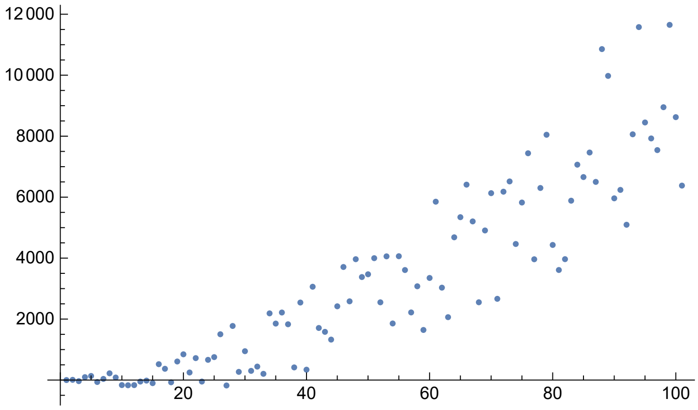
Randomização proporcional ao crescimento de *y*.

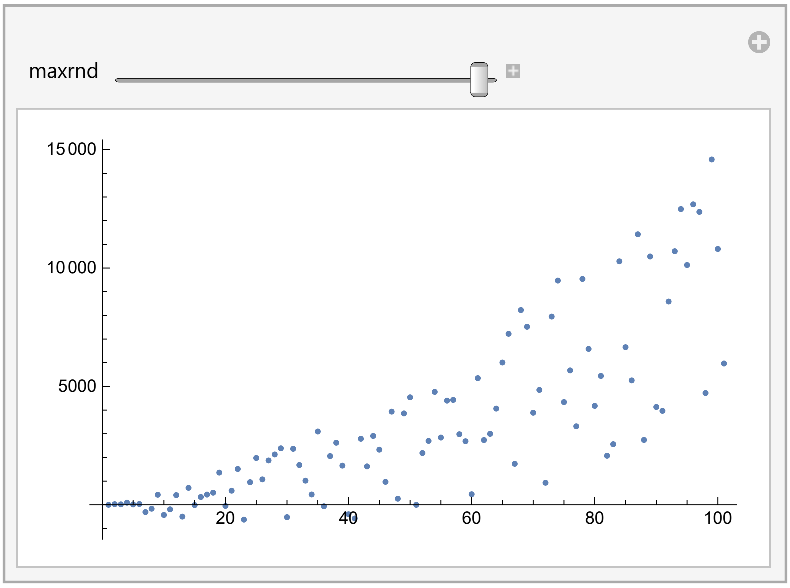
In[ ]:=
```
Clear[points1]
points1 = Table[x^2 + RandomReal[{-40 * x, 40 * x}], {x, 0, 100, 1}];
ListPlot[points1]
```

Out[ ]=



In[ ]:=
```
Manipulate[ListPlot[Table[x^2 + RandomReal[{maxrnd * x * -1, maxrnd * x}], {x, 0, 100, 1}],
  ImageSize → Medium], {maxrnd, 0, 60}]
```

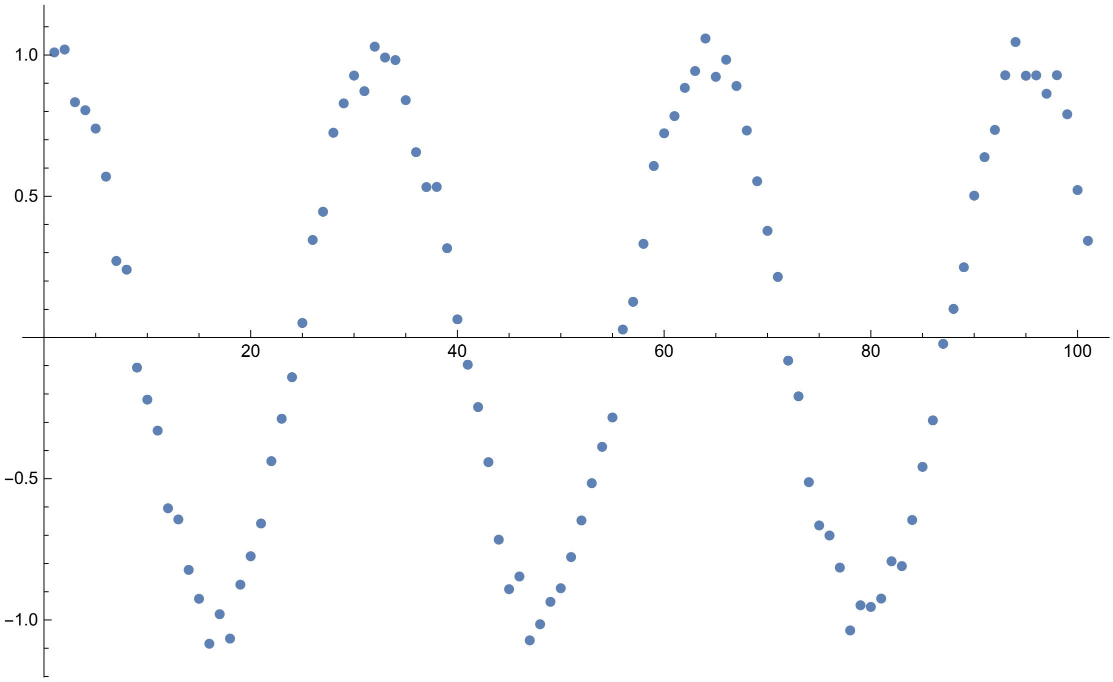
Out[ ]=



In[25]:=

```
Clear[points2]
points2=Table[Cos[x]+RandomReal[{-0.1,0.1}],{x,0,20,0.2}];
ListPlot[points2,ImageSize→Large]
```

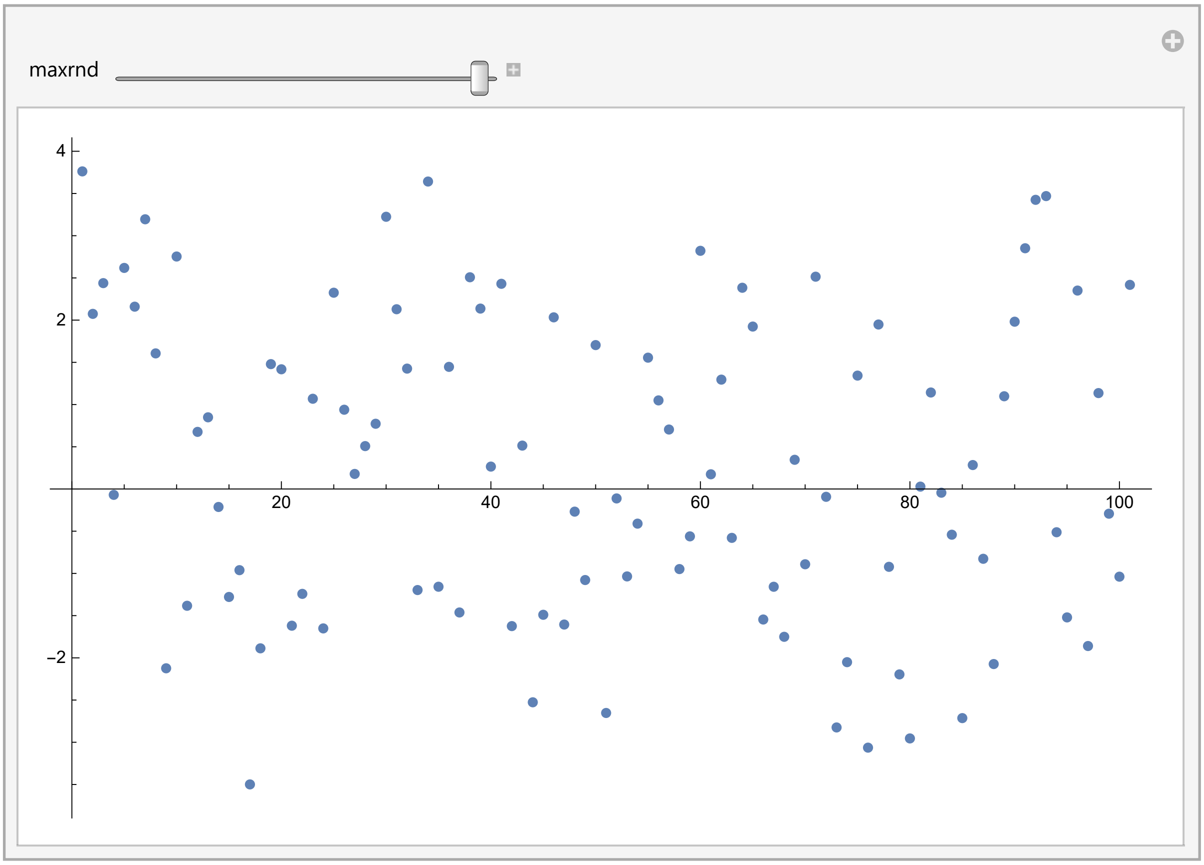
Out[27]=



In[ ]:=

```
Manipulate[ListPlot[Table[Cos[x] + RandomReal[{maxrnd * -1, maxrnd}], {x, 0, 20, 0.2}],
  ImageSize → Large], {maxrnd, 0, 3}]
```

Out[ ]=

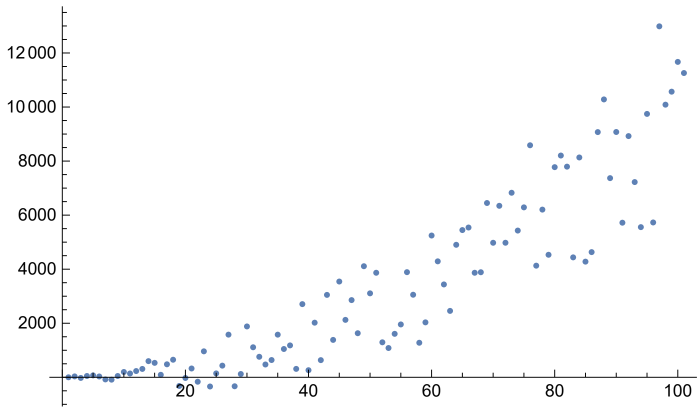


In[28]:=
```
Clear[MakePoints2]
MakePoints2=Function[maxrnd,
    Table[Cos[x]+RandomReal[{maxrnd*-1,0.1}],{x,0,20,0.2}]];
```

Uma série semi-exponencial.

In[ ]:=
```
Clear[exp1];
exp1 = {0.`, 32.896960274254354`, -23.264656131094`, 48.413892090903175`,
   71.04130636629714`, 30.71571787217124`, -78.2400103320191`, -87.31920065999861`,
   47.815364875810246`, 196.62476367032218`, 140.72329802485683`, 231.23106188319366`,
   309.64887890045566`, 597.9161180482959`, 534.4017554894099`, 92.12528425774008`,
   480.88506962469`, 649.8705784591684`, -320.7181849098183`, -24.702074266884892`,
   326.1435564229646`, -163.85167012602165`, 961.1304491019464`, -340.3394164950407`,
   143.82836032124305`, 431.42986255900723`, 1578.4007056162009`,
   -331.77117810482605`, 121.8891793858875`, 1881.7183407592584`,
   1111.0316516201424`, 759.0700853285616`, 473.93543584330837`, 639.1504820253281`,
   1578.0609807200726`, 1047.2069391937493`, 1180.2116238147746`, 311.38953864697305`,
   2707.5507803101746`, 262.25587440577783`, 2020.8812908511163`, 635.1455123643627`,
   3049.2598696308205`, 1383.6834126333952`, 3543.357458164154`, 2124.8960870507462`,
   2856.655474239441`, 1631.1153162436922`, 4110.062929401679`, 3106.6213600204674`,
   3867.8678376996013`, 1294.7531440598386`, 1083.6785921528563`, 1608.3102976058944`,
   1955.8469590368295`, 3890.0753552324204`, 3054.696673072499`, 1278.2091595299044`,
   2032.616653686001`, 5245.8239752182435`, 4290.528814447963`, 3436.116038186936`,
   2456.9595315230745`, 4902.6001502039`, 5447.817208449704`, 5541.796735266875`,
   3867.8474375105534`, 3887.3461606785577`, 6448.922337756563`, 4977.387455625809`,
   6345.161243921662`, 4978.330020108655`, 6828.175823249763`, 5429.309086348778`,
   6287.837027392425`, 8585.229511620408`, 4130.349965977841`, 6204.706332728636`,
   4533.579247688535`, 7775.620642994407`, 8205.230927406965`, 7794.1014083145965`,
   4437.41896966293`, 8134.088842539819`, 4279.5519850953215`, 4631.983781903142`,
   9072.85484720021`, 10281.716367605424`, 7369.244378403393`, 9076.359273899754`,
   5720.2606021228985`, 8924.938288695506`, 7223.622960159175`, 5556.1948215725915`,
   9745.719646746831`, 5729.048433533139`, 12986.045596693588`, 10087.379834632671`,
   10568.321810409445`, 11670.140893906295`, 11258.291964960259`};
ListPlot[
 exp1]
```

Out[ ]=



Descobrir a relação de y com x.

Comparar cada y com a $f(x)$ modelo no mesmo x. Minimizar a soma das diferenças.

Por exemplo:

$f(x) = x^2$

Em $x = 30$, $y = 900$.

Para isso é preciso atribuir *x* para os valores da série.
O índice (do list) começa em 1.

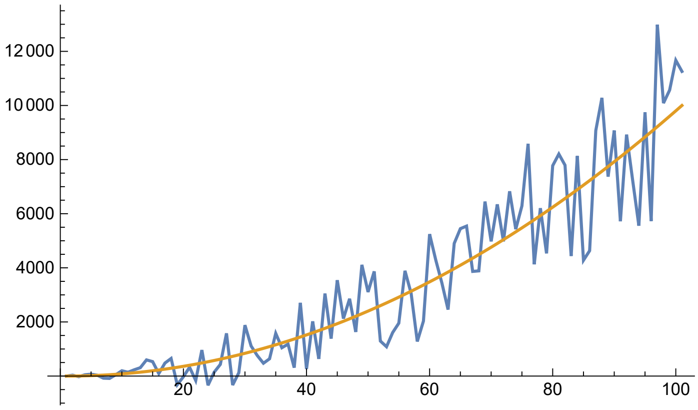
In[ ]:=
```
exp1[[30]]
```
Out[ ]=
```
1881.72
```

In[ ]:=
```
Clear[fexp1model1]
fexp1model1 = Function[x, x^2];
```

In[ ]:=
```
exp1[[30]] - fexp1model1[30]
```
Out[ ]=
```
1248.38
```

In[ ]:=
```
Clear[exp1model1]
exp1model1 = Table[fexp1model1[x], {x, 0, 100, 1}]
```
Out[ ]=
```
{0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361,
 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961, 1024, 1089, 1156, 1225,
 1296, 1369, 1444, 1521, 1600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2304,
 2401, 2500, 2601, 2704, 2809, 2916, 3025, 3136, 3249, 3364, 3481, 3600, 3721,
 3844, 3969, 4096, 4225, 4356, 4489, 4624, 4761, 4900, 5041, 5184, 5329, 5476,
 5625, 5776, 5929, 6084, 6241, 6400, 6561, 6724, 6889, 7056, 7225, 7396, 7569,
 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9604, 9801, 10000}
```

In[ ]:=
```
ListLinePlot[{exp1, exp1model1}]
```
Out[ ]=



In[ ]:=
```
exp1[[30]] - exp1model1[[30]]
```
Out[ ]=
```
1040.72
```

Todos...

