# Report_10657769_10651875

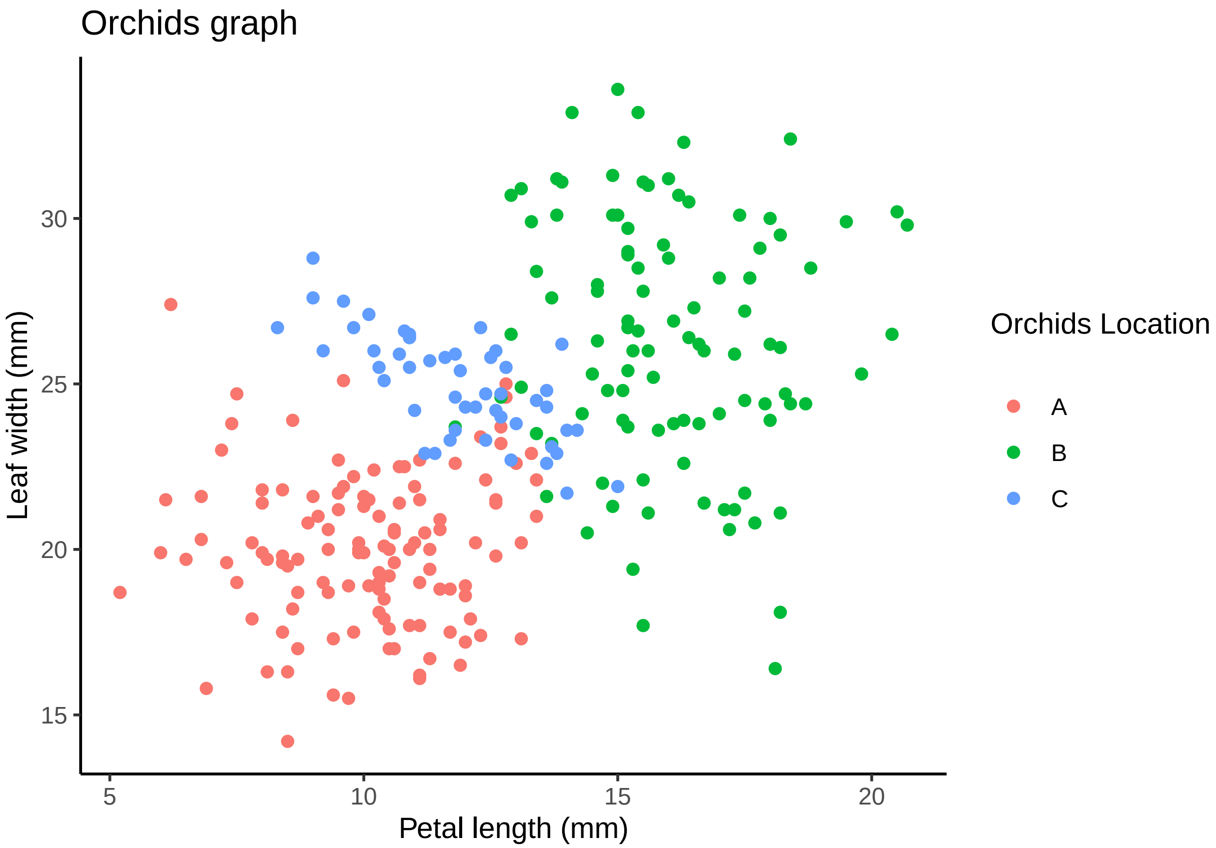
By: Aderibigbe Olugbenga & Paul Hazell

22/04/2020

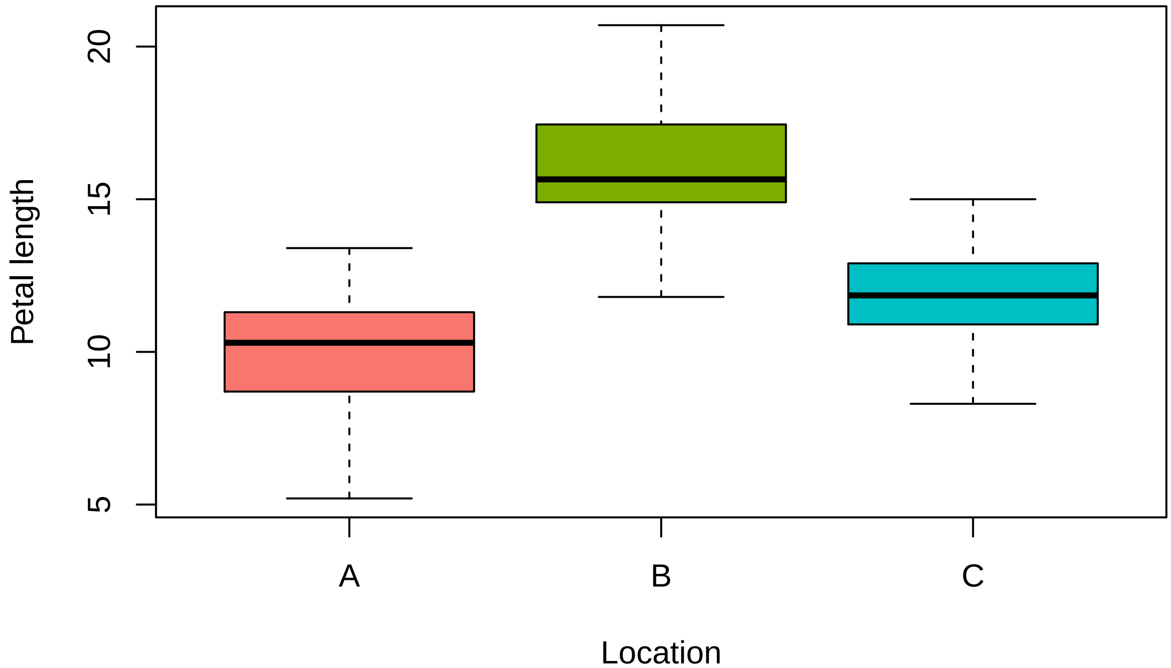
## 3.1 Machine Learning Task

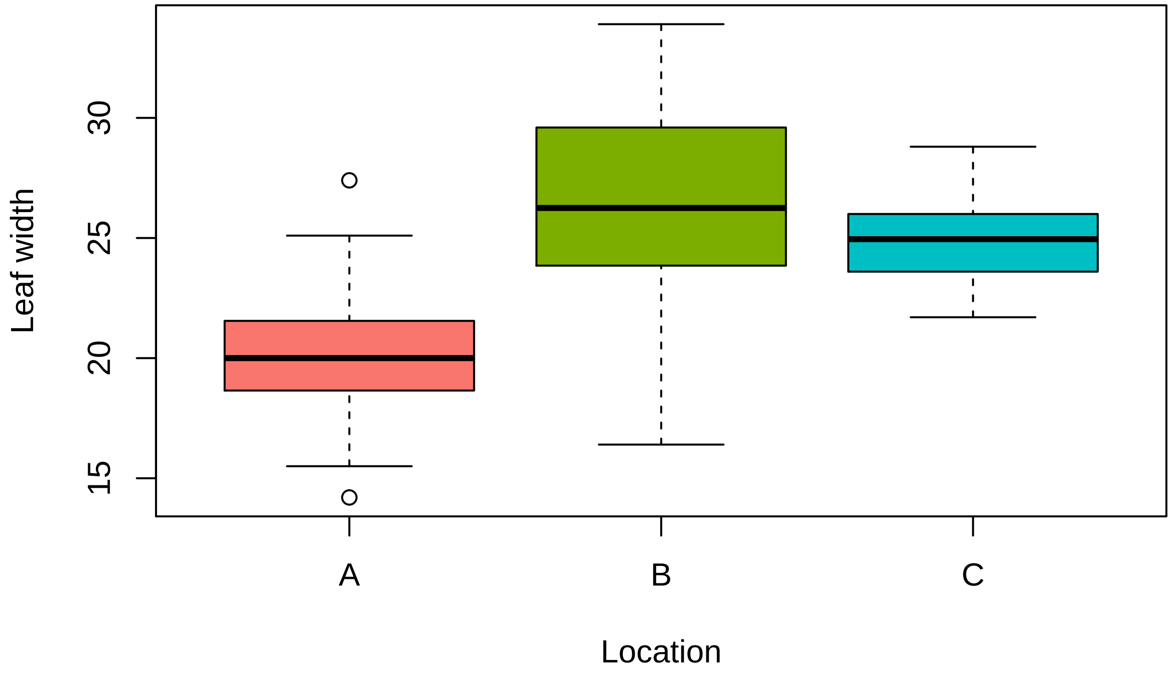
### *Part (a) Selecting parameters*

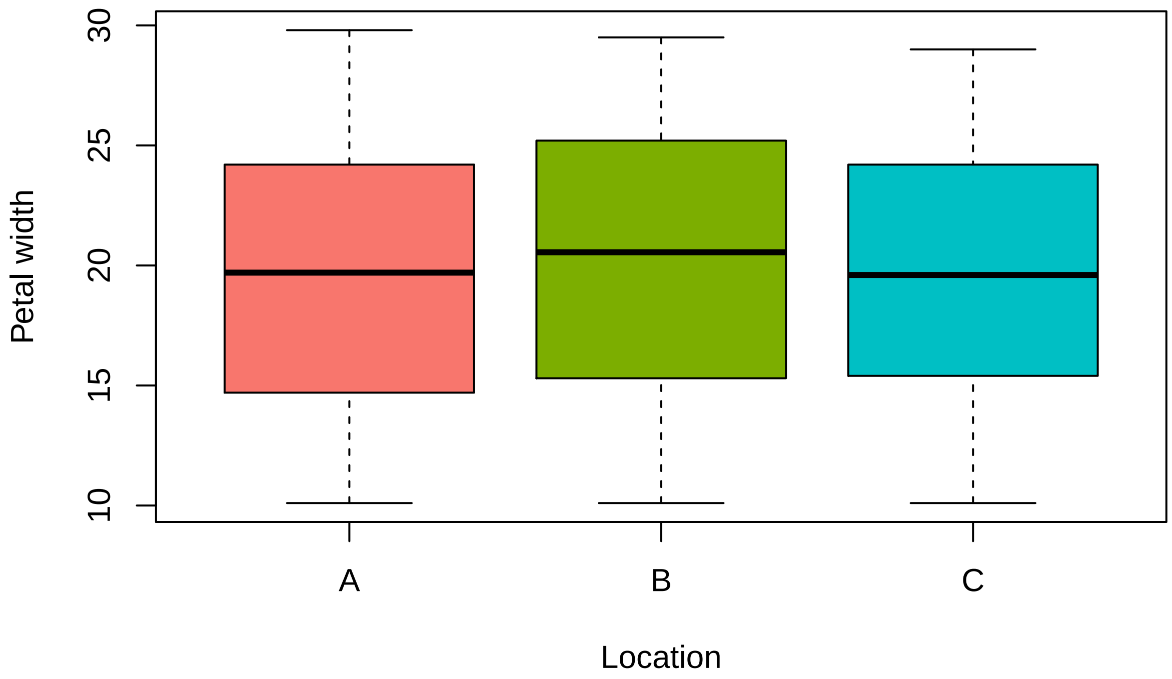
Graph of bivariate scatter plots to distinguish between the three locations of Orchids



Boxplots of the data to choose two characteristcs that should be used as predictors for orchids' locations.


Petal length
20
15
10
5
A
B
C
Location


Leaf width
30
25
20
15
A
B
C
Location



From the graph it can be seen that there is a considerable difference between the mean values of the Petal length (X1) and Leaf width (X2).

```
##   loc       X1
## 1   A 10.05917
## 2   B 16.01400
## 3   C 11.84800
```

```
##   loc       X2
## 1   A 20.01917
## 2   B 26.30500
## 3   C 24.95600
```

Whereas the difference between the mean Petal width (X3) with different locations is not much.

