**BIOL5301 Fruit fly Project**

This project is about designing optimal surveillance strategies for detecting incursions of fruit flies. Fruit flies are a serious biosecurity threat to Australia (and other countries). In areas where fruit fly species are established, they cause huge damages to crop yields and cost a lot of time and money to manage and control. Many areas are still free of many fruit fly species, and we want to keep it that way! For example, Perth is currently free of the very damaging Queensland fruit fly, and Kununurra in the north of WA is still free of the Mediterranean fruit fly that has spread through many areas of the south of WA. State governments have surveillance programs that aim to detect new incursions of fruit fly species as quickly as possible, so that they can then be eradicated before populations build up too far and become fully established. These surveillance programs involve placing fruit fly traps in specific locations. These traps use lures that attract fruit flies into the trap from some distance, and these traps are then checked for fruit flies at regular intervals. For more information on fruit flies in Australia, see the following links:

<https://www.perthnow.com.au/community-news/western-suburbs-weekly/queensland-fruit-fly-infestation-hits-dalkeith-veggie-patches-c-962123>

<http://www.watoday.com.au/wa-news/qfly-outbreak-eight-perth-suburbs-in-quarantine-zone-20151201-glcvz2.html>

<https://www.agric.wa.gov.au/plant-biosecurity/queensland-fruit-fly-qfly-eradicated-perth-suburb>

<https://www.agric.wa.gov.au/citrus/fruit-fly-western-australia>

<https://www.agric.wa.gov.au/plant-biosecurity/queensland-fruit-fly>

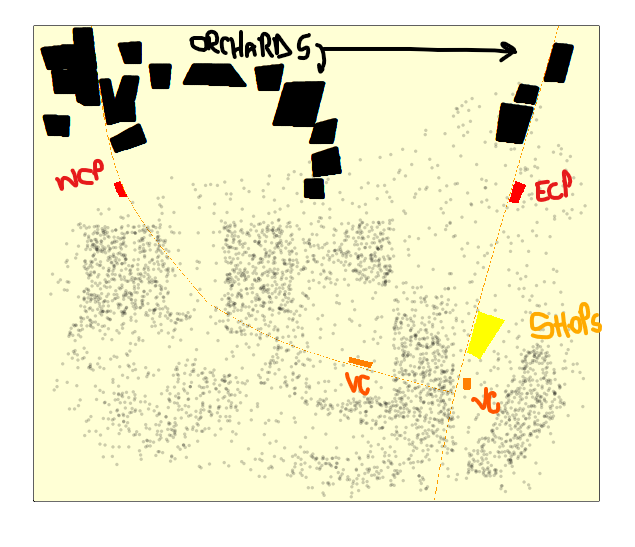
<https://www.agric.wa.gov.au/fruit/mediterranean-fruit-fly>

<http://www.planthealthaustralia.com.au/national-programs/fruit-fly/>

In this project, you will use a simulation model to evaluate different designs for a fruit fly surveillance system aimed at detecting new incursions of fruit fly in a small Australian fruit-growing town.

All the files you need are in the zipped folder. Download the folder and extract it to a suitable location on your computer (or the lab computer).

**First,** open and have a look at the ‘map.pdf’. It shows a map of the town. It is 1000x1000 pixels, with each pixel representing an area 10m by 10m. You can zoom in and look more closely. The state government has mapped the fruit trees in the town, and the locations of these trees are indicated on the map by **small grey semi-transparent dots** (each tree is at the centre of a dot). Note that some suburbs in the town have higher densities of fruit trees; these are the older more established suburbs. **The dark black areas represent orchards**, where there are a lot of these small grey semi-transparent dots close together and overlapping. Fruit flies do not fly long distances by themselves, and mainly spread between towns when people carry infested fruit. The local experts have noticed that past fruit fly incursions have tended to begin close to caravan parks, presumably because this is where tourists passing through town tend to stop and dump their rubbish (presumably including infested fruit). Experts have thus estimated that a new incursion is most likely to start near a caravan park (50% probability), or a visitor centre (20% probability), or the shopping area (12% probability), or along a main road (10% probability), with a remaining 8% chance of starting anywhere else in town. These high introduction risk locations are also shown on the map: the main roads through town (orange lines), caravan parks (red), visitor centre (orange), and shopping area (yellow).



