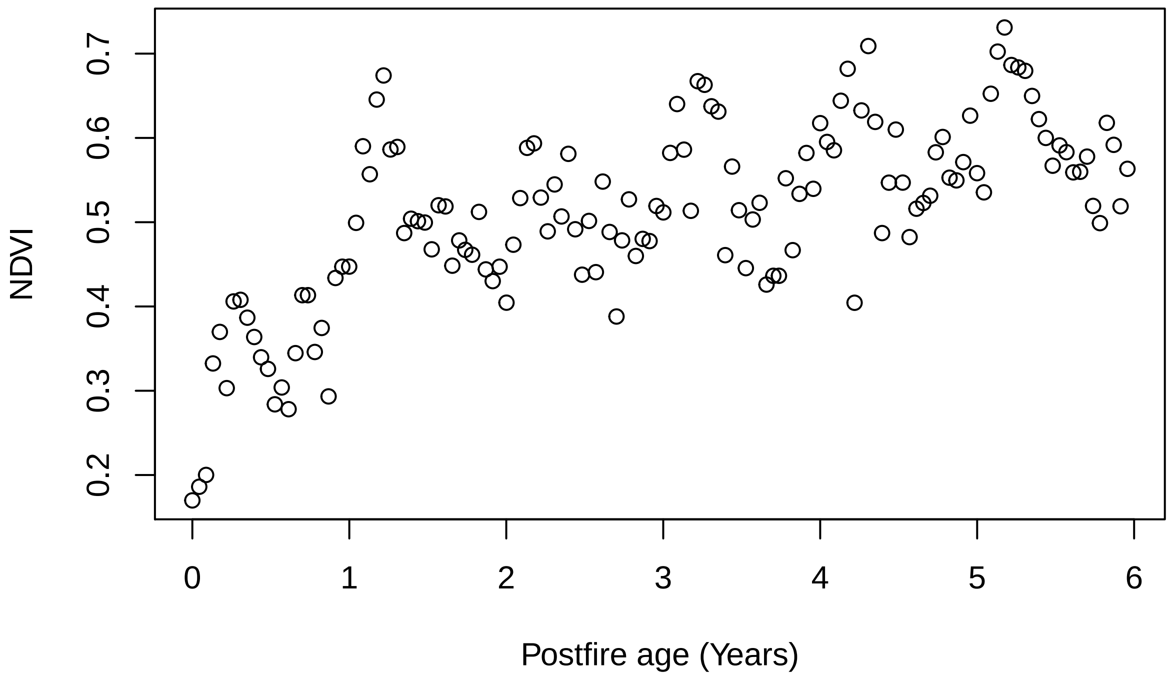
Master script for postfire analysis

1. Source functions, get data and plot

First we'll *source()* (i.e. "run all code in") the scripts with the functions we made. Then we'll set the URL, read in the data with *download.NDVI()*, and plot it with *plot.NDVI()*.

```r
## Load required functions by running source() on the individual function files
if(file.exists("01_download.NDVI.R")) source("01_download.NDVI.R")
if(file.exists("02_plot.NDVI.R"))     source("02_plot.NDVI.R")
if(file.exists("03_negexp.R"))        source("03_negexp.R")
## Download NDVI data
URL = "https://raw.githubusercontent.com/jslingsby/BIO3019S_Ecoforecasting/master/data/modisdata.csv"
dat <- download.NDVI(URL)
# Convert "calendar_date" to postfire age in days since fire - assuming the first date in the times eri
dat$age <- (as.numeric(dat$calendar_date) - min(as.numeric(dat$calendar_date), na.rm = T))/365.25
## Plot overall NDVI time series
plot.NDVI(dat)
```



Q1: This plot suggests that Fynbos greenness (NDVI) as observed from satellite saturates with time since fire. Why do you think it saturates rather than increasing linearly with time?

