# Napolitano Mathematica Primer for Physicists

## Chapter 7 Data Analysis Basics

### Example 7.1

Does the gas mileage of a car vary between summer and winter? A two-column data file mpg.dat has data for a routinely maintained 1994 Honda Accord. The first column is the day since 1 July 2008, and the second is the gas mileage (in miles per gallon) on that date. Find the average gas mileage and its standard deviation, and plot the gas mileage versus time. Then separate the data sets into January/February and July/August. Histogram these two data sets, and find the average gas mileage and the standard deviation.

In [1]:
```
(*DATA ANALYSIS OF GAS MILEAGE*)

(*Read data, extract columns.*)

data = Import["mpg.dat"];
Dimensions[data]
dayVals = data[[All, 1]];
mpgVals = data[[All, 2]];
```

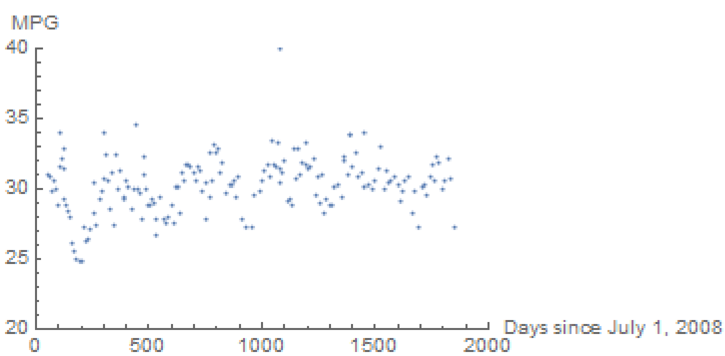
Out[4]:
```
{177, 2}
```

In [7]:
```
(*Mean and SD.*)

Mean[mpgVals]
StandardDeviation[mpgVals]
```

Out[8]:
```
30.2689
2.00811
```

In [10]:
```
(*Plotting*)
ListPlot[data, PlotRange -> {{0, 2000}, {20, 40}},
    AxesLabel -> {"Days since July 1, 2008", "MPG"}]
```

Out[11]:

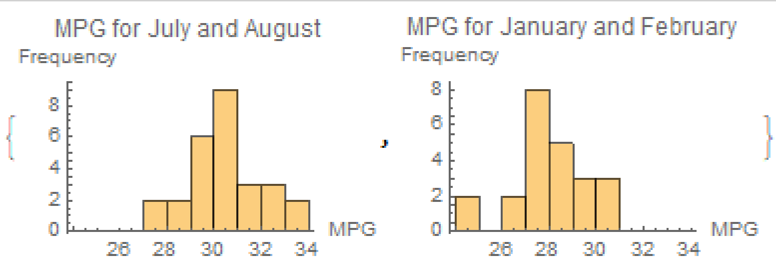


In [12]:
```
(* Extract summer and winter data segments. *)

mpgJulAug = Select[data,
                     Mod[#[[1]], 365] <= 62 &][[All, 2]]
mpgJanFeb = Select[data,
                     Mod[#[[1]], 365] > 184
                     && Mod[#[[1]], 365] <= 245 &][[All, 2]]
```

Out[13]:
```
{31, 31.2, 29.4, 29.2, 30.5, 30.1, 28.6, 29.9, 27.8, 30.4, 29.4, 32.6, 3
0.6, 33.1, 32,

>   29.1, 29.2, 28.9, 32.8, 30.7, 30.3, 30, 30.6, 31.4, 33, 30.7, 27.3}
{24.8, 24.8, 27.3, 26.3, 26.4, 27.1, 27.8, 27.6, 28, 28.9, 27.6, 27.3, 2
7.3, 29.6, 29.3,

>   28.8, 28.9, 30.1, 30.2, 28.3, 29.9, 27.2, 30.1}
```

In [15]:
```
(* Comparison *)

SHist = Histogram[mpgJulAug, {1},  PlotRange -> {{24, 34}, Automatic},
                     PlotLabel -> "MPG for July and August",
                     AxesLabel -> {"MPG", "Frequency"}];

WHist = Histogram[mpgJanFeb, {1},  PlotRange -> {{24, 34}, Automatic},
                     PlotLabel -> "MPG for January and February",
                     AxesLabel -> {"MPG", "Frequency"}];

           (* PlotRange makes comparing the graphs easier. *)

{SHist, WHist}
{Mean@mpgJulAug, Mean@mpgJanFeb}
{StandardDeviation@mpgJulAug, StandardDeviation@mpgJanFeb}
```

