MATH501

10823887

2024-04-27

3.1 Machine Learning Task

Machine Learning Part (a):

Explore and visualise the data to gain insights into the relationships between the two predictors and the output variable type:

```
#Data Exploration:
#Use summary() for numerical summaries and
summary(earthquake_data)
```

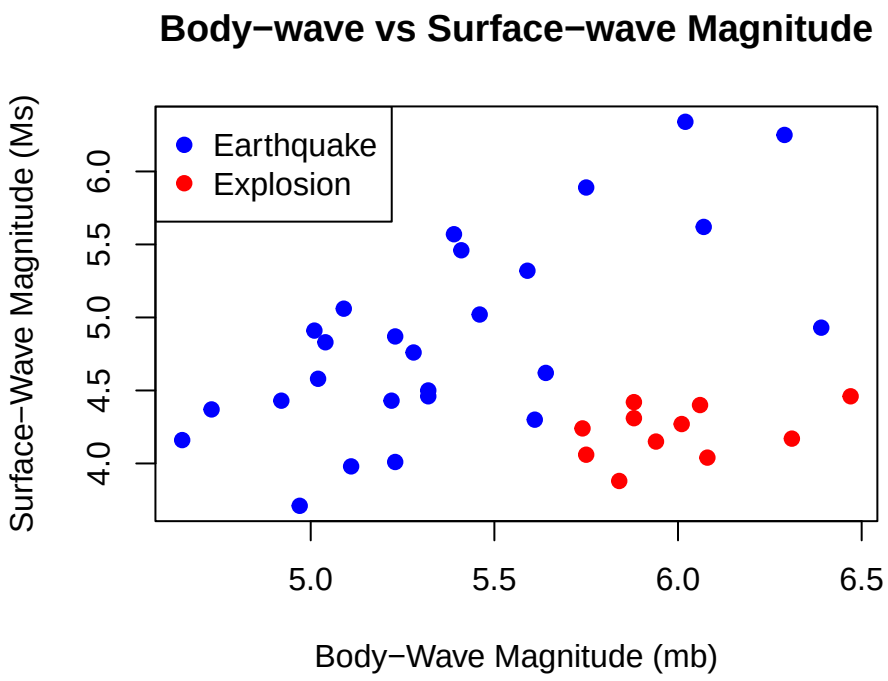
```
##      type                body          surface
##  Length:37          Min.   :4.65   Min.   :3.71
##  Class :character   1st Qu.:5.22   1st Qu.:4.24
##  Mode  :character   Median :5.59   Median :4.46
##                     Mean   :5.56   Mean   :4.67
##                     3rd Qu.:5.94   3rd Qu.:4.93
##                     Max.   :6.47   Max.   :6.34
```

```
nrow(earthquake_data) #number of data points
```

```
## [1] 37
```

Comment on the numerical summaries in the context of the problem:

- The **body** variable has a minimum of 4.65, a maximum of 6.47, a median of 5.59, and a mean of 5.56. This suggests a relatively narrow range of body-wave magnitudes in the dataset, with the majority of observations concentrated around the mean and median values.
- The **surface** variable has a minimum of 3.71, a maximum of 6.34, a median of 4.46, and a mean of 4.67. This shows a quite narrow range, but surface-wave magnitudes tend to be lower than body-wave magnitudes on average.
- The difference between the mean and median for both **body** and **surface** is small, indicating a symmetric distribution of values around the center of the data.



Comment on the graphs in the context of the problem:

- The scatter plot shows a clear distinction in colour between blue points (earthquakes) and red points (explosions). There's a tendency for the red points (explosions) to cluster towards the higher body-wave magnitudes (mb) but lower surface-wave magnitudes (Ms).
- When body-wave magnitudes are the same, the blue points (earthquakes) are more spread out and tend to have higher surface-wave magnitudes than the red points (explosions).
- There seems to be a positive correlation between body-wave and surface-wave magnitudes for earthquakes, which is less pronounced for explosions.

Machine Learning Part(b):

Apply the two selected methods (k-Nearest Neighbors (KNN) and Support Vector Machines (SVM)) to the data to build a classifier to predict the type of explosion(type) based on body and surface.

K-Nearest Neighbors (KNN) Model

Rationale behind choosing k-Nearest Neighbors (KNN) Model:

- KNN is a non-parametric method which is often successful in classification situations where the decision boundary is irregular.
- It requires no assumptions about the shape of the decision boundary, making it suitable for this application where we don't presuppose the relationship between the magnitudes of seismic waves and the type of event.

KNN Model tuning:

For KNN, the main hyperparameter to tune is the number of neighbors k. We can use a simple for-loop to try different values of k and perform Leave-One-Out Cross-Validation (LOOCV) to find the best k.We choose the value of K that gives the lowest error on validation set.

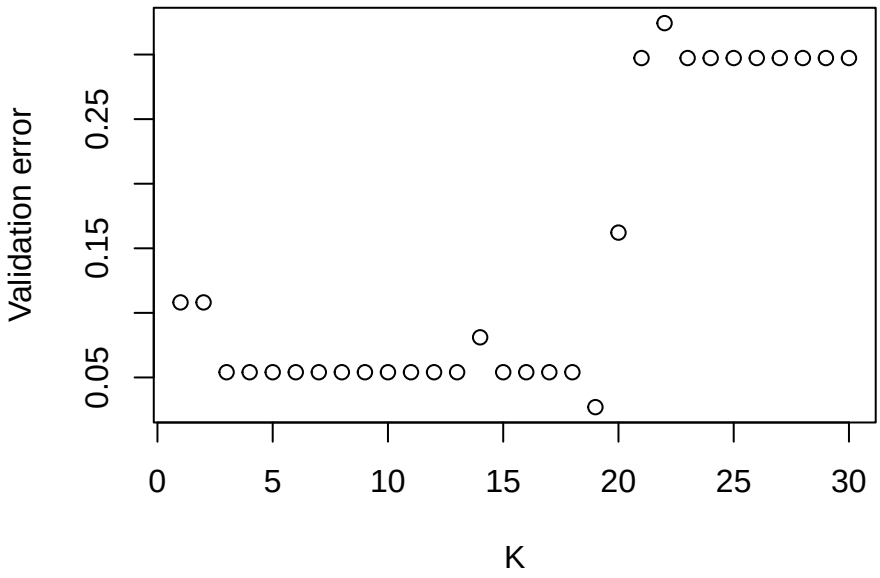
```r
library(class)

set.seed(123)
errors<-rep(0,20) #to store error rates for k from 1 to 20

for(K in 1:30){
  loocv_errors<-rep(0,nrow(earthquake_data)) #to store whether each LOOCV prediction was wrong
  for (i in 1:nrow(earthquake_data)){
    train.data<-earthquake_data[-i,] #leave one out
    test.data<-earthquake_data[i,]
    pred<-knn(train=train.data[,c("body","surface")],
              test=test.data[,c("body","surface")],
              cl=train.data$type,k=K)
    loocv_errors[i]<-as.numeric(pred!=test.data$type) #1 if error, 0 if correct
  }
  errors[K]<-mean(loocv_errors) #LOOCV error rate for k
}
```

Choosing the best k for k-Nearest Neighbors (KNN):

We need to balance between underfitting and overfitting when choosing the best k value. When k is too small, the model can become too complex and sensitive to noise in the data, leading to overfitting. When k is too large, the model can be too simple and may underfit the data, not capturing the underlying trends well enough.

we can use a common **Rule of Thumb** for selecting the number of neighbors k for KNN method, which is to take the square root of the number of data points (37 data points) in the training dataset. Thus, in this case we assume $k = \sqrt{37} \approx 6.08$, but notice that the validation error of k=6 is the same as k=3 (the value when the validation error first reaches the lowest). Despite a smaller k i.e. 3, makes the model sensitive to local variations, which can be good if these variations are meaningful and represent true patterns, but it also makes the model more sensitive to outliers and noise, which can result in a jagged decision boundary that overfits the training data. Meanwhile, a larger k averages more neighbors, which can reduce the effect of noise and lead to a smoother decision boundary. However, we need to bear in mind if k is too large, the model starts to ignore the local structure of the data and may underfit, not capturing important distinctions between classes.

```r
plot(errors[1:30], xlab="K", ylab = "Validation error")
```




```
best_k<-which.min(errors)
print(paste("Minimum Classification error:", min(errors)))
```

```
## [1] "Minimum Classification error: 0.027027027027027"
```

```
print(paste("Best K:", best_k))
```

```
## [1] "Best K: 19"
```