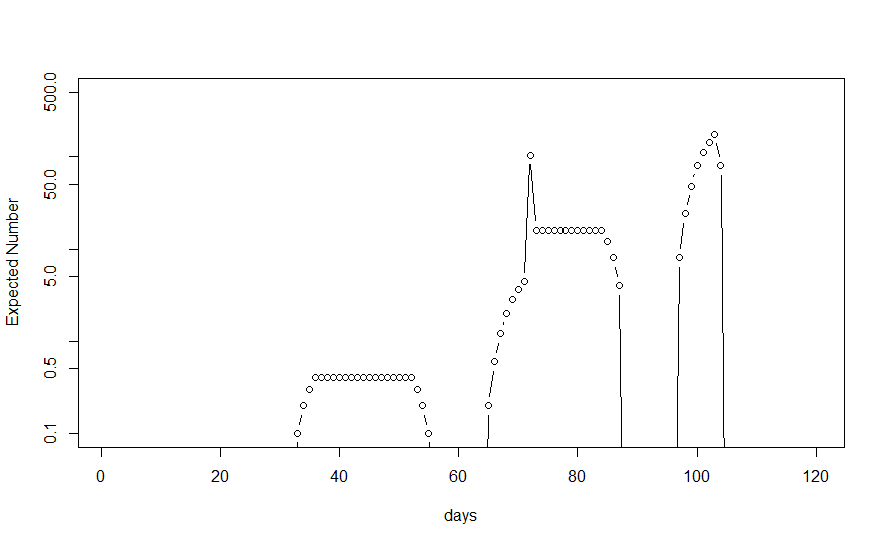
**Next open R and then change the working directory to the extracted folder, and then open the ‘fruit fly pop 2.R’ script within R.**

This script runs a simple fruit fly population model to estimate how many female flies are likely to be leaving an infested fruit tree on any given day after the first female fly arrives at the tree.

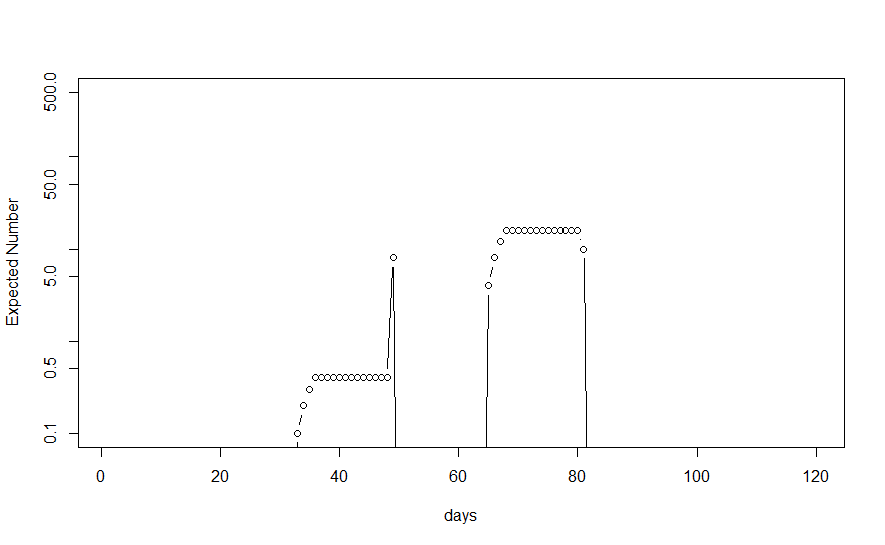
It accounts for the fruit fly life cycle, which means an individual spends a certain number of days as an egg, then as a larva, then as a pupa, then as a flying adult.