Out[19]:




```
{30.363, 27.9826}
{1.4985, 1.55319}
```

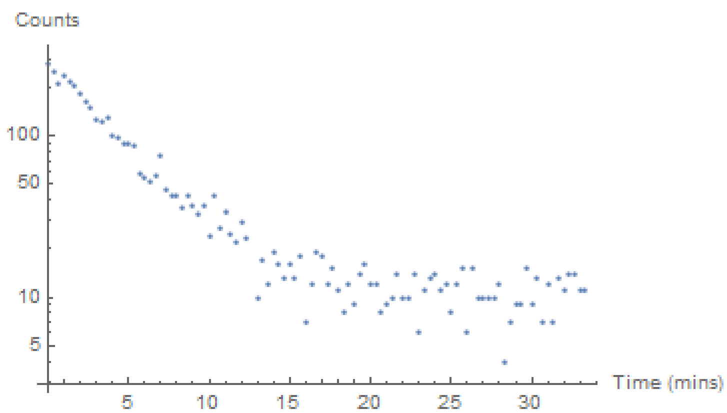
# Chapter 8 Fitting Data

**Example 8.1** The file “Cs137.dat” contains the result of a measurement of the γ decay of 137mBa∗, following the long lived β decay of 137Cs. The data is in two columns representing “time” and “counts,” with time measured in 20-second intervals. This state is known to decay with a half-life of2.552minutes,but there is a constant background due to residual cesium after the chemicalseparation to extract the barium. Analyze this data and confirm the half-life.

In [99]:
```
Remove["Global`*"]
```

In [100]:
```
(*RADIOACTIVE DECAY OF 137Cs*)

(* Loading data. *)
dataFile = Import["Cs137.dat"];
timeTics = dataFile[[All, 1]];
cntsVals = dataFile[[All, 2]];
erroVals = Sqrt[cntsVals];

(* Convert time into minutes and plot. *)
timeVals = timeTics / 3;
data = Transpose[{timeVals, cntsVals}];
ListLogPlot[data, AxesLabel -> {"Time (mins)", Counts}]
```

Out[109]:




In [160]:
```
(* Unweighted fits, with tHalf floating or fixed. *)
funcFull = a 2^(-t / tHalf) + b
parsFull = FindFit[data, funcFull, {a, tHalf, b}, t]

funcFixT = funcFull /. tHalf -> 2.552;
parsFixT = FindFit[data, funcFixT, {a, b}, t]
```

Out[161]:

$2^{-\frac{t}{tHalf}}\, a + b$

{a → 272.064, tHalf → 2.71357, b → 9.90075}

{a → 277.215, b → 11.1862}

In [134]:
```
parsFull2 = NonlinearModelFit[data, funcFull, {a, tHalf, b}, t]
```

Out[134]:

FittedModel[ $9.90075 + 272.064 \times 2^{-0.368518\,t}$ ]

In [146]:
```
parsFull3 = NonlinearModelFit[data, funcFull /. tHalf -> 2.552, {a, b}, t]
```

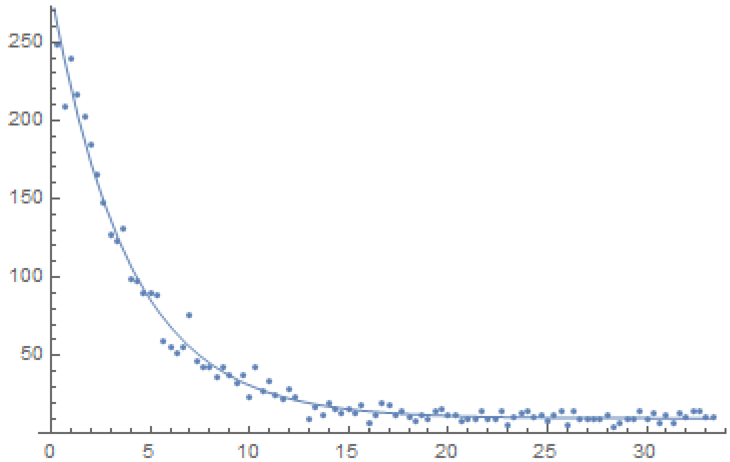
Out[146]:

FittedModel[ $11.1862 + 277.215 \times 2^{-0.39185\,t}$ ]

In [167]:

```
Show[
    ListPlot[data, PlotRange -> {0, a /. parsFull}],
    Plot[funcFull /. parsFull, {t, 0, Max[timeVals]},
            PlotRange -> {0, a /. parsFull}]
    ]
```

