SCIE1000, Solutions to Tutorial Week 4.

(1) This is going to be a rough model, so we will assume the hottest maximum daily temperature Brisbane experiences is 35 °C and the coolest maximum temperature is 20 °C. We will also assume the hottest month is January, as it is in the middle of summer. We will name our variable modelling the maximum daily temperature $T_D(t)$, and we will use a model of the form

$$T_D(t) = A\sin\left(\frac{2\pi}{B}(t+D)\right) + C,$$

where A is the amplitude of the oscillation, B is the period, C is the centre value and D is the phase (horizontal) shift.

Seasons repeat every 12 months, so the period, and thus B, should equal 12. We can use the formulae

$$A = \frac{Max - Min}{2} = \frac{35-20}{2} = 7.5 \qquad \text{and} \qquad C = \frac{Max + Min}{2} = \frac{35+20}{2} = 27.5$$

to find the parameters A and C.

Since January is month 0, then the peak of our model occurs at $t = 0$, however, the peak of a sine curve occurs after a quarter of a period. It appears as if the peak of our model has been shifted to the left by a quarter of a period. The period is 12 months, so a quarter of the period is 3 months, and a shift to the left means the parameter D is positive. So we have $D = 3$. Our model becomes

$$T_D(t) = 7.5\sin\left(\frac{2\pi}{12}(t+3)\right) + 27.5.$$

Note that the term $(t+3)$ could equally be $(t-9)$.

1. 1. The graph is roughly centred around 600000, with an amplitude of approximately 200000. Including $(t-8)$ seems to give a reasonable ‘phase shift’ in the model. Thus the (very rough) model is:

$$E(t) = 200000\sin\left(\frac{2\pi}{12}(t-8)\right) + 600000.$$

The model is not particularly accurate, but is reasonably defensible.

2. Electricity generation should be much lower in June for a range of reasons, including: days are shorter so less sunlight; the sunlight is less intense/strong in winter; temperatures are lower which can reduce generation efficiency. It is perhaps surprising that generation was much lower in December than October and November. Possible explanations include: perhaps December was much more cloudy or rainy than October and November (which it was), or maybe there was an equipment or measuring fault? For reference, in December 2016 Brisbane had 15 rainy days, but in November there were only 6 rainy days, and 4 rainy days in October.

(3) Loops are another important programming tool. A loop can be used to instruct the computer to perform certain calculations or operations repeatedly, until a certain condition is achieved. This condition can be, for example: a certain number of iterations, or when a particular value is ‘good enough’, or when the user inputs a ‘stopping value’.

(4) Program 1 output:

```
0
0.5
1.0
done
```

Program 2 output:

```
6 51
18 52
54 53
```

Program 3 output:

```
54 53
```

(5) 1. Here is the output.

```
1  o'clock
2  o'clock
3  o'clock
4  o'clock
5  o'clock
6  o'clock
7  o'clock
8  o'clock
9  o'clock
10  o'clock
11  o'clock
12  o'clock
We're gonna rock around the clock tonight.
```

2. Here is a solution.

```
i=1
while i <= 12:
    if i==3 or i==4 or i==7 or i==8 or i==11 or i==12:
        print(i," o'clock")
    else:
        print(i)
    i=i+1
print("We're gonna rock around the clock tonight.")
```

3. Here is a solution.

```
i=1
while i <= 12:
    if i==3 or i==4 or i==7 or i==8 or i==11 or i==12:
        print(i," o'clock ",end='')
    else:
        print(i,end=' ')
    if i==4 or i==8 or i==12:
        print("rock")
    i=i+1
print("We're gonna rock around the clock tonight.")
```

(6) This is a discussion question.

(7) 1. This is a discussion question.