```
##   loc       X3
## 1   A 19.48333
## 2   B 20.04700
## 3   C 19.71400
```

So the Petal length and Leaf width data will be used as predictors for the orchids' locations.

---

### *Part (b) Training data*

Creating a training set 210 randomly chosen data points and a test set of 60 data points.

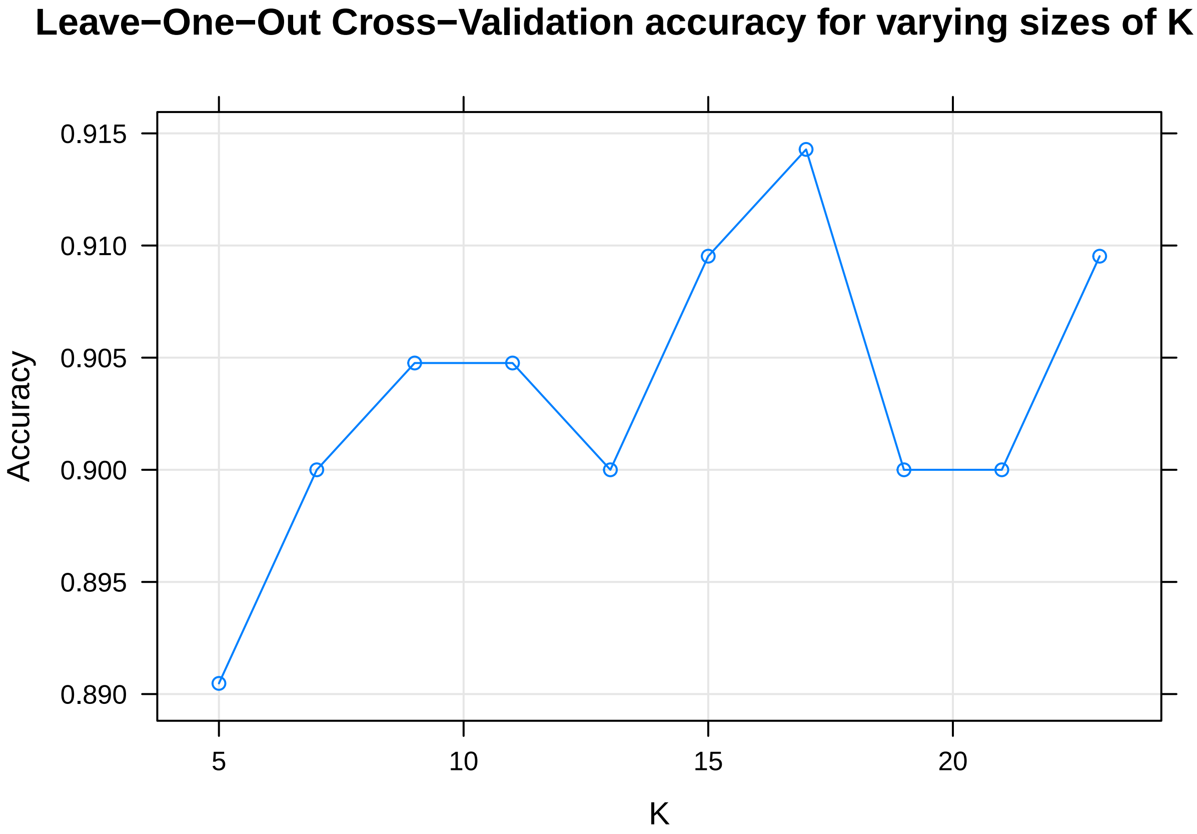
```
set.seed(1)
data.subset <- sample(270, 210)
model.train <- orchid[data.subset,]
model.test <- orchid[-data.subset,]
```

---

### *Part (c) KNN Method*

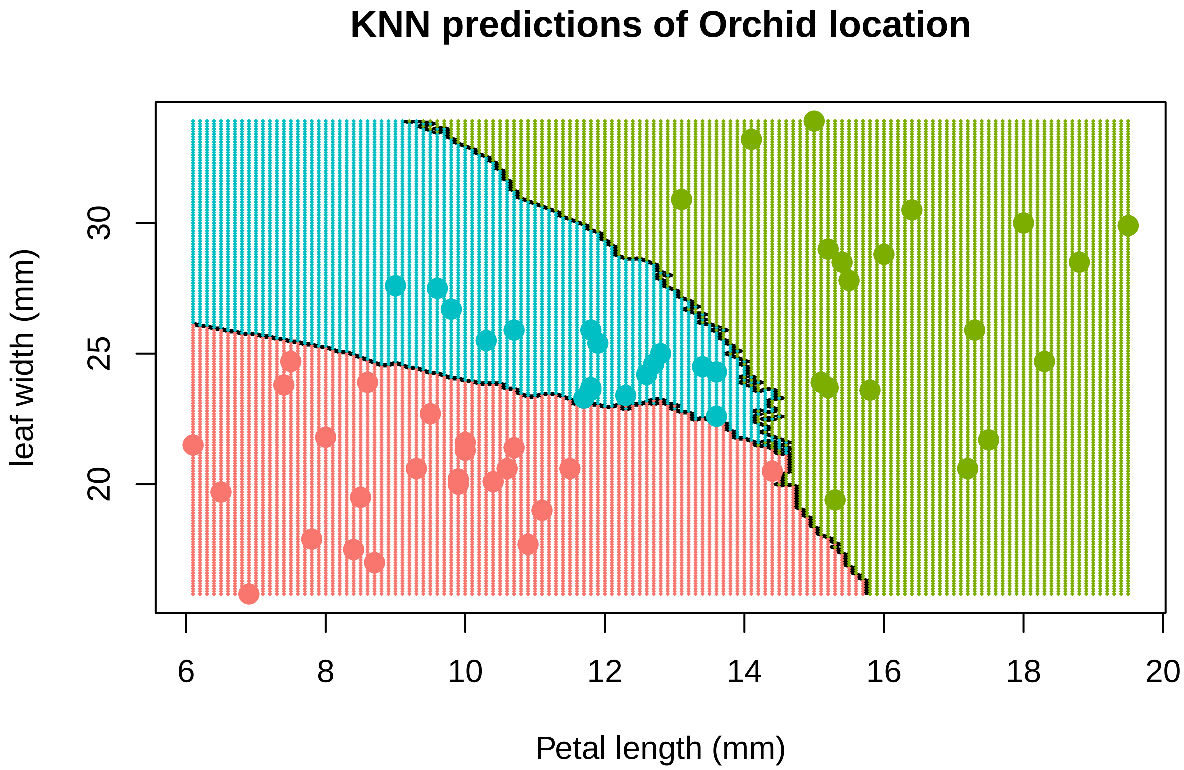
```r
# Create knn model
set.seed(1)
model.knn <- train(loc~.-X3,
                   data = model.train,
                   method = "knn",
                   trControl = trainControl(method  = "LOOCV"),
                   preProcess = c("center", "scale"), # Normalize the data
                   tuneLength = 10) # Number of possible K values to evaluate
```