In[ ]:=
```
exp1model1 - exp1
```

Out[ ]=
```
{0., -31.897, 27.2647, -39.4139, -55.0413, -5.71572, 114.24, 136.319, 16.1846,
 -115.625, -40.7233, -110.231, -165.649, -428.916, -338.402, 132.875, -224.885,
 -360.871, 644.718, 385.702, 73.8564, 604.852, -477.13, 869.339, 432.172, 193.57,
 -902.401, 1060.77, 662.111, -1040.72, -211.032, 201.93, 550.065, 449.85, -422.061,
 177.793, 115.788, 1057.61, -1263.55, 1258.74, -420.881, 1045.85, -1285.26,
 465.317, -1607.36, -99.8961, -740.655, 577.885, -1806.06, -705.621, -1367.87,
 1306.25, 1620.32, 1200.69, 960.153, -865.075, 81.3033, 1970.79, 1331.38, -1764.82,
 -690.529, 284.884, 1387.04, -933.6, -1351.82, -1316.8, 488.153, 601.654, -1824.92,
 -216.387, -1445.16, 62.67, -1644.18, -100.309, -811.837, -2960.23, 1645.65,
 -275.706, 1550.42, -1534.62, -1805.23, -1233.1, 2286.58, -1245.09, 2776.45,
 2593.02, -1676.85, -2712.72, 374.756, -1155.36, 2379.74, -643.938, 1240.38,
 3092.81, -909.72, 3295.95, -3770.05, -678.38, -964.322, -1869.14, -1258.29}
```

In[ ]:=
```
Mean[exp1model1 - exp1]
```

Out[ ]=
```
-80.596
```

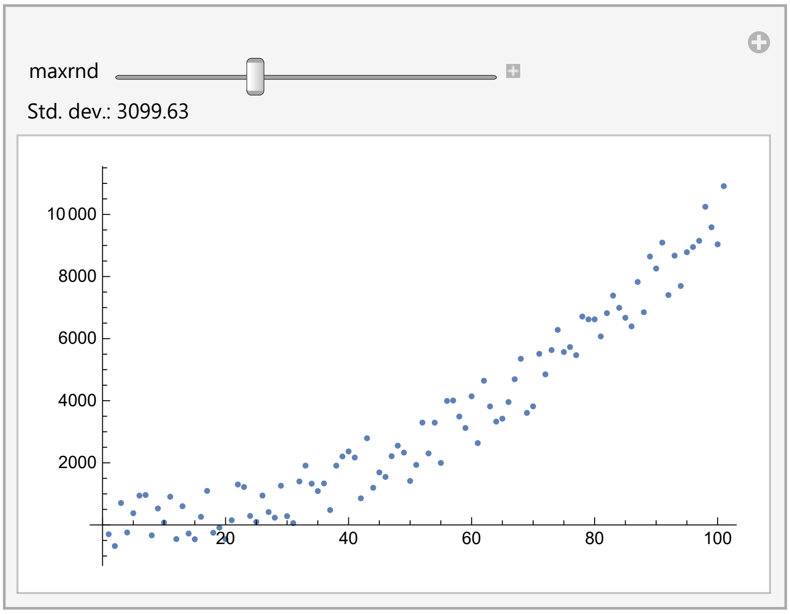
In[ ]:=
```
Total[exp1model1 - exp1]
```

Out[ ]=
```
-8140.2
```

Gerando nova série com ruído linear...

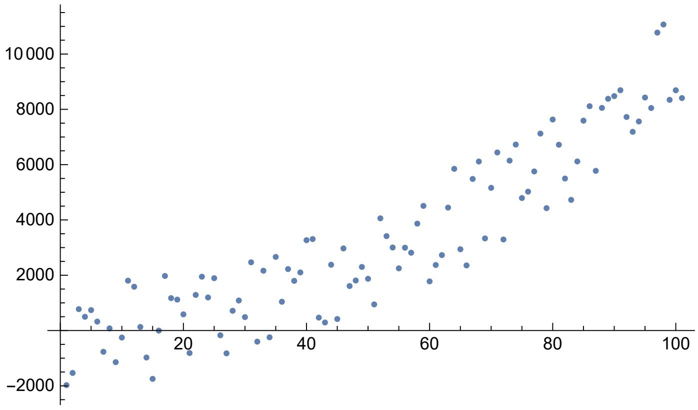
In[ ]:=
```
Manipulate[
 pts = Table[x^2 + RandomReal[{maxrnd * 50 * -1, maxrnd * 50}], {x, 0, 100, 1}];
 ListPlot[pts, ImageSize → Medium],
 {maxrnd, 0, 60},
 Dynamic["Std. dev.: " <> ToString[StandardDeviation[pts]]]
]
```

Out[ ]=



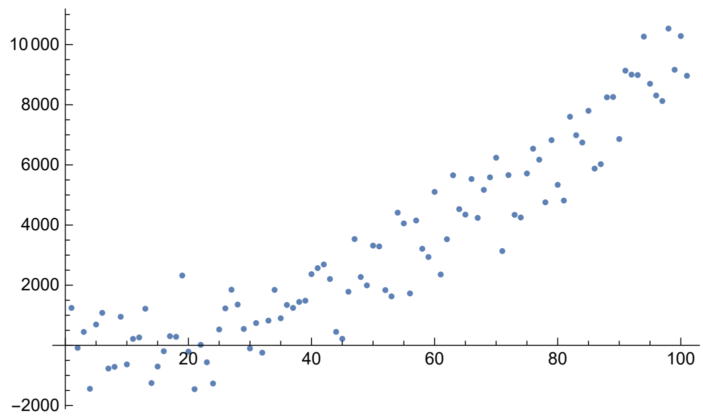
In[∘]:=
```
Clear[points1b]
maxrnd = 40;
points1b = Table[x^2 + RandomReal[{-50 * maxrnd, 50 * maxrnd}], {x, 0, 100, 1}];
ListPlot[points1b]
```

Out[∘]=



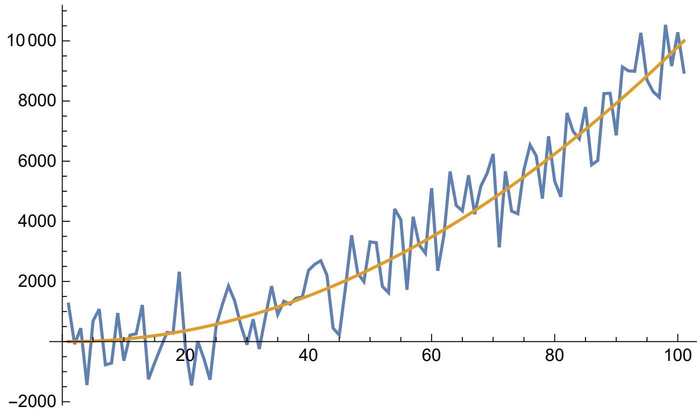
In[∘]:=
```
Clear[exp1b]
exp1b = {1244.87424194898`, -84.07549668914908`, 446.196305181199`,
   -1445.56143785746`, 691.7253234604159`, 1079.512849107322`, -772.0394824305295`,
   -715.4230778863748`, 949.5619808219471`, -634.4450461349288`, 214.0826529340302`,
   261.80841278797925`, 1216.932607167183`, -1254.3873542357223`, -705.6308022026569`,
   -196.1107028344004`, 303.84524463848356`, 283.490928440463`, 2321.607100671051`,
   -209.09335242606994`, -1457.9528099067948`, 15.698955434896561`,
   -563.595983877849`, -1269.3642875359901`, 527.1758285172655`, 1228.3307570996494`,
   1848.9229019542563`, 1355.2573579091522`, 544.9410927327226`, -102.72116444955009`,
   740.3186761160496`, -244.8525747161757`, 821.625866090827`, 1843.6534815803043`,
   899.9709595506529`, 1342.3664042454884`, 1244.7163921065767`, 1442.6187117379768`,
   1483.8808519152353`, 2367.4702382465903`, 2567.4162556504352`, 2689.0151764137117`,
   2204.661970739621`, 448.0435733482964`, 216.8237827710709`, 1782.4212642450093`,
   3533.7986397397763`, 2272.2193838321655`, 1993.4283117021396`,
   3315.912058991301`, 3288.779528755954`, 1836.246206515736`, 1627.777823042592`,
   4411.852085447112`, 4054.149358629752`, 1726.3822902979282`, 4151.099169114793`,
   3210.681375624922`, 2936.9770732570996`, 5102.441058241816`, 2355.0481193807445`,
   3527.790767320227`, 5655.596275031297`, 4529.178812425745`, 4346.91432349158`,
   5530.375435778425`, 4237.76432132417`, 5169.565996768326`, 5586.071883848494`,
   6238.510612683809`, 3134.5682081166206`, 5661.846749056337`, 4339.625473041178`,
   4251.212385776947`, 5714.778540958786`, 6538.698970475642`, 6173.923795745773`,
   4756.123194591116`, 6823.513808723065`, 5336.59445468462`, 4814.385741044567`,
   7601.176251062999`, 6987.024090564625`, 6744.799575502757`, 7799.232146822729`,
   5877.0264661557085`, 6024.6055488778175`, 8247.877769714596`, 8258.305544358993`,
   6859.517481518431`, 9132.841145443253`, 9001.729175113987`, 8988.087300578318`,
   10266.137031833247`, 8697.430703038444`, 8308.61924597661`, 8124.460280667454`,
   10533.323541366877`, 9164.49320485329`, 10283.748430572434`, 8962.765337901586`};
ListPlot[
 exp1b]
```

Out[∘]=



In[∘]:=
```
ListLinePlot[{exp1b, exp1model1}]
```

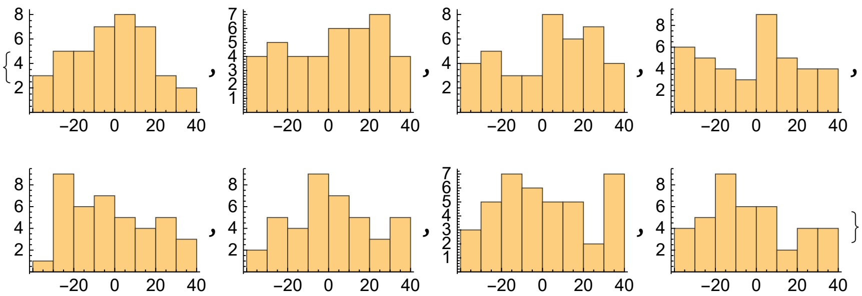
Out[∘]=



Integração: "(...) a smooth curve called a probability density function, usually abbreviated to pdf. A pdf must take only non-negative values, and the total area under the curve must be one. Then, the area under the curve between any two values, a and b, say, gives the probability that the outcome will take a value somewhere between a and b."[1]

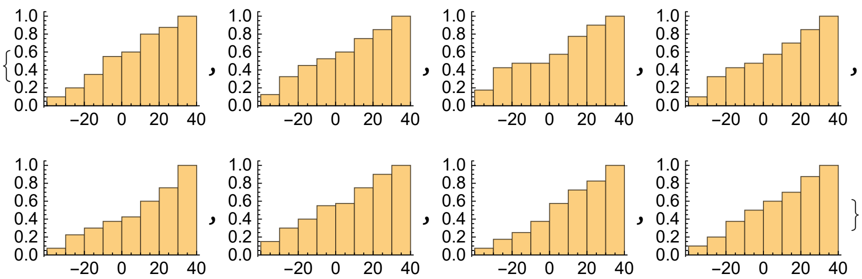
In[∘]:=
```
Table[Histogram[Table[RandomVariate[UniformDistribution[{-40, 40}]], 40],
  {-40, 40, 10}, ImageSize → Tiny], 8]
```

Out[∘]=



In[∘]:=
```
Table[Histogram[Table[RandomVariate[UniformDistribution[{-40, 40}]], 40],
  {-40, 40, 10}, "CDF", ImageSize → Tiny], 8]
```

Out[∘]=



Ainda não estou *especificando* as distribuições, apenas instanciando... as distribuições são (internamente) funções.

Distribuição normal:

$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e - \frac{(x-\mu)^2}{2\sigma^2}, \text{ onde}$$

μ = média

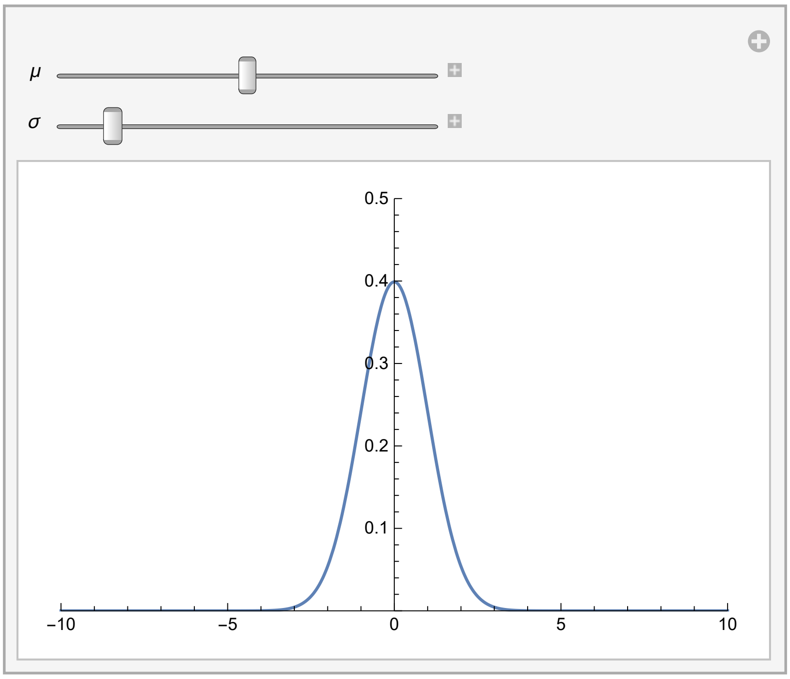
σ = variância

In[30]:=
```
Clear[ϕ0]
ϕ0=Function[{x,μ,σ}, 1/Sqrt[2π*σ^2] e^(-(x-μ)^2/(2σ^2))]
```

Out[31]= $\text{Function}\left[\{x, \mu, \sigma\}, \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\right]$

In[58]:=
```
Manipulate[Plot[ϕ0[x, μ, σ], {x, -10, 10}, PlotRange → {{-10, 10}, {0, 0.5}},
  ImageSize → Medium], {{μ, 0}, -10, 10, .01}, {{σ, 1}, 0.5, 5, .01}]
```

Out[58]=



"Every normal distribution is a version of the standard normal distribution whose domain has been stretched by a factor σ (the standard deviation) and then translated by μ (the mean value)."[2]

Distribuição normal: "(...) the number of cancer-related deaths in the UK next year; strictly this must be a finite number, but its upper bound is hard to determine exactly. For a variable like this, the usual strategy for describing its statistical properties is to specify a probability distribution that allows arbitrarily large outcomes, but with vanishingly small probabilities."[3] (O tail-end.)

