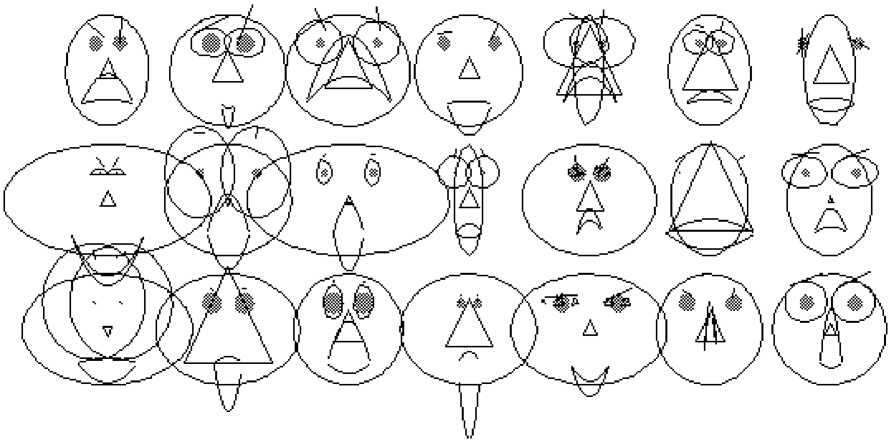
Average faces with slight variations: (μ = 0, σ= 2)



The Mean is again 0 (= average), but σ= 5 causes significant differences:



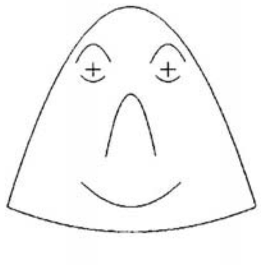
Whats about a standard deviation of 10?:



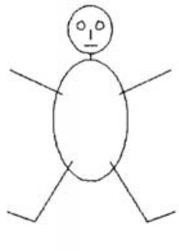
How many strange and grotesque faces are there in the world?

I will close by presenting a couple of other proposals for motivating figures support the student´s feelings for applying parameters.

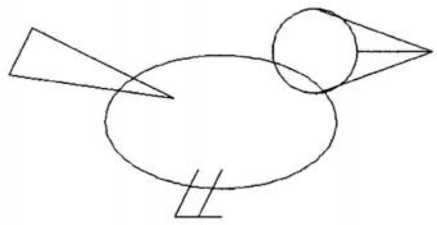
All what to do is, to have an idea for the basic figure and then select those characteristics which can usefully be varied. But from my experience I can recommend to rely on the phantasy and creativity of your students. You surely will be surprised.



Parabolicus



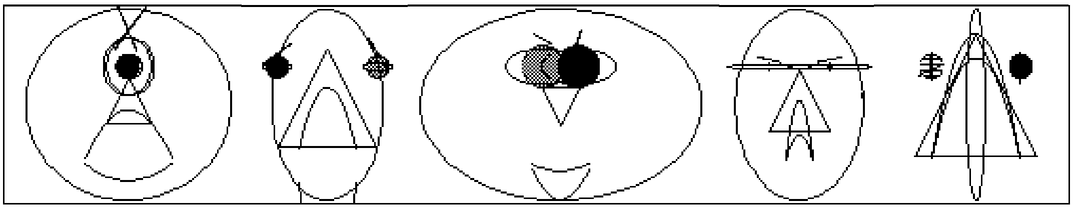
Dot,Dot-Dash,Dash



Wings for the Bird

As a summary I´d like to add that I am really enthusiastic by the connection of creativity, the use of parameters and the visualization of basic statistics concepts.

See finally 5 "Faces" designed by the TI-program



Uncle Cyclopios, Auntie Duck, Willi (why is his nose on the top?), my maths teacher KnowsAllBetter and finally my wealthy aunt Agathe.

I´d like to express my gratefulness to Carl Leinbach, who translated the German original using his German-English-dictionary. He did a great job and once more proved his enthusiasm for using technology in teaching mathematics and also the fact that maths language is international. The next challenge for him should be a Japanese or Chinese text!

Josef Böhm, nojo.boehm@pgv.at