Graph showing the accuracy of K



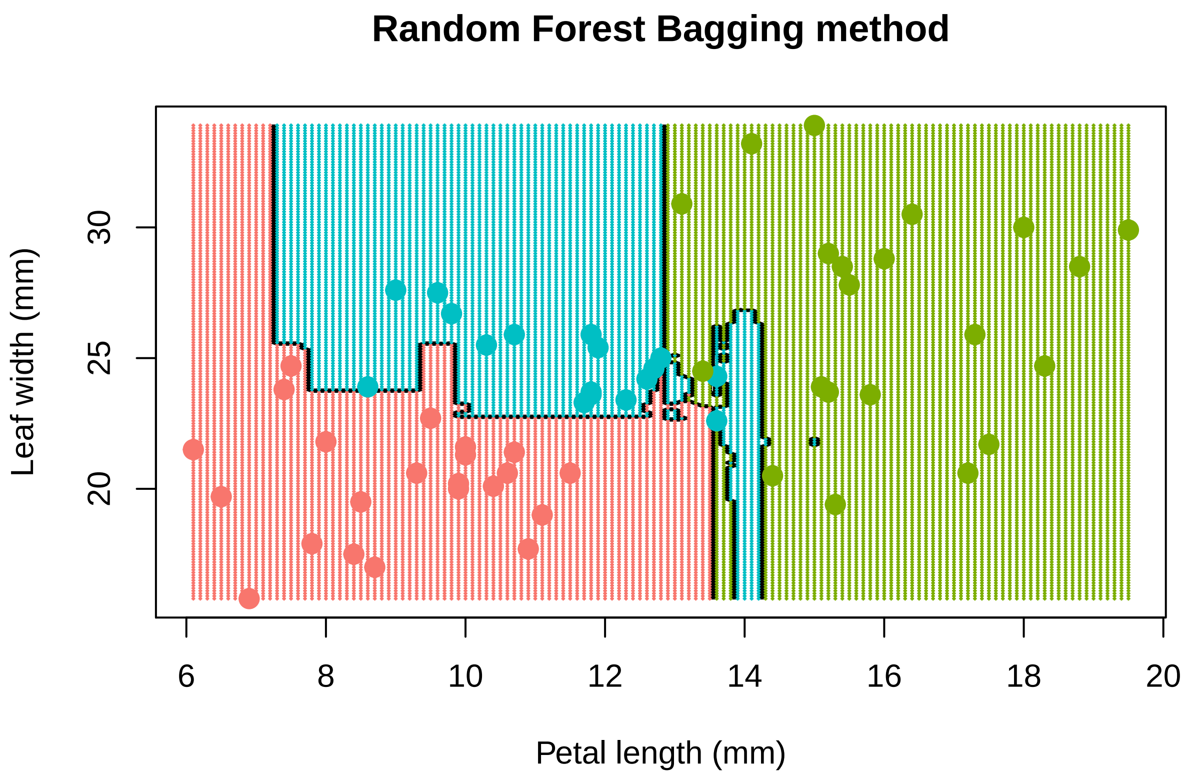
The best optimal k value is 17

```r
# Predicting the test model
predict.knn <- model.knn %>% predict(model.test)
```



***Part (d) Random Forest Bagging method***

```
##         A     B     C MeanDecreaseAccuracy MeanDecreaseGini
## X1 85.75 94.55 36.21               115.84            83.09
## X2 91.83 15.35 48.46               101.67            47.72
```



---

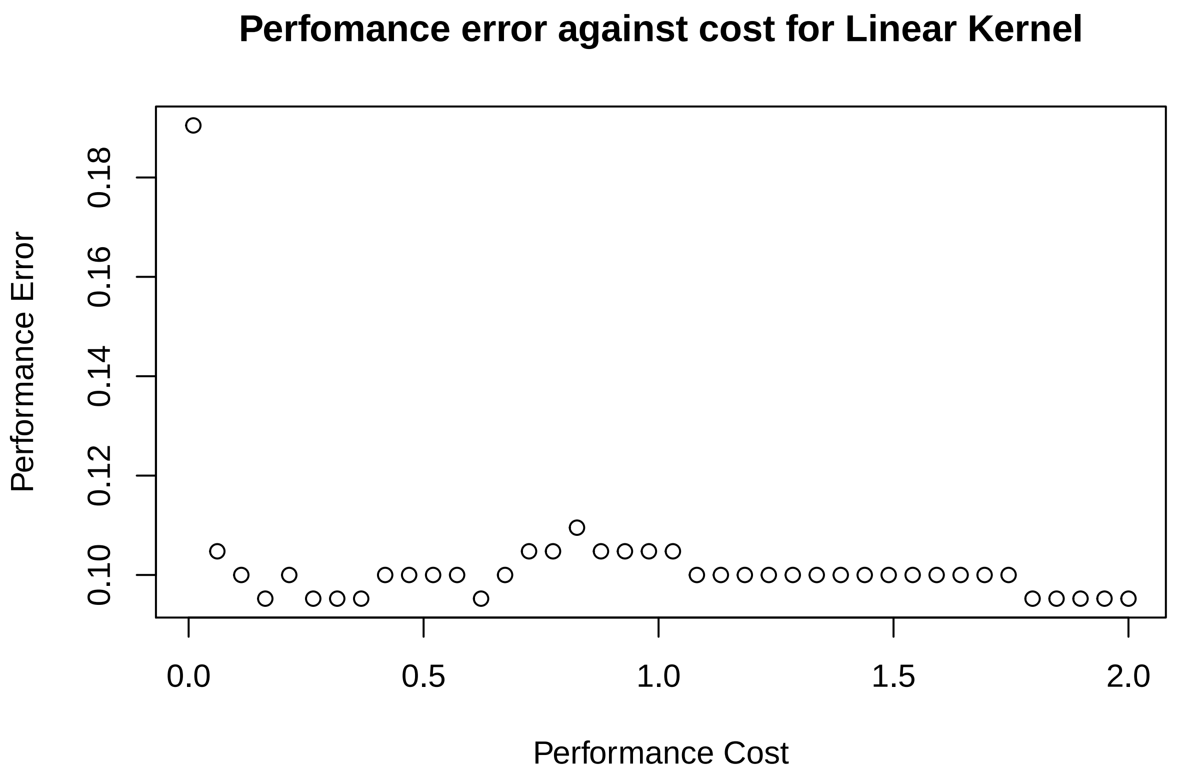
***Part (e) Support Vector Machines***

Linear kernel

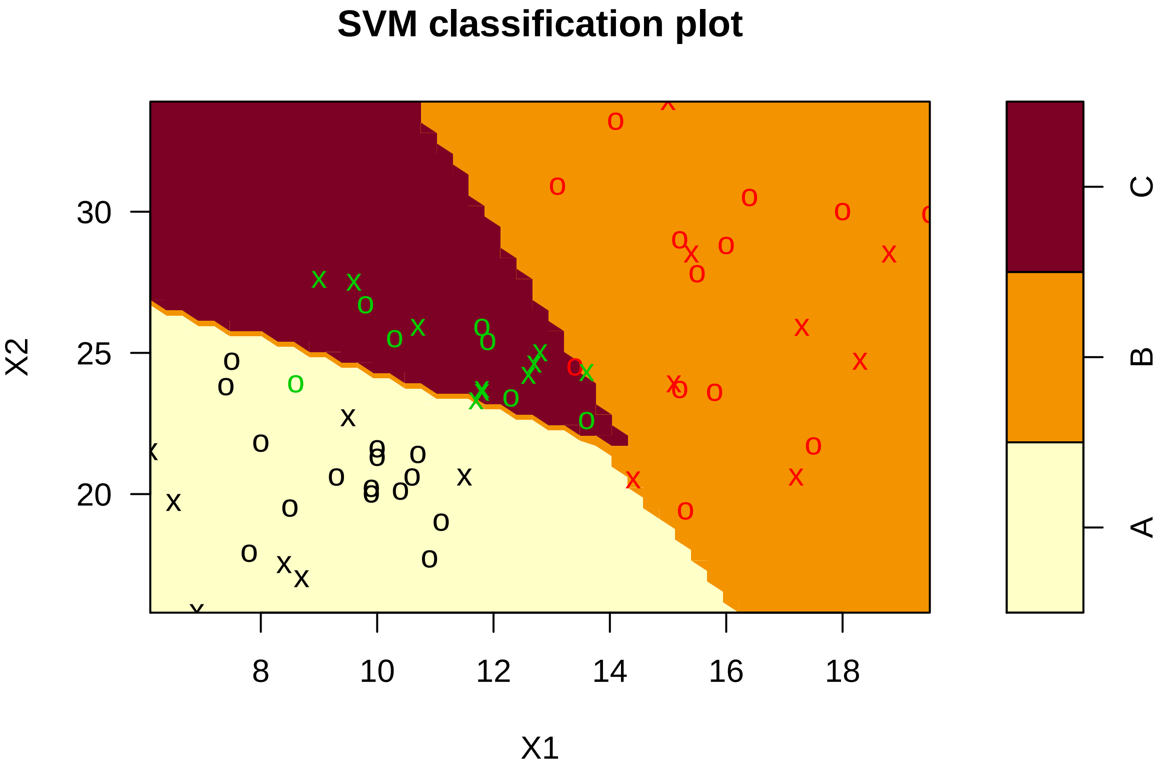
```
set.seed(1)
tune.out = tune(svm, loc ~ X1 + X2, data = orchid[data.subset,],
                kernel ="linear",
                ranges = list(cost = seq(from = 0.01,to = 2, length = 40) ))
```