Variações:

$\phi(x) = \frac{1}{\sqrt{2\pi}} e - \frac{1}{2} x^2$. Distribuição normal "padrão" com $\mu = 0, \sigma = 1$.

In[◦]:=

```
Clear[ϕ1, ϕ2, ϕ3]
```

$$\phi 1 = \text{Function}\left[x, \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\right]$$

$$\phi 2 = \text{Function}\left[x, \frac{e^{-x^2}}{\sqrt{\pi}}\right]$$

$$\phi 3 = \text{Function}\left[x, e^{-\pi * x^2}\right]$$

Out[◦]= $\text{Function}\left[x, \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}\right]$

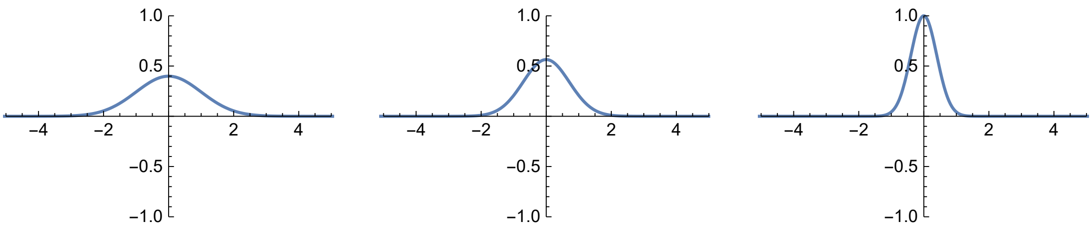
Out[◦]= $\text{Function}\left[x, \frac{e^{-x^2}}{\sqrt{\pi}}\right]$

Out[◦]= $\text{Function}\left[x, e^{-\pi x^2}\right]$

In[◦]:=

```
GraphicsRow[{
  Plot[ϕ1[x], {x, -10, 10}, PlotRange → {{-5, 5}, {-1, 1}}, ImageSize → Small],
  Plot[ϕ2[x], {x, -10, 10}, PlotRange → {{-5, 5}, {-1, 1}}, ImageSize → Small],
  Plot[ϕ3[x], {x, -10, 10}, PlotRange → {{-5, 5}, {-1, 1}}, ImageSize → Small]
 }]
```

Out[◦]=



In[32]:=

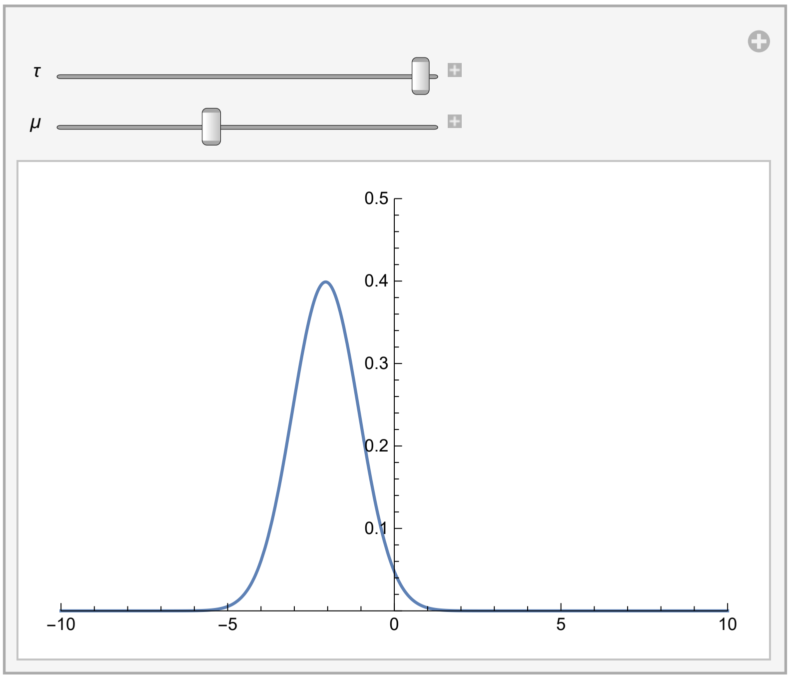
```
Clear[ϕ4,ϕ5]
```

$$\phi 4 = \text{Function}\left[\{x,\tau,\mu\}, \sqrt{\frac{\tau}{2\pi}}\, e^{\frac{-\tau (x-\mu)^2}{2}}\right]$$

Out[33]= $\text{Function}\left[\{x, \tau, \mu\}, \sqrt{\frac{\tau}{2\pi}}\, e^{\frac{1}{2}(-\tau)(x-\mu)^2}\right]$

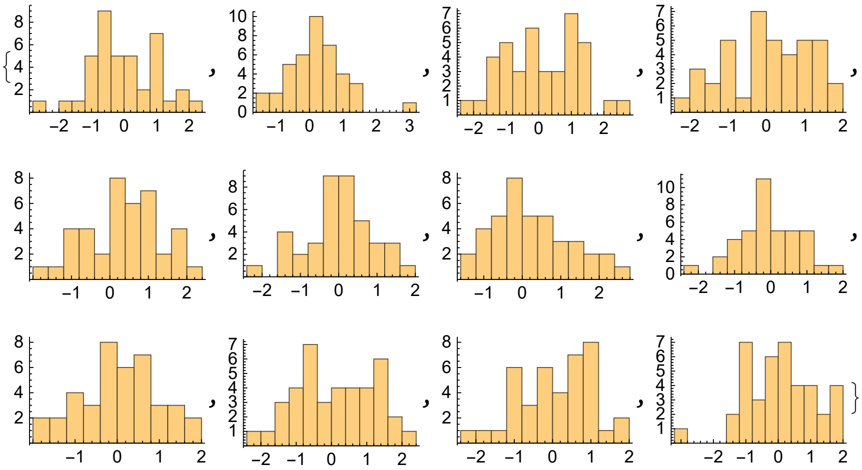
In[∘]:=
```
Manipulate[Plot[ϕ4[x, τ, μ], {x, -10, 10}, PlotRange → {{-10, 10}, {0, .5}},
  ImageSize → Medium], {{τ, 1}, 0.01, 1, .01}, {{μ, 0}, -10, 10, .01}]
```

Out[∘]=



In[∘]:=
```
Table[
 Histogram[Table[RandomVariate[NormalDistribution[]], 40], {.4}, ImageSize → Tiny], 12]
```

Out[∘]=



“A discrete probability distribution can be encoded by a discrete list of the probabilities of the outcomes, known as a **probability mass function**. A continuous probability distribution is typically described by **probability density functions** (with the probability of any individual outcome actually being 0). The normal distribution is a commonly encountered continuous probability distribution.”[5]

Integração: “The values of the **pdf** (as opposed to those of the **pmf**) are not probabilities as such: a pdf must be integrated over an interval to yield a probability.”[6]
“(...) a smooth curve called a probability density function, usually abbreviated to pdf. A pdf must take only non-negative values, and the total area under the curve must be one. Then, the area under the curve between any two values, a and b, say, gives the probability that the outcome will take a value somewhere between a and b.”[7]

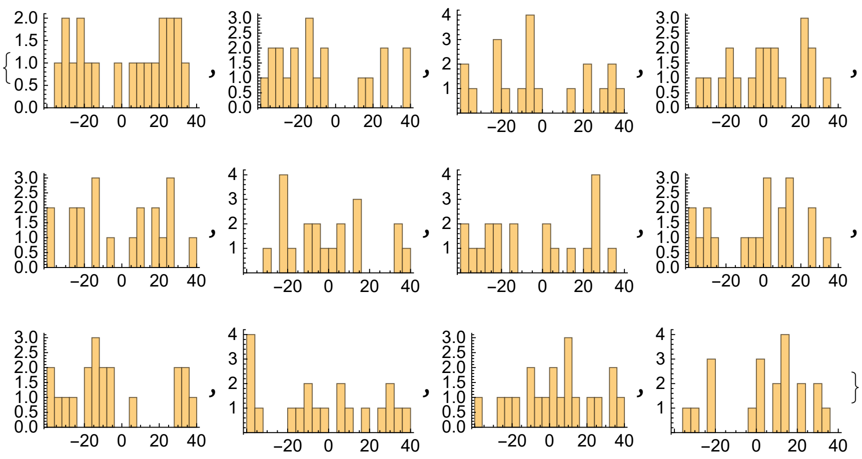
A variável certa é: $\phi(x) = 1$ com domínio em $-40 < x < 40$.

Modelo 1

```
In[ ]:= Clear[exp1ϕ1]
        exp1ϕ1 = ProbabilityDistribution[Function[x, 1][x], {x, -40, 40}]
Out[ ]= ProbabilityDistribution[1, {x, -40, 40}]
```

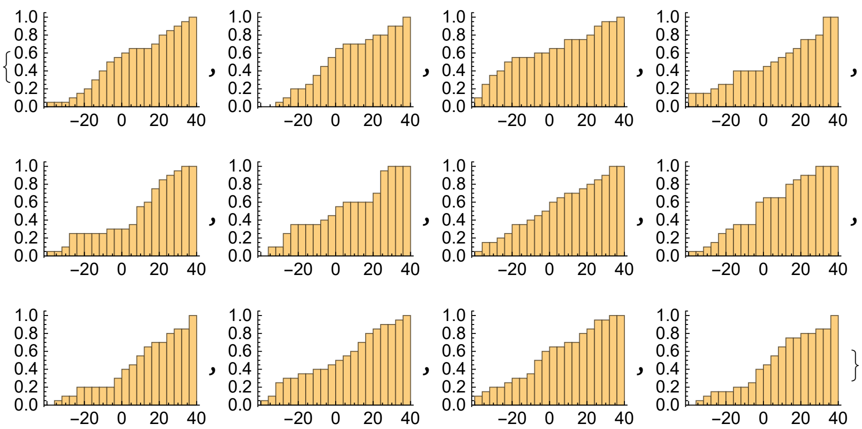
```
In[ ]:= Table[RandomVariate[exp1ϕ1], 20]
Out[ ]= {30.0984, 9.14555, -14.1705, -22.8391, 32.9483, -2.42389,
         -36.5842, -17.203, 35.6297, -19.3011, 20.5506, -27.0192, 12.1335,
         -2.7879, -5.90197, -14.5735, 32.3403, -24.8704, -33.4424, 29.9531}
```

```
In[ ]:= Table[Histogram[Table[RandomVariate[exp1ϕ1], 20], {-40, 40, 4}, ImageSize → Tiny], 12]
Out[ ]=
```



Os números ficam relativamente uniformes. É que os máximos estão em ≈ 4.

```
In[ ]:= Table[Histogram[Table[RandomVariate[exp1ϕ1], 20],
          {-40, 40, 4}, "CDF", ImageSize → Tiny], 12]
Out[ ]=
```



Agora criar outras erradas.

In[34]:=
```
Clear[pdhistos]
pdhistos=Function[{pd,binfrom,binto,binsize,qty},
    Table[Histogram[RandomVariate[pd,binto-binfrom],{binfrom,binto,binsize},
        ImageSize->Tiny],qty]];
```

Modelo 2

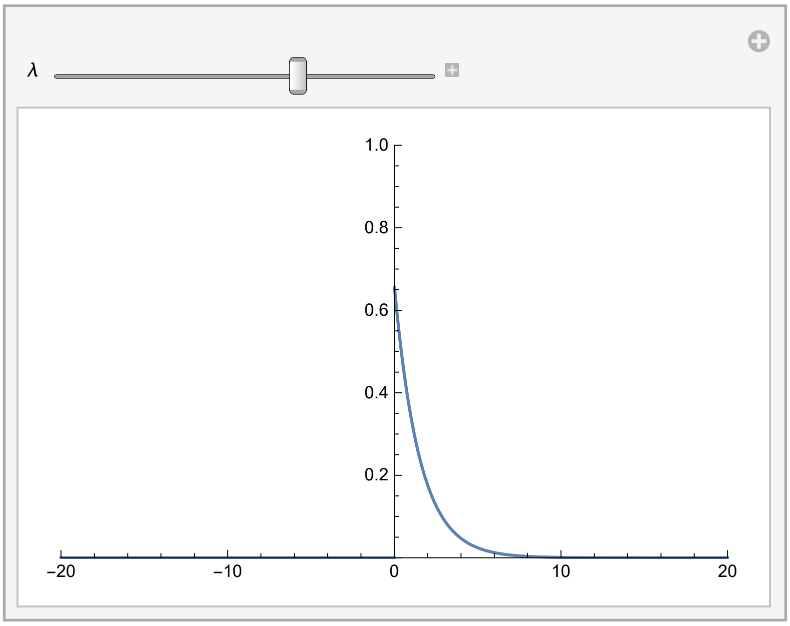
In[36]:=
```
Clear[λ0]
λ0=Function[{x,λ},Piecewise[{{λ*e^(-λ*x),x≥0},{0,x<0}}]]
λ0[10,10]
```

Out[37]= Function[{x, λ}, $\begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$]

Out[38]= $\frac{10}{e^{100}}$

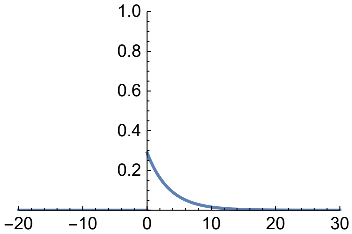
In[ ]:=
```
Manipulate[Plot[λ0[x, λ], {x, -20, 20},
  PlotRange → {{-20, 20}, {0, 1}}, ImageSize → Medium], {{λ, 0.29}, 0.01, 1, .01}]
```

