**2019-09-24**

Recall, I have office hours by appointment also, so if you want to see me on a monday, say, send an email.

Let's figure out how to generate some data to fit.

First, lets make a random line.

> $y = m \cdot x + b$

$$y = m\,x + b \tag{1}$$

Here ae some help pages to look at about random things.

> $?random$

> $?HowDoI,WorkWithRandomGenerators$

> $rand(\,)$

$$395718860534 \tag{2}$$

> $rand(\,)$

$$193139816415 \tag{3}$$

> $rand(1\,..6)$

**proc**( ) (4)

**proc**( ) **option** *builtin* = *RandNumberInterface*; **end proc**(6, 6, 3) + 1

**end proc**

> $dice := rand(1\,..6\,):$

> $dice(\,), dice(\,), dice(\,)$

$$5, 6, 2 \tag{5}$$

> $with(RandomTools[MersenneTwister])$

$$[GenerateData, GenerateFloat, GenerateFloat64, GenerateInteger, GenerateInteger32, GenerateUnsignedInt32, GetState, NewGenerator, SetState] \tag{6}$$

> $GenerateFloat(\,)$

$$0.0809094426 \tag{7}$$

> $slope := 4 \cdot GenerateFloat(\,) - 2$

$$slope := 1.624714887 \tag{8}$$

It gives me a number between 0 and 1, want something between -2 and+2.

> $intercept := 10 \cdot GenerateFloat(\,) - 5$

$$intercept := -1.572161447 \tag{9}$$

> $targetline := slope \cdot x + intercept$

$$targetline := 1.624714887\,x - 1.572161447 \tag{10}$$

Here's another way to generate a bunch of random numbers. It is a little more versatile, since it allows us to have them be drawn from a distribution. That is, instead of having the random numbers spread evenly over some interval, instead we can have them more likely to be in one place than another. A Normal distribution is a "bell-curve" with a given mean an standard deviation.

> $with(Statistics):$

> $Sample(Normal(0, 1))(10)$

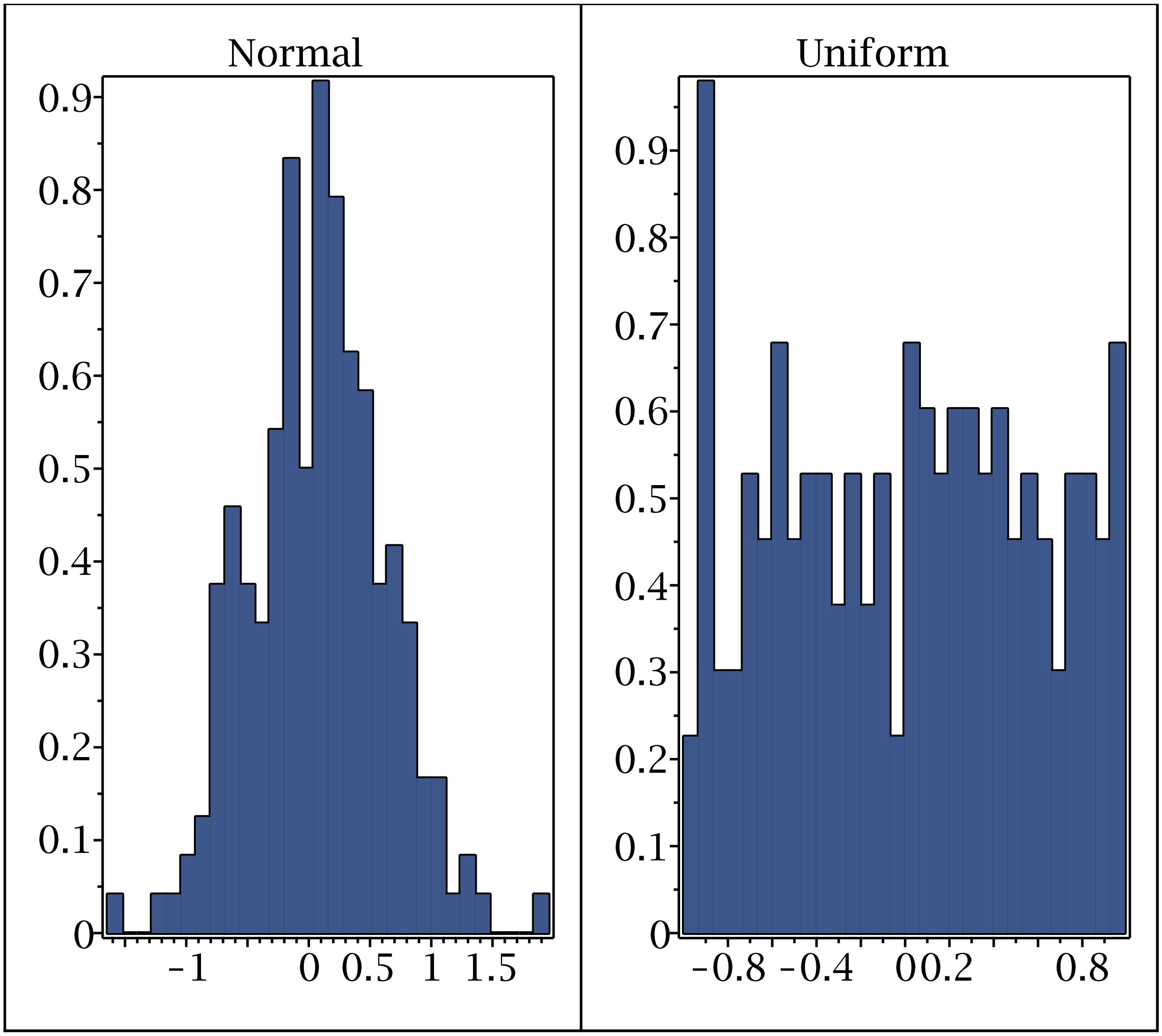
$$[-1.07242412799827, -0.329077870547065, -0.617091936909790, 0.214466745245291, -0.0254281280423846, 1.72882128417783, -1.63485675434390, 1.57117217175430, 0.170358421410408, 1.00647840875181] \quad (11)$$