Choosing the best cost parameter

```
plot(tune.out$performances$cost, tune.out$performances$error,
     main = "Perfomance error against cost for Linear Kernel",
     xlab = "Performance Cost",ylab = "Performance Error")
```



The best cost parameter for the linear kernel is 0.1630769



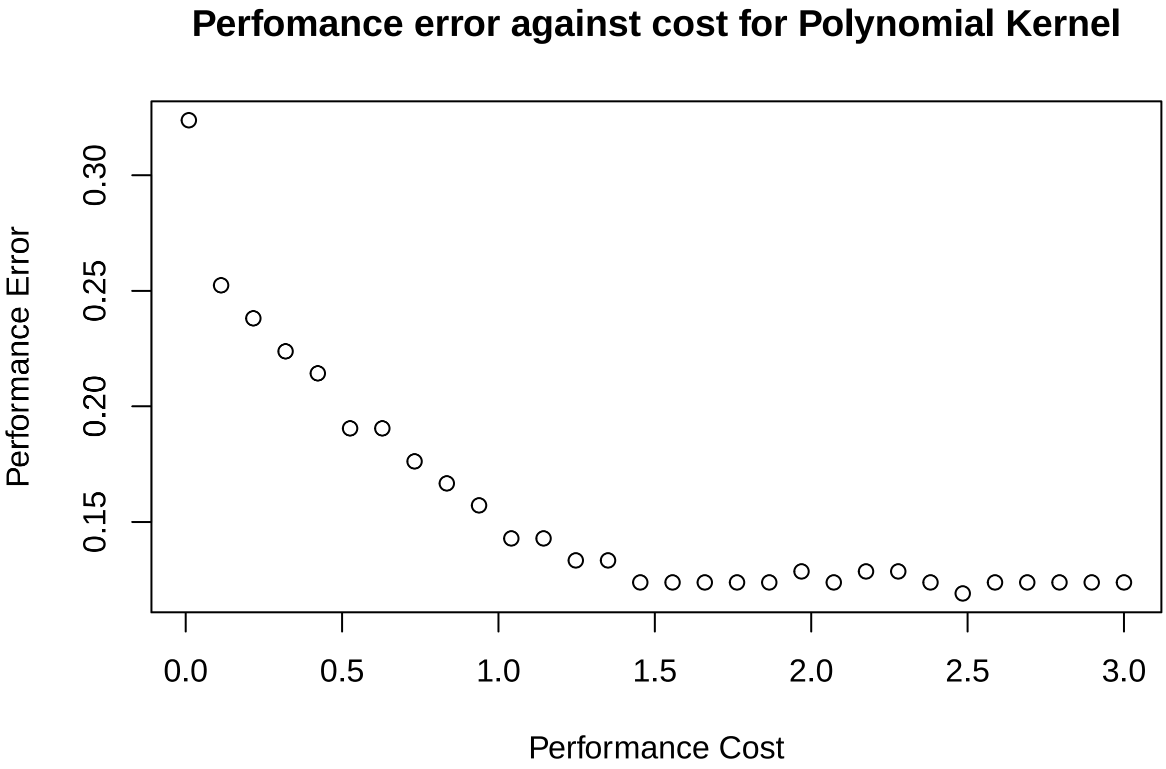
```
# Predicting test result linear
ypred_linear = predict(bestmod, model.test)
```

Polynomial Kernel

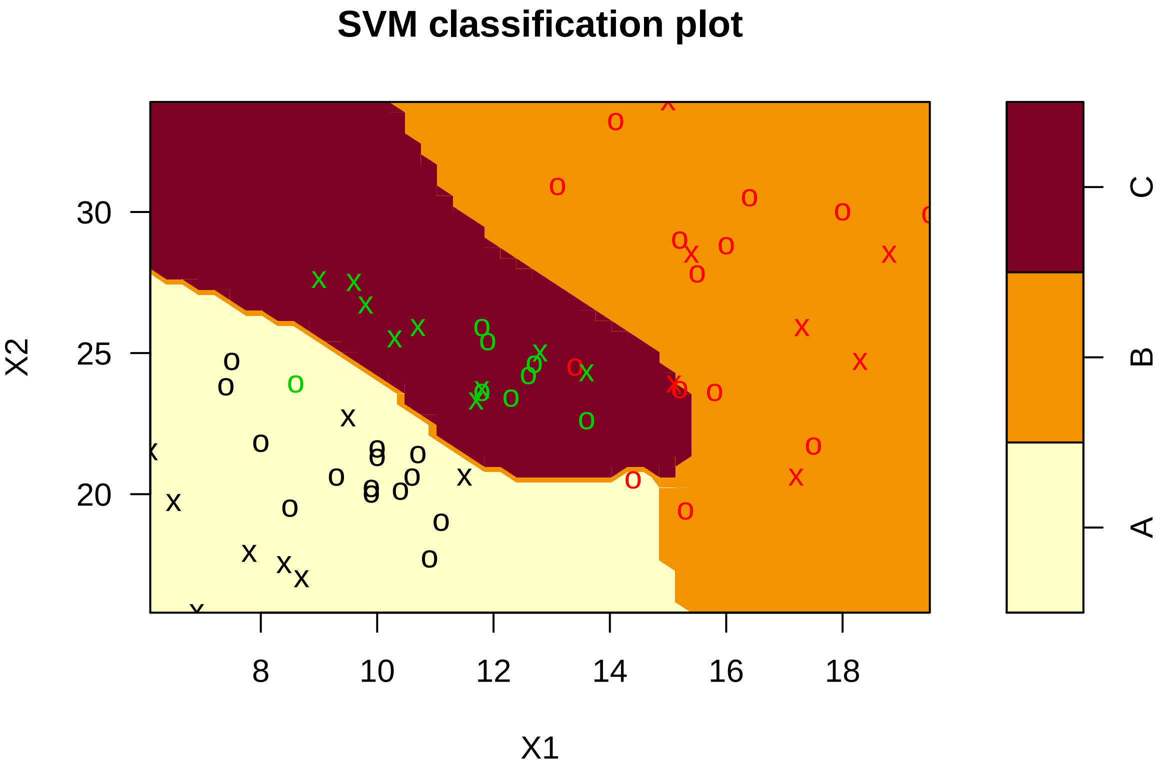
```r
# Create Polynomial Kernel
set.seed(1)
tune.out_poly = tune(svm, loc ~ X1 + X2, data = orchid[data.subset,],
                     kernel ="polynomial",
                     ranges = list(cost = seq(from = 0.01,to = 3, length = 30)))
```

Choosing the best cost parameter

```r
plot(tune.out_poly$performances$cost, tune.out_poly$performances$error,
     main = "Perfomance error against cost for Polynomial Kernel",
     xlab = "Performance Cost",ylab = "Performance Error")
```



The best cost parameter for the Polynomial kernel is 2.4844828



Computing the accuracy from the confusion matrix it can be seen that the accuracy from the Linear Kernel is higher than that of the Polynomial Kernel. It would be best to choose the kernel with the higher accuracy which is the Linear Kernel

***Part (f) Comparison***

**Test Accuracy for KNN Method**

```
##
## predict.knn  1  2  3
##           1 22  1  1
##           2  0 19  0
##           3  0  1 16
```

```
## [1] 0.95
```

**Test Accuracy for Random Forest Bagging Method**

```
##
## bag_predict  1  2  3
##           A 22  0  0
##           B  0 21  0
##           C  0  0 17
```

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

**Test Accuracy for Linear Kernel Support Vector**

```
##
## predict.knn  1  2  3
##           1 22  1  1
##           2  0 19  0
##           3  0  1 16
## [1] 0.9666667
```

**Test Accuracy for Polynomial Kernel Support Vector**

```
##
## ypred  1  2  3
##     A 22  1  1
##     B  0 17  0
##     C  0  3 16
```

```
## [1] 0.9166667
```

From our results it can be seen that the model with the best accuracy is the Random Forest Bagging method. This method came out as the best because the Bagging method helps in increasing the accuracy of the Random Forest method and also helps reduce variance of the model to help in predicting accurate results.