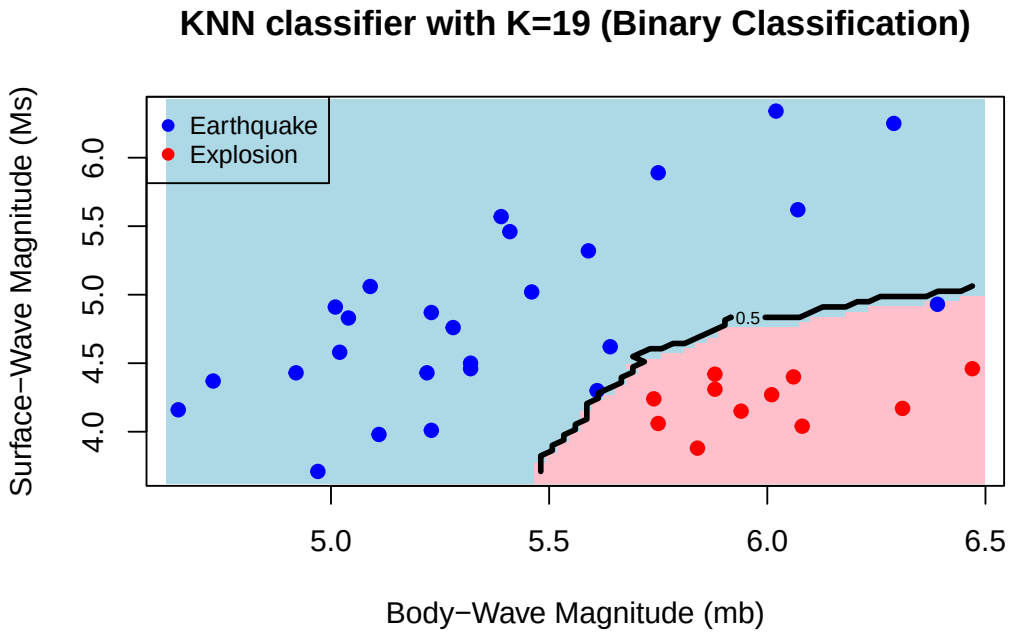
The validation error graph shows that the validation error generally decreases as the number of neighbors k increases up to a point (k=19) where it reaches the lowest validation error. Afterward, increasing k doesn't significantly improve the error rate.

While k=3 might give us a low validation error, it is likely not the global minimum. As we usually following the rule that the validation error as a function of k usually forms a U-shaped curve, showing the bottom of this curve which represents the best trade-off between bias and variance.

In conclusion, while a smaller k could provide a good fit for the training data, it is the generalisability of the model that we care about, which is why we rely on the validation error to guide our choice of k. The lowest validation error indicates the model that is expected to perform the best on data it has not seen before, and thus, k=19 is chosen despite being significantly larger than k=3.

Visualisation of the KNN classifier :

We visualise the resulting classification rule by drawing the class boundary in the scatter plot of the data.



The KNN scatter plot visualises the classification boundary well. The decision regions are coloured accordingly to the predicted class for different points in the feature space. The scatter plot shows the boundary is quite irregular that the relationship between body-wave and surface-wave magnitudes is complex and likely non-linear. KNN is adept at capturing such non-linear relationships because it classifies data points based on the majority vote of their nearest neighbours.

Furthermore, KNN is particularly useful for small datasets. Given that the earthquake dataset comprises only 37 data points, KNN can effectively deal with this limited data size, especially since the feature space is low-dimensional (only two features). The KNN's flexibility is also a key advantage. By tuning the parameter k, we can control the balance between noise sensitivity and the smoothness of the decision boundary.

Lastly, the KNN approach allows for immediate updates to the model as new data is collected, without the need for retraining. This makes KNN an attractive option for evolving datasets where new observations are frequently added.

Support Vector Machine Model

Reasons for Choosing Support Vector Machines (SVM) Model:

- SVM is effective in high-dimensional spaces and in cases where the number of dimensions is greater than the number of samples, which is not exactly our case but still SVMs are known for their effectiveness in binary classification problems.
- SVMs are equipped with the kernel trick, making them versatile for both linear and non-linear decision boundaries. Given that we don't know a priori if the relationship between body-wave and surface-wave magnitudes is linear, this flexibility is beneficial.
- It is robust to outliers and focuses on the points that are the most difficult to classify (support vectors), which can be quite useful when distinguishing between events that are very similar.

Using LOOCV tuning the SVM model to find the best value of Gamma, and best value of cost that gives lowest classification error.

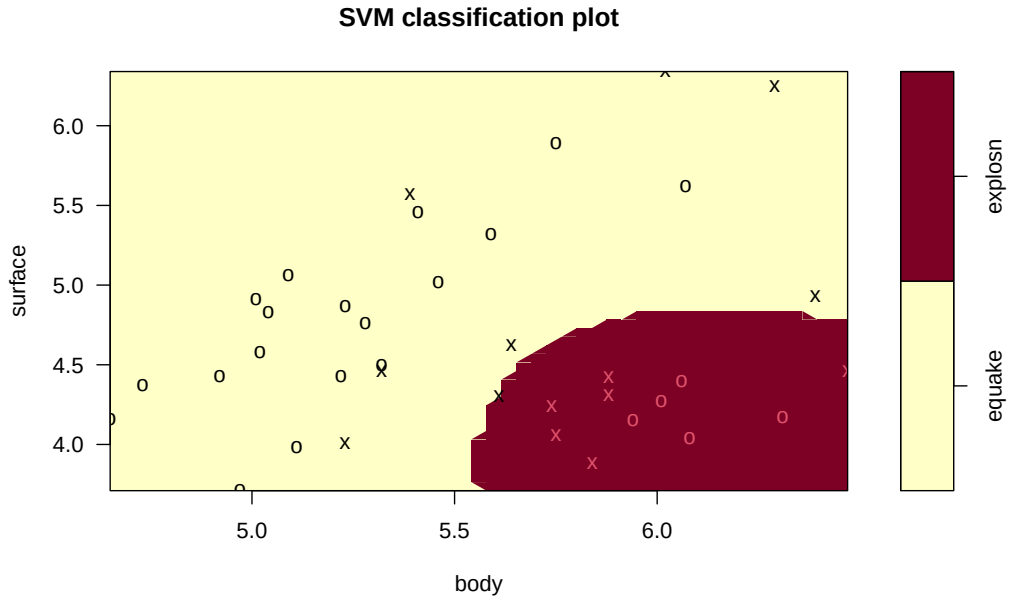
```
library(e1071)

earthquake_data$type <- as.factor(earthquake_data$type)
```

```r
# Define a range for gamma and cost
gamma_range <- 2^(-1:1)
cost_range <- 10^(-1:1)

# Initialise variables for tracking the best parameters and lowest error
best_gamma <- NA
best_cost <- NA
lowest_error <- Inf
all_predictions <- vector("list", length = nrow(earthquake_data)) # To store all predictions

# Leave-One-Out Cross-Validation
for(gamma in gamma_range) {
  for(cost in cost_range) {
    loocv_errors <- numeric(nrow(earthquake_data))
    for(i in 1:nrow(earthquake_data)) {
      # Define training and test sets for LOOCV
      train_set <- earthquake_data[-i, ]
      test_set <- earthquake_data[i, , drop = FALSE]
      # Train the model
      svm_model <- svm(type ~ surface+body, data = train_set, kernel = "radial", gamma = gamma, cost = c
      # Test the model on the left-out data point
      prediction <- predict(svm_model, newdata = test_set)
      all_predictions[[i]] <- prediction  # Correctly store the prediction
      # Record if there was an error
      loocv_errors[i] <- ifelse(prediction != test_set$type, 1, 0)
    }
    # Calculate average error for the current combination of gamma and cost
    avg_error <- mean(loocv_errors)
    # Update best parameters if current error is lower
    if(avg_error < lowest_error) {
      lowest_error <- avg_error
      best_gamma <- gamma
      best_cost <- cost
    }
  }
}

# Output the best parameters
cat("Best Gamma:", best_gamma, "Best Cost:", best_cost, "Lowest Error:", lowest_error)
```

```
## Best Gamma: 0.5 Best Cost: 1 Lowest Error: 0.05405405
```

```r
# Flatten all_predictions since it's stored in a list
all_predictions <- unlist(all_predictions)
```

The SVM model tuning finds the best gamma and cost parameters through a grid search using LOOCV. The chosen gamma and cost reflect the trade-off between model complexity and the risk of overfitting. The lowest error achieved by the SVM model indicates that it performs quite well, though not perfectly. This suggests that while the SVM can capture the distinction between earthquakes and explosions, there may be some overlap in the feature space that makes perfect separation difficult.



The SVM classification plot shows the decision boundary as a relatively smooth curve, which represents the high-dimensional plane found by the SVM to separate the classes. The SVM's ability to create a margin around the decision boundary provides a clear separation between the classes, albeit with some exceptions.

Machine Learning Part (c):

Performance Metrics:

Look at the accuracy, precision, recall, F1 score, and validation errors. The model with better scores in these metrics generally performs better.

```
best_k <- which.min(errors)  # As an example, finding the minimum error across all ks

# Re-run LOOCV for the best_k to collect predictions
predicted_labels <- character(nrow(earthquake_data))  # To store predictions
for (i in 1:nrow(earthquake_data)) {
    train.data <- earthquake_data[-i, ]
    test.data <- earthquake_data[i, , drop = FALSE]
    predicted_labels[i] <- knn(train = train.data[, c("body", "surface")],
                               test = test.data[, c("body", "surface")],
                               cl = train.data$type, k = best_k)
}

# Generate the confusion matrix
conf_matrix_knn <- table(Predicted = predicted_labels, Actual = earthquake_data$type)

# Calculate accuracy
accuracy_knn <- sum(diag(conf_matrix_knn)) / sum(conf_matrix_knn)

# Calculate precision and recall for each class
precision_knn <- diag(conf_matrix_knn) / colSums(conf_matrix_knn)
recall_knn <- diag(conf_matrix_knn) / rowSums(conf_matrix_knn)

# Calculate F1 score for each class
```

```r
f1_score_knn <- 2 * (precision_knn * recall_knn) / (precision_knn + recall_knn)
```

KNN Performance Metrics:

```
## [1] "KNN Accuracy: 0.972972972972973"

## [1] "KNN Precision: 0.961538461538462" "KNN Precision: 1"

## [1] "KNN Recall: 1"                    "KNN Recall: 0.916666666666667"

## [1] "KNN F1 Score: 0.980392156862745" "KNN F1 Score: 0.956521739130435"
```

KNN has shown excellent results with a very low classification error of approximately 0.027 and an accuracy of about 97.3%. It demonstrates high precision and recall, achieving almost perfect scores, particularly with one class where precision and recall are both 100%.