> $Sample(Uniform(-5, 5))(10)$

$$[-0.510481396665271, 0.101089105510362, -2.69368070347827, 2.11880776187563, 4.84452612072164, 1.55213244267934, -2.27333020698658, -0.309844357941371, -4.96998798393336, -0.535760613518974 \quad (12)$$

Let's have a look at how these spread out. I'll use the Statistics[Histogram] to generate a frequency plot comparing 200 points from a Uniform distribution on [-1, 1] to a Normal distribution with mean 0 and std.dev 0.5.

> $plots[display](\langle Histogram(Sample(Normal(0, 0.5))(200), title = \text{"Normal"}) \mid Histogram(Sample(Uniform(-1, 1))(200), title = \text{"Uniform"})\rangle)$




So we can see that the Normal samples cluster near the mean, while the Uniform ones are more spread out.

> $xvals := Sample(Uniform(-10, 10))(20):$

> $xvals[1]$

$$-0.345084972592954 \tag{13}$$

> $eval(targetline, x = xvals[1])$

$$-2.13282613925176 \tag{14}$$

> $targetline(xvals[1])$

$$1.624714887\, x(-0.345084972592954) - 1.572161447 \tag{15}$$

> $linef := unapply(targetline, x)$

$$linef := x \mapsto 1.624714887\, x - 1.572161447 \tag{16}$$

> $linef(xvals[1])$

$$-2.13282613925176 \tag{17}$$

> $xvals$

$[-0.345084972592954, -0.450669011377069, 7.98886497134678, -2.06248162915906, -6.84001854327599, -8.19350925787260, 5.97462610499476, 4.07610812493634, -3.93873746886630, 9.38445141344071, -1.08214844393362, 2.05674139534111, -2.92165125192468, -9.99975711677289, -8.78354922628864, 1.84798854679426, 2.02501114729715, 0.643193515469413, -1.52468265311085, -0.520829021552613$ , **(18)**