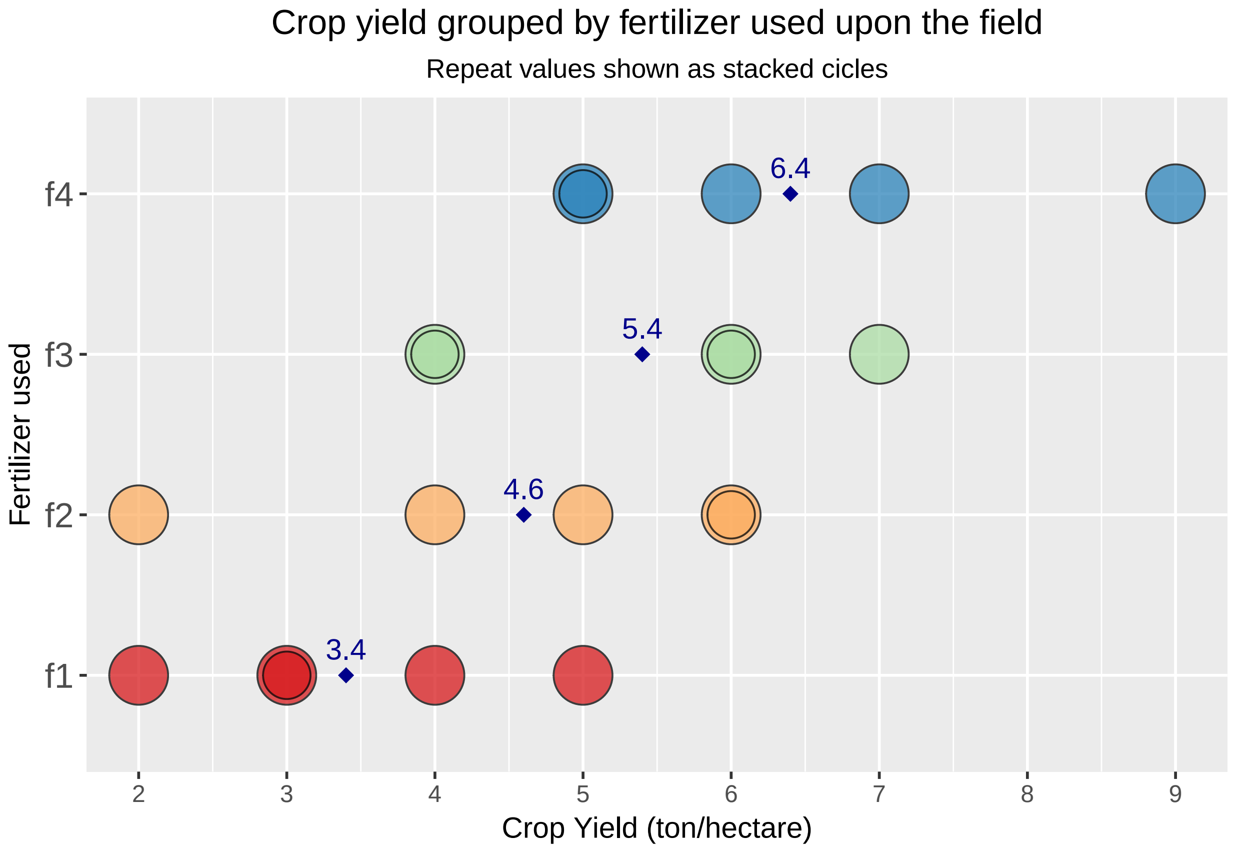
---

## 3.2 Bayesian Tasks

### 3.2.1 Frequentist One-Way Analysis of Variance

***Part (a) Visualise the data***

In the plot below crop yields are grouped by the fertilizer used upon the field and repeated yield values are shown as stacked circles. The mean yield for each fertilizer group is marked by a blue diamond.



Our visualisation shows a trend of rising means across the fertilizer groups supporting the idea that a fields crop yield is influenced by the fertilizer used. However our data set is small, variance is considerable and we can see large overlaps in the ranges of yields for each of the groups.

***Part (b) Interpretation of the parameter $\alpha$***

For a given fertilizer group $\alpha$ represents the difference between the groups mean crop yield and the mean crop yield for group *f1*, for example $\alpha_f > 0$ implies that the field is more productive than *f1*.

If $\alpha = 0$ for all of our groups then our data provides no evidence that these fertilizers have different effects on crop yield. A null hypothesis of there being no difference in crop yield between the fertilizer groups can be articulated as:

$$H_{0(b)} : \alpha_{f2} = \alpha_{f3} = \alpha_{f4} = 0$$

***Part (c) Frequentist ANOVA***

To undertake frequentist ANOVA on our dataset first we generate a linear model for crop yield dependant upon fertilizer group.

```
yield_lm <- lm(Yield ~ Fertilizer,data = df_long)
```

From this model we can derive the mean crop yield value for *f1* and the $\alpha$ values for the other groups.

```
# rename data for a cleaner results table
names(yield_lm$coefficients) <- c("f1_mean","f2_alpha","f3_alpha","f4_alpha")
summary(yield_lm)
```

```
##
## Call:
## lm(formula = Yield ~ Fertilizer, data = df_long)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -2.6   -1.4    0.0    0.8    2.6
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## f1_mean    3.4000     0.6595   5.155 9.59e-05 ***
## f2_alpha   1.2000     0.9327   1.287  0.21656
## f3_alpha   2.0000     0.9327   2.144  0.04772 *
## f4_alpha   3.0000     0.9327   3.216  0.00539 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.475 on 16 degrees of freedom
## Multiple R-squared:  0.4097, Adjusted R-squared:  0.299
## F-statistic: 3.701 on 3 and 16 DF,  p-value: 0.03388
```

An ANOVA test on our linear model returns the following results:

```
anova(yield_lm)
```

```
## Analysis of Variance Table
##
## Response: Yield
##            Df Sum Sq Mean Sq F value  Pr(>F)
## Fertilizer  3  24.15   8.050  3.7011 0.03388 *
## Residuals  16  34.80   2.175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Given that the size of this ANOVA test is 0.05 our p-value of 0.03388 means we can reject $H_{0(b)}$ and conclude that crop yield is different when different fertilizers are used.

---

### *Part (d) Tukey Honest Significant Differences Test*

We will now investigate inter-field crop yield differences using a Tukey Honest Significant Difference test. The null hypotheses for this test are:

$$\begin{gathered}
H_{0(d2:1)} : \alpha_{f2} \text{ - } \alpha_{f1} = 0 \\
H_{0(d3:1)} : \alpha_{f3} \text{ - } \alpha_{f1} = 0 \\
H_{0(d4:1)} : \alpha_{f4} \text{ - } \alpha_{f1} = 0 \\
H_{0(d3:2)} : \alpha_{f3} \text{ - } \alpha_{f2} = 0
\end{gathered}$$

$$H_{0(d4:2)} : \alpha_{f4} \text{ - } \alpha_{f2} = 0$$
$$H_{0(d4:3)} : \alpha_{f4} \text{ - } \alpha_{f3} = 0$$

The code and results of our Tukey test are as follows:

```
yield_aov <- aov(Yield ~ Fertilizer,data = df_long)
TukeyHSD(yield_aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Yield ~ Fertilizer, data = df_long)
##
## $Fertilizer
##       diff        lwr      upr     p adj
## f2-f1  1.2 -1.4685816 3.868582 0.5840061
## f3-f1  2.0 -0.6685816 4.668582 0.1814975
## f4-f1  3.0  0.3314184 5.668582 0.0249664
## f3-f2  0.8 -1.8685816 3.468582 0.8261588
## f4-f2  1.8 -0.8685816 4.468582 0.2550618
## f4-f3  1.0 -1.6685816 3.668582 0.7107552
```

Whilst all groups have a non zero difference in mean yields only the difference between *f4* and *f1* passes our significance value of 0.05 hence the only null hypothesis that we can reject is:

$$H_{0(d4:1)} : \alpha_{f4} \text{ - } \alpha_{f1} = 0$$

Whilst we can be confident that *f4* is more productive than *f1* we have little certainity in the extent of the increase we are seeing. The mean increase of 3 ton/hectare would represent a near doubling of productivity for *f1* but the confidence intervals of our Tukey test show the true value lies somewhere between the negligible 0.33 ton/hectare and an impressive 5.6 ton/hectare. Achieving a higher accuracy on this figure would require further data collection and analysis.

---

***Part (e) Is the underlying crop yield level for* `f4` *more than 0.5 units greater than the average of the underlying crop yield levels obtained using the other three fertilizers?***

In order to answer this question we will look to reject the following null hypothesis:

$$H_{0(e)} : \mu_4 \text{ - } (\mu_1 + \mu_2 + \mu_3/3) \leq 0.5$$

To test $H_{0(e)}$ we will run a mean parameterised version of our linear model through a generalised linear hypothesis test, the results of which can be seen below.