- **Advantages:** Highly effective in capturing local variations in the data due to its instance-based learning approach. This is evident from its higher precision and recall scores.
- **Disadvantages:** Computationally intensive at the prediction phase, as it must compute the distance between test points and all training points. It also suffers from the curse of dimensionality, although less of a concern here with only two features.

```r
# Now calculate the performance metrics for the best parameters
conf_matrix_svm <- table(Predicted = all_predictions, Actual = earthquake_data$type)

# Calculate accuracy
accuracy_svm <- sum(diag(conf_matrix_svm)) / sum(conf_matrix_svm)

# Calculate precision and recall for each class
precision_svm <- diag(conf_matrix_svm) / colSums(conf_matrix_svm)
recall_svm <- diag(conf_matrix_svm) / rowSums(conf_matrix_svm)

# Calculate F1 score for each class
f1_score_svm <- 2 * (precision_svm * recall_svm) / (precision_svm + recall_svm)
```

SVM Performance Metrics:

```
## [1] "SVM Accuracy: 0.945945945945946"

## [1] "SVM Precision: 0.961538461538462" "SVM Precision: 0.909090909090909"

## [1] "SVM Recall: 0.961538461538462" "SVM Recall: 0.909090909090909"

## [1] "SVM F1 Score: 0.961538461538462" "SVM F1 Score: 0.909090909090909"
```

SVM has a slightly higher classification error of about 0.054 and an accuracy of about 94.6%. While still high, its precision and recall scores are slightly lower compared to k-NN, indicating a minor reduction in its ability to identify both classes perfectly.

- **Advantages:** Provides a smooth and generalisable decision boundary, which helps in making it robust against overfitting. It can be useful when expecting data with less clear-cut distinctions or more noise.
- **Disadvantages:** Parameter tuning (gamma and cost) can be complex and computationally expensive. The smooth boundary may also oversimplify some local variations in the data.

Recommendation:

- **KNN** would be recommended as the better classifier for the analysis of this dataset. It has superior accuracy, precision, recall, and F1 scores, which proves its robustness in correctly identifying and

classifying seismic events. This model's ability to closely follow the intricate patterns in the seismic magnitudes makes it a more precise tool for this task, especially when precision in capturing local patterns is crucial.

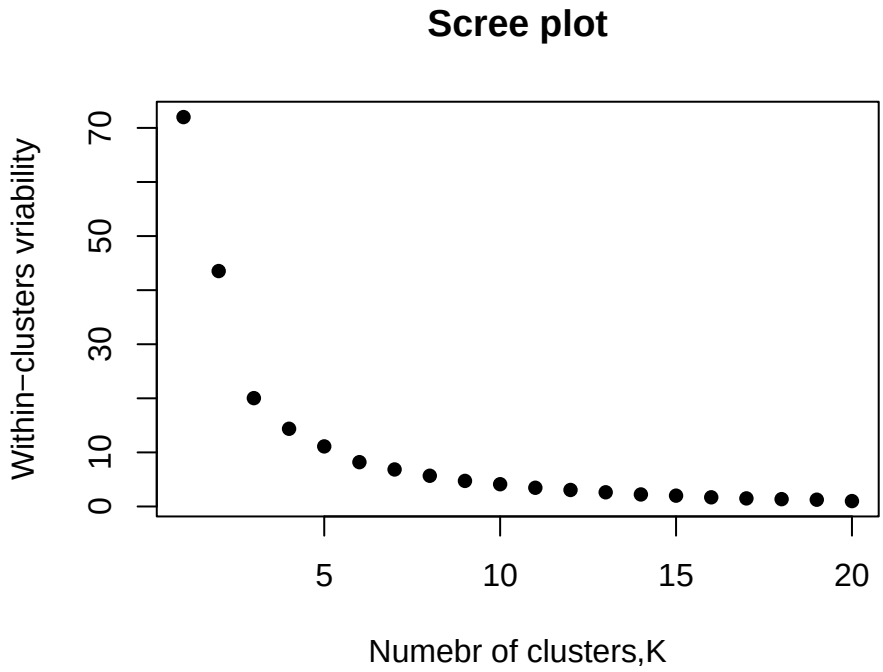
- However, **SVM** is also a strong contender, particularly if future applications of the model require a balance between accuracy and computational efficiency during training, or if the dataset size significantly increases.

In conclusion, while both models are good options, KNN's performance in this case is more aligned with the need for high accuracy and detailed recognition of patterns in seismic data, making it the preferred choice for this specific application.

Machine Learning Part (d):

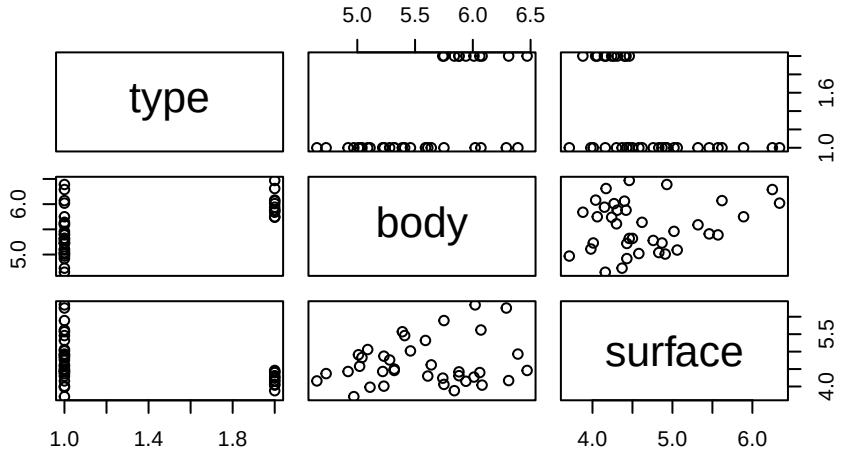
Apply the K-means algorithm using the two variables body and surface and ignoring the variable type to cluster the data.

```
plot(v,xlab="Numebr of clusters,K",
     ylab="Within-clusters vriability",
     pch=16,
     main="Scree plot")
```



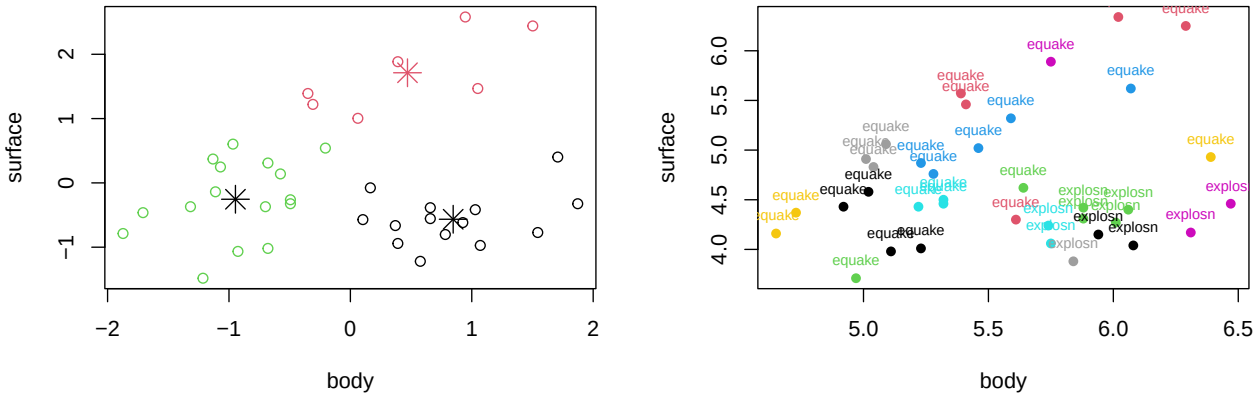
The scree plot suggests that the optimal number of clusters for the earthquake dataset lies around $k = 3$ or $k = 4$. This is indicated by the 'elbow' where the rate of decrease in within-cluster variability levels off, implying that additional clusters beyond this point yield minimal improvement in data segregation. Consequently, setting k to 3 or 4 would provide a balance between over-segmenting the data and failing to distinguish between different seismic events, thus presenting a reasonable choice for effective clustering with K-means.

```
#Create pairwise scatter plots of all variables with the command paris(earthquake_data)
pairs(earthquake_data)
```



The pairwise scatter plot analysis of body and surface wave magnitudes reveals distinct distributions and relationships pivotal for K-means clustering. It suggests potential patterns which K-means can exploit to differentiate seismic events. Visual clusters within the plot hint at the algorithm's capability to segregate data effectively, even without known labels, and outlier detection within these plots is critical for refining cluster quality. This preliminary visualisation underpins K-means as a promising tool for classifying seismic events when the type of explosion is unknown.

