LS30A - Lab 6

Aaron

Review about derivatives

1. **(12/73 points on Summer 2016 midterm) In mammals, resting metabolic rate M is related to body mass b as approximately M=0.8b*3/4.**

    a. **(6 points) Find the linear approximation to this function for a newborn black kitten with a body mass of 100 grams.**

    b. **(6 points) A different breed of black cats has an average mass of 110 grams. Use the linear approximation method to estimate how much its metabolic rate would change from the original black kitten. (You must use the linear approximation method to get credit for this problem.)**

Review about derivatives

a. **(6 points) Find the linear approximation to this function for a newborn black kitten with a body mass of 100 grams.**

b. (6 points) A different breed of black cats has an average mass of 110 grams. Use the linear approximation method to estimate how much its metabolic rate would change from the original black kitten. (You must use the linear approximation method to get credit for this problem.)

Review about derivatives

a. **(6 points) Find the linear approximation to this function for a newborn black kitten with a body mass of 100 grams.**

$$M'(b) = \frac{dM}{db} = \frac{3}{4} \cdot 0.8b^{-0.25}$$
$$M'(b = 100) = 0.190$$
$$\Delta M \approx M'(b = 100)\Delta b = 0.190\Delta b$$

b. (6 points) A different breed of black cats has an average mass of 110 grams. Use the linear approximation method to estimate how much its metabolic rate would change from the original black kitten. (You must use the linear approximation method to get credit for this problem.)

$$\Delta M \approx M'(b = 100)\Delta b = 0.190\Delta b$$
$$\Delta b = 110 - 100 = 10grams$$
$$\Delta M = 0.190 * 10 = 1.90$$

Note: students will not earn credit for simply calculating M(110)-M(100)since this doesn't use the linear approximation method. However, this is a great way to check their calculations since they should be fairly close!

Review about Lab6

1. Command: **soln = desolve_odeint(des , ics, times, dvars)**
2. **Solve a systems (Shark-Tuna)**
3. **Min and Max**
4. **Plot vector field**

```
var("S","T")
t = srange(0, 60, 0.1)
sol = desolve_odeint([0.005*S*T-0.2*S, 0.5*T-0.01*S*T], ics=[30, 40], times=t, dvars=[S, T])
list_plot(zip(t, sol[:,0])) + list_plot(zip(t,sol[:,1]), color='red')
```