```
yield_lm_mp <- lm(Yield ~ Fertilizer -1, data = df_long)
# rename data for a cleaner results table
names(yield_lm_mp$coefficients) <- c("f1","f2","f3","f4")
ght <- glht(yield_lm_mp,linfct="f4 - ((f1 + f2 + f3)/3) <= 0.5")
summary(ght)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = Yield ~ Fertilizer - 1, data = df_long)
##
## Linear Hypotheses:
##                                Estimate Std. Error t value Pr(>t)
## f4 - ((f1 + f2 + f3)/3) <= 0.5   1.9333     0.7616   1.882 0.0391 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

The above results state that if we pressume $H_{0(e)}$ to be true then the probability of $\mu_4$ - $(\mu_1 + \mu_2 + \mu_3/3) \geq 1.882$ is 0.0391.

Taking the complement of this probability we can state that if we pressume $H_{0(e)}$ to be false then the probability $\mu_4$ - $(\mu_1 + \mu_2 + \mu_3/3) \geq 1.882$ is 0.9609.

Since $1.882 > 0.5$ we can reasonably conclude that the $\mu_4$ is more than 0.5 units greater than the average of the other underlying yields and reject $H_{0(e)}$.

---

### 3.2.2 Bayesian One-way ANOVA

***Part (f) One way ANOVA using JAGS***

The JAGS code we will use to undertake our ANOVA is as follows:

```
# yield values
y <- df_long$Yield
# fertilizer groups - convert group to numeric factor
g1 <- df_long$Fertilizer
g2 <- substr(g1,2,2)
group <-as.factor(as.numeric(g2))
# number of data points
n <- length(y)
# number of groups
g <- 4
data_anova <- list("y", "group", "n", "g")

# JAGS model
Bayesian_anova <- function(){
  # Likelihood function
  for(k in 1:n){
    y[k] ~ dnorm(mu[k], tau)
    mu[k] <- m + alpha[group[k]]
  }

  # Priors - objective with uniform variance
  m ~ dnorm(0.0, 0.0001)
  alpha[1] <- 0
  for(i in 2:g){
    alpha[i] ~ dnorm(0.0,0.0001)
  }
  tau ~ dgamma(0.001,0.001)
  sd = 1/sqrt(tau)

  # Tracking the mean yield values
  mean_yield[1] = m + alpha[1]
  mean_yield[2] = m + alpha[2]
  mean_yield[3] = m + alpha[3]
  mean_yield[4] = m + alpha[4]
}
```

```r
Bayesian_anova_inference <- jags(data = data_anova,
                                 parameters.to.save = c("alpha","mean_yield"),
                                 n.iter = 100000, n.chains = 3,
                                 model.file = Bayesian_anova)
```

```
## module glm loaded
```

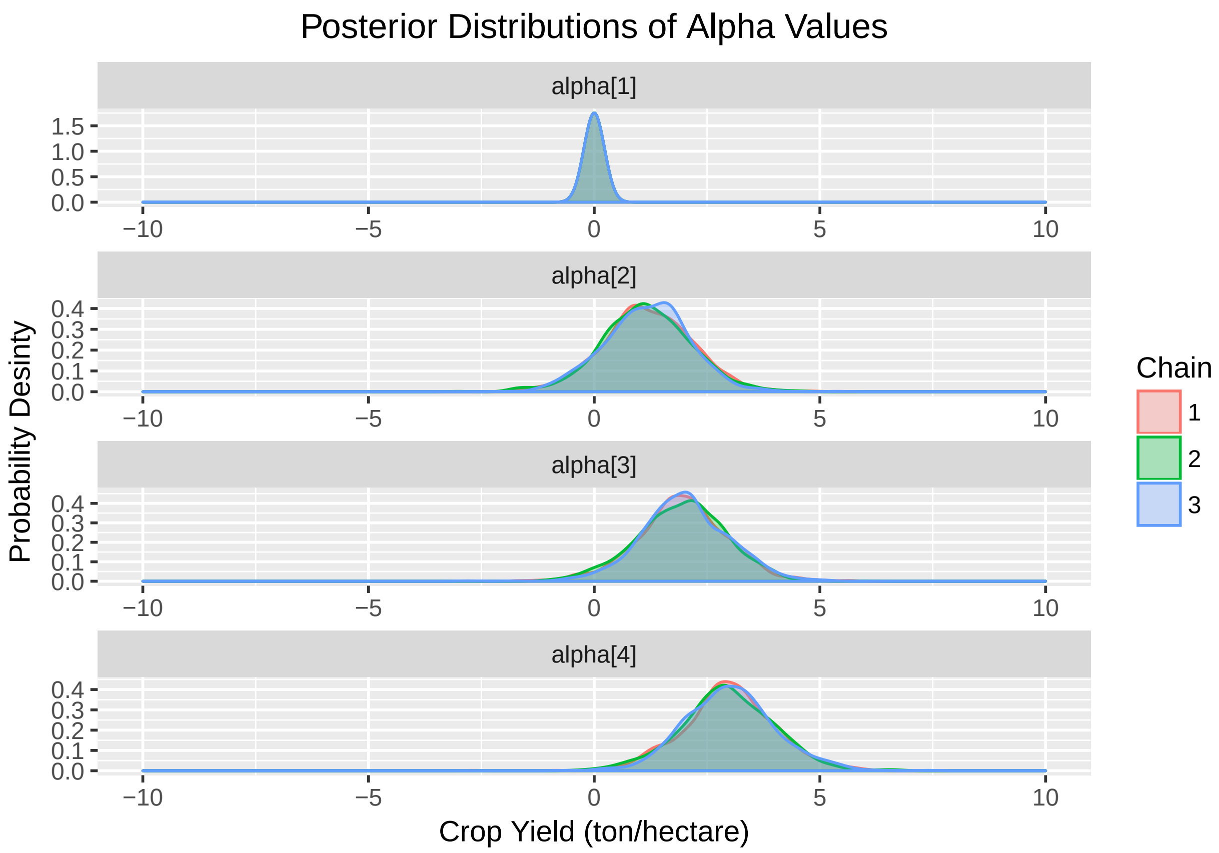
```r
anova_data <- Bayesian_anova_inference[[2]]$summary[c(-1,-5),c(1,2,3,5,7)]
print(anova_data)
```

```
##                    mean        sd       2.5%      50%    97.5%
## alpha[2]      1.172705 0.9699793 -0.7691654 1.159624 3.073878
## alpha[3]      1.970505 0.9866318 -0.0555495 1.983943 3.862402
## alpha[4]      2.983319 1.0158225  0.9788451 2.968416 5.047429
## mean_yield[1] 3.425581 0.6989486  2.0577499 3.431811 4.807692
## mean_yield[2] 4.598285 0.6868939  3.2587851 4.605113 5.949318
## mean_yield[3] 5.396086 0.6949018  3.9892778 5.399576 6.815709
## mean_yield[4] 6.408899 0.7169188  4.9920564 6.400326 7.811384
```

---

### ***Part (g) Graphical representation of posterior densities***

```r
mcmc_data <- as.mcmc(Bayesian_anova_inference)
ggs_data <- ggs(mcmc_data)
ggs_density(ggs_data, family = "^alpha")+xlim(-10,10)+
  theme(plot.title =element_text(hjust = 0.5))+
  ggtitle("Posterior Distributions of Alpha Values")+
  xlab("Crop Yield (ton/hectare)")+
  ylab("Probability Desinty")
```



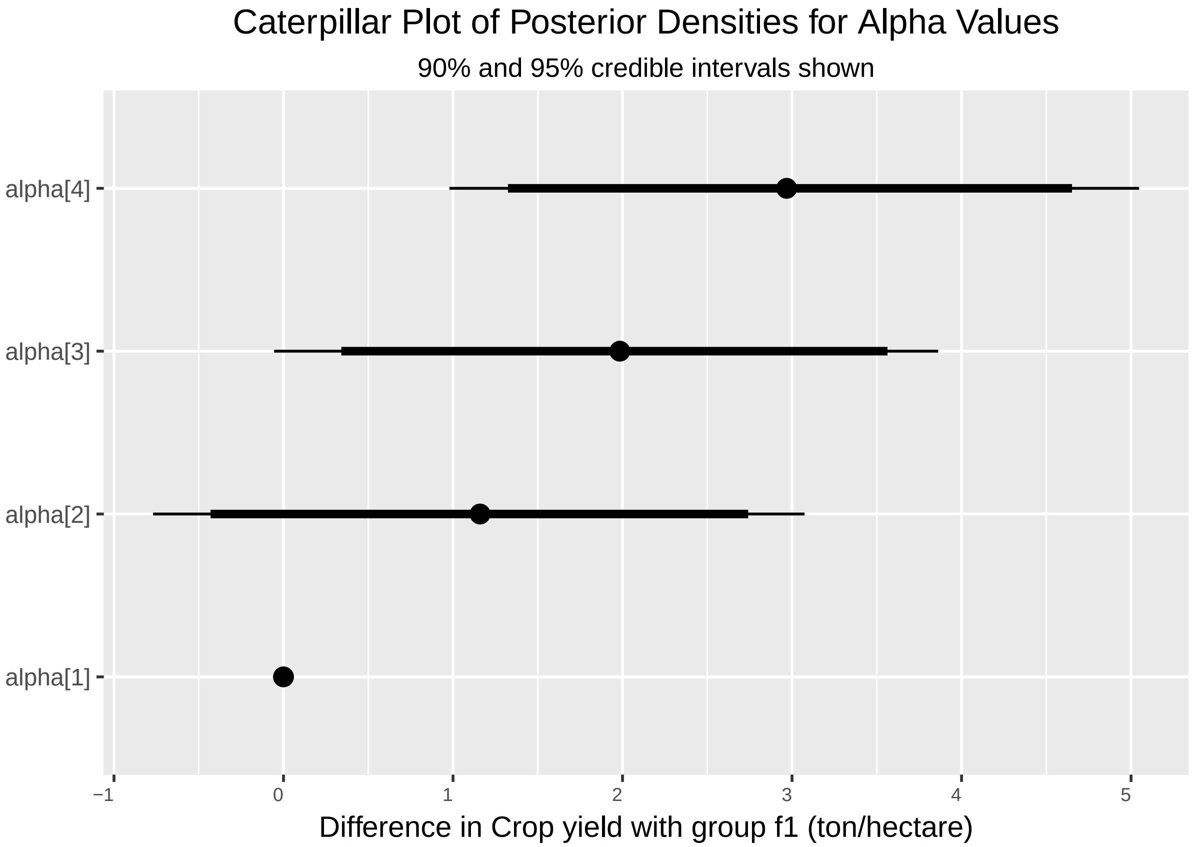
The first thing to note from the above plot is that the posterior distributions of the three chains are closely alligned, this shows our JAGS code was able to consistently sample and gives us confidence in our results. $\alpha_1$ is of course centered around zero and by scanning the peaks of probability density across each of the distributions we can see a similar pattern to the rising means observed in section (a).

