Planetary Orbits

Mathematica has some basic planetary data built in. We can use it to re-discover Kepler's third law, and, from that, infer the form of Newton's law of Universal Gravitation. Since the orbits are not quite circular, the appropriate quantity to use is the semi-major axis (i.e. half of the major axis of the orbital ellipse.)

In[26]:=
```
Clear["Global`*"];
```

In[27]:=
```
planets =
  {"Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"};
```

In[28]:=
```
data = Table[{PlanetData[p, "MajorAxis"] / 2.0,
    PlanetData[p, "OrbitPeriod"]}, {p, planets}];
```

Pluto is technically classified as a "Dwarf Planet", so we have to add it in separately.

In[29]:=
```
AppendTo[planets, "Pluto"];
AppendTo[data, {MinorPlanetData["Pluto", "MajorAxis"] / 2.0,
  MinorPlanetData["Pluto", "OrbitPeriod"]}];
```

In[30]:=
```
TableForm[data,
 TableHeadings → {planets, {"Orbital Radius", "Orbital Period"}}]
```

Out[30]//TableForm=

	Orbital Radius	Orbital Period
Mercury	0.387099 au	87.96926 days
Venus	0.723332 au	224.70080 days
Earth	1. au	365.25636 days
Mars	1.52366 au	1.8808476 a
Jupiter	5.20336 au	11.862615 a
Saturn	9.53707 au	29.447498 a
Uranus	19.1913 au	84.016846 a
Neptune	30.069 au	164.79132 a
Pluto	39.5886 au	249. a

For plotting and computation, we should switch all data to uniform units of meters and seconds.

In[31]:=
```
data = UnitConvert[data, "SIBase"];
TableForm[data,
 TableHeadings → {planets, {"Orbital Radius", "Orbital Period"}}]
```

Out[32]//TableForm=

	Orbital Radius	Orbital Period
Mercury	5.79092×10^{10} m	7.600544×10^{6} s
Venus	1.08209×10^{11} m	1.9414149×10^{7} s
Earth	1.49598×10^{11} m	3.1558149×10^{7} s
Mars	2.27937×10^{11} m	5.9355036×10^{7} s
Jupiter	7.78412×10^{11} m	3.7435566×10^{8} s
Saturn	1.42673×10^{12} m	9.2929236×10^{8} s
Uranus	2.87097×10^{12} m	2.6513700×10^{9} s
Neptune	4.49825×10^{12} m	5.2004186×10^{9} s
Pluto	5.92237×10^{12} m	7.86×10^{9} s

Animation (inner planets only)

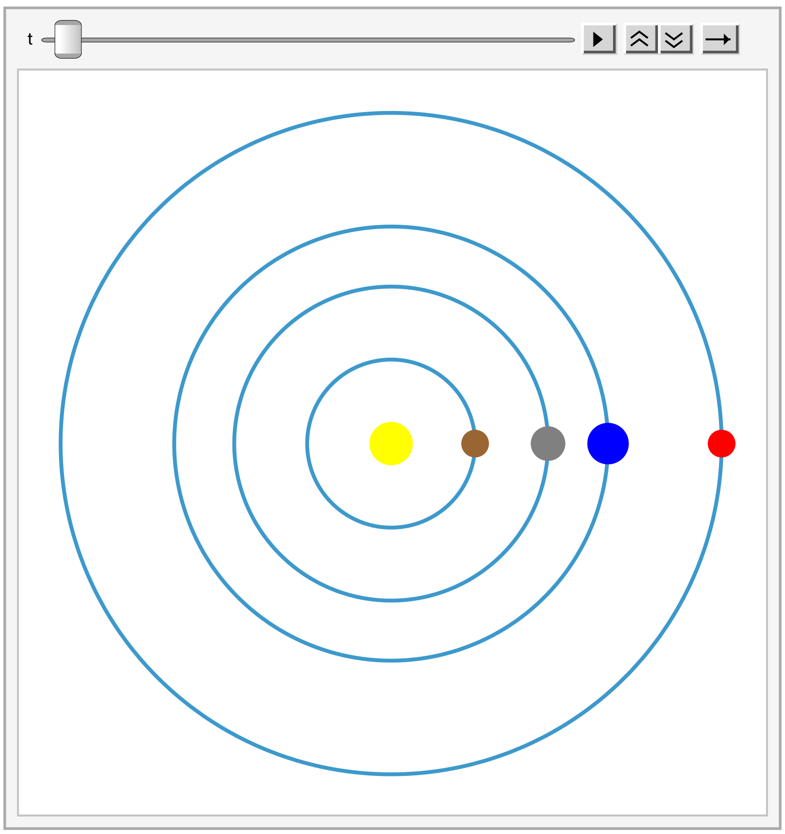
For animation, it is convenient to scale all lengths by Earth’s orbital radius, and all times by Earth’s orbital period (1 year):