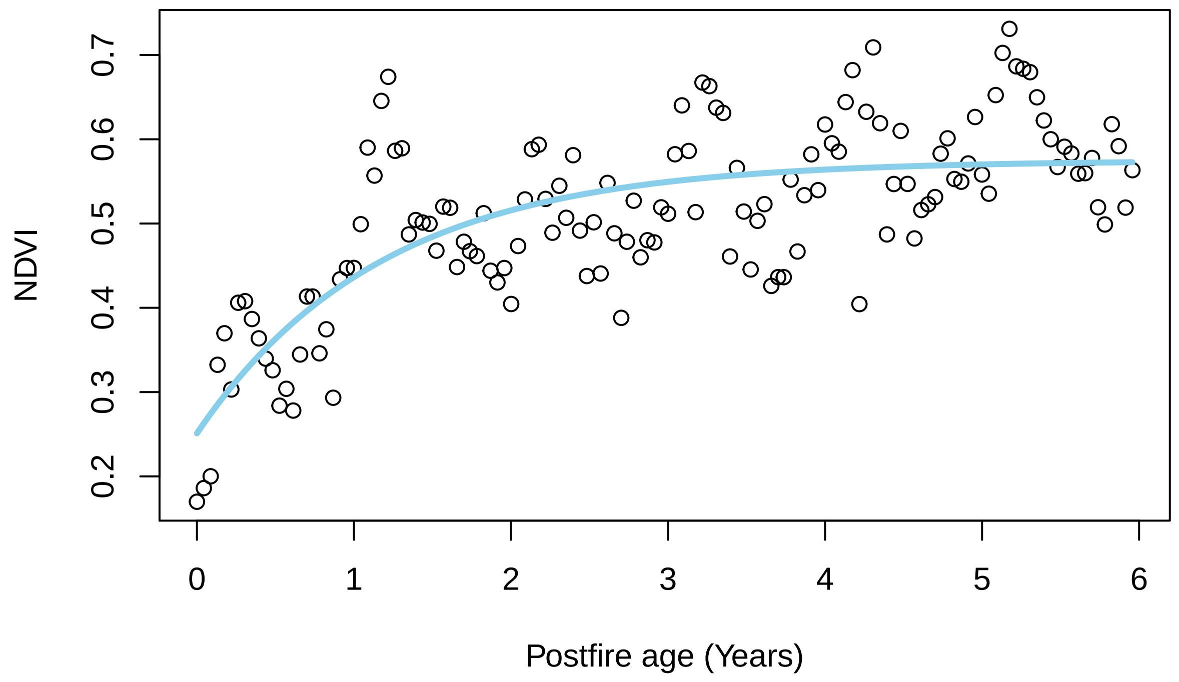
Answer 1: I think NDVI doesnt increase linearly with time since fire, because fynbos establishes quite quickly initially post fire (e.g. adaptations such as resprouting and reseeding) and thus has a rapid increase in NDVI. The change in the plot steepness could be due to the NDVI saturating over denser vegetation. After an older post fire age, the ground cover of fynbos might remain similar (most of the area is covered already) and thus not show an increase in NDVI similar to that seen at a young post fire age. Another possible reason for the non-linearity and saturation could be because there is environmental variation in the fynbos landscapes. This plot shape might be due to differences in post fire succession that causes a non linear/saturting increase in NDVI. Every year post fire there will be different environmental, seasonal, anthropogencic etc. conditions that will influence the NDVI each year. Thus, some form of non linear increases would be expected.

2. Fit models using Non-linear Least Squares (NLS)

Now we'll fit the simple and full negative exponential models using Non-linear Least Squares (NLS).