Out[ ]=



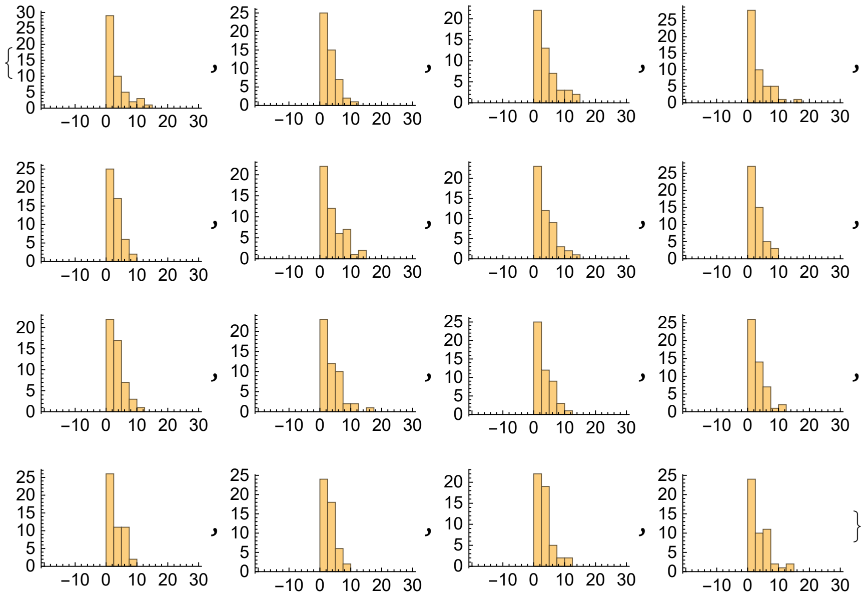
In[∘]:=
```
Clear[f, exp1λ0]
f = λ0[x, 0.29];
Plot[f, {x, -20, 30}, PlotRange → {{-20, 30}, {0, 1}}, ImageSize → Small]
exp1λ0 = ProbabilityDistribution[f, {x, -20, 20}];
RandomVariate[exp1λ0, 25] * 100
pdhistos[exp1λ0, -20, 30, 2.5, 16]
```

Out[∘]=

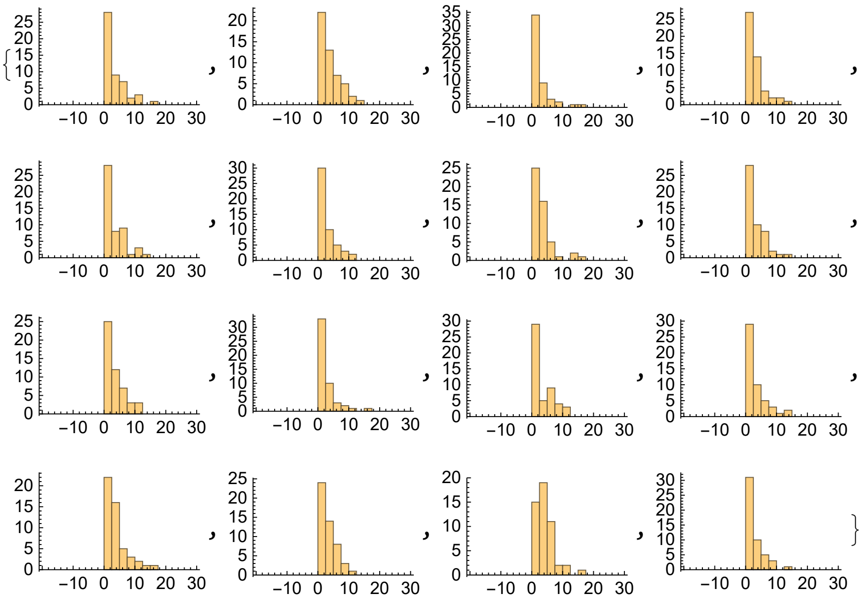


Out[∘]=
```
{219.659, 1206.71, 729.922, 369.698, 601.465, 547.536, 19.553, 305.196,
 684.44, 215.503, 163.274, 1269.12, 411.061, 576.239, 85.722, 820.902, 429.772,
 29.2015, 339.954, 83.2262, 361.013, 149.535, 172.121, 440.485, 364.107}
```

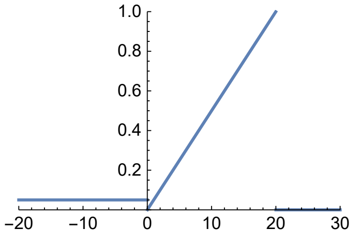
Out[∘]=



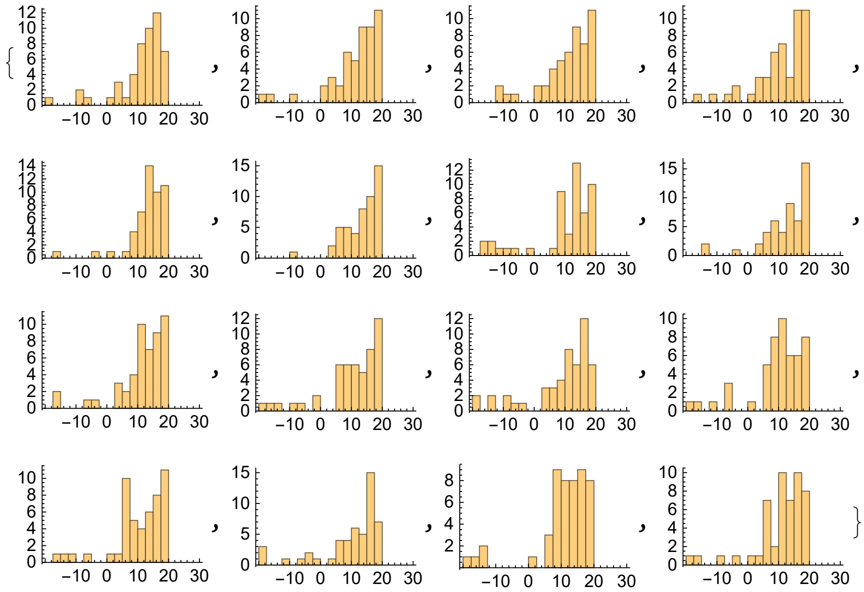
Out[ ]=
```
{53.93704642007098`, 97.01729297965684`, 97.13977946406729`,
 314.0676767080434`, 67.6758201021542`, 188.10817539770585`, 501.34818183458094`,
 78.09059895020968`, 108.19778444830526`, 331.2147942366284`, 108.82309314954296`,
 142.18042480265785`, 262.01073835320017`, 830.6028310787516`,
 113.56663174391315`, 256.4344040636772`, 50.52626180857285`, 232.26042748283095`,
 226.96681454537355`, 198.44608507807263`, 448.47767989590545`,
 735.5179208084074`, 64.30122525802435`, 146.46880668600488`, 280.55245258111336`}
```

Out[ ]=


```
In[◦]:= Clear[f, exp1ϕ2]
f = Piecewise[{{0.05, x ≤ 0}, {x / 20, x > 0 && x ≤ 20}, {0, x > 20}}];
Plot[f, {x, -20, 30}, PlotRange → {{-20, 30}, {0, 1}}, ImageSize → Small]
exp1ϕ2 = ProbabilityDistribution[f * maxrnd, {x, -20, 20}];
RandomVariate[exp1ϕ2, 25] * 100
pdhistos[exp1ϕ2, -20, 30, 2.5, 16]
```

Out[◦]=



```
Out[◦]= {1185.05, 1155.79, 216.681, 405.419, 1122.26, 1462.41, 1937., 1252.01,
 -175.097, 1637.82, -1119.62, 1771.6, 717.675, -1676.25, 1171.06, 1141.8,
 902.915, 1232.29, 1048.82, 1979.7, 1272.4, 1293.44, 1143.43, 1210.04, 1437.26}
```

Out[◦]=


A distribuição é exatamente a mesma coisa que a função definidora... (em termos de valores retornados). Portanto o intervalo da distribuição e domínio da função são teoricamente coincidentes (para representar a variação na variável randômica).

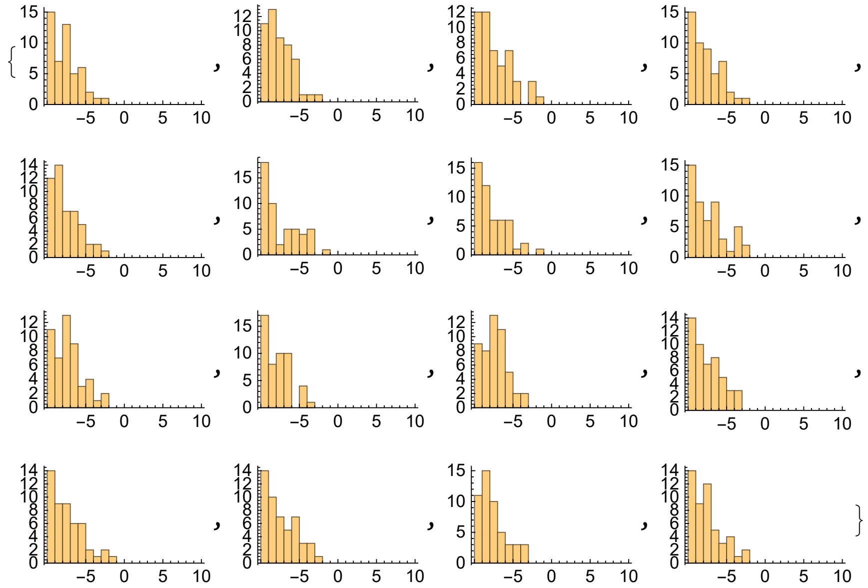
Neste primeiro modelo, estamos intencionalmente usando um domínio de intervalo menor que o da variável correta, e apenas no sentido positivo.

Modelo 3

Uma exponencial invertida também com o domínio menor.

```
(*pdhistos[exp1ϕ3,0,100,1,16]*) (*Quebra*)
```

Out[ ]=



Integrar

Integrar ambas como variável randômica de **points1** e comparar as variações nas distâncias.

In[40]:=

```
(*TODO: Dicretize retorna uma lista com length steps + 1. Seria legal corrigir para retornar lista com length steps e retirar os Length[...]-1 das chamadas de Discretize que passam o tamanho de uma lista gerada por Discretize como parâmetro steps.*)
Clear[Discretize]
Discretize=Function[{f,steps,x1},Table[f[x],{x,0,x1,Floor[x1/steps]}]];
Table[x^3,{x,0,100,10}]
Discretize[Function[x,x^3],10,100]
```

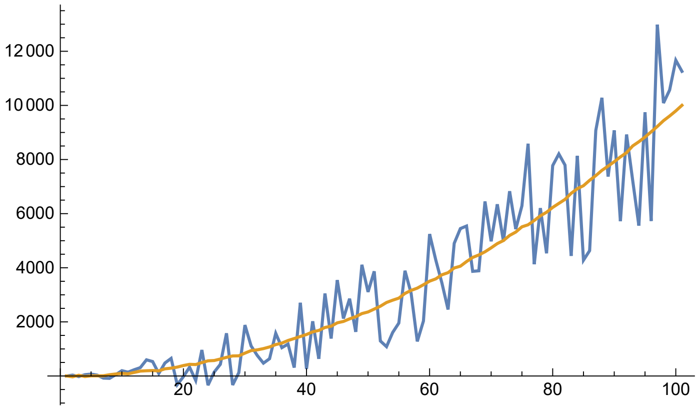
Out[42]= {0, 1000, 8000, 27 000, 64 000, 125 000, 216 000, 343 000, 512 000, 729 000, 1 000 000}

Out[43]= {0, 1000, 8000, 27 000, 64 000, 125 000, 216 000, 343 000, 512 000, 729 000, 1 000 000}

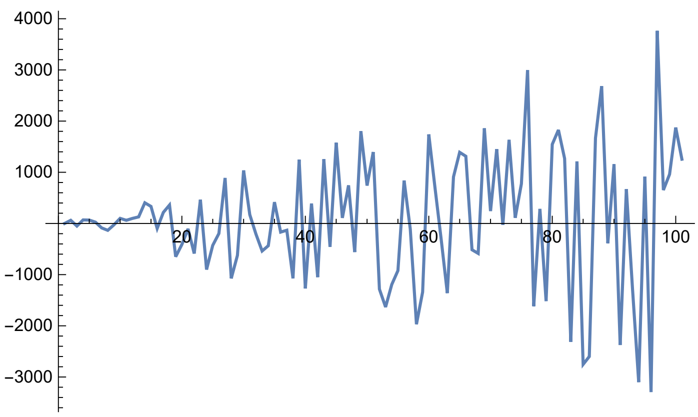
Modelo 1

In[ ]:=
```
Clear[fexp1model1, exp1model1, exp1model1diff]
(*model1=Function[x,x^2+RandomReal[{-40*x,40*x}]]*)
fexp1model1 = Function[x, x^2 + RandomVariate[exp1ϕ1]];
exp1model1 = Discretize[fexp1model1, 100, 100];
exp1model1diff = exp1 - exp1model1;
ListLinePlot[{exp1, exp1model1}, ImageSize → Medium]
ListLinePlot[exp1model1diff, ImageSize → Medium]
```

Out[ ]=



Out[ ]=



Modelo 2

Esse modelo engloba dois modelos, um exponencial e um linear.