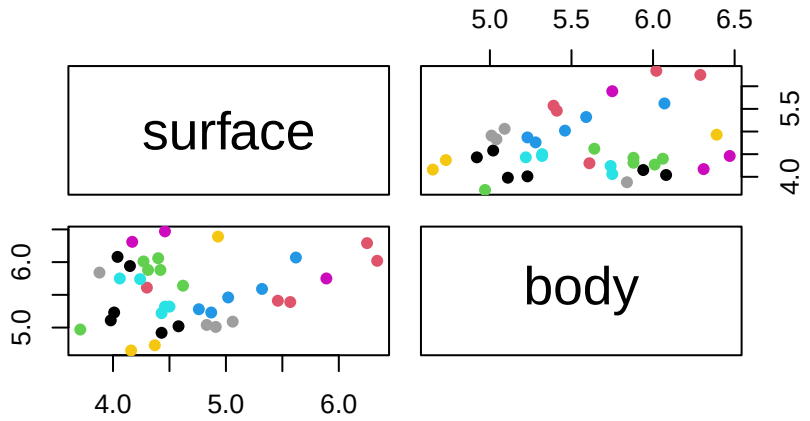
```
par(mfrow=c(1,2))
# Visualise the clusters
plot(sd.earthquake_data, col = km.sd.earthquake_data.k3$cluster)
points(km.sd.earthquake_data.k3$centers, col = 1:2, pch = 8, cex = 2)

#Explore the resulting clusters by analysing pairwise scatter plots of the four variables with added co

plot(body,surface,col=km.type$cluster,pch=16)
text(earthquake_data$body, earthquake_data$surface, labels=earthquake_data$type,
     cex=0.7, pos=3, col=km.type$cluster)
```



The cluster visualisations with $k = 3$ show a well-defined grouping of data points, with cluster centroids indicating the core of each cluster. We can tell that the K-means cluster assignments overlay with the actual labels on the scatter plot , this alignment suggests that the K-means clustering is effectively distinguishing between earthquakes and explosions. This evidence supports the effectiveness of K-means in classifying seismic events using body and surface wave measurements alone, validating its use as a reliable tool for unsupervised classification in the absence of explicit event labels.

```
# Adding cluster assignments to the earthquake_data dataset for plotting
earthquake_data$Cluster <- km.type$cluster
```

```r
# Use the pairs() function to create pairwise scatter plots
pairs(~surface+body, data = earthquake_data, col = earthquake_data$Cluster , pch = 16)
```



The colour-coded scatter plots with cluster assignments offer a clear visual distinction between groups, indicating that K-means clustering has successfully identified separable clusters in the dataset. We can conclude that even without explicit event type labels, K-means can differentiate between seismic events based on body and surface wave measurements, which is promising for unsupervised classification tasks such as distinguishing earthquakes from explosions.

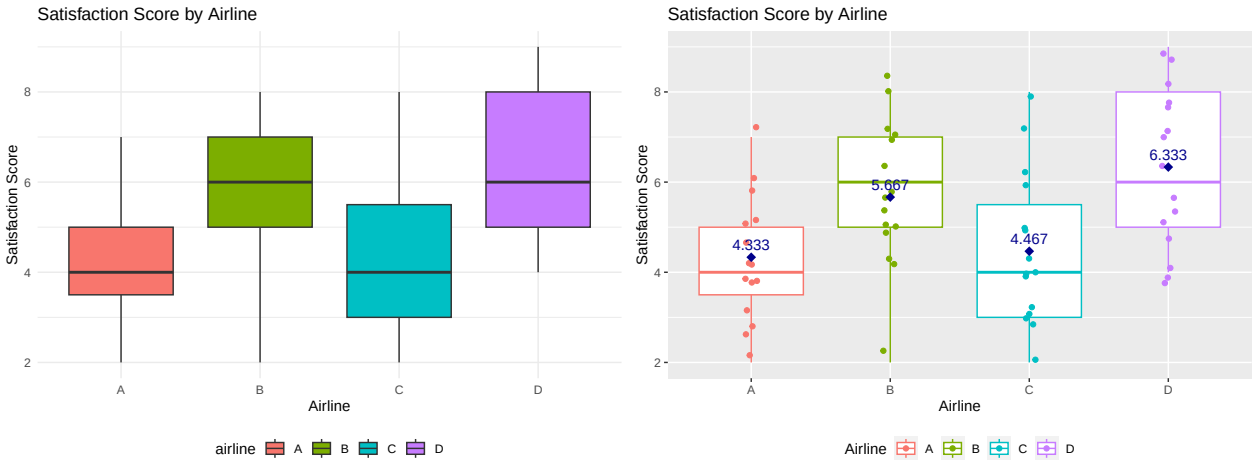
3.2 Bayesian Statistics Task (with some frequentist analysis)

Bayesian Statistics Part (a)∗:

Use ggplot2 to visualise insightfully these data. What can you conclude from the plot(s)?

```r
library(ggplot2)
library(dplyr)
library(patchwork)
airline_data <- read.csv("airline.csv")

#Use ggplot2 to visualize the data:
p1 <- ggplot(airline_data, aes(x = airline,
                          y = satisfactionscore,
                          fill = airline)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Satisfaction Score by Airline",
       x = "Airline",
       y = "Satisfaction Score")+
  theme(legend.position = "bottom")

#enhanced boxplot
p2 <- ggplot(airline_data, aes(x = airline, y = satisfactionscore,
                    colour = airline)) +
  geom_boxplot(varwidth = TRUE) +
  geom_jitter(width = 0.05) +
  labs(title = "Satisfaction Score by Airline",
      x = "Airline",
       y = "Satisfaction Score",
       colour = "Airline") +
```

```
  stat_summary(fun = mean,
               colour="darkblue",
               geom = "point",
               shape = 18,
               size = 3,
               show.legend = FALSE) +
  stat_summary(fun = mean,
               colour = "darkblue",
               geom = "text",
               show.legend = FALSE,
               vjust = -0.7,
               aes(label = round(..y.., digits = 3))) +
  theme(legend.position = "bottom")

# Combine plots side by side
p1 + p2
```



From the plots, we can gain a nuanced understanding of customer satisfaction across the four airlines:

- Airline D has the highest median satisfaction score at 6.333, which suggests that the central customer experience is most favorable with this airline. Airline B follows with a median score of 5.667. Airline C's median is slightly above Airline A's, at 4.467 compared to 4.333, indicating that customers' general experiences with these airlines are less favorable compared to Airlines B and D.
- The interquartile range (IQR) for Airline D is larger than the others, implying that customer satisfaction is more varied. Some customers are very satisfied, while others much less so. In contrast, Airlines B and C have a narrower IQR, suggesting that customers generally have a more consistent experience.
- Airline B displays the widest range of satisfaction scores, indicating a diverse set of customer experiences from low to high satisfaction. Airlines A and C exhibit potential outliers, with a few scores significantly lower than the rest, indicating some extremely dissatisfied customers.
- The consistency of customer satisfaction can be inferred from the compactness of the boxes. Airline C shows a relatively compact box, suggesting a more uniform customer satisfaction level. Meanwhile, the spread of scores in Airline B, while having a wide range, centers around a higher median, which may reflect generally favorable experiences despite some dissatisfied customers.

These insights provide a preliminary understanding of customer satisfaction. While Airline D leads in terms of median satisfaction, its wider variability suggests that customer experiences are not consistently high. Airlines B and C appear to offer a more uniform level of service, as reflected in their narrower IQRs. Airline

A has the lowest median satisfaction and exhibits some outliers indicating particularly dissatisfied customers. However, without statistical testing, we cannot determine if these observed differences are significant, particularly between the lower scoring airlines (A and C) and the higher scoring ones (B and D). To make definitive conclusions about the differences in satisfaction scores between airlines, a statistical test such as ANOVA is required.

Bayesian Statistics Part (b)∗:

The parameter α_4 represents the difference between the mean satisfaction score of the fourth airline (μ_{4j}) and the overall mean satisfaction score (μ_1) across all airlines.

The model suggests that:

- μ_{1j} (mean satisfaction score for airline 1) is set as the baseline (μ_1).
- μ_{2j} (mean satisfaction score for airline 2) is the baseline plus α_2, the effect of being airline 2 as opposed to airline 1.
- μ_{3j} (mean satisfaction score for airline 3) is the baseline plus α_3, the effect of being airline 3.
- μ_{4j} (mean satisfaction score for airline 4) is the baseline plus α_4, which is the specific effect of being airline 4.

So, α_4 quantifies how the average score of airline 4 deviates from the baseline average score of airline 1. If α_4 is positive, it means that airline 4 has a higher average satisfaction score than the baseline, and if it is negative, then airline 4 has a lower average satisfaction score than the baseline. If α_4 is zero, it would suggest that the average satisfaction score of airline 4 is the same as the baseline. This is under the assumption that the satisfaction scores are normally distributed with a common variance σ^2 for all airlines.