> $yvals := [seq(linef(xvals[i]), i = 1..20)]$

$yvals := [-2.13282613925176, -2.30437009889390, 11.4074664021799, -4.92310605405874, -12.6852414016166, -14.8842779150379, 8.13490253004381, 5.05035210460572, -7.97148684865177, 13.6748964707453, -3.33034413380284, 1.76945691671985, -6.31901173062422, -17.8189157010051, -15.8429246356485, 1.43029305598213, 1.71790431035463, -0.527155367194980, -4.04933605145985, -2.41836011189817]$ **(19)**

or I could use map

> $yvals := map(linef, xvals)$

$yvals := [-2.13282613925176, -2.30437009889390, 11.4074664021799, -4.92310605405874, -12.6852414016166, -14.8842779150379, 8.13490253004381, 5.05035210460572, -7.97148684865177, 13.6748964707453, -3.33034413380284, 1.76945691671985, -6.31901173062422, -17.8189157010051, -15.8429246356485, 1.43029305598213, 1.71790431035463, -0.527155367194980, -4.04933605145985, -2.41836011189817]$ **(20)**

> $pairXY := x \to [x, linef(x)]$

$$pairXY := x \mapsto [x, linef(x)] \quad \textbf{(21)}$$

> $XYvals := map(pairXY, xvals)$

$XYvals := [[-0.345084972592954, -2.13282613925176], [-0.450669011377069, -2.30437009889390], [7.98886497134678, 11.4074664021799], [-2.06248162915906, -4.92310605405874], [-6.84001854327599, -12.6852414016166], [-8.19350925787260, -14.8842779150379], [5.97462610499476, 8.13490253004381], [4.07610812493634, 5.05035210460572], [-3.93873746886630, -7.97148684865177], [9.38445141344071, 13.6748964707453], [-1.08214844393362, -3.33034413380284], [2.05674139534111, 1.76945691671985], [-2.92165125192468, -6.31901173062422], [-9.99975711677289, -17.8189157010051], [-8.78354922628864, -15.8429246356485], [1.84798854679426, 1.43029305598213],$ **(22)**

$[2.02501114729715, 1.71790431035463], [0.643193515469413, -0.527155367194980], [-1.52468265311085, -4.04933605145985], [-0.520829021552613, -2.41836011189817]]$

or, more efficiently

> $XYvals := map(\, x \rightarrow [x, linef(x)], xvals)$

$$XYvals := [[-0.345084972592954, -2.13282613925176], [-0.450669011377069, -2.30437009889390], [7.98886497134678, 11.4074664021799], [-2.06248162915906, -4.92310605405874], [-6.84001854327599, -12.6852414016166], [-8.19350925787260, -14.8842779150379], [5.97462610499476, 8.13490253004381], [4.07610812493634, 5.05035210460572], [-3.93873746886630, -7.97148684865177], [9.38445141344071, 13.6748964707453], [-1.08214844393362, -3.33034413380284], [2.05674139534111, 1.76945691671985], [-2.92165125192468, -6.31901173062422], [-9.99975711677289, -17.8189157010051], [-8.78354922628864, -15.8429246356485], [1.84798854679426, 1.43029305598213], [2.02501114729715, 1.71790431035463], [0.643193515469413, -0.527155367194980], [-1.52468265311085, -4.04933605145985], [-0.520829021552613, -2.41836011189817]] \quad \textbf{(23)}$$