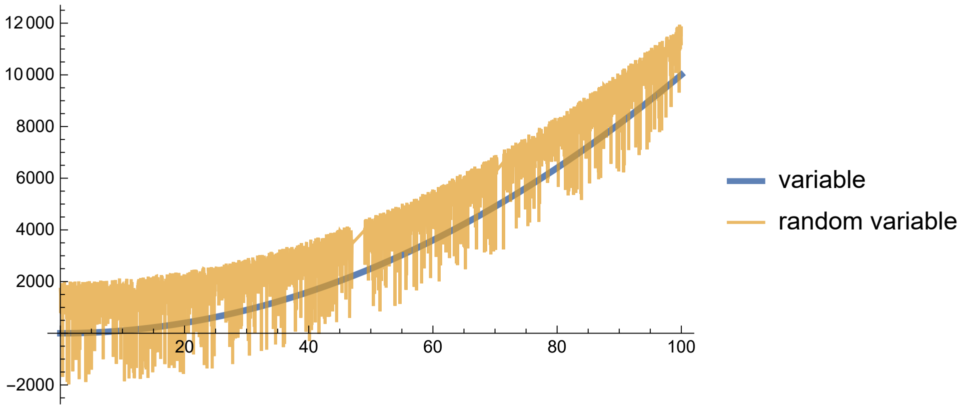
In[44]:=
```
m=Directive[Opacity[1],Thickness[.01]];
s=Directive[{Opacity[.7]}];
```

In[ ]:=
```
exp1ϕ2
```

Out[ ]= ProbabilityDistribution$\left[40\left(\begin{cases} 0.05 & x \le 0 \\ \frac{x}{20} & x > 0 \,\&\&\, x \le 20 \\ 0 & \text{True} \end{cases}\right), \{x, -20, 20\}\right]$

In[∘]:=
```
Plot[{x^2, x^2 + RandomVariate[exp1ϕ2] * 100}, {x, 0, 100},
 PlotStyle → {m, s}, PlotLegends → {"variable", "random variable"}]
```

Out[∘]=

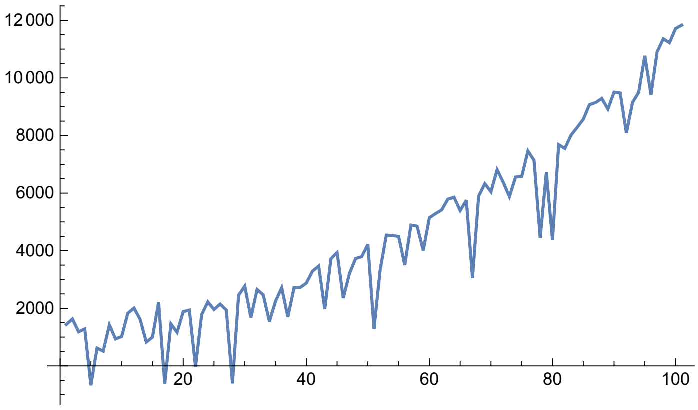


In[∘]:=
```
Discretize[Function[x, x^2 + RandomVariate[exp1ϕ2] * 100], 100, 100]
```

Out[∘]=
```
{1754.06, 1022.67, 1584.27, 676.094, -1466.59, 1176.81, 1720.88, 1649.46, 1799.01,
 984.223, 1077.02, -1047.44, 933.075, 1434.13, 1971.79, 835.687, 2147.7, 1735.01,
 1602.12, 1477.21, 2224.7, 1979.45, 2120.87, 1766.67, 2108.78, 1448.37, 1444.58,
 231.966, 2740.82, 1181.61, 2656., 187.353, 1506.13, 2236.44, 2587.7, 196.05, 3136.,
 5.50086, 3095., 3179.63, 3061.8, 2896.24, 2653.49, 2959.38, 3362.6, 2739.7, 3822.3,
 2608.47, 2981.86, 1447.52, 3425.92, 3375.82, 3412.67, 3846.15, 4714.2, 3641.09,
 4314.73, 2104.69, 4753.65, 4714.99, 4812.36, 4898.12, 4512.79, 5622.29, 5944.37,
 5226.15, 4556.51, 5980.46, 5878.43, 5805.7, 5767.19, 6978.16, 7046.64, 6437.92,
 6440.07, 7537.1, 6903.35, 7266.31, 6750.02, 7334.13, 6950.29, 7033.34, 7691.1,
 6653.47, 8456.79, 8187.95, 8247.07, 9502.47, 8805.96, 8973.31, 9649.97, 9426.6,
 9398.64, 10403.4, 10432.4, 10709., 9820.06, 10588., 10594.4, 11703.5, 11170.4}
```

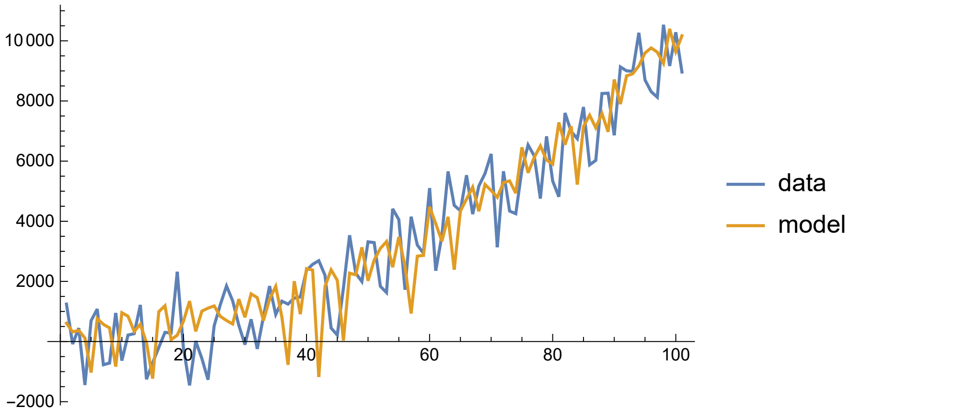
In[∘]:=
```
ListLinePlot[Discretize[Function[x, x^2 + RandomVariate[exp1ϕ2] * 100], 100, 100]]
```

Out[∘]=

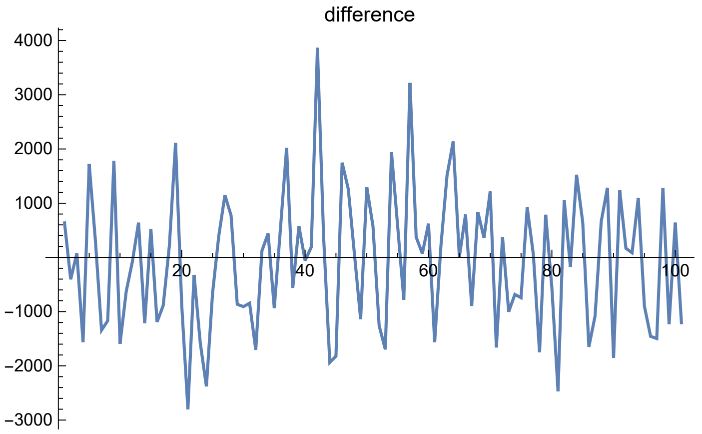


In[∘]:=
```
Clear[exp1model2ϕ, exp1model2ϕdiff]
exp1model2ϕ = Discretize[Function[x, x^2 + (RandomVariate[exp1ϕ2] - 10) * 100], 100, 100];
exp1model2ϕdiff = exp1b - exp1model2ϕ;
ListLinePlot[{exp1b, exp1model2ϕ},
 ImageSize → Medium, PlotLegends → {"data", "model"}]
ListLinePlot[exp1model2ϕdiff, ImageSize → Medium, PlotLabel → "difference"]
```

Out[∘]=



Out[∘]=

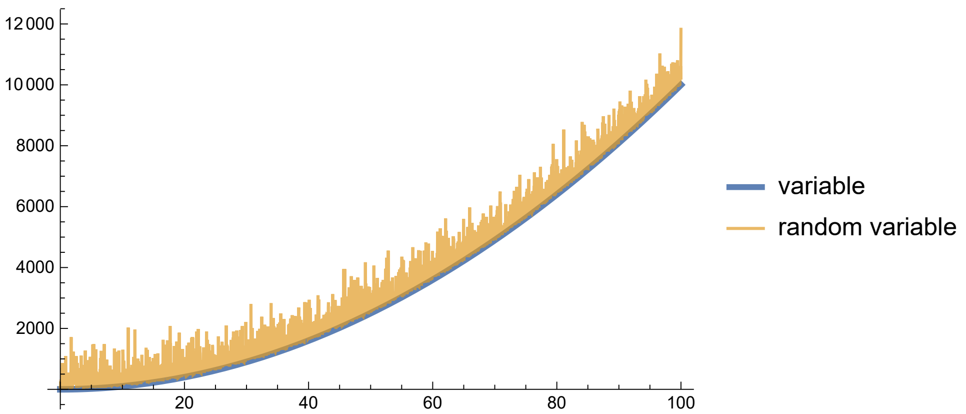


In[∘]:=
```
Mean[exp1model2ϕdiff]
```

Out[∘]= -248.851

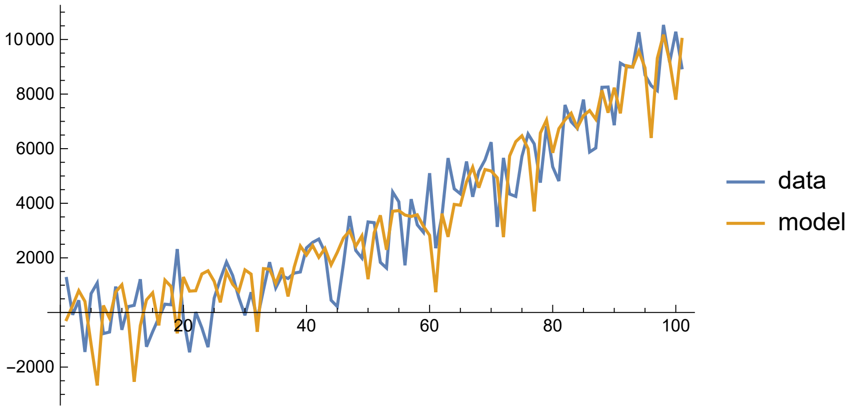
In[∘]:=
```
Plot[{x^2, x^2 + RandomVariate[exp1λ0] * 100}, {x, 0, 100},
 PlotStyle → {m, s}, PlotLegends → {"variable", "random variable"}]
```

Out[∘]=

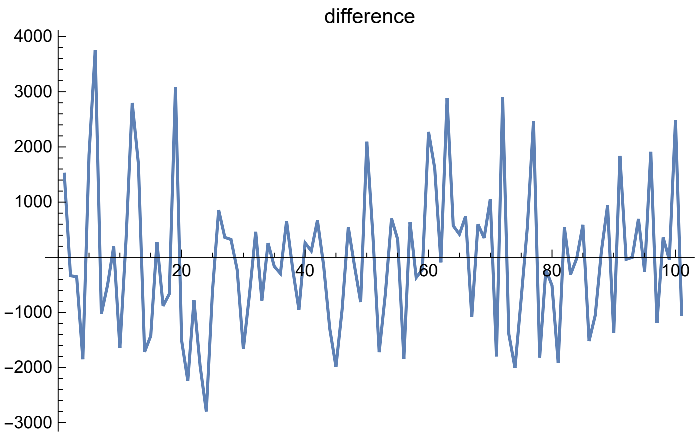


In[ ]:=
```
Clear[exp1model2λ, exp1model2λdiff]
exp1model2λ = Discretize[Function[x, x^2 + (RandomVariate[exp1ϕ2] - 10) * 100], 100, 100];
exp1model2λdiff = exp1b - exp1model2λ;
ListLinePlot[{exp1b, exp1model2λ},
 ImageSize → Medium, PlotLegends → {"data", "model"}]
ListLinePlot[exp1model2λdiff, ImageSize → Medium, PlotLabel → "difference"]
```

Out[ ]=



Out[ ]=



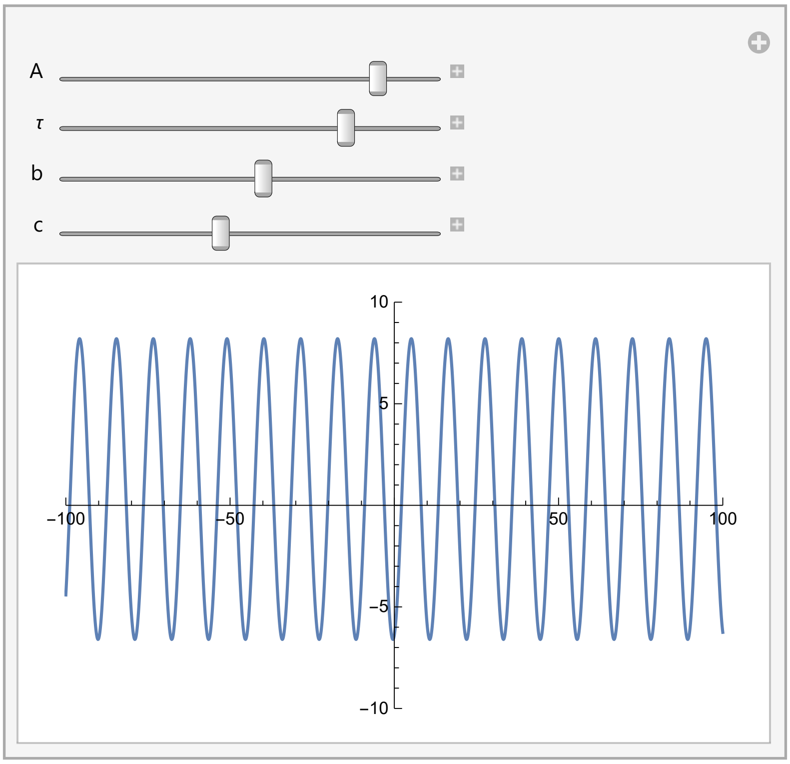
In[ ]:=
```
Mean[exp1model2λdiff]
```

Out[ ]= 51.9879

Este segundo é mais aproximado.

In[ ]:=
```
Manipulate[Plot[A * Cos[τ * (x + c)] + b, {x, -100, 100},
  PlotRange → {{-100, 100}, {-10, 10}}], {{A, 5.3}, -10, 10, .1},
 {{τ, 0.43}, -1, 1, .01}, {{b, -1}, -10, 10, .1}, {{c, 0.5}, -100, 100, .1}]
```

Out[ ]=



Multiplicadores (coeficientes):

A: amplitude

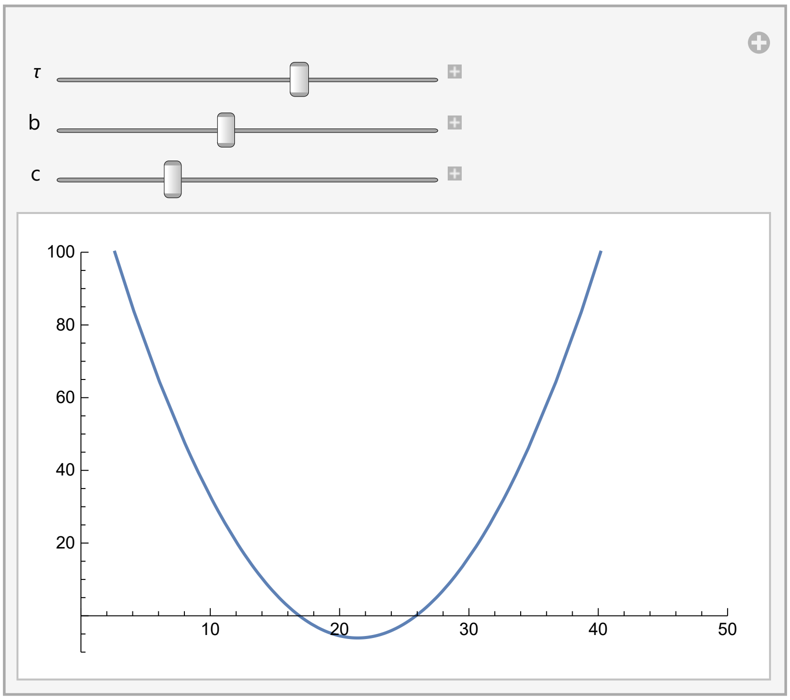
τ: período

Deslocamentos:

b: vertical

c: horizontal (fase)

In[ ]:=
```
Manipulate[Plot[(τ * (x + c)^2) + b, {x, 0, 100}, PlotRange → {{0, 50}, {-10, 100}}],
 {{τ, 0.05}, -1, 1, .01}, {{b, 0}, -50, 50, .1}, {{c, 0}, -50, 50, .1}]
```

Out[ ]=


Multiplicadores (coeficientes):

τ: abertura

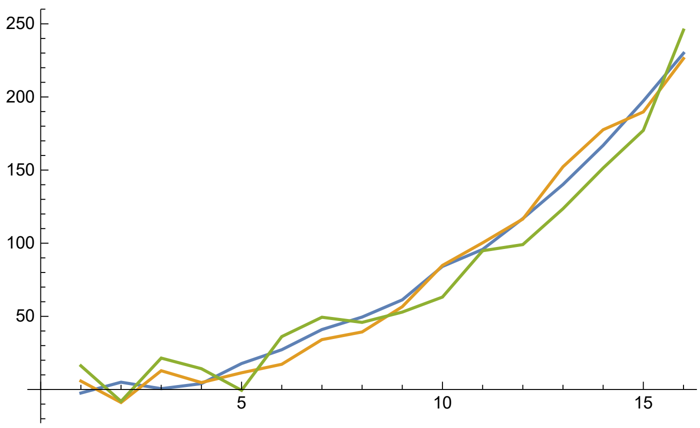
Deslocamentos:

b: vertical

c: horizontal

Estes parâmetros terão que ter seus limites especificados para não produzir funções “desproporcionais” à série.