2. This is a discussion question.

3. If average Brisbane daily temperatures were to rise by 4 °C, the model could reflect this by increasing the centre value, C, by 4 °C. The annual variation is twice the amplitude. If the annual variation were to decrease by 2 °C, then the amplitude, A, would only decrease by 1 °C. So our new model would be

$$T_D(t) = 6.5 \sin\left(\frac{2\pi}{12}(t+3)\right) + 31.5.$$

(8) 1. There appears to be a peak at Month 9 and the minimum of the trough (although not the centre of the minimum) is at Month 3, which are fortunately 6 months apart, so we will use the values to make our model. A sine graph has peak a quarter of the way through the period, and a trough 3 quarters of the way through the

year, which is the opposite to what we have in the graph. Hence our model needs to have a phase shift of one half of a cycle, so so our model will look like

$$P(t) = A \sin\left(\frac{2\pi}{B}(t-6)\right) + C.$$

The maximum value is 16 GW and the minimum value is about 4 GW, so

$$A = \frac{Max - Min}{2} = \frac{16-4}{2} = 6 \qquad \text{and} \qquad C = \frac{Max + Min}{2} = \frac{16+4}{2} = 10.$$

Finally, the period is a year, so $B = 12$ again. Our model becomes

$$P(t) = 6 \sin\left(\frac{2\pi}{12}(t-6)\right) + 10.$$

2. This is an activity question.

3. A. Less rain and snow means less water, which means less power. Assuming the drought is even over the whole year, the whole would graph shift down by some amount. The model could represent this by decreasing the value of the centre position, C.

   B. This would shift the whole graph to the right by some value, so $(t-6)$ might become $(t-8)$.

(9) 1. This can be solved with straight forward calculation.

$$B_1 = (1+r) \times B_0 = 1.08 \times 100 = 108$$

$$B_2 = (1+r) \times B_1 = 1.08 \times 108 = 116.64$$

2. Here is one possible solution

```
# A program to estimate the biomass of a fish species in a reef
from pylab import *

#The initial biomass
biomass=100

# monthly increase
r=0.08

i=0
while i<12:
    biomass=(1+r)*biomass
    i=i+1
    print("The biomass at month", i, "is", biomass, "kg.")
```

The output of this program agrees with Part 1.

```
The biomass at month 1 is 108.0 kg.
The biomass at month 2 is 116.64 kg.
The biomass at month 3 is 125.9712 kg.
The biomass at month 4 is 136.048896 kg.
The biomass at month 5 is 146.93280768 kg.
The biomass at month 6 is 158.687432294 kg.
The biomass at month 7 is 171.382426878 kg.
The biomass at month 8 is 185.093021028 kg.
The biomass at month 9 is 199.90046271 kg.
The biomass at month 10 is 215.892499727 kg.
The biomass at month 11 is 233.163899705 kg.
The biomass at month 12 is 251.817011682 kg.
```

3. Here is one possible solution

```
# A program to estimate the biomass of a fish species in a reef
from pylab import *

#Asks the user for the initial biomass
biomass=eval(input("What is the initial biomass of the species? (kg) "))

#Asks the user for the number of months
# the biomass should be calculated
months=eval(input("How many months do you want to calculate the biomass? "))

#The monthly population increase
r=0.08

i=0
while i<months:
    #Calculates the new biomass
    biomass=(1+r)*biomass
    i=i+1
    print("The biomass at month", i, "is", biomass, "kg.")
```

The output of this program is almost identical to the output of the program in Part 2.

(10) 1. A sine function achieves its maximum when its argument is equal to $\frac{\pi}{2}$. Solving

$$\frac{\pi}{2} = \frac{2\pi(i-9)}{12}$$

for i we get $i = 12$. This is not a number between 0 and 11, but because the period is 12, month 12 is the same as month 0. Furthermore, a sine function achieves its minimum when its argument is equal to $-\frac{\pi}{2}$. Solving

$$-\frac{\pi}{2} = \frac{2\pi(i-9)}{12}$$

for i we get $i = 6$. We can calculate the maximum and minimum growth rates

$$r_0 = 0.04 + 0.06\sin\left(\frac{2\pi(0-9)}{12}\right) = 0.1 \qquad \text{and} \qquad r_0 = 0.04 + 0.06\sin\left(\frac{2\pi(6-9)}{12}\right) = -0.02.$$

If January is month 0, then the highest growth rate occurs in summer, when there is plenty of sunlight to help plants grow, which in turn feeds the fish. This may also be the breeding season for the fish. The lowest growth rate occurs in winter, when food may be scarce. The negative growth rate is possible, as fish could be starving, or dying from the cold, which would cause the biomass to decrease.