By observing the proportion of each distributions that is $< 0$ we can also gain an idea of whether or not we can reject $H_{0(b)}$. For example its quite clear $\alpha_2$ does not allow us to reject $H_{0(b)}$ but there is a good chance $\alpha_4$ and perhaps even $\alpha_3$ will allow us to do so.

---

### ***Part (h) 95% Credible Interval Analysis***

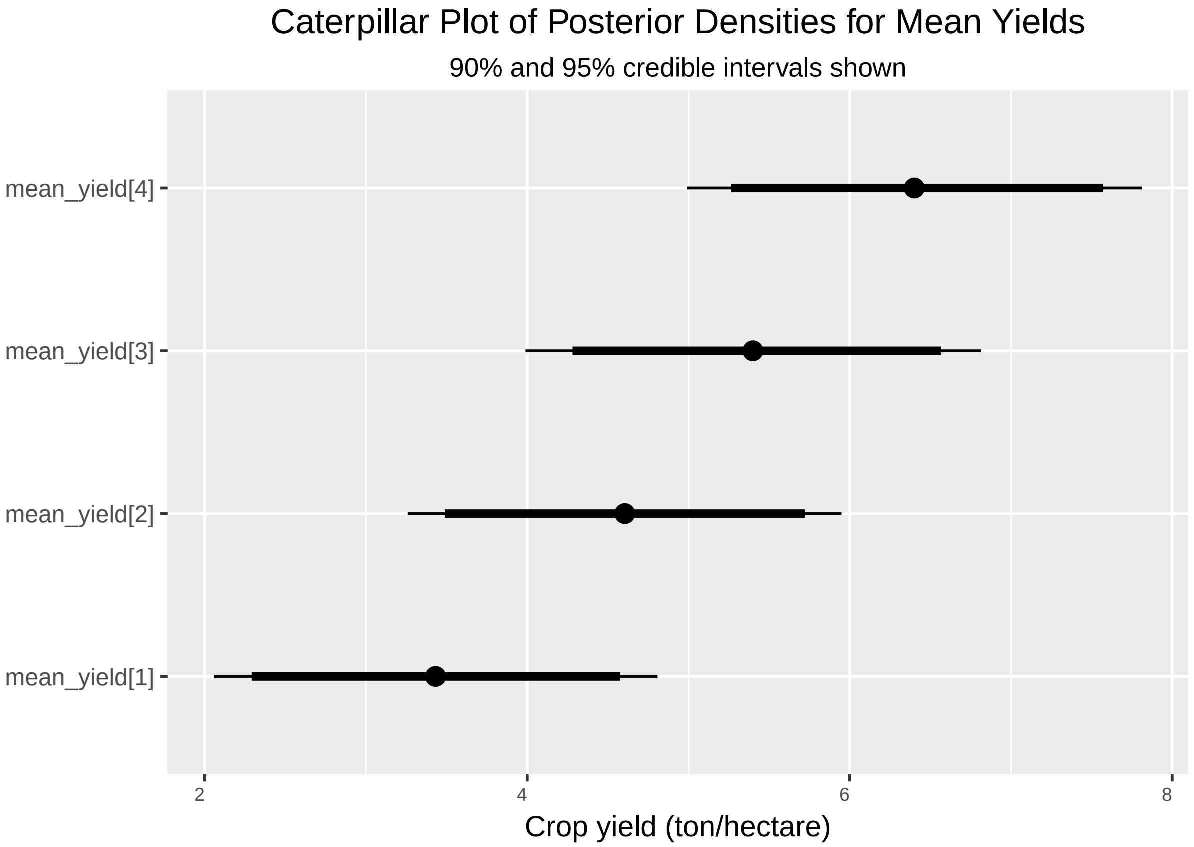
```r
ggs_caterpillar(ggs_data,family = "^alpha")+
  ylab("")+
  xlab("Difference in Crop yield with group f1 (ton/hectare)")+
  labs(title = "Caterpillar Plot of Posterior Densities for Alpha Values",
       subtitle = "90% and 95% credible intervals shown")+
  theme(plot.title =element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```



Above we see a caterpillar plot for the 95% and 90% credible intervals of the distributions shown in section (g). Here we can see that the 95% credible intervals for $\alpha_4$ and $\alpha_3$ only contain values $> 0$ thereby allowing us to reject $H_{0(b)}$, pressuming that crop yield is a continuous parameter.

Whilst intuitively we would expect crop yield to be continuous this fact is not given nor supported by the data. If we take crop yield to be discrete then we can no longer use $\alpha_3$ to reject $H_{0(b)}$ as a significant proportion of its credible interval is $< 1$, however we are still able to reject $H_{0(b)}$ on the basis that the credible interval of $\alpha_4$ is completely $> 1$.

For a wider view of the data we present a caterpillar plot of credible intervals for mean yields.



Visually we can deduce similarity between groups as overlapping credible intervals. As expected *f2* crosses all groups and once again the significant difference between *f4* and *f1* can be seen as there is no overlap between their 95% credible intervals.

---

***Part (i) Compare Frequentist and Bayesian 95% confidence intervals***

Both ANOVA analysis methods have allowed us to reject $H_{0(b)}$ and the implied mean yield values from section (c) ($\mu_f = \mu_1 + \alpha_f$) agree with the mean yields computed in section (f).

We can also see little difference between the confidence intervals of the frequentist approach and the credible intervals of the Bayesian approach in the following table:

```
##         conf_lower  cred_2.5%         diff conf_upper cred_97.5%
## alpha2 -0.77731603 -0.7691654 -0.008150583   3.177316   3.073878
## alpha3  0.02268397 -0.0555495  0.078233475   3.977316   3.862402
## alpha4  1.02268397  0.9788451  0.043838838   4.977316   5.047429
##               diff
## alpha2  0.10343816
## alpha3  0.11491450
## alpha4 -0.07011256
```

---

***Part (j) Bayesian analysis of differences in $\alpha_f$***

We will use the following JAGS code to infer the posterior distributions that allow us to test all of the null hypotheses from section (d) and $H_{0(e)}$

```r
delta_alpha_model <- function(){

  # Likelihood function
  for(k in 1:n){
    y[k] ~ dnorm(mu[k], tau)
    mu[k] <- m + alpha[group[k]]
  }

  # Priors - objective with uniform variance
  m ~ dnorm(0.0, 0.0001)
  tau ~ dgamma(0.001,0.001)

  for(i in 1:g){
    alpha[i] ~ dnorm(0.0,0.0001)
    for (n in 1:(i-1)) {
      AlphaDelta[n,i] <- alpha[i]-alpha[n] # Track values for (d) hypotheses
    }
  }

  # Track value for part (e) hypothesis
  mu_four_test <- alpha[4] - ((alpha[1] + alpha[2] + alpha[3])/3)
}

delta_alpha_inference <- jags(data = data_anova,
                              parameters.to.save = c("AlphaDelta","mu_four_test"),
                              n.iter = 3000000, n.chains = 1,
                              model.file = delta_alpha_model)
```

```r
da_data <- delta_alpha_inference[[2]]$summary[-7,c(1,2,3,5,7)]
print(da_data)
```