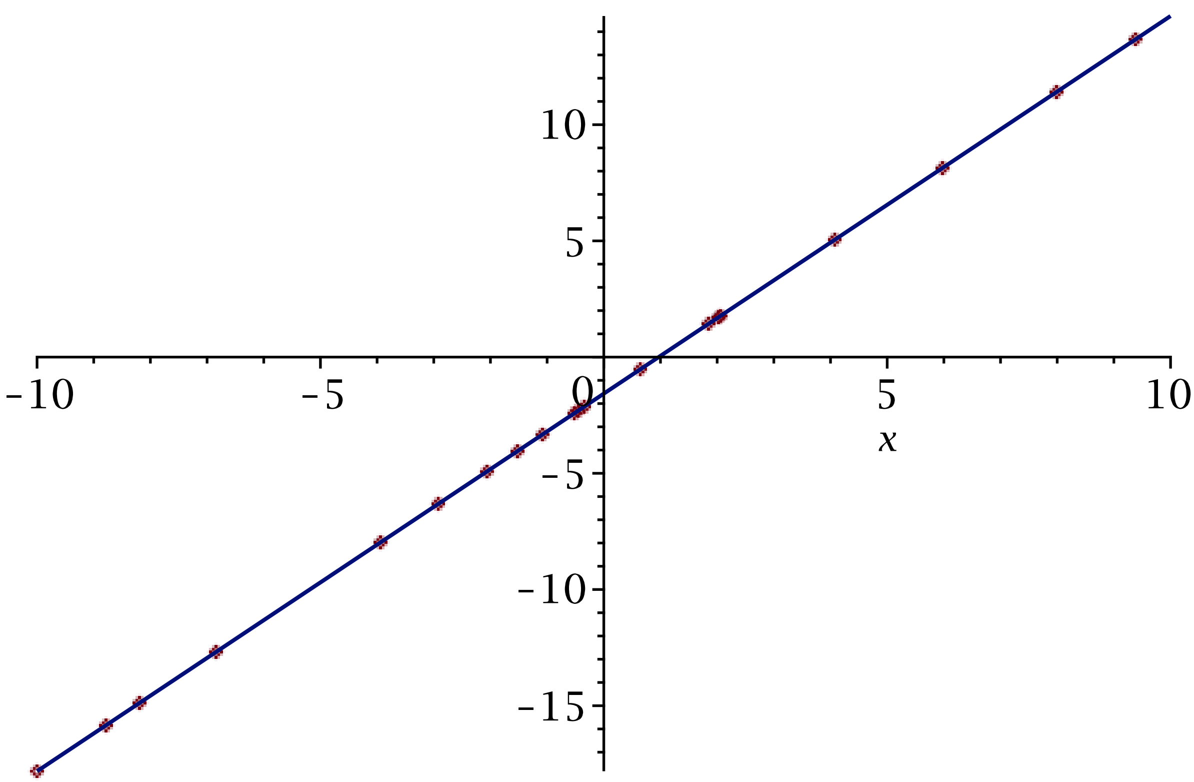
> $plot(\, [XYvals, linef(x)], x = -10..10, style = [point, line])$

```
Error, (in plot) incorrect first argument [Vector[row](20, {(1)
= [HFloat(-0.3450849725929537), HFloat(-2.132826139251759)],
(2) = [HFloat(-0.4506690113770695), HFloat(-2.304370098893897)
], (3) = [HFloat(7.988864971346782), HFloat(11.407466402179946)
], (4) = [HFloat(-2.062481629159059), HFloat
(-4.923106054058737)], (5) = [HFloat(-6.840018543275992),
HFloat(-12.685241401616558)], (6) = [HFloat
(-8.193509257872599), HFloat(-14.884277915037934)], (7) =
[HFloat(5.974626104994758), HFloat(8.13490253004381)], (8) =
[HFloat(4.0761081249363365), HFloat(5.0503521046057225)], (9) =
[HFloat(-3.9387374688662984), HFloat(-7 ... 9021552613), HFloat
(-2.418360111898174)]}), 1.624714887*x-1.572161447]
```

Ooops, I forgot that Sample gives  me a Vector instead of a list; let's covert it into something plot can deal with.

> $XYvalsL := convert(XYvals, listlist)$ :

> $plot(\, [XYvalsL, linef(x)], x = -10..10, style = [point, line])$




> $Noise := Sample(Normal(0, 0.5))(20)$

$Noise := [0.500432244431485, -0.566470167940209, 0.267772261909304, -0.187214187911572, 0.442386705419446, -0.491470956527422, -0.937864453301053, -0.0321257433690264, 0.137194506456044, 0.396159675432445, -0.118407965413127, -0.464284114895288, -0.310629495524704, 0.446284927640902, -0.126982831806239, 0.135742216006806, 0.448691808227012, 0.675057879136975, 0.811324418836003, -0.121833519713645]$ **(24)**