In[46]:=
```
Clear[MakePoints1]
MakePoints1=Function[var,Table[x^2+RandomReal[{-var,var}],{x,0,15,1}]];
ListLinePlot[{MakePoints1[5],MakePoints1[10],MakePoints1[25]}]
```

Out[48]=


In[49]:=
```
Clear[points1c]
points1c=MakePoints1[15]
```

Out[50]= {-10.3885, 4.03241, 2.24165, 22.4584, 3.70584, 26.5119, 24.2396, 59.3081, 66.0442, 87.5995, 107.741, 118.057, 140.017, 156.015, 205.251, 224.677}

In[ ]:=
```
Table[Function[x, x^2][x], {x, 0, 15, 1}]
```

Out[ ]= {0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225}

In[ ]:=
```
Discretize[Function[x, x^2], 15, 15]
```

Out[ ]= {0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225}

In[ ]:=
```
Length[points1c]
```

Out[ ]= 16

In[ ]:=
```
N[Discretize[Function[x, x^2], Length[points1c] - 1, Length[points1c] - 1]]
```

Out[ ]= {0., 1., 4., 9., 16., 25., 36., 49., 64., 81., 100., 121., 144., 169., 196., 225.}

In[ ]:=
```
Total[points1c]
```

Out[ ]= 1304.52

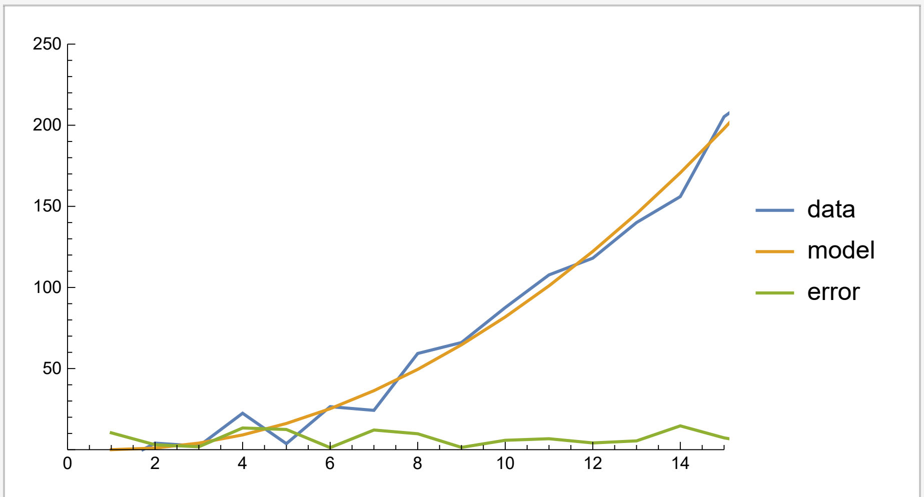
In[ ]:=
```
Manipulate[
 GetDiff = Function[
   Total[dta - mdl]
  ];
 GetSqDiff = Function[
   Total[(dta - mdl)^2]
  ];
 dta = points1c;
 mdl = Discretize[Function[x, τ * x^2], Length[dta] - 1, Length[dta] - 1];
 ListLinePlot[{dta, mdl, Abs[dta - mdl]}, PlotRange → {{0, Length[dta] - 1}, {0, 250}},
  PlotLegends → {"data", "model", "error"}],
 {{τ, 1.01}, .01, 3, .01},
 Dynamic[
  diff = GetDiff[];
  sqDiff = GetSqDiff[];
  "τ: " <> ToString[τ] <>
   (*"\nΣdata-Σmodel: "<>ToString[diff]*)
   "\nΣ(data-model)^2: " <> ToString[sqDiff]
 ]
]
```

Out[ ]=

τ

τ: 1.01

Σ(data–model)²: 54.5824

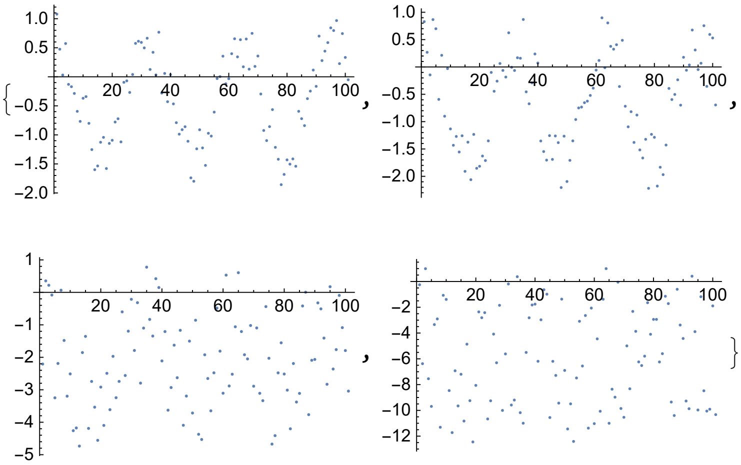


In[•]:=
```
Manipulate[
 GetDiff = Function[
   Total[dta - mdl]
  ];
 GetSqDiff = Function[
   Total[(dta - mdl)^2]
  ];
 dta = points2;
 mdl =
  Discretize[Function[x, N[A * Cos[τ * (x + ϕ)] + b]], Length[dta] - 1, Length[dta] - 1];
 ListLinePlot[{dta, mdl, Abs[dta - mdl]},
  PlotRange → {{0, Length[dta] - 1}, {-1.2, 1.4}},
  PlotLegends → {"data", "model", "error"},
  ImageSize → Medium],
 {{A, .96}, .75, 1.5, .01},
 {{τ, .2}, .025, .5, .001},
 {{ϕ, 0}, -10, 10, .1},
 {{b, 0}, -.5, .5, .1},
 Dynamic[
  diff = GetDiff[];
  sqDiff = GetSqDiff[];
  "A: " <> ToString[A] <>
   "\nτ: " <> ToString[τ] <>
   "\nc: " <> ToString[ϕ] <>
   "\nb: " <> ToString[b] <>
   (*"\nΣdata-Σmodel: "<>ToString[diff]*)
   "\nΣ(data-model)^2: " <> ToString[sqDiff]
 ]
]
```

In[•]:=
```
Clear[points2b]
points2b = Table[MakePoints2[mr], {mr, {0.92, 1.285, 4, 12}}];
Table[ListPlot[points2b[[i]]], {i, 1, 4, 1}];
```

In[51]:=
```
Clear[points2b1,points2b2,points2b3,points2b4]
points2b1={1.0783077614849716`,0.4748249277119898`,0.030138017340957335`,0.5728095061482488`,-
points2b2={0.8284486071053614`,0.2673766376785629`,-0.14587103848031902`,0.8659704680415028`,0
points2b3={-2.2078705031789028`,0.3532694263384413`,0.21874066160618977`,-0.0825627277014358`,
points2b4={-0.2531977741929943`,-6.377176597278109`,0.9939290360511164`,-7.5449543069063605`,-
Table[ListPlot[pts],{pts,{points2b1,points2b2,points2b3,points2b4}}]
{Length[points2b1],Length[points2b2],Length[points2b3],Length[points2b4]}
```

Out[56]=



Out[57]= {101, 101, 101, 101}

In[ ]:=
```
Manipulate[
 GetDiff = Function[
   Total[dta - mdl]
  ];
 GetSqDiff = Function[
   Total[(dta - mdl)^2]
  ];
 dta = points2b1;
 mdl =
  Discretize[Function[x, N[A * Cos[τ * (x + ϕ)] + b]], Length[dta] - 1, Length[dta] - 1];
 ListLinePlot[{dta, mdl, Abs[dta - mdl]},
  PlotRange → {{0, Length[dta] - 1}, {-2, 2}},
  PlotLegends → {"data", "model", "error"},
  ImageSize → Medium],
 {{A, 1.28}, .75, 1.5, .01},
 {{τ, .186}, .025, .5, .001},
 {{ϕ, 3.5}, -10, 10, .1},
 {{b, -0.5}, -1.5, .5, .1},
 Dynamic[
  diff = GetDiff[];
  sqDiff = GetSqDiff[];
  "A: " <> ToString[A] <>
   "\nτ: " <> ToString[τ] <>
   "\nc: " <> ToString[ϕ] <>
   "\nb: " <> ToString[b] <>
   (*"\nΣdata-Σmodel: "<>ToString[diff]<>*)
   "\nΣ(data-model)²: " <> ToString[sqDiff]
 ]
]
```

A

τ

ϕ

b

A: 1.28

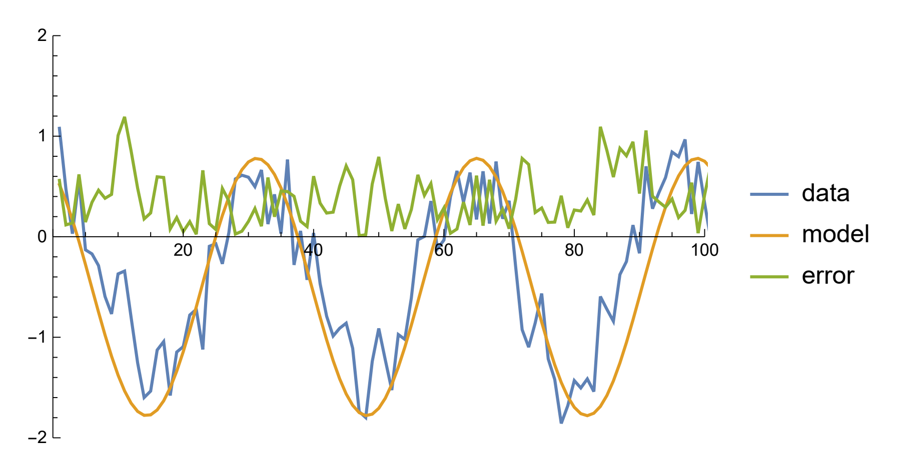
τ: 0.186

c: 3.5

b: −0.5

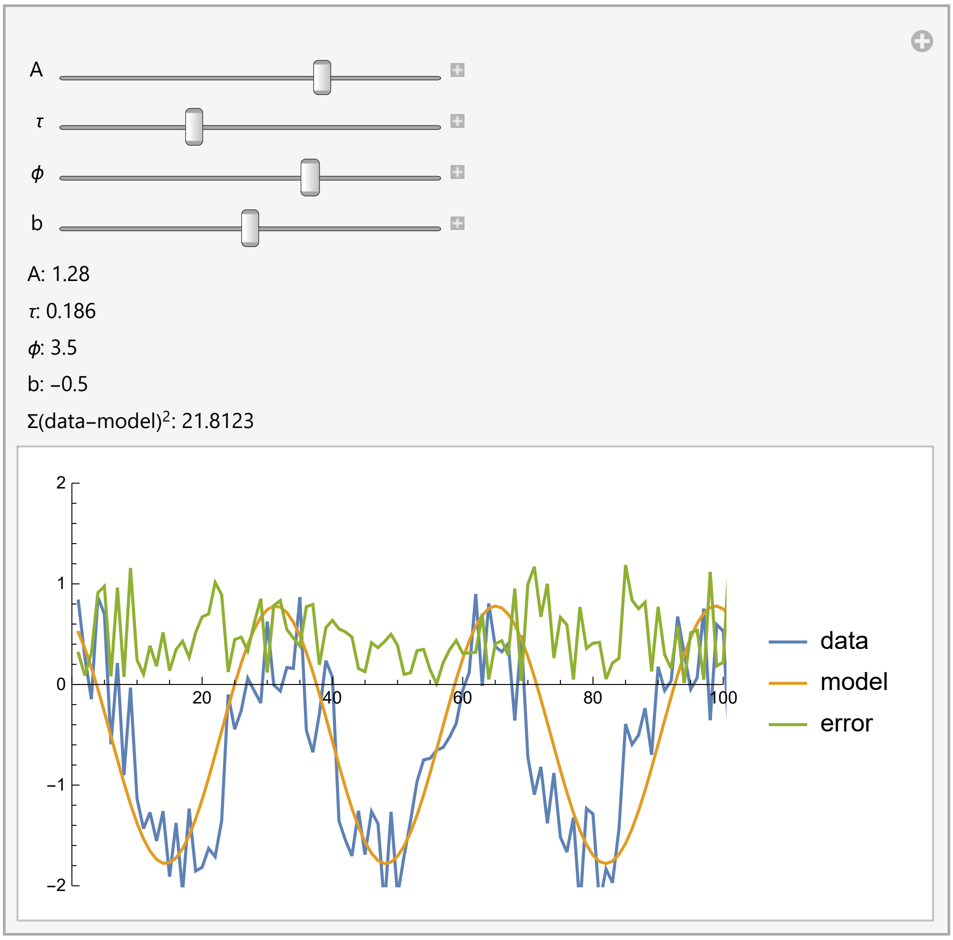
Σ(data−model)^2: 1101.73

Out[ ]=



In[•]:=
```
Manipulate[
  GetDiff = Function[
    Total[dta - mdl]
   ];
  GetSqDiff = Function[
    Total[(dta - mdl)^2]
   ];
  dta = points2b2;
  mdl =
   Discretize[Function[x, N[A * Cos[τ * (x + ϕ)] + b]], Length[dta] - 1, Length[dta] - 1];
  ListLinePlot[{dta, mdl, Abs[dta - mdl]},
   PlotRange → {{0, Length[dta] - 1}, {-2, 2}},
   PlotLegends → {"data", "model", "error"},
   ImageSize → Medium],
  {{A, 1.28}, .75, 1.5, .01},
  {{τ, .186}, .025, .5, .001},
  {{ϕ, 3.5}, -10, 10, .1},
  {{b, -0.5}, -1.5, .5, .1},
  Dynamic[
   diff = GetDiff[];
   sqDiff = GetSqDiff[];
   "A: " <> ToString[A] <>
    "\nτ: " <> ToString[τ] <>
    "\nϕ: " <> ToString[ϕ] <>
    "\nb: " <> ToString[b] <>
    (*"\nΣ(data-model): "<>ToString[diff]<>*)
    "\nΣ(data-model)^2: " <> ToString[sqDiff]
  ]
 ]
```