First the simpler model:

```r
## Simple model
# set parameters
par <- c(alpha = 0.2, gamma = 0.4, lambda = 0.5)
# fit model
fit_negexp <- nls(NDVI ~ alpha + gamma * (1 - exp(- age/lambda)),
                  data = dat, start = par, trace = F,
                  control = nls.control(maxiter = 500))
# plot
plot.NDVI(dat = dat, fit = fit_negexp)
```



And let's look at the model summary with parameter estimates

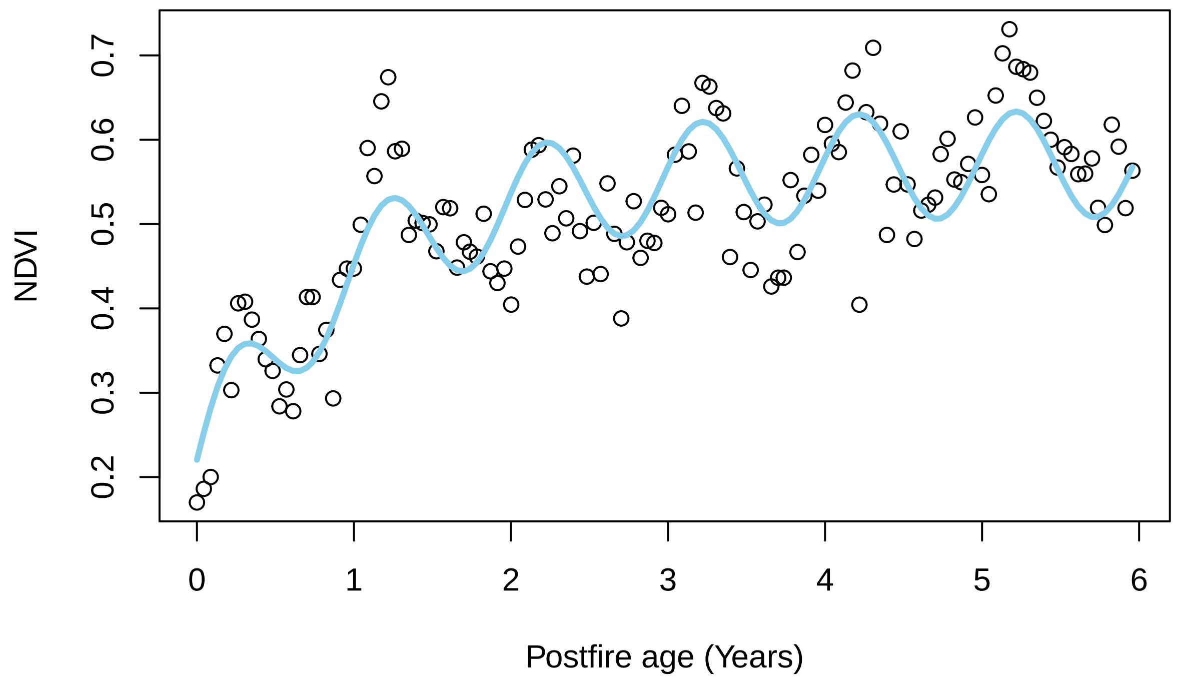
```r
# print model summary
summary(fit_negexp)
```

```
##
## Formula: NDVI ~ alpha + gamma * (1 - exp(-age/lambda))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## alpha   0.25107    0.02887   8.695 1.04e-14 ***
## gamma   0.32371    0.02723  11.887  < 2e-16 ***
## lambda  1.17687    0.21396   5.500 1.84e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07302 on 135 degrees of freedom
##
## Number of iterations to convergence: 12
## Achieved convergence tolerance: 3.924e-06
```

Now the full model:

```r
## Full model
# set parameters
```

```r
par <- c(alpha = 0.2, gamma = 0.4, lambda = 0.5, A = 0.6, phi = 0)
# fit model
fit_negexpS <- nls(NDVI ~ alpha + gamma * (1 - exp(- age/lambda))
                   + A*sin(2*pi*age + (phi + pi/6*(3 - 1))),
                   data = dat, start = par, trace = F,
                   control = nls.control(maxiter = 500))
# plot
plot.NDVI(dat = dat, fit = fit_negexpS)
```



```r
# print model summary
summary(fit_negexpS)
```

```
##
## Formula: NDVI ~ alpha + gamma * (1 - exp(-age/lambda)) + A * sin(2 * pi *
##     age + (phi + pi/6 * (3 - 1)))
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
## alpha   0.207522   0.024948   8.318 9.31e-14 ***
## gamma   0.364746   0.023926  15.245  < 2e-16 ***
## lambda  0.989154   0.126064   7.846 1.25e-12 ***
## A       0.063136   0.007114   8.875 4.12e-15 ***
## phi    -0.839167   0.111887  -7.500 8.10e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.05835 on 133 degrees of freedom
##
## Number of iterations to convergence: 15
## Achieved convergence tolerance: 6.927e-06
```