In[33]:=
```
comp =
  Table[{data⟦i, 1⟧ / data⟦3, 1⟧, data⟦i, 2⟧ / data⟦3, 2⟧}, {i, 1, Length[data]}];
```

In[34]:=
```
x[r_, T_, t_] := r Cos[2 π t / T]
y[r_, T_, t_] := r Sin[2 π t / T]
pos[r_, T_, t_] := {x[r, T, t], y[r, T, t]}
```

In[37]:=
```
planets = {1, 2, 3, 4};  (* Just show the 4 inner planets *)
Animate[Show[{
   Graphics[{Yellow, Disk[{0, 0}, 0.1]}] (* Sun *),
   ParametricPlot[
    Table[pos[comp[[i, 1]], 1, ϕ], {i, planets}], {ϕ, 0, 1}, PlotRange → All],
   ListPlot[Table[{pos[comp[[i, 1]], comp[[i, 2]], t]}, {i, planets}],
    PlotStyle → {{Brown, PointSize[0.04]}, {Gray, PointSize[0.05]},
      {Blue, PointSize[0.06]}, {Red, PointSize[0.04]}
      (*, {Brown, PointSize[0.10]}, {Yellow, PointSize[0.08]} *)}]
  }
 ], {t, 0, 5, 0.01, DefaultDuration → 60, AnimationRunning → False}]
```

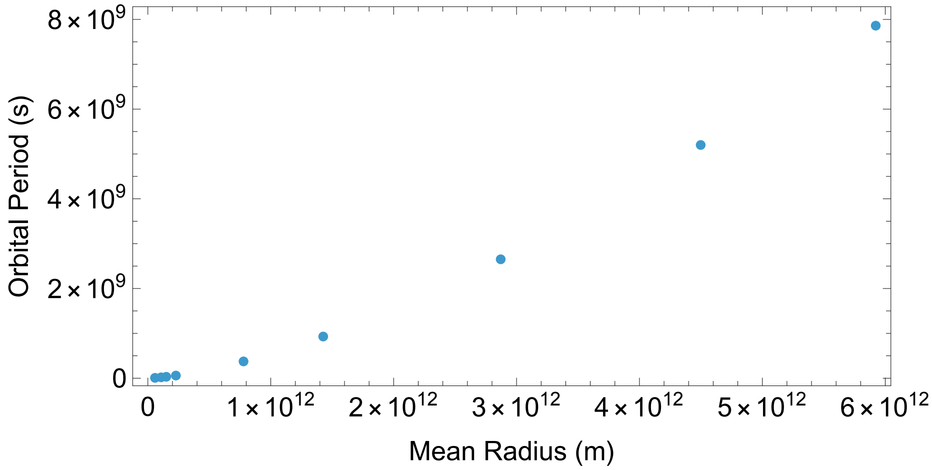
Out[38]=



In[39]:=
```
opts = {AspectRatio → 1 / 2, LabelStyle → Larger, Frame → True,
   FrameLabel → {"Mean Radius (m)", "Orbital Period (s)"},
   ImageSize → Scaled[0.75]}; (* Common plot options *)
```