Bayesian Statistics Part (c)∗:

The results of fitting a linear model to satisfaction scores based on airlines, and the subsequent ANOVA test for differences in means between the airlines.

```
##
## Call:
## lm(formula = satisfactionscore ~ airline, data = airline_data)
##
## Coefficients:
## (Intercept)     airlineB     airlineC     airlineD
##      4.3333       1.3333       0.1333       2.0000
```

```
## (Intercept)    airlineB    airlineC    airlineD
##   4.3333333   1.3333333   0.1333333   2.0000000
```

```
## Analysis of Variance Table
##
## Response: satisfactionscore
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## airline    3  41.867 13.9556    5.29 0.00278 **
## Residuals 56 147.733  2.6381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## [1] 0.002780377
```

```
## [1] "0.002780377"
```

```
## [1] "0.00278"
```

Here's what we can conclude from the output:

Model Coefficients:

- The (**Intercept**) coefficient is 4.333, which is the estimated mean satisfaction score for the baseline category (presumably airline A).
- The coefficients for **airlineB** (1.333), **airlineC** (0.133), and **airlineD** (2.000) represent the estimated differences in mean satisfaction scores compared to airline A.

In terms of the model:

- $\widehat{\mu}_1 = 4.333$ for airline A.
- $\widehat{\mu}_2 = \widehat{\mu}_1 + \widehat{\alpha}_2 = 4.333 + 1.333 = 5.666$ for airline B.
- $\widehat{\mu}_3 = \widehat{\mu}_1 + \widehat{\alpha}_3 = 4.333 + 0.133 = 4.466$ for airline C.
- $\widehat{\mu}_4 = \widehat{\mu}_1 + \widehat{\alpha}_4 = 4.333 + 2.000 = 6.333$ for airline D.

ANOVA Test:

- The F-statistic is 5.29, and the associated p-value is 0.0028. This p-value is less than the 0.05 alpha level, indicating that there are statistically significant differences in satisfaction scores among the airlines.

The coefficients suggest that airline B and airline D have higher satisfaction scores than airline A by 1.333 and 2.000 points, respectively, on average. Airline C is almost similar to airline A with only a 0.133 point difference on average. Given the low p-value in the ANOVA test, we have strong evidence to reject the null hypothesis that all airlines have the same mean satisfaction score. The conclusion is that there are significant differences in the average satisfaction scores across airlines. These results, along with the estimated coefficients, provide grounds to conclude that the satisfaction scores do indeed vary by airline.

Bayesian Statistics Part (d)∗:

```
## (Intercept)    airlineB    airlineC    airlineD
##     4.33333     1.33333     0.13333     2.00000
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## airline      3   41.9   13.96    5.29 0.0028 **
## Residuals   56  147.7    2.64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = satisfactionscore ~ airline, data = airline_data)
##
## $airline
##         diff      lwr     upr   p adj
## B-A  1.33333 -0.23708 2.90375 0.12297
## C-A  0.13333 -1.43708 1.70375 0.99595
## D-A  2.00000  0.42959 3.57041 0.00722
## C-B -1.20000 -2.77041 0.37041 0.19178
## D-B  0.66667 -0.90375 2.23708 0.67635
## D-C  1.86667  0.29625 3.43708 0.01366
```

Hypotheses for Each Pair of Airlines

1. **B vs. A**
   - **Null Hypothesis (H0)**: $\mu_B = \mu_A$ (There is no difference in satisfaction scores between airline B and airline A)

- **Alternative Hypothesis (H1)**: $\mu_B \neq \mu_A$ (There is a difference in satisfaction scores between airline B and airline A)

2. **C vs. A**
   - **H0**: $\mu_C = \mu_A$
   - **H1**: $\mu_C \neq \mu_A$
3. **D vs. A**
   - **H0**: $\mu_D = \mu_A$
   - **H1**: $\mu_D \neq \mu_A$
4. **C vs. B**
   - **H0**: $\mu_C = \mu_B$
   - **H1**: $\mu_C \neq \mu_B$
5. **D vs. B**
   - **H0**: $\mu_D = \mu_B$
   - **H1**: $\mu_D \neq \mu_B$
6. **D vs. C**
   - **H0**: $\mu_D = \mu_C$
   - **H1**: $\mu_D \neq \mu_C$

Conclusions Based on the Tukey HSD Test Results

- **B vs. A**: The p-value of 0.12297 suggests that we **fail to reject the null hypothesis**. There is no statistically significant difference in satisfaction scores between airlines B and A.
- **C vs. A**: The p-value of 0.99595 suggests that we **fail to reject the null hypothesis**. There is no statistically significant difference in satisfaction scores between airlines C and A.
- **D vs. A**: The p-value of 0.00722 suggests that we **reject the null hypothesis**. There is a statistically significant difference in satisfaction scores, with airline D having higher scores compared to airline A.
- **C vs. B**: The p-value of 0.19178 suggests that we **fail to reject the null hypothesis**. There is no statistically significant difference in satisfaction scores between airlines C and B.
- **D vs. B**: The p-value of 0.67635 suggests that we **fail to reject the null hypothesis**. There is no statistically significant difference in satisfaction scores between airlines D and B.
- **D vs. C**: The p-value of 0.01366 suggests that we **reject the null hypothesis**. There is a statistically significant difference in satisfaction scores, with airline D having higher scores compared to airline C.

Overall Summary

The significant differences are observed only in comparisons involving Airline D against Airlines A and C, suggesting that Airline D provides a superior customer satisfaction experience compared to these airlines. No significant differences were detected between the other pairs, indicating similar levels of satisfaction among them. These insights can be useful for strategic improvements and targeted customer service enhancements where necessary.

Bayesian Statistics Part (e)∗:

Hypotheses Formula:

- **Null Hypothesis ($H0$):**
  - $\mu_D \leq (\mu_B + \mu_C)/2 + 3$
  - The satisfaction score for airline D is three points or less higher than the average satisfaction score for airlines B and C.
- **Alternative Hypothesis ($H1$):**
  - $\mu_D > (\mu_B + \mu_C)/2 + 3$
  - The satisfaction score for airline D is more than three points higher than the average satisfaction score for airlines B and C.

```r
library(multcomp)

# Fit the linear model without an intercept
m_mu <- lm(satisfactionscore ~ airline - 1, data = airline_data)

# Is mu_D is 3 greater than
# the average (B + C) / 2 of mu_2 and mu_3?
# H_0: mu_D - (mu_B + mu_C) / 2 <= 3
# H_1: mu_D - (mu_B + mu_C) / 2 > 3
ght <- glht(m_mu,
            linfct = c("(airlineD - (airlineB + airlineC)/2) <= 3"))

# Display the results
summary(ght)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = satisfactionscore ~ airline - 1, data = airline_data)
##
## Linear Hypotheses:
##                                            Estimate Std. Error t value Pr(>t)
## (airlineD - (airlineB + airlineC)/2) <= 3    1.267      0.514   -3.37      1
## (Adjusted p values reported -- single-step method)
```

```
## [1] "Do not reject the null hypothesis:Airline D's mean satisfaction score is not more than 3 points
```

3.2.2 Second Sub-Task: Bayesian Two-ways Analysis of Variance

Bayesian Statistics Part (f)***:

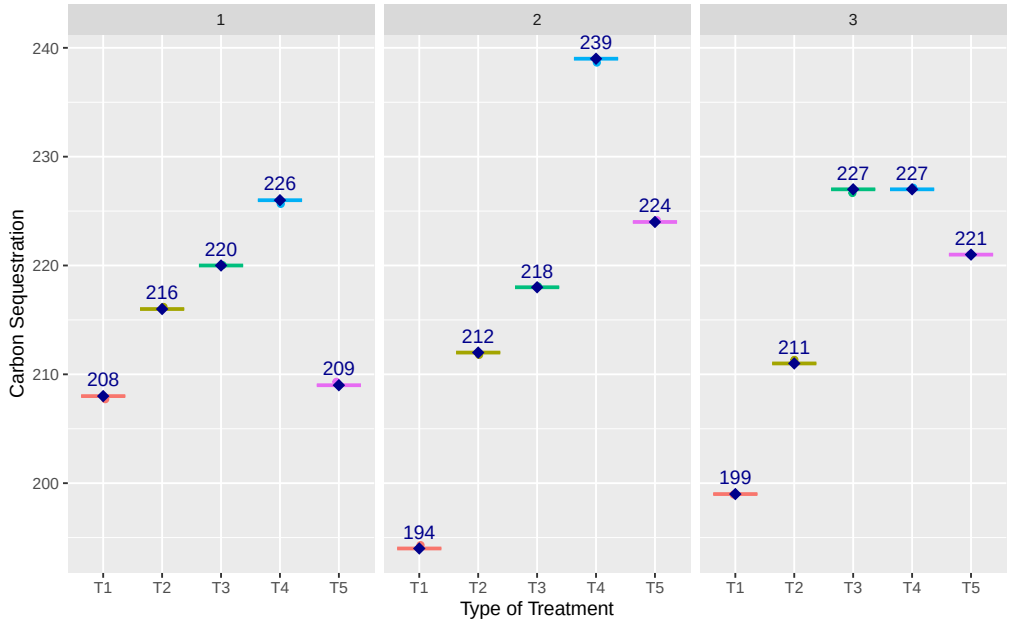
```r
# Carbon sequestration scores
carbon_sequestration <- c(
  208, 216, 220, 226, 209, # Field 1
  194, 212, 218, 239, 224, # Field 2
  199, 211, 227, 227, 221  # Field 3
)

# Corresponding fields and treatments
fields <- factor(rep(1:3, each = 5)) # 'each = 5' because there are 5 treatments per field

# Treatment factor (assuming the order T1 to T5 for each field as per the screenshot)
treatments <- factor(rep(c("T1", "T2", "T3", "T4", "T5"), times = 3)) # 'times = 3' because there are 3

# Create the data frame
carbon_data <- data.frame(carbon_sequestration, fields, treatments)
head(carbon_data)
```

```
##   carbon_sequestration fields treatments
## 1                  208      1         T1
## 2                  216      1         T2
## 3                  220      1         T3
## 4                  226      1         T4
## 5                  209      1         T5
```

```
## 6                 194      2         T1
```

```
str(carbon_data)
```

```
## 'data.frame':    15 obs. of  3 variables:
##  $ carbon_sequestration: num  208 216 220 226 209 194 212 218 239 224 ...
##  $ fields              : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 2 2 2 2 2 ...
##  $ treatments          : Factor w/ 5 levels "T1","T2","T3",..: 1 2 3 4 5 1 2 3 4 5 ...
```