It also accounts for the number or size of trees at a location, and thus the number of fruits to infest.

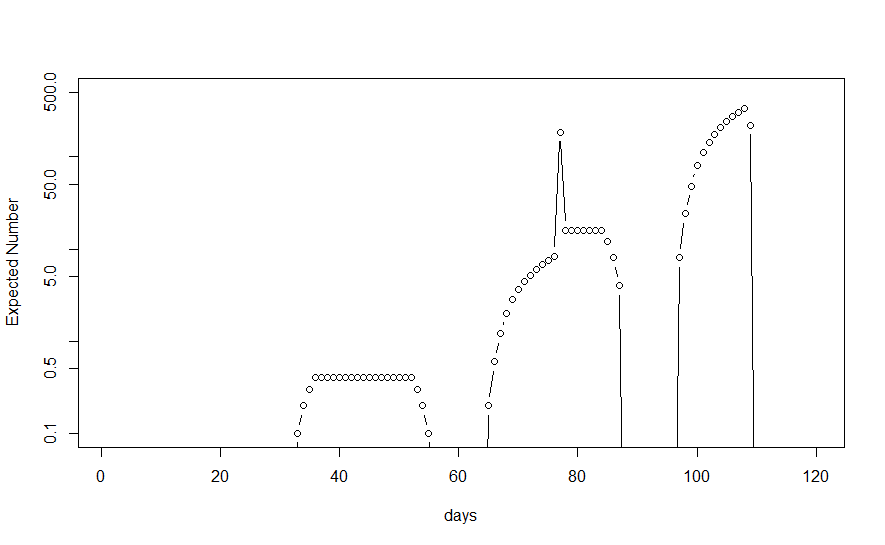
1. First run the whole script to generate and save the population predictions.
2. Then plot some of them. Look for the line ‘ntrees=4’ towards the end and enter the following plot command.
   1. You should see a plot of the expected number of female fruit flies that will be leaving an infested trees over time, following first infestation. Note how the lifecycle of the fruit fly leads to pulses of emigration as generations mature.



1. Change the ‘ntrees=4’ to ‘ntrees=1’ and replot to see what happens if there was just one small tree at a location.



1. Change it to ‘ntrees=10’ to see what happens if there were ten small trees at a location (or equivalently one very large one).



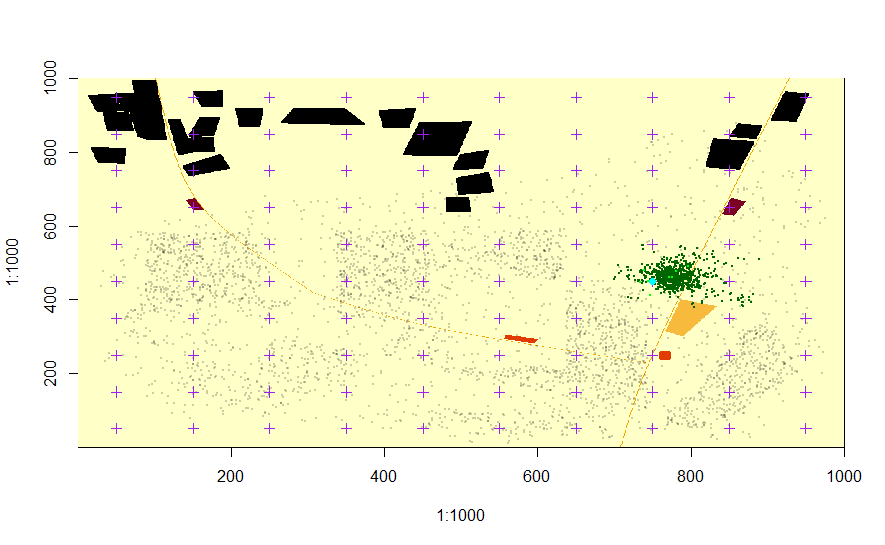
**Next, open the script ‘calc distances.R’ and run it.**

