Linearna regresija

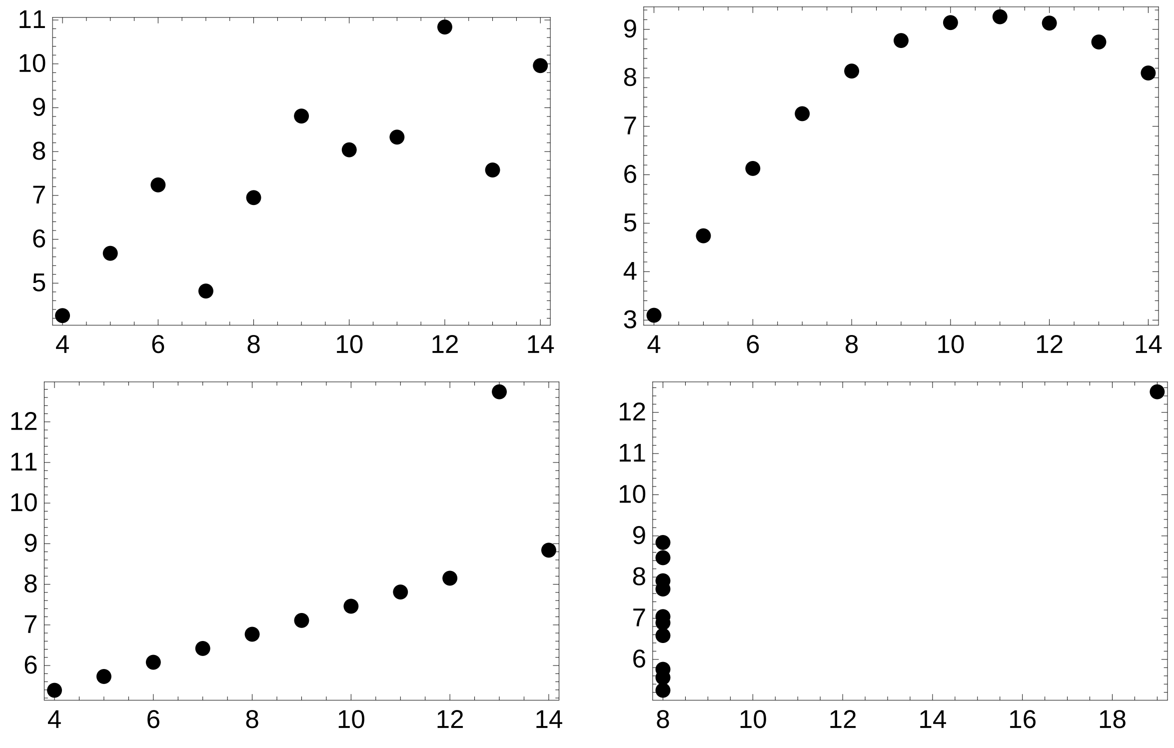
In[1]:=
```
SetOptions[ListPlot, ImageSize → 5 × 72, Frame → True,
  GridLinesStyle → Directive[Gray, Dashed],
  Method → {"GridLinesInFront" → True},
  PlotStyle → Directive[Thickness[Medium], PointSize[Large], Black],
  TicksStyle → Thickness[Medium], AxesStyle → Thickness[Medium],
  LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 12], Axes → False];
SetOptions[Plot, ImageSize → 5 × 72, Frame → True,
  GridLinesStyle → Directive[Gray, Dashed],
  Method → {"GridLinesInFront" → True},
  PlotStyle → Directive[Thickness[Medium], PointSize[Large], Black],
  TicksStyle → Thickness[Medium], AxesStyle → Thickness[Medium],
  LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 12], Axes → False];
Needs["ErrorBarPlots`"]
```

In[4]:=
```
anscombe = ReadList["~/vaje/rovf12/vaja7/anscombe.dat", Table[Real, {i, 8}]];
data1 = anscombe[[All, 1 ;; 2]];
data2 = anscombe[[All, 3 ;; 4]];
data3 = anscombe[[All, 5 ;; 6]];
data4 = anscombe[[All, 7 ;; 8]];
```

