Applied Numerical Integration

MAT 229, Spring 2021

Week 6

Cadaver Temperature:

An Exercise in Modeling Post-Mortem Temperatures in the Human Body

---

The Data

The data for this modeling exercise is taken from the article Post-Mortem Temperature and the Time of Death (G. S. W. De Saram, G. Webster, N. Kathirgamatamby, Post-Mortem Temperature and the Time of Death, 46 J. Crim. L. Criminology & Police Sci. 562 (1955-1956)).

We digitized the data (pulled it from tables) based on the deaths of 40 executed prisoners, and records of their bodies' rectal temperatures at hourly intervals for 12 hours following their deaths.

Per the model you'll see in a moment, we scaled the data using the initial temperature of the body (at 9am) and the ambient, background temperature (we might expect the body's temperature to approach this temperature, if given enough time).
Then we averaged the hourly temperatures for all of the 28 time series with complete records, to get the following temperature data over time:

```
temps = {1., 0.9222684153743212, 0.8512044304340011,
   0.7870267914889395, 0.7261989779202581, 0.6776207521636185,
   0.6237002302952035, 0.5709996871736893, 0.5254784973612153,
   0.4817907940680118, 0.43677400085447377, 0.3942083969507442};
times = {9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
 (* 9am and every hour after until 8pm (20 hour) *)
```

The Model

Newton's law of cooling suggests that a body will equilibrate with its environment by slowly losing heat in what might appropriately be called a "dying (exponential) fashion" (the following model for rectal temperature is featured in the same article):

TempRectal(*t; s, r*)-TempAmbient =(TempInitial-TempAmbient) $s\,e^{rt}$

where

1. TempRectal is the observed temperature of the body, which is a function of time;
2. TempAmbient is the average ambient temperature in the area where the body is stored;
3. TempInitial is the initial temperature of the body, at 9:00 am; while executions took place at 8:00 am, there was a delay to make sure that the prisoner was officially dead, etc. It was usually 15 minutes or so before the person was actually declared dead. Then they were removed from the premises, taken to the morgue, etc.

So this scaled data is defined as:

$$\text{temp}(t; s, r) = \frac{\text{TempRectal}(t;s,r)-\text{TempAmbient}}{\text{TempInitial}-\text{TempAmbient}};$$

This new variables temp should therefore be an exponential function:

$temp(t; s, r) = s\,e^{rt}$

where *r* should be negative. We often need to make some good guesses for the parameters of our model, *s* and *r*, before doing our model building.

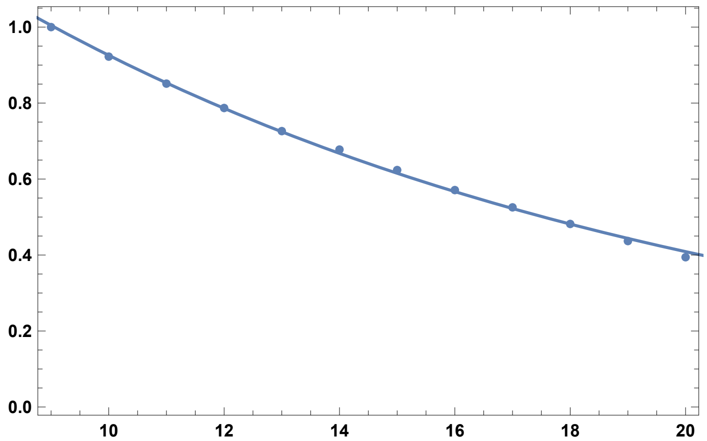
1. We know that the temperature ratio is roughly 40% (or .40) of its original value by 12 hours later, so $r \cong$ Log[.40]/12 = -0.0763576
2. Notice that $s\,e^{9r} \cong 1$; so $s \cong e^{-9r} = e^{9*0.0763576} \cong 2$

So we're ready to find a model!

Using a process called "non-linear Regression" (which is based strongly on calculus!) we obtain the following

FittedModel[$2.0953\, e^{-0.0817016\, t}$]

Notice that we were pretty close in our parameter guesses! Notice also that the model seems to fit the data pretty well:



Computing heat loss as an integral, in 3 different ways:

The body contains its full complement of heat at the outset, then slowly loses it to the environment over time. How much heat is lost over time? The **rate** at which it loses heat **decreases** over time (the **slope** of the curve).

We will compute the fraction lost over the first 10 hours as the integral

$$\text{HeatLoss}=\left(\int_9^{19}\text{temp}[t]\,dt\right)/(-s/r).$$

(Don't worry about where that came from! We choose to look at the first **10** hours because we want to be able to use Simpson's Rule, which requires an even number of subintervals....)

The integral may be computed in three different ways (at least -- probably an infinite number of different ways, but we only have time for three!:).