Out[ ]=



A
τ
ϕ
b
A: 1.28
τ: 0.186
ϕ: 3.5
b: −0.5
Σ(data−model)²: 21.8123
2
1
0
−1
−2
20
40
60
80
100
data
model
error

In[•]:=
```
Manipulate[
  GetDiff = Function[
    Total[dta - mdl]
   ];
  GetSqDiff = Function[
    Total[(dta - mdl)^2]
   ];
  dta = points2b3;
  mdl =
   Discretize[Function[x, N[A * Cos[τ * (x + ϕ)] + b]], Length[dta] - 1, Length[dta] - 1];
  ListLinePlot[{dta, mdl, Abs[dta - mdl]},
   PlotRange → {{0, Length[dta] - 1}, {-5.5, 4}},
   PlotLegends → {"data", "model", "error"},
   ImageSize → Medium],
  {{A, .94}, 0, 3.5, .01},
  {{τ, .186}, .025, .5, .001},
  {{ϕ, 2.2}, -20, 20, .1},
  {{b, -2.1}, -5.2, 1, .1},
  Dynamic[
   diff = GetDiff[];
   sqDiff = GetSqDiff[];
   "A: " <> ToString[A] <>
    "\nτ: " <> ToString[τ] <>
    "\nϕ: " <> ToString[ϕ] <>
    "\nb: " <> ToString[b] <>
    (*"\nΣ(data-model): "<>ToString[diff]<>*)
    "\nΣ(data-model)^2: " <> ToString[sqDiff]
  ]
 ]
```

A

τ

ϕ

b

A: 0.94

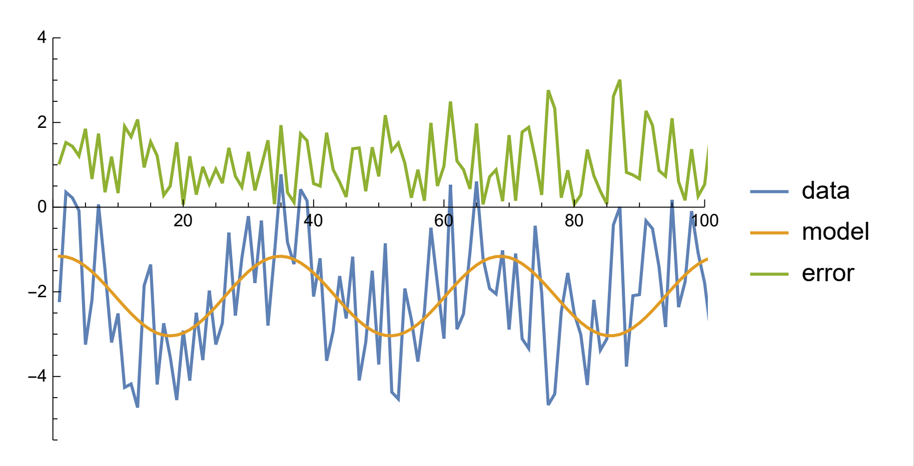
τ: 0.186

ϕ: -0.1

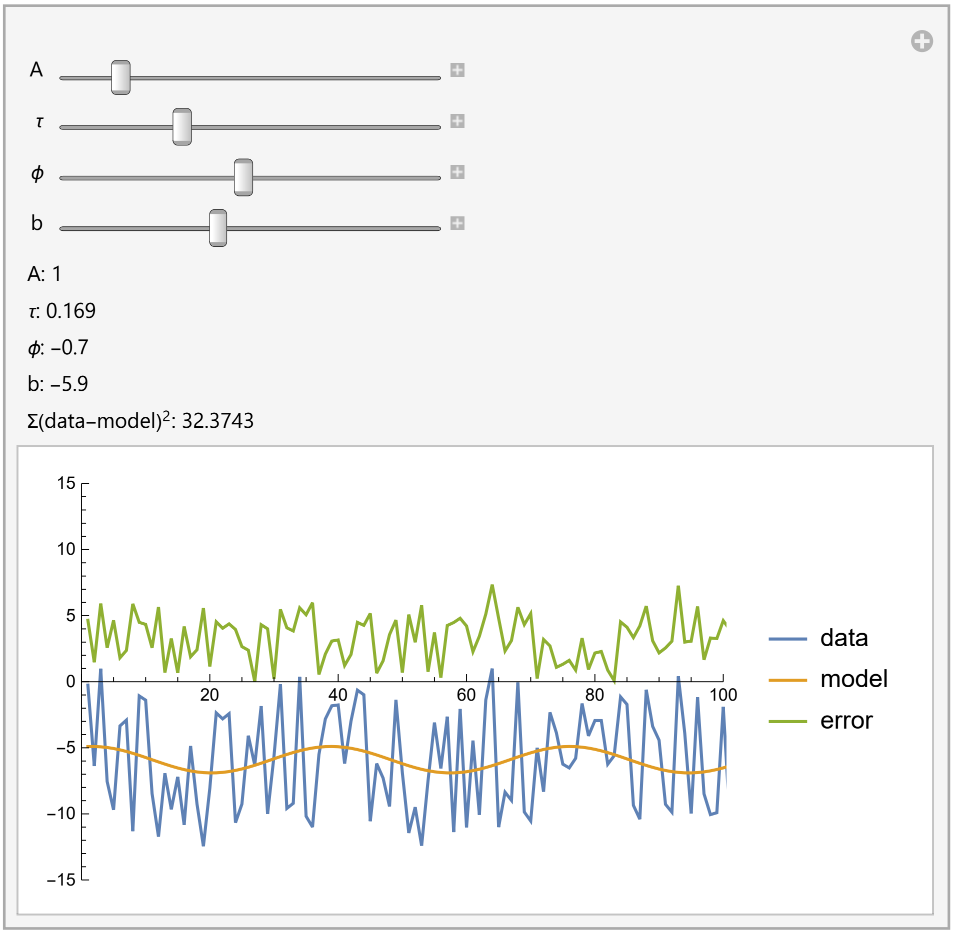
b: -2.1

Σ(data-model)^2: 32.3743

Out[ ]=



```
In[•]:= Manipulate[
  GetDiff = Function[
    Total[dta - mdl]
   ];
  GetSqDiff = Function[
    Total[(dta - mdl)^2]
   ];
  dta = points2b4;
  mdl =
   Discretize[Function[x, N[A * Cos[τ * (x + ϕ)] + b]], Length[dta] - 1, Length[dta] - 1];
  ListLinePlot[{dta, mdl, Abs[dta - mdl]},
   PlotRange → {{0, Length[dta] - 1}, {-15, 15}},
   PlotLegends → {"data", "model", "error"},
   ImageSize → Medium],
  {{A, 1}, 0, 8, .01},
  {{τ, .169}, .025, .5, .001},
  {{ϕ, -0.7}, -20, 20, .1},
  {{b, -5.9}, -12, 3, .1},
  Dynamic[
   diff = GetDiff;
   tss = GetTSS;
   mss = GetMSS;
   rss = GetRSS;
   "A: " <> ToString[A] <>
    "\nτ: " <> ToString[τ] <>
    "\nϕ: " <> ToString[ϕ] <>
    "\nb: " <> ToString[b] <>
    (*"\nΣ(data-model): "<>ToString[diff]<>*)
    "\nΣ(data-model)^2: " <> ToString[sqDiff]
  ]
 ]
```



Verificação da soma dos quadrados das diferenças:

```
points2b3;
```

```
Clear[mdl, A, τ, c, b]
A = 2.9; τ = .186; c = 3.5; b = -2.1;
mdl =
  Discretize[Function[x, N[A * Cos[τ * (x + c)] + b]], Length[dta] - 1, Length[dta] - 1];
```

```
points2b3 - mdl;
```

```
(points2b3 - mdl)^2;
```

```
Total[(points2b3 - mdl)^2];
```

Medir as curvas utilizadas para definir “menos definida”.

```
{Variance[points2b1], Variance[points2b2], Variance[points2b3], Variance[points2b4]}
```

```
{0.597806, 0.743393, 2.01128, 14.7029}
```

```
{StandardDeviation[points2b1], StandardDeviation[points2b2],
 StandardDeviation[points2b3], StandardDeviation[points2b4]}
```

```
{0.773179, 0.862203, 1.41819, 3.83444}
```

F-test

```
In[]:= Manipulate[
  GetTSS = Total[(dta - Mean[dta])^2];
  GetMSS = Total[(mdl - Mean[mdl])^2];
  GetRSS = Total[(dta - mdl)^2];
  dta = points2b4;
  mdl =
   Discretize[Function[x, N[A * Cos[τ * (x + ϕ)] + b]], Length[dta] - 1, Length[dta] - 1];
  ListLinePlot[{dta, mdl, Abs[dta - mdl]},
   PlotRange → {{0, Length[dta] - 1}, {-15, 15}},
   PlotLegends → {"data", "model", "error"},
   ImageSize → Medium],
  {{A, 1}, 0, 8, .01},
  {{τ, .169}, .025, .5, .001},
  {{ϕ, -0.7}, -20, 20, .1},
  {{b, -5.9}, -12, 3, .1},
  Dynamic[
   tss = GetTSS;
   mss = GetMSS;
   rss = GetRSS;
   "A: " <> ToString[A] <>
    "\nτ: " <> ToString[τ] <>
    "\nϕ: " <> ToString[ϕ] <>
    "\nb: " <> ToString[b] <>
    "\nΣ(data-mean data)^2 (TSS): " <> ToString[tss] <>
    "\nΣ(model-mean model)^2 (MSS): " <> ToString[mss] <>
    "\nΣ(data-model)^2 (RSS): " <> ToString[rss] <>
    "\nMSS/RSS (basic significance): " <> ToString[mss / rss]
  ]
 ]
```

A

τ

ϕ

b

Out[ ]=

A: 1

τ: 0.169

ϕ: –0.7

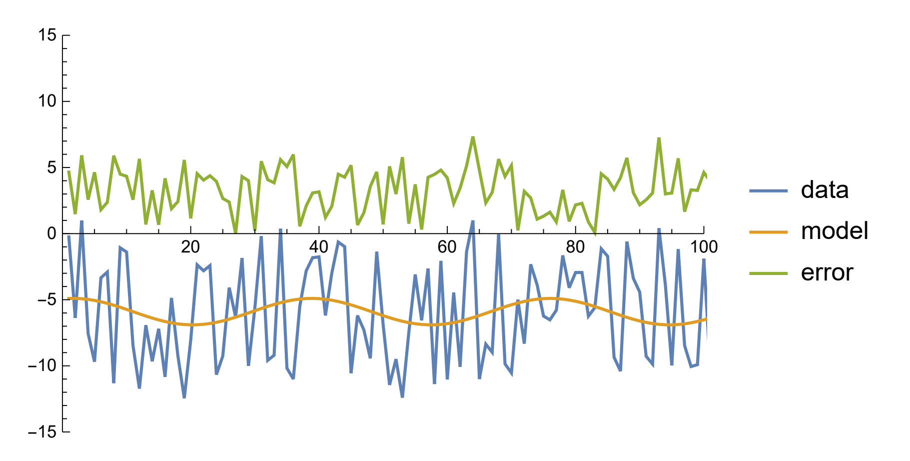
b: –5.9

$\Sigma(\text{data}-\text{mean data})^2$ (TSS): 1470.29

$\Sigma(\text{model}-\text{mean model})^2$ (MSS): 52.0009

$\Sigma(\text{data}-\text{model})^2$ (RSS): 1401.36

MSS/RSS (basic significance): 0.0371073

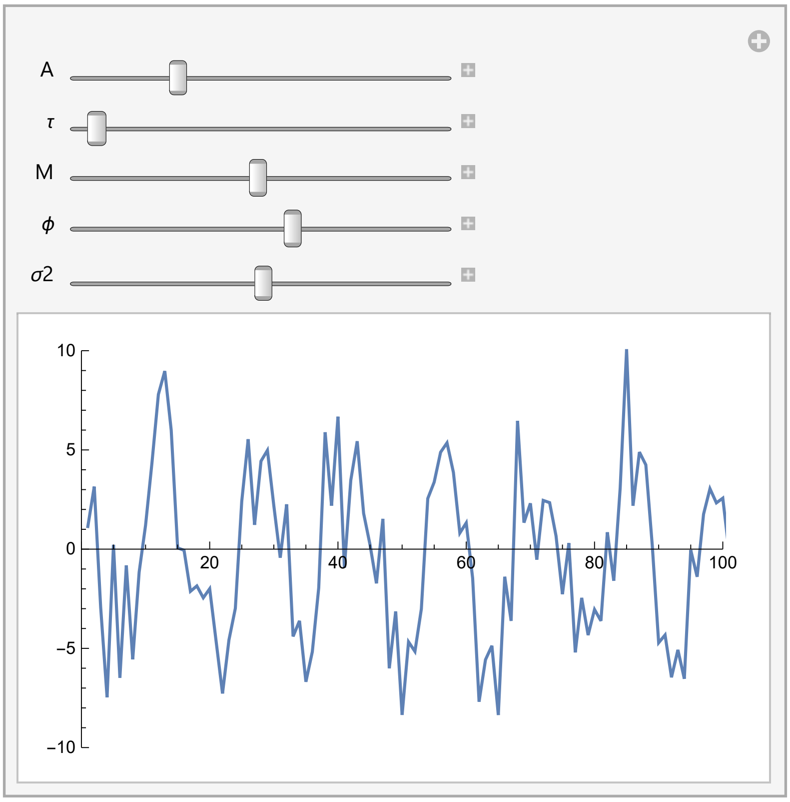


In[ ]:= **Discretize**

Out[ ]=
```
Function[{f, steps, x1}, Table[f[x], {x, 0, x1, Floor[x1/steps]}]]
```

In[•]:=
```
Manipulate[ListLinePlot[Discretize[
    Function[t, M + A Cos[2 π t / τ + ϕ] + RandomVariate[NormalDistribution[0, σ2]]], 100, 100],
   PlotRange → {{0, 100}, {-10, 10}}], {{A, -4.8}, -10, 10, .1},
  {{τ, -14.6}, -15, 0, .1}, {{M, -0.1}, -10, 10, .1},
  {{ϕ, 1.9}, -10, 10, .1}, {{σ2, 2.6}, 0.1, 5, .1}]
```

Out[•]=

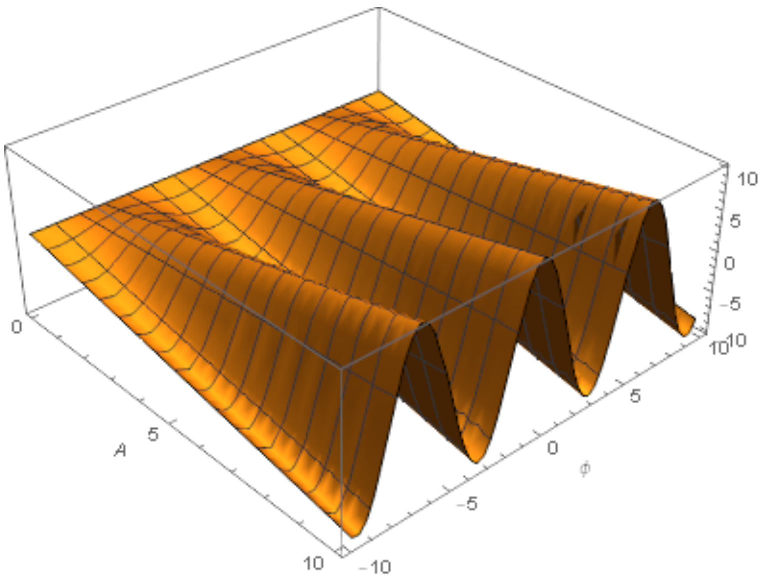


Este modelo são os quatro parâmetros normais mais o erro com variância desconhecida.