In[9]:=
```
p1 = ListPlot[data1];
p2 = ListPlot[data2];
p3 = ListPlot[data3];
p4 = ListPlot[data4];
```

In[13]:=
```
GraphicsGrid[{{p1, p2}, {p3, p4}}, ImageSize → 8 × 72]
```

Out[13]=



In[14]:=
```
fit1 = LinearModelFit[data1, x, x]
fit2 = LinearModelFit[data2, x, x]
fit3 = LinearModelFit[data3, x, x]
fit4 = LinearModelFit[data4, x, x]
```

Out[14]= FittedModel[$3.00009 + 0.500091\,x$]

Out[15]= FittedModel[$3.00091 + 0.5\,x$]

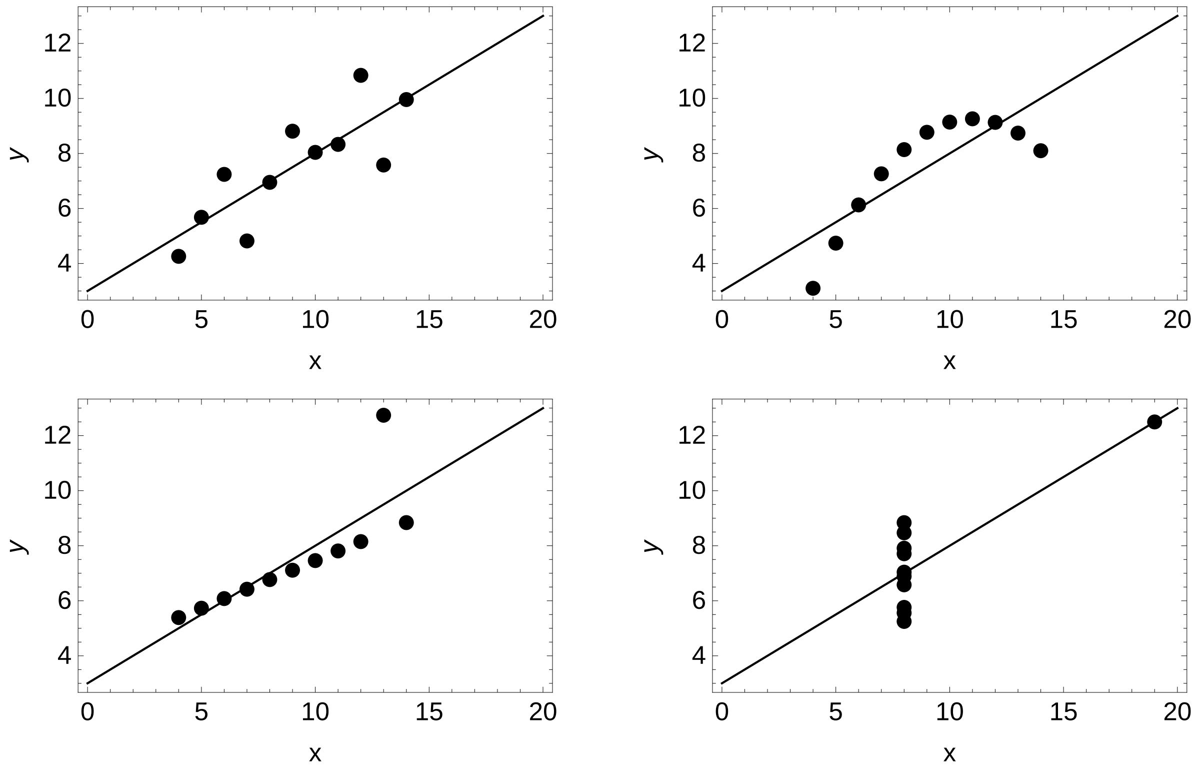
Out[16]= FittedModel[$3.00245 + 0.499727\,x$]