In[40]:=
```
ListPlot[data, opts]
```

Out[40]=



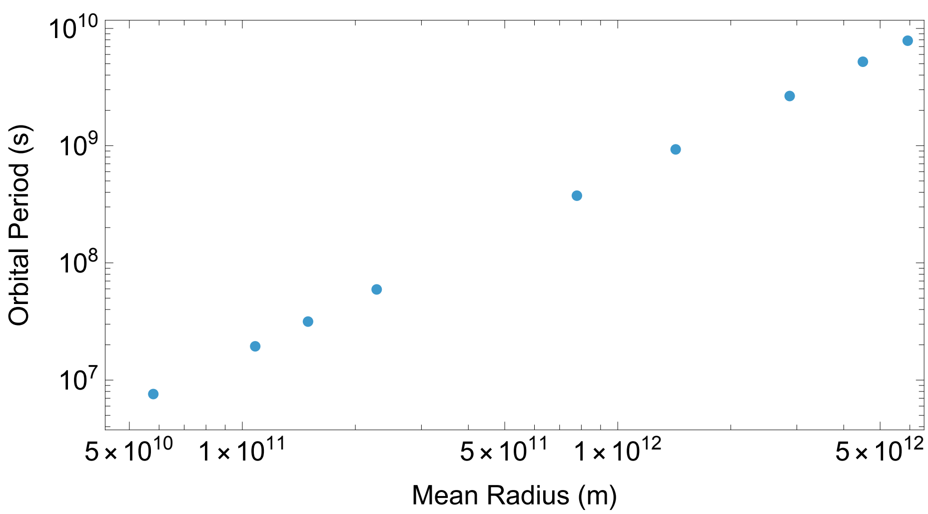
This plot is not terribly useful for visualization, since all of the inner planets are squished down in the lower left corner, and it doesn't follow any obvious curve.

Power-Laws and Log-Log Plots

The T vs. R plot is not terribly useful for visualization, since all of the inner planets are squished down in the lower left corner. Using a Log-Log plot spreads things out nicely.

In[41]:=
```
ListLogLogPlot[data, opts]
```

Out[41]=



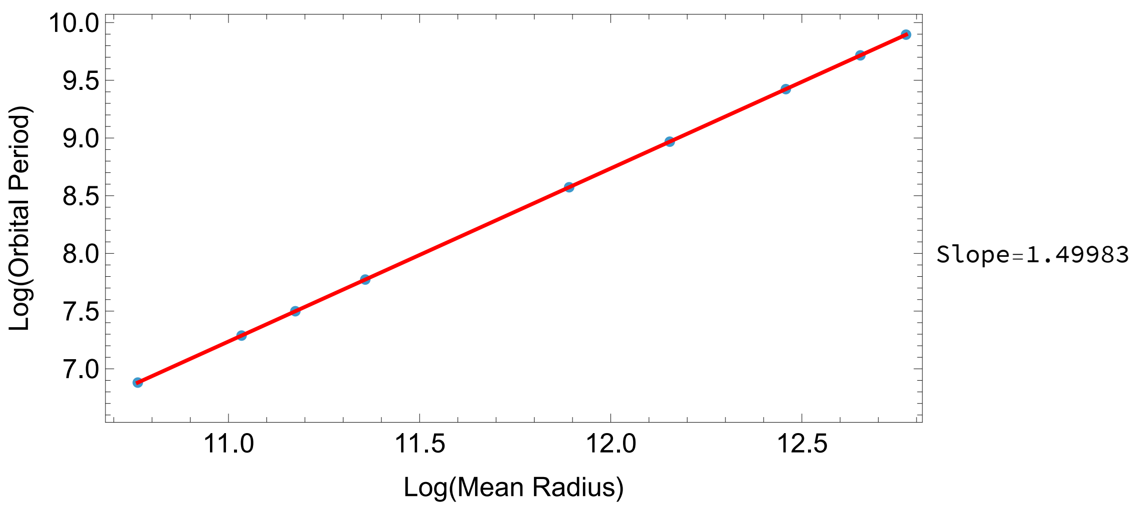
This plot shows a suggestive linear trend. Try fitting it to find the slope:

In[42]:=
```
logdata = Log[10, QuantityMagnitude[data]]
```
Out[42]=
```
{{10.7627, 6.8808447}, {11.0343, 7.28811836}, {11.1749, 7.49911152},
 {11.3578, 7.77345757}, {11.8912, 8.57328440}, {12.1543, 8.96815237},
 {12.458, 9.42347034}, {12.653, 9.71603830}, {12.7725, 9.895}}
```

In[43]:=
```
llfit[x_] = Fit[logdata, {1, x}, x]
```
Out[43]=
```
-9.26216 + 1.4999 x
```

The slope is very nearly 1.5.