Write jags/BUGS code to perform inference about the model:

```
# Now we can code our model in BUGS
#
# We will use y for satisfaction score, and
# group for treatments:
# Extract the carbon sequestration data for the BUGS model
y <- carbon_data$carbon_sequestration
y
```

```
##  [1] 208 216 220 226 209 194 212 218 239 224 199 211 227 227 221
```

```
# Extract the numeric codes for the fields
field <- as.numeric(carbon_data$fields)
field
```

```
##  [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
```

```
# Extract the numeric codes for the treatments
treatment <- as.numeric(carbon_data$treatments)
treatment
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

```
#
# Prepare the data for the JAGS sampler
```

```r
n <- length(y)  # Total number of observations

# Number of fields and treatments
F <- 3  # Assuming there are 3 fields
T <- 5  # Assuming there are 5 types of treatments

# All the required data
data_anova <- list(
  y = y,
  field = field,
  treatment = treatment,
  n = n,
  F = F,
  T = T
)

# Here is the ****** BUGS code *******
# in which m stands for mu
#
Bayesian_anova <- function() {
  # Data model part
  for (k in 1:n) {
    # Response variable,, y, is a vector, not a matrix
    y[k] ~ dnorm(mu[k], tau)

    # Underlying mean for combination of field and treatment
    mu[k] <- m + alpha[field[k]] + beta[treatment[k]]
  }

  # Priors on unknown parameters
  m ~ dnorm(0.0, 1.0E-4) # Prior on m

  # Constraints for identifiability
  # On alpha (field effects), with alpha[1] being the reference level
  alpha[1] <- 0  # Corner constraint for reference field
  for (i in 2:F) {
    alpha[i] ~ dnorm(0.0, 1.0E-4)  # Prior on non-constrained alphas (field effects)
  }

  # On beta (treatment effects), with beta[1] being the reference level
  beta[1] <- 0  # Corner constraint for reference treatment
  for (j in 2:T) {
    beta[j] ~ dnorm(0.0, 1.0E-4)  # Prior on non-constrained betas (treatment effects)
  }

  # Prior on tau (precision of the model)
  tau ~ dgamma(1.0E-3, 1.0E-3)  # Prior on tau

  # Derived quantity: sigma (standard deviation)
  sigma <- 1.0 / sqrt(tau)  # Convert precision to standard deviation
}

library(R2jags)
```

```r
# Specify the model file or directly use the function containing the model
# If Bayesian_anova is a function that defines the model, use type='source'
Bayesian_anova_inference <- jags(
  data = data_anova,
  parameters.to.save = c("m", "alpha", "beta", "sigma", "tau"),
  n.iter = 100000, # Number of iterations per chain
  n.chains = 3, # Number of chains
  n.burnin=5000, # Number of burn-in samples
  n.thin=10, #Thinning rate
  DIC=TRUE,
  model.file = Bayesian_anova
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 15
##    Unobserved stochastic nodes: 8
##    Total graph size: 77
##
## Initializing model
```

```r
# Check output, diagnostics, and summary of the MCMC results
# Posterior mean, standard deviation,
# posterior median from the 50% column
# 95% credible intervals from the columns 2.5% and 97.5%
print(Bayesian_anova_inference,intervals = c(0.025, 0.5, 0.975))
```

```
## Inference for Bugs model at "/var/folders/ls/d_6cnrhs3yz5fw356d6dgpqr0000gn/T//Rtmpk537vd/model23cc5
##  3 chains, each with 1e+05 iterations (first 5000 discarded), n.thin = 10
##  n.sims = 28500 iterations saved
##          mu.vect sd.vect    2.5%     50%   97.5%  Rhat n.eff
## alpha[1]   0.000   0.000   0.000   0.000   0.000 1.000     1
## alpha[2]   1.797   5.026  -8.078   1.765  11.892 1.001 18000
## alpha[3]   1.452   5.036  -8.589   1.443  11.460 1.001 28000
## beta[1]    0.000   0.000   0.000   0.000   0.000 1.000     1
## beta[2]   12.849   6.412   0.117  12.837  25.760 1.001  9100
## beta[3]   21.469   6.399   8.942  21.403  34.337 1.001 26000
## beta[4]   30.452   6.416  17.644  30.389  43.391 1.001 17000
## beta[5]   17.829   6.475   4.888  17.850  31.013 1.001 10000
## m        199.069   5.367 188.248 199.133 209.626 1.001 28000
## sigma      7.623   2.185   4.657   7.206  12.983 1.001 28000
## tau        0.021   0.010   0.006   0.019   0.046 1.001 28000
## deviance 102.477   5.715  94.278 101.514 116.203 1.001 28000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 16.3 and DIC = 118.8
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Summarising Posterior Probability Density Functions: Here is a summary of the posterior probability density functions for the parameters μ, αi, βj, and τ, which include the posterior means, medians, and 95% credible intervals:

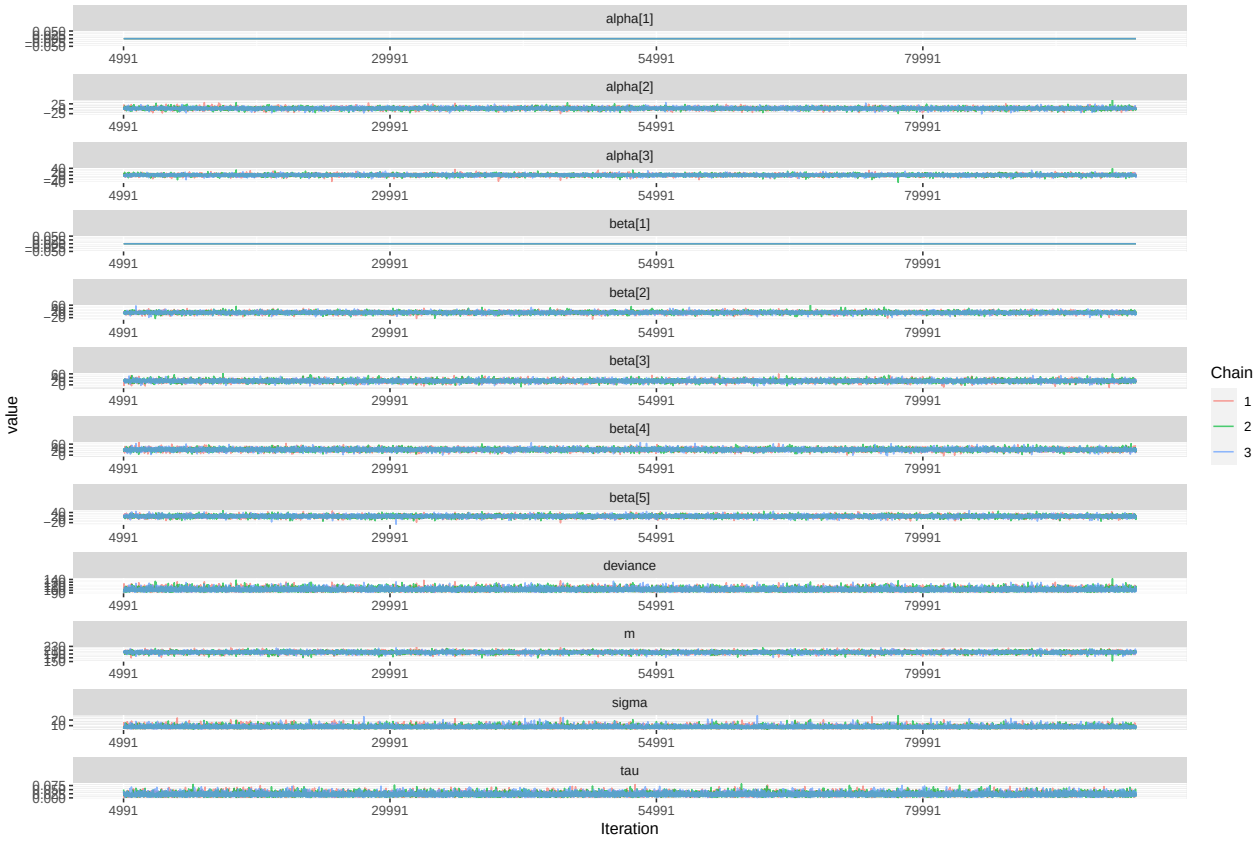
- **Overall Mean (μ)**: Posterior mean carbon sequestration is 198.944 units, highly likely between 188.048 and 209.540 units (95% CI).
- **Field Effects ($\alpha 2$ and $\alpha 3$)**: Show a positive deviation from the baseline ($\alpha 1$ set at 0), with credible intervals suggesting field location influences carbon sequestration but with some uncertainty, given the intervals include negative values.
- **Treatment Effects ($\beta 2$-$\beta 5$)**: All treatments differ significantly from the baseline ($\beta 1$), with $\beta 4$ showing the largest increase in sequestration, all credible intervals being above zero.
- **Precision (τ)**: The mean of 0.021 indicates moderate variability around the mean carbon sequestration levels, confirmed by a standard deviation (σ) mean of 7.659.
- **Model Diagnostics**: Rhat values near 1 and high effective sample sizes demonstrate a good chain convergence and reliable parameter estimates.
- **Model Fit**: A DIC of 119.0 suggests a good fit, with lower values indicating better predictive accuracy.