Out[17]= FittedModel[$3.00173 + 0.499909\,x$]

In[18]:=
```
p1 =
  Show[ListPlot[data1, FrameLabel → {"x", "y"}], Plot[Normal[fit1], {x, 0, 20}]];
p2 = Show[ListPlot[data2, FrameLabel → {"x", "y"}],
   Plot[Normal[fit2], {x, 0, 20}]];
p3 = Show[ListPlot[data3, FrameLabel → {"x", "y"}],
   Plot[Normal[fit3], {x, 0, 20}]];
p4 = Show[ListPlot[data4, FrameLabel → {"x", "y"}],
   Plot[Normal[fit4], {x, 0, 20}]];
```

In[22]:=
```
GraphicsGrid[{{p1, p2}, {p3, p4}}, ImageSize → 8 × 72]
```

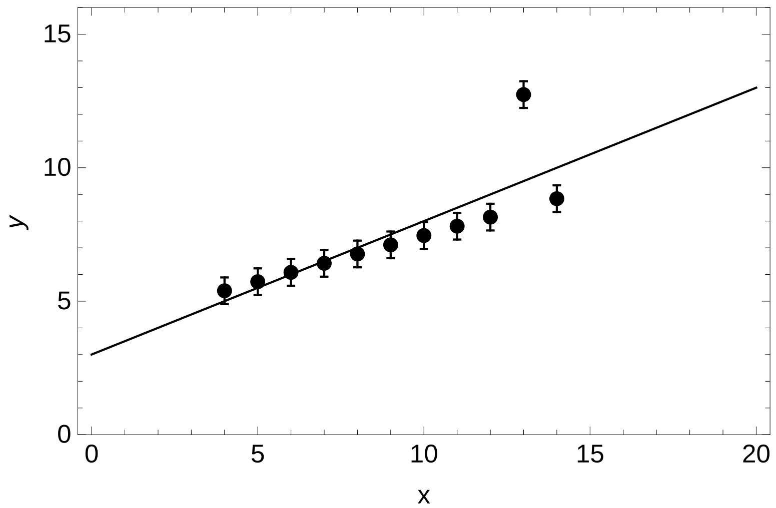
Out[22]=



In[23]:=
```
Manipulate[
 Module[{data, fit},
  data = Map[{#[[1]], #[[2]], 0.5} &, data3];
  data[[3, 3]] = σ;
  fit = LinearModelFit[data[[All, 1 ;; 2]],
    x, x, Weights → Map[1 / #^2 &, data[[All, 3]]]];
  Column[
   {Show[ErrorListPlot[data, PlotRange → {0, 16}, FrameLabel → {"x", "y"}],
     Plot[Normal[fit], {x, 0, 20}]],
    "y = " <> ToString[Normal[fit]],
    "χ^2/N = " <> ToString[fit["EstimatedVariance"]] <>
     " [" <> ToString[1. - Sqrt[2 / Length[data]]] <>
     ", " <> ToString[1. + Sqrt[2 / Length[data]]] <> "]"
   }
  ]
 ], {{σ, 0.5}, Table[x, {x, 0.5, 3, 0.5}]}
]
```

Out[23]=

σ 0.5



y = 3.00245 + 0.499727 x

χ^2/N = 6.11386 [0.573599, 1.4264]