In[44]:=
```
Show[{ListPlot[logdata], Plot[llfit[x], {x, logdata[[1, 1]], logdata[[-1, 1]]},
    PlotStyle → Red, PlotLegends → "Slope=1.49983"]}, LabelStyle → Larger,
 Frame → True, FrameLabel → {"Log(Mean Radius)", "Log(Orbital Period)"},
 ImageSize → Scaled[0.75], AspectRatio → 1 / 2
]
```
Out[44]=



Log-Log Plots and Power Laws

Suppose the period T is related to the orbital radius R by a power law $T = c\,R^m$. How would that show up in a plot? Try taking logs of both sides:

$$T = c\,R^m$$

$$\log(T) = \log(c) + \log(R^m) = \log(c) + m\,\log(R)$$

This suggests that if you plot log(T) vs. log(R), you will get a straight line with slope ‘m’. In this case, we have m = 1.5. This means

$$T = c\,R^{1.5} = c\,R^{3/2}$$

squaring both sides gives

$$T^2 = c\,R^3$$

which is Kepler’s 3rd law.

Kepler's Third Law

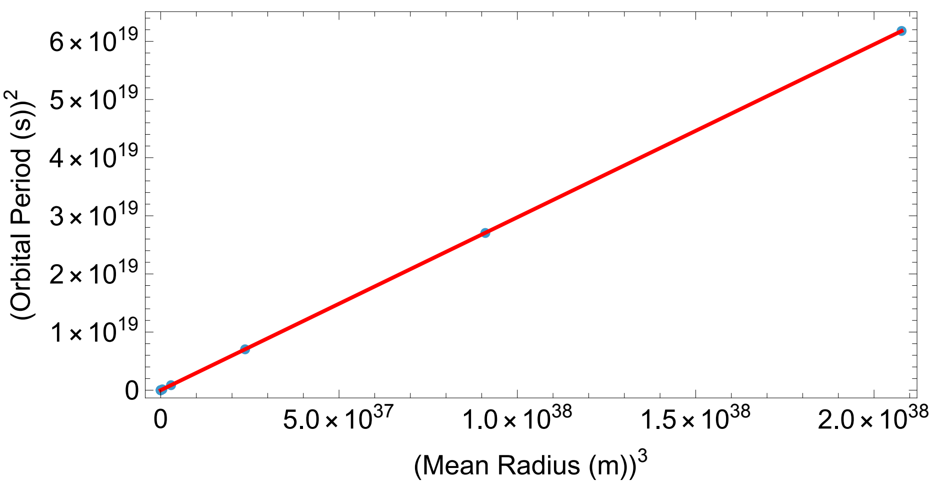
The Log-Log analysis suggests that another way to plot it is to plot T^2 vs. R^3. It shows a linear trend, so we can fit it with a straight line. We will see that the slope is given by $\frac{4\pi^2}{G\,M_{sun}}$

In[45]:=
```
data32 = Table[{QuantityMagnitude[data[[i, 1]]]^3,
    QuantityMagnitude[data[[i, 2]]]^2}, {i, 1, Length[data]}];
```

In[46]:=
```
fit = LinearModelFit[data32, { x}, x, IncludeConstantBasis → False ];
```

In[47]:=
```
Show[{ListPlot[data32],
   Plot[fit[x], {x, data32[[1, 1]], data32[[-1, 1]]}, PlotStyle → Red]},
  LabelStyle → Larger, Frame → True,
  FrameLabel → {"(Mean Radius (m))^3", "(Orbital Period (s))^2"},
  ImageSize → Scaled[0.75], AspectRatio → 1 / 2
]
```

Out[47]=



In[48]:=
```
slope = fit'[x]
```

Out[48]=

2.97414×10^{-19}

In[49]:=
```
G = Quantity[1, "GravitationalConstant"];
Msun = Quantity[1, "SolarMass"];
UnitConvert[4 π^2 / (G * Msun), "SIBase"]
```

Out[51]=

$2.975 \times 10^{-19}\ s^2/m^3$