2. This can be solved with straight forward calculation.

$$r_0 = 0.04 + 0.06\sin\left(\frac{2\pi(0-9)}{12}\right) = 0.1$$

$$r_1 = 0.04 + 0.06\sin\left(\frac{2\pi(1-9)}{12}\right) = 0.0920$$

3. Here is one possible solution

```
# A program to estimate the growth rate of the biomass of a species
from pylab import *

#Asks the user for the initial biomass
biomass=eval(input("What is the initial biomass of the fish species? (kg) "))

#Asks the user for the number of months
# the biomass should be calculated
```

```
months=eval(input("How many months do you want to calculate the biomass? "))

i=0
while i<months:
  # Calculates growth rate
    r=0.04+0.06*sin(pi*2*(i-9)/12)

    print("The growth rate at month", i, "is", r)
    i=i+1
```

The output of this program agrees with 2.

```
The growth rate at month 0 is 0.1
The growth rate at month 1 is 0.0919615242271
The growth rate at month 2 is 0.07
The growth rate at month 3 is 0.04
The growth rate at month 4 is 0.01
The growth rate at month 5 is -0.0119615242271
The growth rate at month 6 is -0.02
The growth rate at month 7 is -0.0119615242271
The growth rate at month 8 is 0.01
The growth rate at month 9 is 0.04
The growth rate at month 10 is 0.07
The growth rate at month 11 is 0.0919615242271
```

4. For a months larger than 11, the values for the growth rate above will repeat, matching those for 12 months earlier. For example, Month 21 will be the same as Month $21 - 12 = 9$.

(11) Here is one possible solution

```
# A program to estimate the biomass of a fish species in a reef
from pylab import *

#Asks the user for the initial biomass
biomass=eval(input("What is the initial biomass of the fish species? (kg) "))

#Asks the user for the number of months
# the biomass should be calculated
months=eval(input("For how many months do you want to calculate the biomass? "))

i=0
while i<months:
    # Calculates the growth rate for this month
    r=0.04+0.06*sin(pi*2*(i-9)/12)

    #Calculates the new biomass
    biomass=(1+r)*biomass
    i=i+1

# Prints the final biomass of fish
print("The biomass at month", i, "is", biomass, "kg.")
```

The output of this program when used to find biomass of a population of fish after 24 months, for which the initial biomass is 500 kg, is as follows.

```
What is the initial biomass of the fish species? (in kg) 500
For how many months do you want to calculate the biomass? 24
The biomass at month 24 is 1256.27946671 kg.
```

(12) 1. Given $W(t, v) = 13.12 + 0.6251t - 11.37v^{0.16} + 0.3965tv^{0.16}$, we substitute in the parameter $v = 76.1$, and obtain:

$$\begin{aligned} W(t, 76.1) &= 13.12 - 11.37(76.1)^{0.16} + 0.6251t + 0.3965t(76.1)^{0.16} \\ &= 13.12 - 22.7396 + 0.6251t + 0.793t \\ &\approx -9.62 + 1.41t \end{aligned} \tag{1}$$

2. We wish to find a temperature, T, such that when the wind speed is 76.1 km/h, an increase in the ambient temperature of 5 degrees Celsius results in the wind-chill doubling. Mathematically, this corresponds to solving $W(T + 5, 76.1) = 2 \times W(T, 76.1)$. Using our approximation we derived in Equation 1, we get:

$$\begin{aligned} -9.62 + 1.41(T + 5) &= 2 \times (-9.62 + 1.41T) \\ -9.62 + 1.41T + 7.05 &= -19.24 + 2.82T \\ -9.62 + 7.05 + 19.24 &= 2.82T - 1.41T \\ \therefore\ T &= \frac{16.67}{1.41} = 11.8227 \end{aligned}$$

Therefore, at an ambient temperature of 11.8 degrees Celsius, and a wind-speed of 76.1 km/h, an increase in ambient temperature of 5 degrees (to 16.8) will result in the wind-chill temperature doubling.