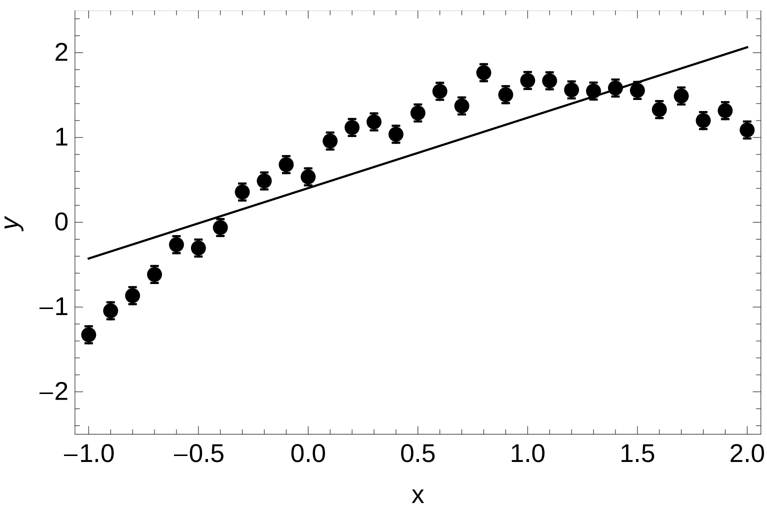
In[24]:=

```
Manipulate[
 Module[{data, fit1, fit2, fit},
  data = Table[{x, 0.8 + 1.5 x - 0.7 x^2 + RandomReal[NormalDistribution[0, σ]], σ},
    {x, -1, 2, 0.1}];
  fit1 = LinearModelFit[data[[All, 1 ;; 2]], x, x,
    Weights → Map[1 / #^2 &, data[[All, 3]]]];
  fit2 = LinearModelFit[data[[All, 1 ;; 2]], {x, x^2}, x,
    Weights → Map[1 / #^2 &, data[[All, 3]]]];
  fit = If[n == 1, fit1, fit2];
  Column[
   {Show[ErrorListPlot[data, PlotRange → {-2.5, 2.5}, FrameLabel → {"x", "y"}],
     Plot[Normal[fit], {x, -1, 2}]],
    "y = " <> ToString[StandardForm[Normal[fit]]],
    "χ²/N = " <> ToString[fit["EstimatedVariance"]] <>
     " [" <> ToString[1. - Sqrt[2 / Length[data]]] <>
     ", " <> ToString[1. + Sqrt[2 / Length[data]]] <> "]"
   }
  ]
 ], {{σ, 0.1}, Table[x, {x, 0.1, 0.5, 0.1}]}, {{n, 1}, {1, 2}}
]
```