Lots more parameters…

Q2: How do the estimates for the common parameters compare?

> *Answer 2:* Both the alpha and lambda estimates decrease in the full model, compared to the simple model (0.25 to 0.21; 1.18 to 0.99 respectively). In contrast, Gamma increases from the simple model to the full model (0.32 to 0.36). Therefore, gamma and lambda changed by less than alpha.

3. Compare NLS models using ANOVA

Modelers often want to know which of a set of models are better. One way to do this when comparing nested* models using least squares is using analysis of variance (ANOVA). In this case the `anova()` function will take the model objects as arguments, and return an ANOVA testing whether the full model results in a significant reduction in the residual sum of squares (and thus is better at capturing the data), returning an F-statistic, Degrees of Freedom (the difference in the number of parameters between the models) and p-value.

*i.e. one model is a subset of the other, as in our case

```r
anova(fit_negexp, fit_negexpS)
```

```
## Analysis of Variance Table
##
## Model 1: NDVI ~ alpha + gamma * (1 - exp(-age/lambda))
## Model 2: NDVI ~ alpha + gamma * (1 - exp(-age/lambda)) + A * sin(2 * pi * age + (phi + pi/6 * (3 - 1
##   Res.Df Res.Sum Sq Df  Sum Sq F value   Pr(>F)
## 1    135    0.71976
## 2    133    0.45280  2 0.26696  39.207 4.12e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q3: Which model is better?

> *Answer 3:* The results suggest that the full model is better at capturing the data than the simpler model. This is due to the full model resulting in a significant reduction in the residual sum of squares (RSS) (reduction from the simple model's RSS). i.e. the p-value is lower than 0.05 which means that the null hypothesis (simple model is better) is unlikely.

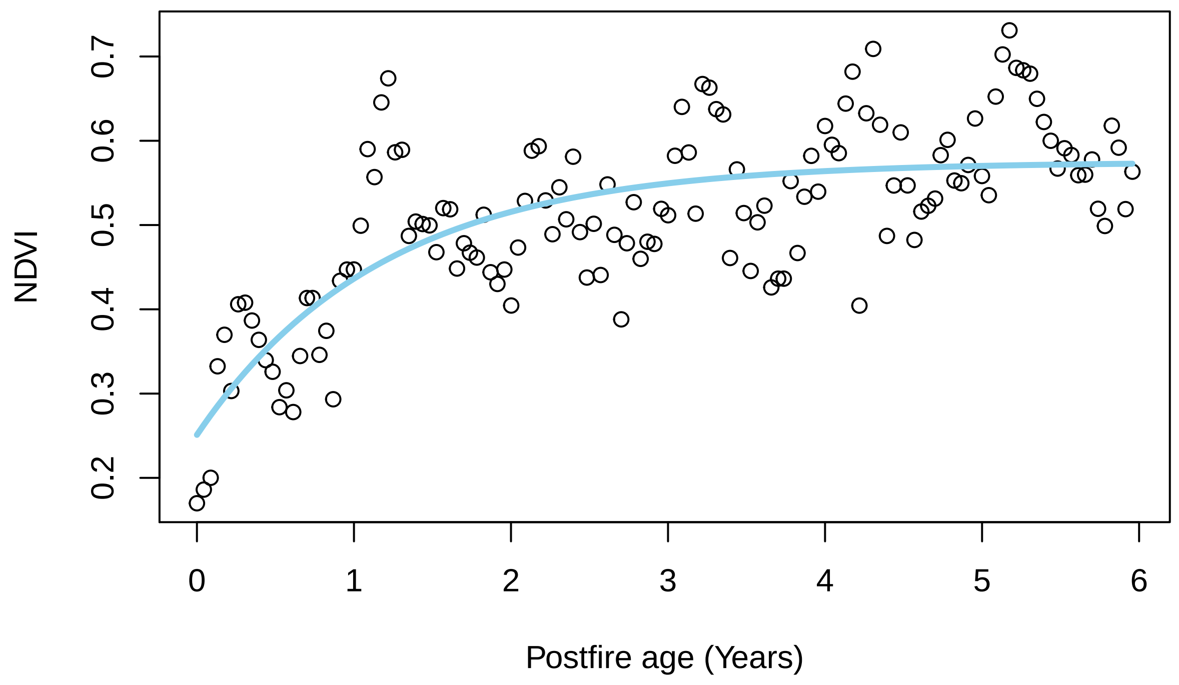
Q4: How many degrees of freedom are there in this ANOVA and why (i.e. what are they)?

> *Answer 4:* Degrees of freedom represent the difference in the number of parameters between the models. There are 268 degrees of freedom in total in this ANOVA (135, 133). Degrees of freedom are the number of subjects minus the number of treatments.

4. Fit models using Maximum Likelihood Estimation (MLE)

First let's fit the simpler model:

```r
## Fit the simpler model using MLE
# set parameters
par <- c(alpha = 0.2, gamma = 0.4, lambda = 0.5)
# fit model
fit_negexpMLE <- fit.negexp.MLE(dat, par)
# plot
plot.NDVI(dat)
# add curve with MLE parameters
lines(dat$age, pred.negexp(fit_negexpMLE$par,dat$age), col = 'skyblue', lwd = 3)
```

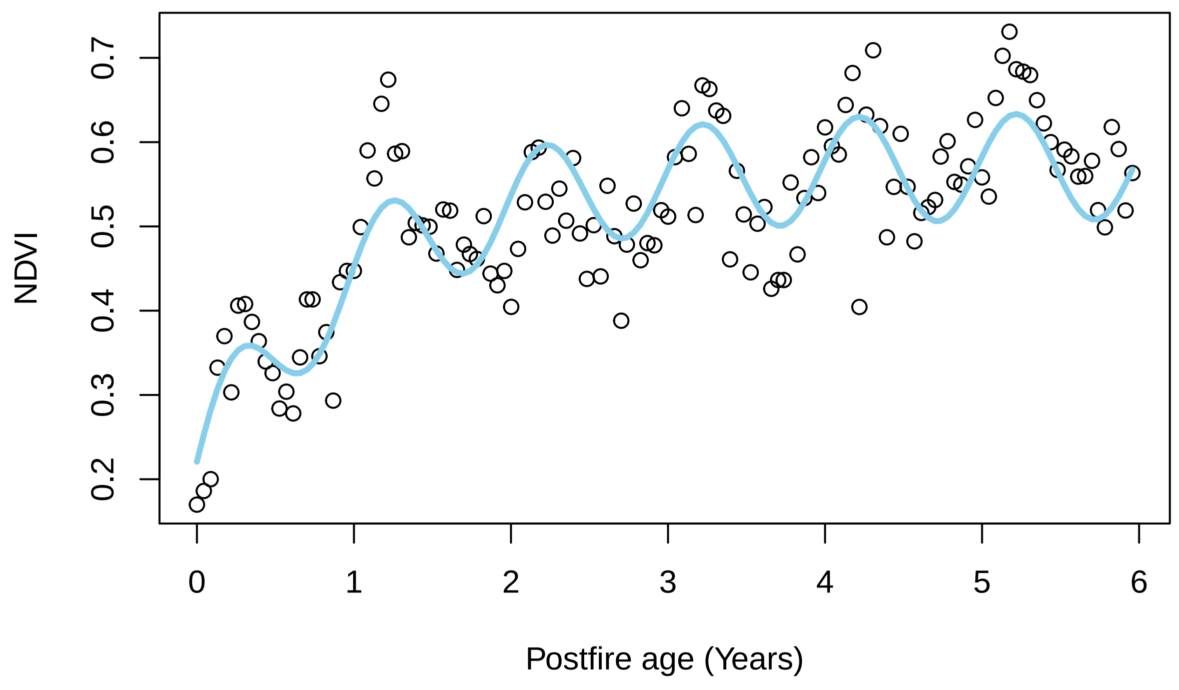