Method 1: using the model as though it's perfect:

In[185]:=
```
s = 2.095301183703536;
r = -0.08170156048049892;
temp[t_] := s E^(r t)
```

Our model is just an exponential, so it's easy to compute the answer:

$$temp(t) = s\,e^{rt} = 2.0953\,e^{-0.0817016\,t}$$

Okay then: we have a model (which we derived from the data). Because of the form of the model, notice that we can simplify this to

$$\text{HeatLoss}=\left(\int_9^{19}\text{temp}[t]\,dt\right)/(-s/r) = -r\int_9^{19} e^{rt}\,dt$$

We integrate that easily to get an anti-derivative: $-e^{rt}$, which we evaluate at its endpoints to get about 0.267601:

```
In[188]:= Integrate[temp[t], {t, 9, 19}] / (-s / r)
          (* and, as we suspected, we get *)
          -(E^(r 19) - E^(r 9))
Out[188]= 0.267601
Out[189]= 0.267601
```

Method 2: using the model, but approximating its integral with Simpson’s rule:

Note: *we’re going to be introducing you to the “Sum” command in Mathematica in this week’s lab, which will simplify these calculations when the number of subintervals becomes large.*

Now let's use Simpson’s rule to evaluate this integral, with 10 subdivisions. So we will need to evaluate the integrand (temp(t)) at the times 9, 10, ..., 19, and then weight them in the appropriate way. But recall that

$$S_{2n}=\frac{2M_n+T_n}{3}$$

We want S_{10}: so let’s compute Midpoint M_5 and Trapezoid T_5 approximations first. Remember, however, that Trapezoid Rule T_n is just the average of the left and right rectangle rules.

$$T_n=\frac{\text{Left}_n+\text{Right}_n}{2}$$

So, with n=5, we have the following:

a=9

b=19

n=5

$\Delta x = 2\ \left(=\frac{b-a}{n}\right)$

```
deltaX = 2;
LRR = deltaX (temp[9] + temp[11] + temp[13] + temp[15] + temp[17])
RRR = deltaX (temp[11] + temp[13] + temp[15] + temp[17] + temp[19])
trap = (LRR + RRR) / 2
trapApprox = trap / (-s / r)
```

```
Out[47]= 2
Out[48]= 7.4388
Out[49]= 6.31739
Out[50]= 6.8781
Out[51]= 0.268196
```

So the estimate from the trapezoidal method is about 0.268196. Let's move on to the midpoint method: the Δx=2, still, but the choice of data changes: we look at

```
In[53]:= mid = deltaX (temp[10] + temp[12] + temp[14] + temp[16] + temp[18])
         midApprox = mid / (-s / r)
Out[53]= 6.85521
Out[54]= 0.267303
```

The estimate from the midpoint rule is about 0.267303. Now we're ready for the computation of Simpson's, which will use all of the data, from temp[9] to temp[19]:

```
In[55]:= simp = (2 * mid + trap) / 3
         simpApprox = simp / (-s / r)
Out[55]= 6.86284
Out[56]= 0.267601
```

Hence our final (and hopefully best) estimate is Simpson's S_{10}, which is approximately 0.267601. Recall that, using the model and integrating it exactly we obtained 0.267601. Exactly the same thing!

Method 3: using the data directly, without using the model at all (other than for the computation of *s* and *r*, which are playing a role in the heat loss calculation).

We can proceed to Simpson's method using the data, rather than the function value calculations.

Again, let's compute Midpoint M_5 and Trapezoid T_5 approximations first. The only difference in this method is that, rather than computing using temp, we will be soliciting the data from the list of temps. Again,

```
In[190]:= temps
Out[190]= {1., 0.922268, 0.851204, 0.787027, 0.726199, 0.677621,
  0.6237, 0.571, 0.525478, 0.481791, 0.436774, 0.394208}
```

Notice how we access an element of a list in Mathematica -- again, we'll see more of this in the weeks ahead. Note that lists in Mathematica are indexed beginning at 1. So temps[[1]] is the first element in the list, corresponding to 9am.

```
In[191]:= deltaX = 2;
LRR = deltaX (temps[[1]] + temps[[3]] + temps[[5]] + temps[[7]] + temps[[9]])
RRR = deltaX (temps[[3]] + temps[[5]] + temps[[7]] + temps[[9]] + temps[[11]])
trap = (LRR + RRR) / 2
trapApprox = trap / (-s / r)

Out[192]= 7.45316

Out[193]= 6.32671

Out[194]= 6.88994

Out[195]= 0.268658
```

So the estimate from the trapezoidal method is about 0.268658. Now to the midpoint method:

```
In[196]:= mid = deltaX (temps[[2]] + temps[[4]] + temps[[6]] + temps[[8]] + temps[[10]])
midApprox = mid / (-s / r)

Out[196]= 6.87941

Out[197]= 0.268247
```

The estimate from the midpoint rule is about 0.268247. Now we're ready for the computation of Simpson's, which will use all of the data:

```
In[198]:= simp = (2 * mid + trap) / 3
simpApprox = simp / (-s / r)

Out[198]= 6.88292

Out[199]= 0.268384
```

Hence our final estimate is Simpson's S_{10}, computed from the data directly, which is approximately 0.268384. Recall that, using the model and integrating it exactly we obtained 0.267601. **Nearly** the same thing.

This method doesn't require the construction of a model, which makes it much simpler! Numerical integration techniques are particularly important when we have data -- often recorded at regular intervals, so Δx is constant -- but no "function" lurking in the background.

Many industrial processes produce data like this, or sensing software, etc. So these numerical routines are important!