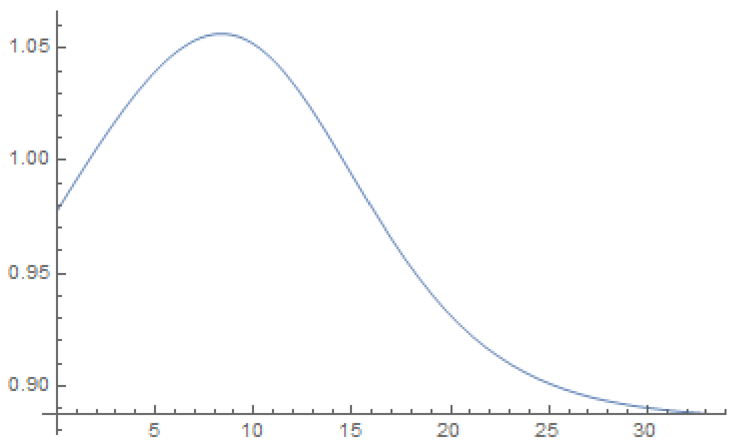
Out[167]:




In [174]:

```
rFloat2Fixed =
    (funcFull /. parsFull) / (funcFixT /. parsFixT);
Plot[rFloat2Fixed, {t, 0, Max[timeVals]}]
```

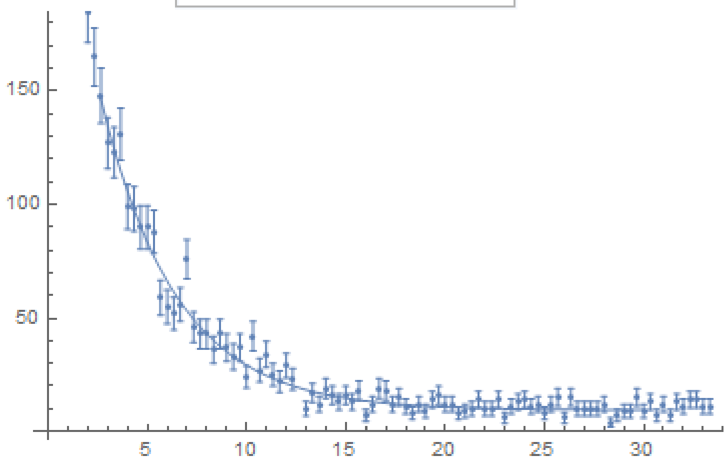
Out[175]:



In [233]:

```
(* Weighted fit with halflife parameter. *)

efit = NonlinearModelFit[data, funcFull, {a, tHalf, b}, t,
                            Weights -> 1 / erroVals^2]
Needs["ErrorBarPlots`"]
eBars = Table[ErrorBar[erroVals[[i]]],
                {i, 1, Length[timeTics]}];
edata = Partition[Riffle[data, eBars], 2];

Show[ErrorListPlot[edata], Plot[Normal[efit], {t, 0, 30}]]
```

Out[234]: FittedModel[ $9.36367 + 275.517 \times 2^{-0.379363\,t}$ ]



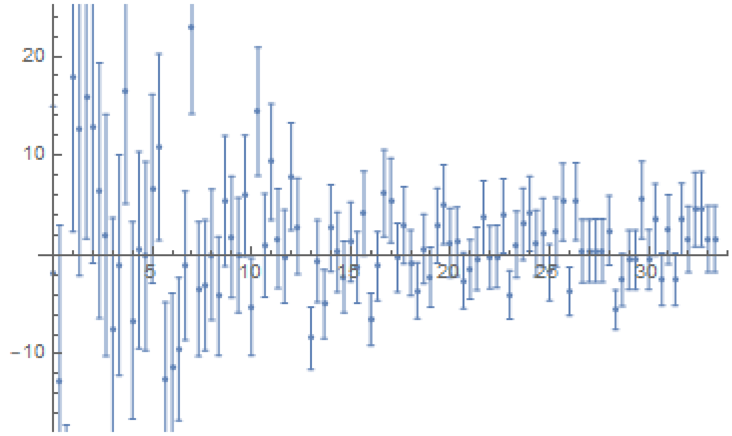

In [243]:

```
(* Learn about fit and parameters. *)

efit["Properties"];
efit["BestFitParameters"]
efit["ParameterErrors"]
```

Out[245]: {a → 275.517, tHalf → 2.63599, b → 9.36367}
{7.6324, 0.0736614, 0.483013}

In [247]:
```
diffData = Partition[Riffle[Transpose[{timeVals, efit["FitResiduals"]}],
                                        eBars], 2];
ErrorListPlot[diffData]
```

Out[248]:




In [251]:
```
Total[efit["StandardizedResiduals"]^2] / (Length[timeVals] - 3)
```

Out[251]: 1.03762