```r
fit_negexpMLE
```

```
## $par
##     alpha     gamma    lambda
## 0.2510442 0.3237419 1.1767370
##
## $value
## [1] 359053.6
##
## $counts
## function gradient
##      118       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Then the full model:

```
## Fit the full model using MLE
# set parameters
par <- c(alpha = 0.2, gamma = 0.4, lambda = 0.5, A = 0.6, phi = 0)
# fit model
fit_negexpMLES <- fit.negexpS.MLE(dat, par)
# plot
plot.NDVI(dat)
# add curve with MLE parameters
lines(dat$age, pred.negexpS(fit_negexpMLES$par,dat$age), col = 'skyblue', lwd = 3)
```



```
fit_negexpMLES
```

```
## $par
##       alpha       gamma      lambda           A         phi
##  0.20772317  0.36449293  0.98919689  0.06310554 -0.83741663
```

```
##
## $value
## [1] 225574.7
##
## $counts
## function gradient
##      914       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

5. Compare MLE models using Akaike's information criterion (AIC)

Note that we can't compare our MLE models using ANOVA because our custom functions do not return full model fits like the `nls()` function - only the parameter estimates, negative log-likelihoods and a few other diagnostics.

Another way to compare models (and probably the most common) is using the Akaike information criterion (AIC), which is an estimator of prediction error (i.e. relative quality) of statistical models for a given set of data.

The formula for the Akaike information criterion is:

$AIC = 2K - 2(ln(L))$

Where:

- k = the number of estimated parameters in the model
- L = maximum value of the likelihood function for the model

Since we have our negative log likelihoods (i.e. $-ln(L)$ in the formula above), we can calculate the AICs and compare them.

```
AIC_simple = 6 + 2*fit_negexpMLE$value
AIC_simple
```

```
## [1] 718113.1
```

```
AIC_full = 6 + 2*fit_negexpMLES$value
AIC_full
```

```
## [1] 451155.3
```

When comparing models, the lower the AIC the better, and in general a difference in AIC of 3 or more is analagous to the models being significantly different at an α of $p < 0.05$.

```
AIC_simple - AIC_full
```

```
## [1] 266957.8
```

Q5: Is there a preferred model and if so, which one?

> *Answer 5:* From the AIC one can see that the full model has a lower AIC value. This suggests that the full model is preferred as it is 'better' than the simple model i.e. indicates the full model is a better-fit. Since the difference in AIC between the two models is very high, this relates to the models being significantly different (at an alpha of p<0.05).

The nice thing about AIC is that the models you compare do not have to be nested like they do for ANOVA, as long as the data are the same. There are a few other constraints however…

Here are the AIC scores for our pair of NLS models:

```r
AIC(fit_negexp, fit_negexpS)
```

```
##             df       AIC
## fit_negexp   4 -325.7135
## fit_negexpS  6 -385.6718
```

You'll notice that these are completely different to the AICs for the MLE models…

Q6: Why is it not okay to compare the AIC of these NLS models with the AIC of the MLE models? Hint: type `?AIC` into the R console and do some reading.

> *Answer 6:* It is not okay to compare the AIC of these NLS models with the MLE models. You can only compare models that have been fitted to the same data i.e. where a response is transformed you can't compare the two models. This is because while AIC can be produced for models not using maximum likelihood, these models shouldn't be compared with those using maximum likelihood.