This will take a while, so be patient. It is calculating the distances between all the trees in the town, for use in the simulations of spread later on. It takes quite a lot of time to do these calculations now in advance, but it saves a lot of time later on when the simulations run. When it finishes, you should have a new data file called ‘allcloseones’ in your folder. This is quite a large file (~40MB), so make sure you have room for it on your disk or drive.

**Next, open the script ‘run sim.R’ and have a look at it**

* Note that there is a line in the script that allows you to specify the ‘luredist’; this is the maximum distance in metres from which a lure is able to attract a fly.
* Also note that it reads in a surveillance trap design file called ‘design\_grid.csv’.
* And then note that it loads a number of data files at the start, including the tree locations, the tree distances we just calculated, the fruit fly population estimates we calculated earlier, and also the introduction risk map.
* You should also notice the ‘distwtfunc’ function definition… this is the dispersal kernel used in the model. A dispersal kernel is a probability density function showing how the likelihood of a fly spreading a certain distance declines as the distance increases.
* Run those lines to get a plot of the dispersal kernel.
* Now run the whole script. Make sure you are using an interface that allows the animation to show… RStudio may not work, but the basic interface in Mac and Windows should work fine, as should running from a terminal.
  + You should see a map of the town and green dots spreading across the town. Light green dots indicate infested fruit trees, and dark green dots indicate the position of male fruit flies.
  + If you look carefully at the start, you should also see the initial infestation tree, and then observe how the invasion progresses from there.
  + The initial infestation tree is likely to be near a caravan park or visitors centre, or maybe a road or the shopping area, but not always of course.
  + Sometimes when there are many infected trees around the start point, it might not be visible (eg. when it starts in an orchard).
  + Eventually a male fly gets close enough to one of the traps (the purple crosses) and the invasion is detected. The trap that detected the fly is then shown with a light blue dot, and if you look at the output in the R console you can see how many days it took to detect the invasion.

**Simulation 1 –** Infestation detected at y= 450, x = 750 on day 75



Run the model at least 10 more times, watching carefully how the invasion unfolds each time.

1. Detected day 75 – Shopping Centre
2. Detected day 74 - West Caravan Park
3. 75 – West Caravan Park
4. 68 – Shopping Centre
5. 73 – West Caravan Park
6. 73 – East CVP
7. 60 – Shopping Centre
8. 70 – East CVP
9. 79 – Other visitor centre?/ shopping area
10. 77 – SE of Orchard (just NE of centre of map)

As you watch, you could start thinking about how YOU would design a surveillance system to detect an invasion like this as quickly as possible. Where would you place the fruit fly traps?

* Around Shopping Centre, Visitor Centre & Caravan Parks, and a few along the main roads

Would it be different if you wanted a surveillance system to detect an invasion with as few trees infected as possible?

* Grid system like that in this experiment for fast detection & eradication – Not different in either of thse

**To start to address this question, open the ‘run many sims.R’ script**.

This is basically the same as the previous one, except it runs a large number of spread simulations one after the other automatically. Its graphical output is a bit different. It doesn’t show the flies; instead it shows each initial incursion tree as a dark blue dot and the trap that detects the incursion as a larger light blue dot. It also saves the results to data files.

1. Check that the number of **runs is set to 20** at the start of the file, and then run the script. It should take a little while, but not too long. After it has run, you should see that it has produced an output file called ‘results design\_grid.csv’.
2. Open this file in Excel and have a look at it. You should see the results for the 20 runs; for each run you should see the number of days until detection (time), the trap.id of the trap that made the detection, the number of male flies at the time of detection, and the number of trees infested at the time of detection.