Field and treatment variations both play roles in sequestration levels, as shown by non-zero intervals for $\alpha 2$, $\alpha 3$, and all β parameters except the baseline.

Bayesian Statistics Part (g)**:

Traceplots:

```
# Graphical summaries
library(ggmcmc)
library(dplyr)
#
# Prepare the jags samples
Bayesian_anova_inference.mcmc <- as.mcmc(Bayesian_anova_inference)
Bayesian_anova_inference.ggs <- ggs(Bayesian_anova_inference.mcmc)
#
# Traceplots of samples from the posterior distribution
#
ggs_traceplot(Bayesian_anova_inference.ggs)
```

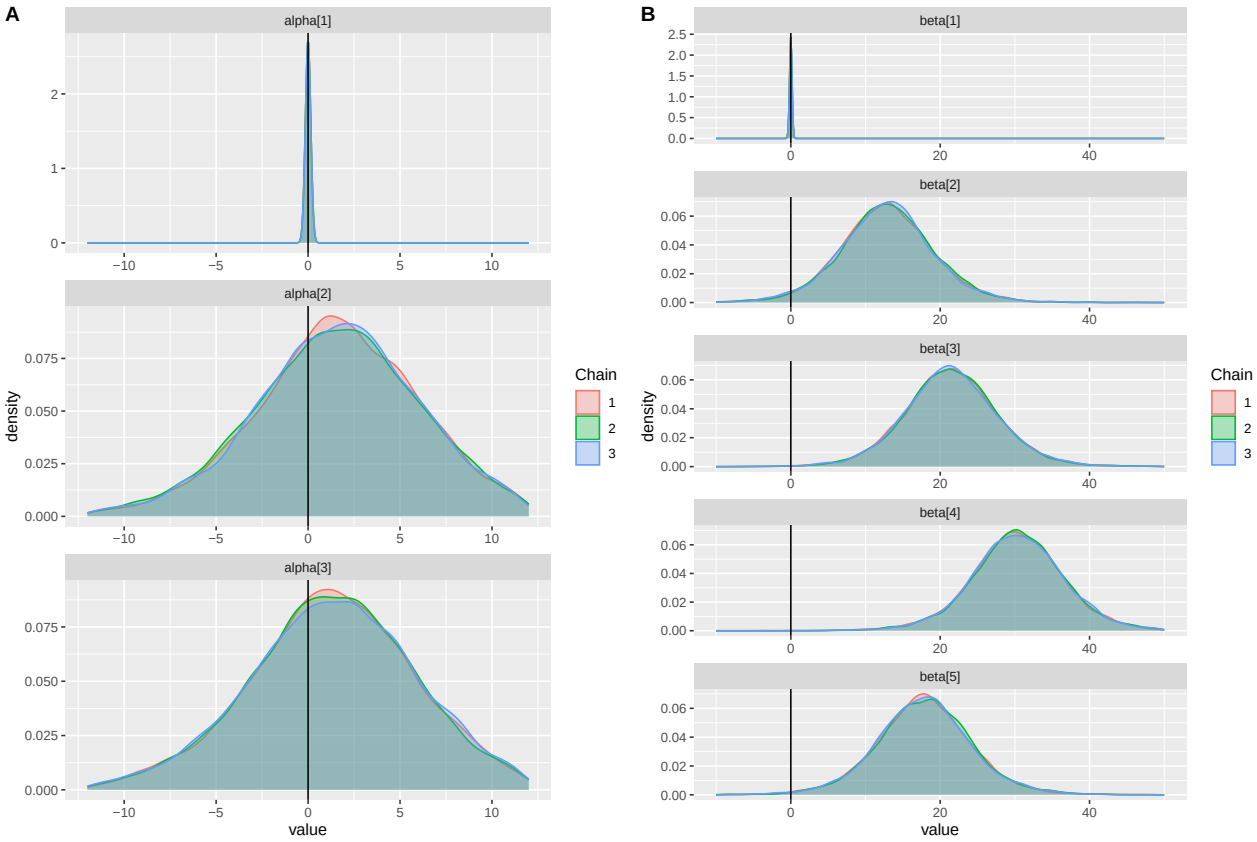


Posterior Densities:

```
ggs_density_alpha<-ggs_density(Bayesian_anova_inference.ggs,family = "alpha")+xlim(-12,12)+ geom_vline(
ggs_density_beta<-ggs_density(Bayesian_anova_inference.ggs,family = "beta")+xlim(-10,50)+ geom_vline(xi

library(cowplot)

# Arrange the 'alpha_caterpillar' and 'beta_caterpillar' plots side by side
plot_grid(ggs_density_alpha, ggs_density_beta, nrow = 1, labels = "AUTO")
```

```
## Warning: Removed 1654 rows containing non-finite values (`stat_density()`).
```

```
## Warning: Removed 191 rows containing non-finite values (`stat_density()`).
```



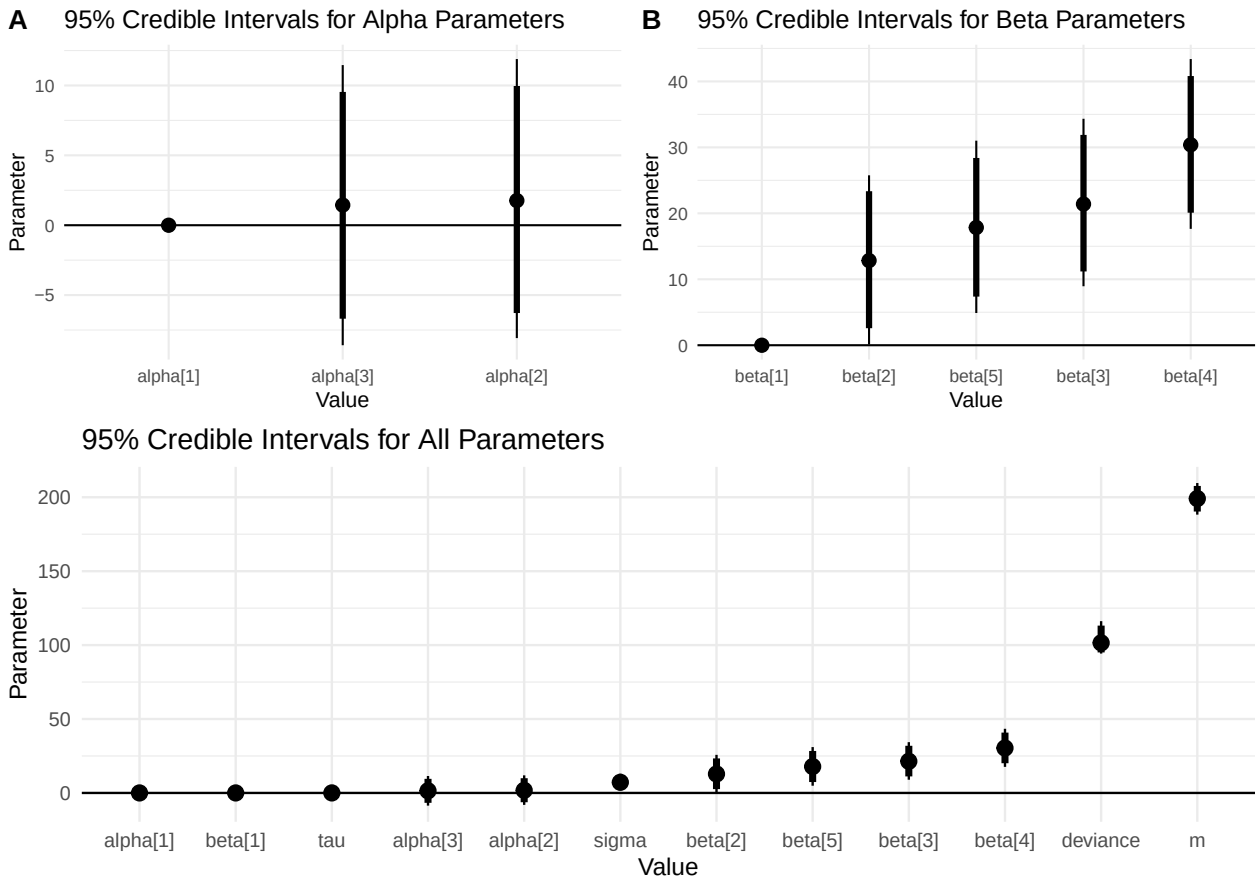
We can conclude that both field locations ($\alpha 2$ and $\alpha 3$) and treatments ($\beta 2$ to $\beta 5$) significantly affect carbon sequestration, as their posterior distributions are centered away from zero. The traceplots demonstrate good mixing and convergence of the MCMC chains, reinforcing the reliability of these findings.