(13) 1. First the researchers calculated c from their existing data, S_0 and A_0, using the estimate $z = 0.25$. To do this they rearranged the equation $S_0 = cA_0^z$ for c and then calculated

$$c = \frac{S}{A^z} = \frac{37}{17274^{0.25}} = 3.227.$$

Now they are able to use the same equation to find S_2

$$S_2 = cA_2^z = 3.227 \times 151^{0.25} = 11.3135.$$

As species are discrete quantities, this should be rounded. This gives the answer the researchers got, $S_2 = 11$.

2. Rearranging the equation for A gives $A = \sqrt[z]{\frac{S}{c}}$. Again, using $z = 0.25$ gives $A = \sqrt[0.25]{\frac{S}{c}} = \frac{S^4}{c^4}$. The area can be found by subbing in the remaining values $A = \frac{5^4}{3.227^4} = 5.76\,\mathrm{km}^2$. It seems that almost $6\,\mathrm{km}^2$ is too small for any species of vertebrate to comfortably be restricted to, let alone 5 different species. This model appears to break down at low values of S.

3. See Figure 1. Of course, your answer will be hand-drawn.

(14) The first piece of code is not an infinite loop; it produces the integers from 1 to 10, then stops. The loop is not infinite because the Boolean test is for $j \leq 10$, and within the loop, j increases by 1 each time.

The second piece of code forms an infinite loop. The problem is that 20 is not a power of 2, and so a will never equal 20.

(15) Here is a solution:

```
t=10
while t>=0:
    if t==3:
        print(t, "Ignition")
    elif t==0:
        print t,("We have lift off!")
    else:
        print(t)
    t=t-1
```

Here is the output.

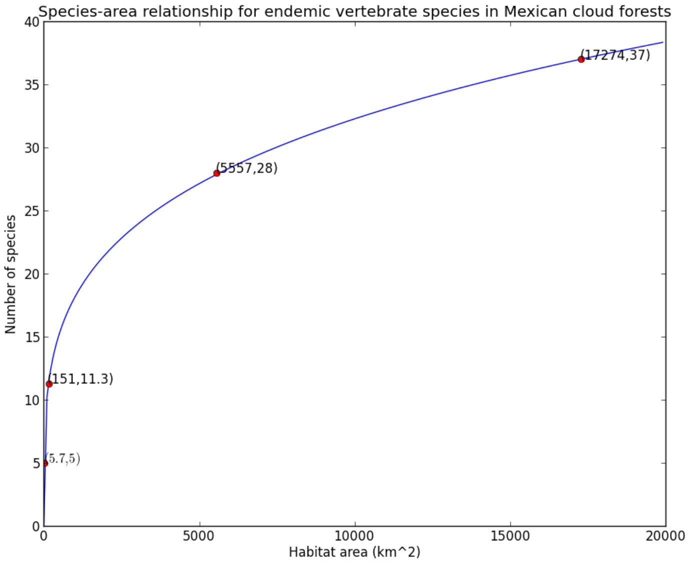
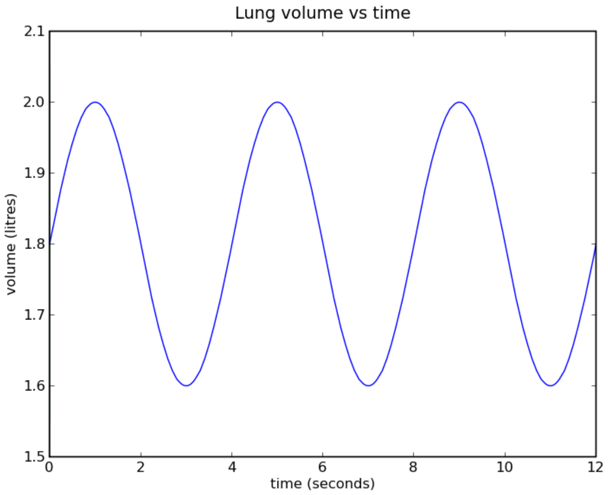