**Now open the script ‘make grid design.R’ and run it**.

This is a very short script that makes a few simple surveillance trapping designs, and saves them to csv files.

* Open the ‘design1.csv’ file in Excel and have a look at it.

**Now open the ‘show design.R’ script, and run it**.

1. You should see the trap locations from 'design1.csv' shown on the town map.
2. Change one of the trap locations in 'design1.csv' and save that file and then run the ‘show design.R’ script again.
3. You should see that the trap location has changed.
4. Now change the name of the file at the start of the ‘show design.R’ script so that you can see the other designs that were created with the ‘make grid design.R’ script.

Your job is to compare the efficacy of a number of surveillance strategies. To do that properly, you need to do a large number of spread simulations for each design (much more than 20). This is because you need to capture the full range of possibilities of how a new invasion will unfold in space and time. You should aim to do 1000 or even 5000 for each design, but be aware that this could take quite a long time to run, depending on how powerful your computer is. Maybe try to do 100 simulation runs at first and time how long it takes, so you have an idea of how long to do a 1000 or 5000.

You should then get your surveillance strategies represented in csv files. Some of the surveillance designs/strategies csv files are provided for you and some you need to create yourself. You can create these using R, or directly in Excel, but either way you should end up with each design represented in a separate csv file.

1. **Create Surveillance Designs in .csv files**
   1. Standard square grid (design\_grid)
   2. Reduced density square grid (25 traps)

The first design should be the standard square grid we have already created.

The second design should be the same standard square grid, but with a reduced density ie only 25 traps in total.

All other designs should have the same number of total traps as the first design ie 100.

You should next create three different ‘random’ designs, where each of the 100 traps is placed at random within the simulated area.

Two more designs are provided for you, by a team of experts who think these designs will outperform the square grid and random designs.

And then it’s your turn - you should create one design that you think may be more efficient than the square grid and random designs, and one more design that you think will be less efficient than the square grid and random designs.

You should thus end up with 9 surveillance strategies (2 grids, 3 random, 2 provided by experts, and 2 created by you), with each design represented in a separate csv file.

Now you need to run the simulations for each design. These will take time to run! You might try to set up some R loops perhaps to automate running multiple designs overnight. Or you could try to save some time by opening R a few times and running simulations for a different design in each instance of R (this will allow you to use multiple cores/processors on your computer, which should speed things up). If you usually use RStudio, then you should switch to the basic R interface for running the simulations, or run them from the command line. It seems RStudio can be much much slower for intensive computation – with people reporting simulations that take days to run in RStudio can take only hours to run outside RStudio.

You should then compare your surveillance designs in terms of the time to first detection. This takes some thought, because you have 1000, or 2000 or even 5000 times to first detection for each design, representing a full range of possible outcomes for that design. You should compare the designs in terms of the mean time to first detection, but you should also consider other criteria. For example, you could consider the 90% and 99% ‘worst-case’ scenarios. (For 99% this would mean the time by which we can be 99% sure that the invasion will be detected, or in other words, the time by which detection has occurred in 99% of our spread simulations.) If possible, you could compare the whole distribution of possible outcomes for each design as well.

For example, you could create and compare histograms of the results, or probability density estimates, or cumulative probability curves.

You could present results graphically, and/or summarised in tables, but it may also be appropriate to do some kind of statistical test as well. For some ideas on presenting results, see the presentation on fruit fly models in this unit, and the paper by Triska et al 2018, “Accounting for spatially heterogeneous conditions in local‐scale surveillance strategies: case study of the biosecurity insect pest, grape phylloxera (*Daktulosphaira vitifoliae* (Fitch))”.

Once you have done that all for time to first detection, you should then also evaluate and compare the surveillance designs in terms of the number of trees infested at first detection in a similar way.

You may find that one design is the best in all regards, or you may find that different designs are best according to different criteria.