```
##                      mean        sd        2.5%       50%    97.5%
## AlphaDelta[1,2] 1.1676585 1.0257611 -0.86504444 1.1651724 3.046142
## AlphaDelta[1,3] 1.9344091 0.9878502  0.00340792 1.9509141 3.855207
## AlphaDelta[2,3] 0.7667506 0.9997119 -1.10450026 0.7534417 2.657421
## AlphaDelta[1,4] 2.9671405 0.9752423  0.93335811 2.9397876 4.993539
## AlphaDelta[2,4] 1.7994820 0.9960897 -0.13743482 1.7989628 3.752102
## AlphaDelta[3,4] 1.0327314 0.9850069 -0.88840419 1.0108970 2.955650
## mu_four_test    1.9331180 0.7967388  0.47537031 1.9154314 3.501559
```

First lets evaluate the null hypotheses from section (d), for these to be rejected we are looking for the entirety of the 95% credible interval to be $> 0$. From this criteria the result from row Alpha[1,4] allows us to confidently reject the following null hypotheses:

$$H_{0(d4:1)} : \alpha_{f4} \text{ - } \alpha_{f1} = 0$$

If you investigate the result Alpha[1,3] you may well see that its 95% credible interval is $> 0$ and conclude that $H_{0(d3:1)}$ can be rejected. However througout the writing of this report we have re-run this model many times and have observed that the absolute value of the 2.5% credible interval is always very close to zero, sometimes it is $> 0$ and we can reject $H_{0(d3:1)}$ and sometimes we cannot. Given this inconsistency we conclude that $H_{0(d3:1)}$ cannot be rejected by our results.

For our Bayesian analysis of $H_{0(e)}$ we first consider the 95% credible interval for mu_four_test presented above. The low end has a value that is $< 0.5$ so we could decide not to reject our null hypothesis. However if we apply the reported $\mu$ and sd values from this test as parameters to a normal distribution we can investigate further.

For example the probability that the underlying difference between $\mu_4$ and the average of the other means is greater than 0.5 can be calculated as:

```
mu <- delta_alpha_inference$BUGSoutput$summary[8,1]
sd <- delta_alpha_inference$BUGSoutput$summary[8,2]
1 - pnorm(0.5,mu,sd)
```

```
## [1] 0.9639693
```

Which is greater than the 95% confidence/credibility we have been applying elsewhere. If we decide to disregard the upper 2.5% of the distribution as outliers then our probability drops to:

```
(1-pnorm(0.5,mu,sd)) - (1-(pnorm((qnorm(0.975,mu,sd)),mu,sd)))
```

```
## [1] 0.9389693
```

For consistency we will maintain a centralised 95% confidence interval and accept $H_{0(e)}$, however the marginality of this decision is of interest and should be reported to stakeholders.

Above we have seen some contrasting conclusions between frequentist and Bayesian methods for the hypotheses $H_{0(d)}$ and $H_{0(e)}$. The authors opinion is that the Bayesian analysis should be followed as our data set is small and the frequentist methods are more susceptible to variance upon re-sampling.

---

### 3.2.3 Simpler Bayesian model

#### *Part (k) JAGS code for simple Bayesian model*

The JAGS code we will use to infer posterior distributions based upon the given simple model is as follows:

```
simple_bayes_model <- function(){

  # Likelihood function
  for(k in 1:n){
    y[k] ~ dnorm(mu[k], tau)
    mu[k] <- fertilizer[group[k]]
  }

  # Priors - objective with uniform variance
  for(i in 1:g){
    fertilizer[i] ~ dnorm(0.0,0.0001)
  }

  tau ~ dgamma(0.001,0.001)
  sigma = 1/sqrt(tau)
}

simple_jags <- jags(data = data_anova,
                    parameters.to.save = c("fertilizer","sigma","tau"),
                    n.iter = 100000, n.chains = 3, model.file = simple_bayes_model)
```

```
simp_data <- simple_jags[[2]]$summary[c(2:6),c(1,2,3,5,7)]
row.names(simp_data) <- c("mean_yield_f1","mean_yield_f2",
                          "mean_yield_f3","mean_yield_f4","standard_deviation")
print(simp_data)
```

```
##                         mean         sd      2.5%       50%     97.5%
## mean_yield_f1      3.395403 0.7085346 2.033709 3.396412 4.784431
## mean_yield_f2      4.598582 0.6745529 3.277520 4.594866 5.946839
## mean_yield_f3      5.397019 0.7014910 3.999238 5.400924 6.793537
## mean_yield_f4      6.392579 0.6974185 4.960344 6.394153 7.814542
## standard_deviation 1.543600 0.2890685 1.111819 1.500689 2.200826
```

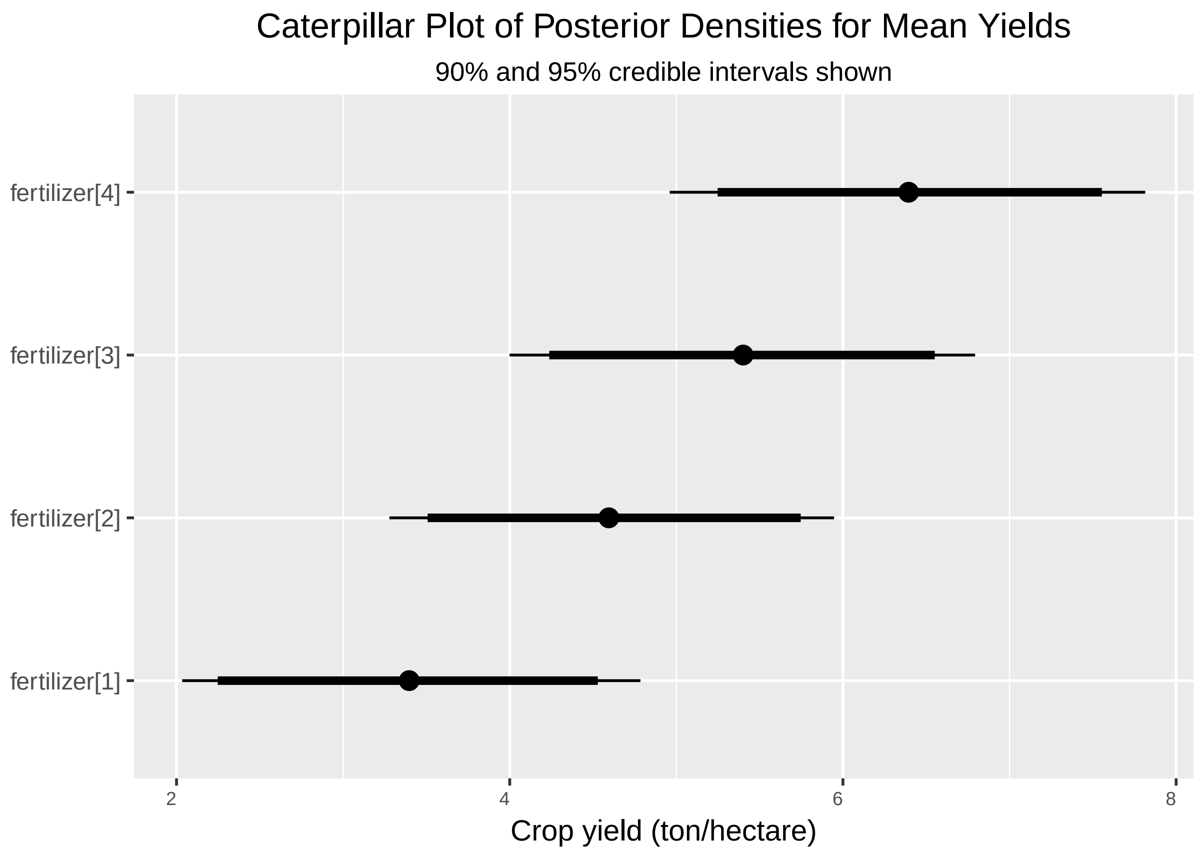
---

### *Part (l) Graphical representations of the simple model*

Above we can see the 95% credible intervals for our simpler model. This model forgoes alpha values and instead leaves us to directly interpret inter-group differences by analysing the mean yield results. We can see that the $\mu$ and $\sigma$ values for the posterior densities of the mean yields closely match their counterpart values from section (f).

As before we can visualise the credible intervals of our posterior distributions using a caterpillar plot:

```
ggs_simple <- ggs(as.mcmc(simple_jags))
ggs_caterpillar(ggs_simple,family = "^fertilizer")+
  ylab("")+
  xlab("Crop yield (ton/hectare)")+
  labs(title = "Caterpillar Plot of Posterior Densities for Mean Yields",
       subtitle = "90% and 95% credible intervals shown")+
  theme(plot.title =element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```



Once again by analysing the credible intervals and caterpillar plot we can deduce that *f4* offers a significantly increased mean yield ovr group *f1*.

### *Part (m) Comparison of Bayesian models*

When comparing Bayesian models of sections (f) and (k) the authors prefer the simpler model of section (k) as it allows us to reach the same conclusions using cleaner, more readable code. The model in section (f) is specifically written to test $H_{0(b)}$ however this can still be tested with our simpler model if we re-write it to the equivalent form of:

$$H_{0(b)} : \mu_1 = \mu_2 = \mu_3 = \mu_4$$

As shown in section (l) both methods produce comparable results, we can support this by seeing how similar the Deviance Information Criterion (DIC) scores for the two models are:

```
Bayesian_anova_inference[[2]]$DIC
```

```
## [1] 80.03357
```

```
simple_jags[[2]]$DIC
```

```
## [1] 79.72612
```

Since neither model is significantly more or less prone to error then we can conclude that the simpler model is preffered thanks to its increased clarity.