Figure 1: *A graph of the species area curve for vertebrates in Mexican cloud forests. Points calculated in Parts 1 and 2 are in red.*

```
10
9
8
7
6
5
4
3 Ignition
2
1
0 We have lift off!
```

(16) (Note that your graph may be hand drawn, of course. We have shown more than two full inhalation/exhalation cycles; you only needed to show two cycles, each taking 4 seconds.)



1.

2.

$$V(t) = 0.2\sin\left(\frac{2\pi t}{4}\right) + 1.8$$

3.
$$V(t) = 0.3\sin\left(\frac{2\pi t}{6}\right) + 1.9$$

4. Consider the equation
$$V(t) = A\sin\left(\frac{2\pi t}{B}\right) + C$$
Changing the value of A will vary the amplitude of the function, reflecting changes in the tidal volume. Changing the value of B will vary the frequency of the function, reflecting the length of time over which a breath occurs. As the value of A changes, the value of C will need to change to keep the same residual volume.

(17) First, we need to calculate the time for Neptune to orbit the sun, in Neptune days. The orbit time is 165 earth years × 365 earth days per earth year × 24 earth hours per earth day ÷ 16 earth hours per Neptune day, which is about 90337 Neptune days.

Hence the amount of daytime $D(t)$ at locations on the surface of Neptune is given by the equation:

$$D(t) = 8 + K \times \sin\left(\frac{2\pi t}{90337}\right)$$

where K is a constant that depends on the latitude of the location.

(18) Here is a possible solution;

```
# Scissors, Paper, Rock
from pylab import *

choice = eval(input("Enter: 1 for scissors, 2 for paper or anything else for rock: ")
computer = randint(1,4)
if computer == 1:
    print("Computer chooses scissors")
elif computer == 2:
    print("Computer chooses paper")
else:
    print("Computer chooses rock")

if choice==computer:
    print("Match is tied!")
elif choice==1:
    if computer==2:
        result = 1
    else:
        result = 2
elif choice==2:
    if computer==1:
        result = 2
    else:
        result = 1
elif computer==1:
    result = 1
else:
    result = 2

if result == 0:
    print("Game is tied!")
elif result == 1:
    print("You win this game!")
```

```
else:
    print("Computer wins this game!")
```

(19) Here is a possible solution;

```
# Scissors, Paper, Rock
from pylab import *

i=0
CompWin=0
PlayWin=0
while(i<5):
    choice = eval(input("Enter: 1 for scissors, 2 for paper or anything else for rock
    computer = randint(1,4)
    if computer == 1:
        print("Computer chooses scissors")
    elif computer == 2:
        print("Computer chooses paper")
    else:
        print("Computer chooses rock")

    if choice==computer:
        print("Match is tied!")
    elif choice==1:
        if computer==2:
            result = 1
        else:
            result = 2
    elif choice==2:
        if computer==1:
            result = 2
        else:
            result = 1
    elif computer==1:
        result = 1
    else:
        result = 2

    if result == 0:
        print("Game is tied!")
    elif result == 1:
        print("You win this game!")
        PlayWin = PlayWin + 1
    else:
        print("Computer wins this game!")
        CompWin = CompWin + 1
    i=i+1
print()
print("Overall summary: ")
print("    You won ",PlayWin,"times, and the computer won",CompWin,"times.")
if PlayWin > CompWin:
    print("    You are the overall winner!")
elif PlayWin < CompWin:
    print("    The computer is the overall winner!")
else:
    print("    The overall result was a tie!")
```