Out[24]=

σ 0.1 | 0.2 | 0.3 | 0.4 | 0.5

n 1 | 2

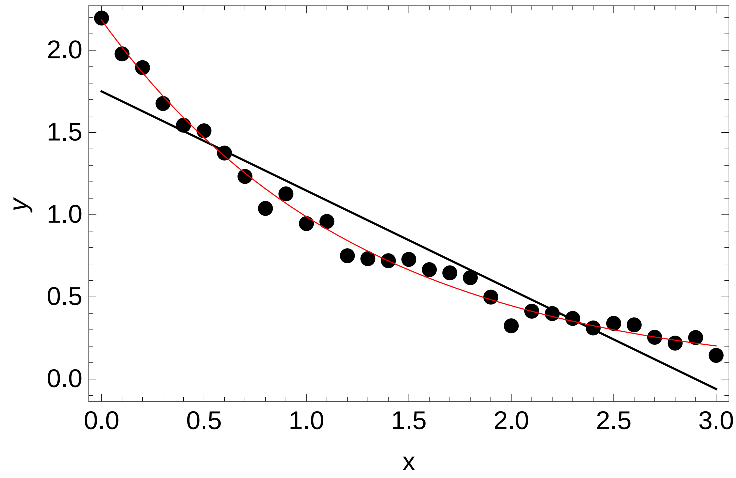


y = 0.403088 + 0.830615 x

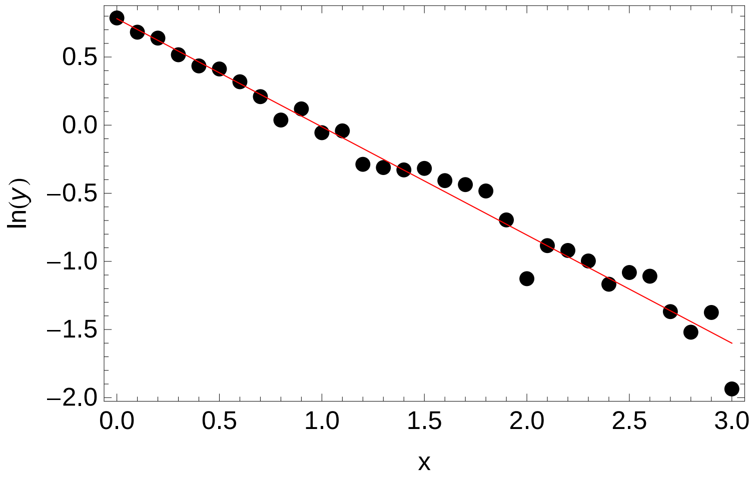
χ^2/N = 24.2762 [0.746, 1.254]

In[25]:=

```
Module[{data, data1, fit, fit1, p2},
 data = Table[
   {x, 2.2 Exp[-0.8 x] + Random[NormalDistribution[0, 0.05]]}, {x, 0, 3, 0.1}];
 data1 = Map[{#[[1]], Log[#[[2]]]} &, data];
 fit = LinearModelFit[data, x, x];
 fit1 = LinearModelFit[data1, x, x];
 Column[{
   Show[ListPlot[data, FrameLabel → {"x", "y"}], Plot[Normal[fit], {x, 0, 3}],
    Plot[Exp[Normal[fit1]], {x, 0, 3}, PlotStyle → Red],
    ImagePadding → {{50, 10}, {35, 10}}],
   Show[ListPlot[data1, FrameLabel → {"x", "ln(y)"}], Plot[Normal[fit1],
     {x, 0, 3}, PlotStyle → Red], ImagePadding → {{50, 10}, {50, 10}}]
  }]
]
```



Out[25]=



In[26]:=

1. naloga: V datoteki *Ljubljana-Bežigrad.zip* so zbrani vremenski podatki z merilne postaje Ljubljana Bežigrad za obdobje od 1. 1. 1900 do 31. 1. 2013. Poišči premico, ki najbolje opiše odvisnost vlažnosti od povprečne dnevne temperature v letu 2010.

2. naloga: Teorija kemijske kinetike napove za sigmoidno krivuljo iz podatkov *Adrenalin.dat* naslednjo odvisnost $F / F_{max} = c / (a + c)$, kjer pomeni a koncentracijo s polovičnim maksimalnim učinkom. Določi koeficienta F_{max} in a.

Pretvori v linearno zvezo – ena pot je uvedba recipročnih spremenljivk 1 / *F* in 1 / *c*, druga pa je uvedba spremenljivke *c* / *F*.

3. naloga: V nalogi 3.2 smo za vsak mesec v letu izračunali povprečje in disperzijo povprečnih dnevnih temperatur z merilne postaje Ljubljana Bežigrad v obdobju 2000-2009. Časovno odvisnost teh povprečij modeliramo s funkcijama $T(t) = T_0 + T_1 \cos\left(2\pi \frac{t}{t_0}\right)$ in $T(t) = T_0 + T_1 \cos\left(2\pi \frac{t-t_1}{t_0}\right)$, kjer je t čas, merjen od začetka zime, in t_0 eno leto. D o l o č i parametre T_0, T_1 in t_1 ter primerjaj vrednosti χ^2 za obe funkciji. Teorija napove, da je ocena napake, ki jo naredimo pri izračunu povprečja količine, porazdeljene po Gaussovi porazdelitvi, enaka $\frac{\sigma}{\sqrt{N}}$, kjer je σ disperzija porazdelitve in N število podatkov v našem vzorcu. V nalogi 3.2 smo sicer opazili, da nekatere mesečne porazdelitve odstopajo od Gaussove, *a* je odstopanje dovolj majhno, da lahko kjub temu uporabimo tako oceno.