Segundo modelo expandido.

O quanto A e ϕ contribuem para β.

In[•]:=
```
Plot3D[A Cos[ϕ], {A, 0, 10}, {ϕ, -10, 10}, AxesLabel → Automatic]
```

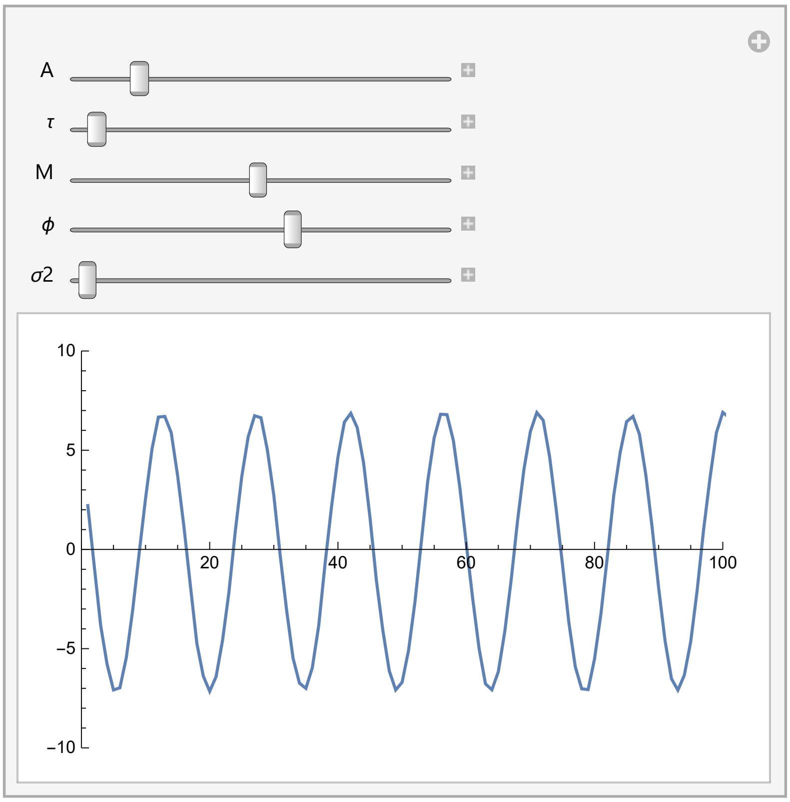


Iniciando em 0, β assume um intervalo de valores proporcionalmente maior conforme a amplitude aumenta, e oscila entre estes extremos (simétricos em torno de 0) periodicamente conforme a fase aumenta.

Portanto o sinal de β... Indendente da **amplitude,** é determinado pelo ponto na fase **fase = ponto no período**.

In[ ]:=
```
Manipulate[
 ListLinePlot[Discretize[Function[t, M + (A Cos[ϕ] Cos[2 π t/τ]) - (A Sin[ϕ] Sin[2 π t/τ]) +
     RandomVariate[NormalDistribution[0, σ2]]], 100, 100],
  PlotRange → {{0, 100}, {-10, 10}}], {{A, -7}, -10, 10, .1},
 {{τ, -14.6}, -15, 0, .1}, {{M, -0.1}, -10, 10, .1},
 {{ϕ, 1.9}, -10, 10, .1}, {{σ2, 0.1}, 0.1, 5, .1}]
```

Out[ ]=



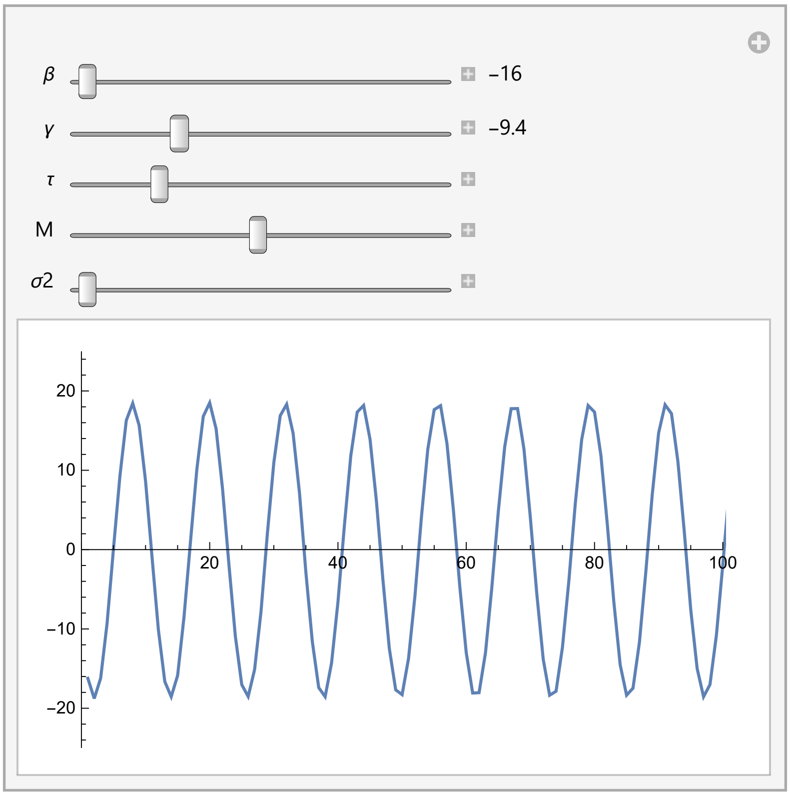
Ainda mesma coisa.

Agora período “conhecido”, $\tau = -15$, por exemplo.

Linearidade do modelo: modelo 1 não é, modelo 2 é.[11]

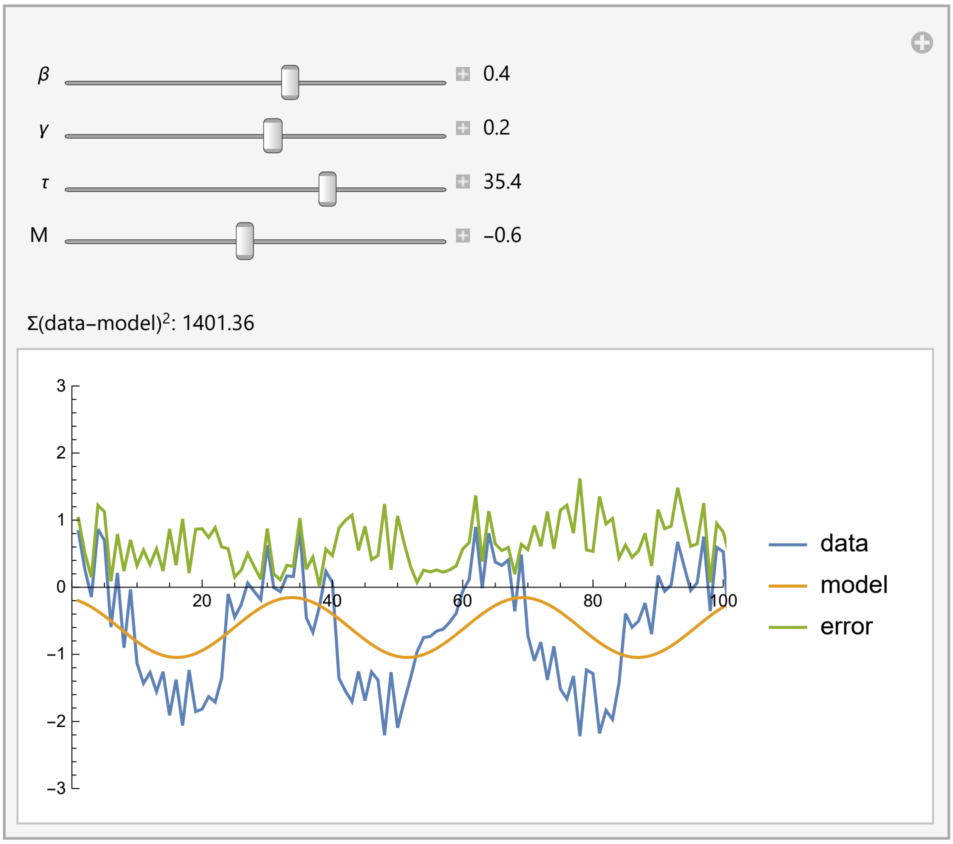
In[59]:=
```
Manipulate[ListLinePlot[Discretize[Function[t,
    M + (β Cos[2 π t / τ]) - (γ Sin[2 π t / τ]) + RandomVariate[NormalDistribution[0, σ2]]],
   100, 100], PlotRange → {{0, 100}, {-25, 25}}],
 {{β, -16}, -16, 16, .1, Appearance → "Labeled"},
 {{γ, -9.4}, -20, 20, .1, Appearance → "Labeled"},
 {{τ, -11.9}, -15, 0, .1}, {{M, -0.1}, -10, 10, .1}, {{σ2, 0.1}, 0.1, 5, .1}]
```

Out[59]=



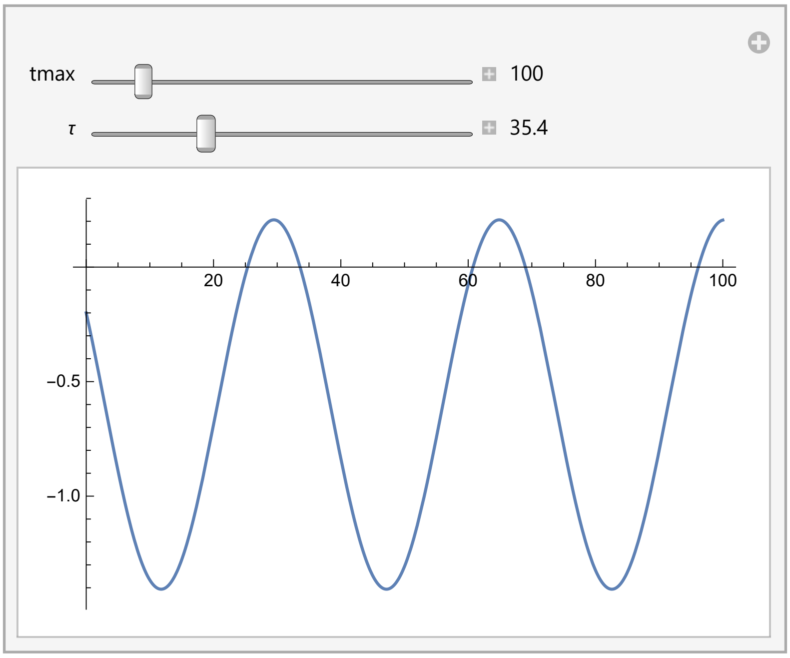
In[ ]:=
```
Manipulate[
 GetSqDiff = Total[(dta - mdl)^2];
 dta = points2b2;
 mdl = Discretize[
   Function[t, M + (β Cos[(2 π t)/τ]) - (γ Sin[(2 π t)/τ])], Length[dta] - 1, Length[dta] - 1];
 (*Print[mdl];*)
 ListLinePlot[{dta, mdl, Abs[dta - mdl]},
  PlotRange → {{0, Length[dta] - 1}, {-3, 3}},
  PlotLegends → {"data", "model", "error"},
  ImageSize → Medium],
 {{β, 0.4}, -2, 2, .1, Appearance → "Labeled"},
 {{γ, 0.7}, -2, 2, .1, Appearance → "Labeled"},
 {{τ, 35.4}, 0, 50, .1, Appearance → "Labeled"},
 {{M, -0.6}, -10, 10, .1, Appearance → "Labeled"},
 Dynamic[
  sqDiff = GetSqDiff;
  "\nΣ(data-model)^2: " <> ToString[sqDiff]
 ]
]
```

Out[ ]=



In[•]:=
```
Manipulate[
 β = 0.4; γ = 0.7; (*τ=35.4;*) M = -0.6;
 Plot[M + (β Cos[2 π t / τ]) - (γ Sin[2 π t / τ]), {t, 0, tmax},
  ImageSize → Medium],
 {{tmax, 100}, 0.1, 1000, .1, Appearance → "Labeled"},
 {{τ, 35.4}, 10, 100, .1, Appearance → "Labeled"}]
```

Out[•]=



τ é o período da onda.

t muda a escala do gráfico.

---

1 Diggle, Chetwynd. Statistics and Scientific Method (2011). Oxford University Press.

2 https://en.wikipedia.org/wiki/Normal_distribution

3 Diggle, Chetwynd. Statistics and Scientific Method (2011). Oxford University Press.

4 https://en.wikipedia.org/wiki/Normal_distribution

5 https://en.wikipedia.org/wiki/Probability_distribution

6 https://en.wikipedia.org/wiki/Probability_mass_function

7 Diggle, Chetwynd. Statistics and Scientific Method (2011). Oxford University Press.

8 https://www.mathsisfun.com/algebra/amplitude-period-frequency-phase-shift.html

9 https://www.cs.sfu.ca/~tamaras/sinusoids318/sinusoids318_4up.pdf

10 http://www2.clarku.edu/faculty/djoyce/trig/ptolemy.html

11 Barnett, Dobson. Analysing Seasonal Health Data (2010). Springer-Verlag.