Completing all the directions above will be enough to pass and get a good mark, but for top marks, you could also explore how the efficacies of your surveillance designs depend on characteristics of the flies and/or the traps in some way. For example, you could consider one of the following questions as well.

1. How do the efficacies of your surveillance designs depend on the trap lure strength? What if the lure distance was 10m instead of 20m? What if it was 30m instead of 10m? Does it change which strategy is the best?
2. If the standard grid design with 100 traps is the one being used currently, can we save money by achieving the same detection efficacy with fewer traps, by using a smarter design? What is the minimum number of traps required to match the efficacy of the current standard grid design with 100 traps?
3. How do the efficacies of your surveillance designs depend on the biological characteristics of the fruit fly? What if the flies:
   1. laid fewer or more eggs each day?
   2. spent shorter or longer if each life stage?
   3. dispersed further, or not so far?

Does it change which strategy is the best?

1. How do the efficacies of your surveillance designs depend on the size of the fruit trees ie the number of fruits they hold? Does it change which strategy is the best?
2. How do the efficacies of your surveillance designs depend on the number of the fruit trees? Someone might propose removing every second fruit tree in the town, because it would help detect new fruit fly incursions more quickly. Would the model agree? Does it change which strategy is the best?

These are just suggestions – you might also think of another question that you wish to explore with the model.

You will submit a report for this project, which should be no more than 1500 words (not including tables and figure captions). It should be concise, clear and accurate – if you can present a clear and accurate description of a sound study in less than 1500 words then please do! The report should have four sections: Introduction, Methods, Results and Discussion.

* The Introduction should very briefly introduce the topic and the issue and finish by clearly stating your aims. A paragraph or at most two should suffice. You do not need to give as much background in this report as you normally would, as everyone reading it will know what the project is about.
* The Methods section should clearly define and show exactly what strategies/designs you tested and explain exactly how you tested them. A brief explanation of how or why you created your own strategies would be useful. Note that ideally it should be possible for someone else to do exactly what you did, based on your Methods.
* The Results section should present the results using figures, tables and/or text, as appropriate. It does not have to be long; you may just need a paragraph or two of text referring to the figures/tables. You should also present any results of statistical tests of difference clearly and concisely – a few p-values in the text or within a table or figure is probably enough, definitely do NOT include whole ANOVA tables or anything like that.
* Discussion should include your main conclusions, along with thoughts about what you thought was interesting or surprising or exactly as expected, possible implications, discussion about the limitations of what you did (or of the model itself), and what you would like to do next or would recommend others to do next if the study was to continue.

You are welcome to work with other people as much (or as little) as you like. But you must write and submit your own report, and you should make sure you consider at least two completely different strategies of your own. If you have time to explore how the efficacies of your surveillance designs depend on characteristics of the flies and/or the traps, then the way you do this should not be the same as the people you are working with. Cases where reports or all tested strategies are very similar will be penalised. On average, you should plan to spend at least 18-20 hours total on the project and writing the report. It may take more time depending on how fast you work of course.

I will create a specific forum for questions and discussion about the fruit fly project. Please use this in preference to direct email if you think your question may have any general relevance, so that everyone has access to questions and replies.

The assessment guidelines to be used for the project are below. I will apply these based very closely on the specific directions above.

**Assessment Guidelines:** In general a mark of 80% should be awarded when an assessment or part of an assessment clearly meets all requirements. Extra marks can be awarded for extra insight or originality, up to a possible 100% for outstanding work. Marks can be lost for flaws, errors, omissions, lack of clarity etc. So for example:

less than 50% – doesn’t meet requirements

50% – barely meets requirements, with some fairly major flaws, omissions, lack of clarity

60% – meets the basic requirements, but with minor flaws and lack of clarity, or with one or two more major flaws

70% – meets requirements but with minor flaws or lack of clarity

80% – clearly meets all requirements

90% – clearly meets all requirements, with some extra insight or originality

100% – very clearly meets all requirements, with significant extra insight and originality