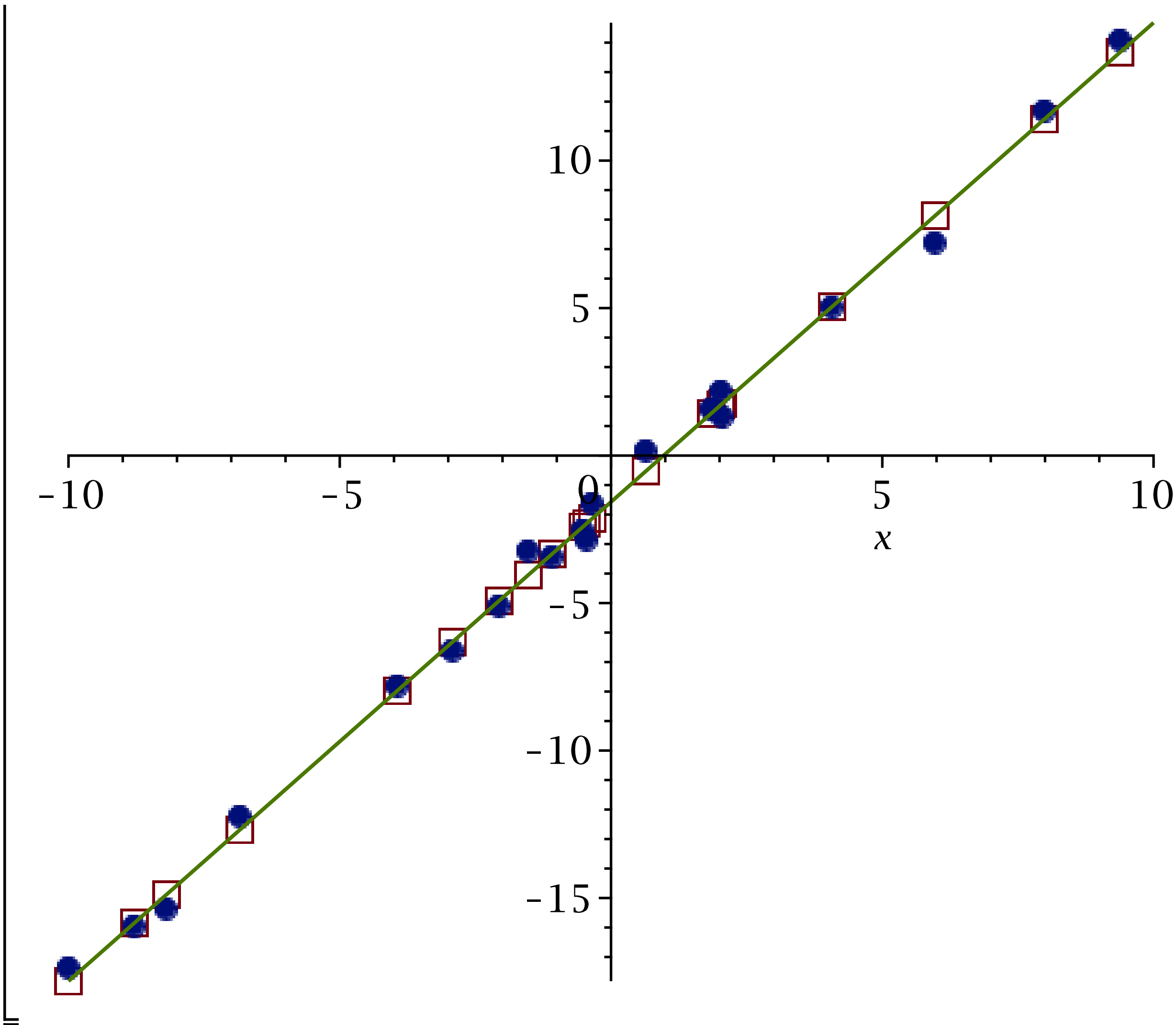
make some noisy data, as xy list

> $[xvals[10], linef(xvals[10]) + Noise[10]]$

$[9.38445141344071, 14.0710561461778]$ **(25)**

> $Noisy := [\ seq(\ [xvals[i], linef(xvals[i]) + Noise[i]], i = 1..20)\ ]:$
> \# I don't have to convert since I'm building it as a list.

> $plot(\ [XYvalsL, Noisy, linef(x)], x = -10..10, style = [point\$2, line], symbolsize = [20, 15], symbol = [box, solidcircle])$



10
5
0
-5
-10
-15
-10
-5
5
10
x

>