In summary, fields 2 and 3 show a positive difference in carbon sequestration compared to the baseline field 1. Similarly, treatments 2 to 5 are estimated to have a positive impact on carbon sequestration relative to the baseline treatment $T1$, with treatment 4 showing the most considerable effect albeit with the greatest uncertainty. The robustness of the MCMC diagnostics indicates confidence in the model's estimates.

Bayesian Statistics Part (h)**:

```
library(ggmcmc)
library(ggplot2)

# First, ensure that Bayesian_anova_inference output has been converted using the ggs() function.
# Now, we can plot the 95% credible intervals for alpha parameters
alpha_caterpillar <- ggs_caterpillar(Bayesian_anova_inference.ggs, family = "alpha") +
  coord_flip() +
  geom_vline(xintercept = 0)+
  labs(title = "95% Credible Intervals for Alpha Parameters",
       x = "Parameter",
       y = "Value") +
  theme_minimal()

# Plot the 95% credible intervals for beta parameters
beta_caterpillar <- ggs_caterpillar(Bayesian_anova_inference.ggs, family = "beta") +
  coord_flip() +
```

```
  geom_vline(xintercept = 0)+
  labs(title = "95% Credible Intervals for Beta Parameters",
       x = "Parameter",
       y = "Value") +
  theme_minimal()

# plot all parameters together
all_parameters_caterpillar <- ggs_caterpillar(Bayesian_anova_inference.ggs) +
  coord_flip() +
  geom_vline(xintercept = 0)+
  labs(title = "95% Credible Intervals for All Parameters",
       x = "Parameter",
       y = "Value") +
  theme_minimal()
```



The credible intervals for (field effects) and (treatment effects) parameters provide evidence for assessing the influence on total carbon values due to different treatments and field locations.

- **Field Effects (α parameters)**:
  - $\alpha 1$ is set to zero; this acts as a baseline.
  - $\alpha 2$ and $\alpha 3$ have credible intervals that do not contain zero, indicating a statistically significant effect of field location on carbon sequestration compared to the baseline.
- **Treatment Effects (β parameters)**:
  - $\beta 1$ serves as a baseline (set to zero) against which other treatments are compared.
  - $\beta 2$ to $\beta 5$ show credible intervals that do not overlap zero, suggesting that these treatments are significantly different from the baseline treatment $T1$ in their effect on carbon sequestration.

- **Overall Mean (μ)**:
  - The credible interval for the overall mean (μ) is well above zero and does not overlap with zero, which confirms that carbon sequestration is taking place.

Conclusion: The underlying total carbon value in the soil does indeed vary when different treatments are applied and when different field locations are used. The analysis proves that treatments $T2$ to $T5$ have a distinguishable and positive effect on carbon sequestration compared to $T1$. Similarly, fields 2 and 3 have different carbon sequestration levels than field 1. These conclusions are drawn from the fact that the 95% credible intervals for these parameters do not include zero, indicating a statistically significant effect.

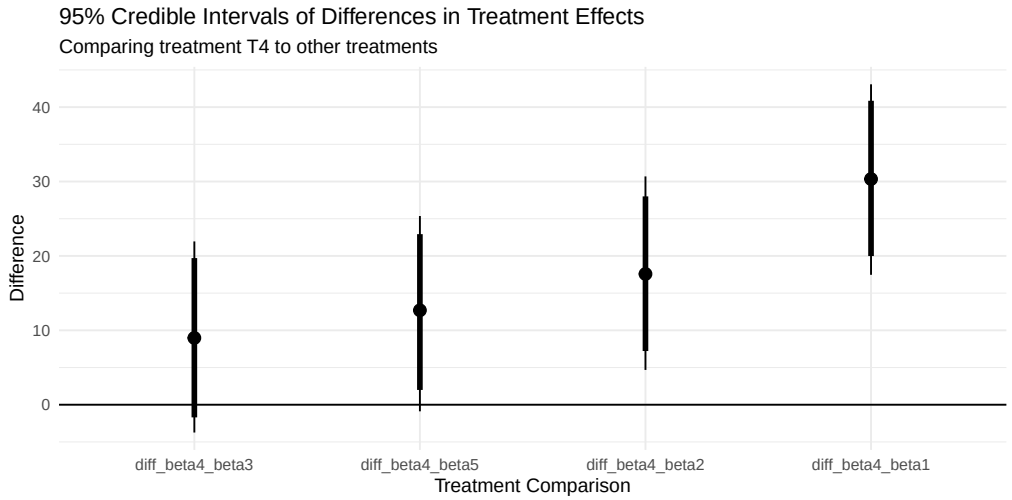
Bayesian Statistics Part (i)**:

```
## [1] "mcmc.list"
```

```
# Calculate the summary statistics for these differences
summary_differences <- summary(differences)
summary_differences
```

```
##  diff_beta4_beta1 diff_beta4_beta2 diff_beta4_beta3 diff_beta4_beta5
##  Min.   :-3.72    Min.   :-21.5    Min.   :-22.11   Min.   :-25.70
##  1st Qu.:26.43    1st Qu.: 13.7    1st Qu.:  4.93   1st Qu.:  8.66
##  Median :30.33    Median : 17.6    Median :  8.97   Median : 12.69
##  Mean   :30.36    Mean   : 17.6    Mean   :  8.94   Mean   : 12.58
##  3rd Qu.:34.28    3rd Qu.: 21.5    3rd Qu.: 12.87   3rd Qu.: 16.56
##  Max.   :71.87    Max.   : 61.0    Max.   : 51.10   Max.   : 66.97
```

```
library(ggmcmc)
library(ggplot2)


# Convert 'differences' into a format suitable for ggmcmc
differences_ggmcmc <- as.mcmc(differences)

# Use ggs() to convert the MCMC object into a data frame for plotting
differences_ggs <- ggs(differences_ggmcmc)

# Use ggs_caterpillar to create the plot
# Create the caterpillar plot for the differences data frame
caterpillar_plot <- ggs_caterpillar(differences_ggs) +
  coord_flip()  +
  geom_vline(xintercept = 0)+
  labs(title = "95% Credible Intervals of Differences in Treatment Effects",
       x = "Difference",
       y = "Treatment Comparison",
       subtitle = "Comparing treatment T4 to other treatments") +
  theme_minimal()

caterpillar_plot
```



The farmer's belief that treatment $T4$ should yield a higher level of carbon sequestration is evaluated by calculating the differences in effect between treatment $T4$ and the other treatments ($\beta 1$, $\beta 2$, $\beta 3$, and $\beta 5$). These differences represent how much more (or less) effective $T4$ is compared to each of the other treatments in sequestering carbon.

Difference between $T4$ and $T1$ (baseline):

The credible interval is positive and does not include zero, indicating that $T4$ is significantly more effective than $T1$.

Differences between $T4$ and $T2$, $T3$, and $T5$:

- For $T3$, the credible intervals include zero, suggesting that there may not be a significant difference in effectiveness compared to $T4$.
- For $T2$, the credible interval is positive and does not include zero, indicating that $T4$ is significantly more effective than $T2$.
- For $T5$, the bottom tail of the interval touches zero, it implies a potential advantage for $T4$, but not with enough confidence to rule out the possibility that the treatments could be equally effective or that $T5$ could even be more effective.

Interpretation: The summary statistics and the credible interval plot consistently suggest that treatment $T4$ has a superior effect on carbon sequestration compared to $T1$ and $T2$. The comparison with $T3$ and $T5$ is less clear, as the credible intervals cross zero, indicating that the evidence is not strong enough to confirm a significant difference between these treatments and $T4$. The farmer's expectation about $T4$ is partially supported by the data, particularly when compared to $T1$ and $T5$.

Bayesian Statistics Part (j)**:

```
I<-3
J<-5
y_matrix <- matrix(carbon_sequestration, nrow = I, byrow = TRUE)


# List of data for JAGS
data_simpler <- list(
  y = y_matrix,
  I = I,
  J = J
)
```

```r
# Simpler Bayesian model function
Simpler_Bayesian_Model <- function() {
  # Prior for overall mean level
  m ~ dnorm(0.0, 1.0E-4)

  # Treatment effects, beta[1] fixed at 0 as reference level
  beta[1] <- 0
  for (j in 2:J) {
    beta[j] ~ dnorm(0.0, 1.0E-4)
  }

  # Linking treatment means to overall mean and treatment effects
  for (j in 1:J) {
    mu[j] <- m + beta[j]
  }

  # Model definition for observed data
  for (i in 1:I) {
    for (j in 1:J) {
      # Response variable
      y[i,j] ~ dnorm(mu[j], tau)
    }
  }

  # Prior for precision of the model
  tau ~ dgamma(1.0E-3, 1.0E-3)

  # Derived quantity: standard deviation
  sigma <- 1.0 / sqrt(tau)
}


# Running the JAGS model
library(R2jags)
Simpler_Bayesian_inference <- jags(
  data = data_simpler,
  parameters.to.save = c("m", "beta", "sigma", "tau"),
  n.iter = 100000, # Number of iterations per chain
  n.chains = 3, # Number of chains
  n.burnin=5000, # Number of burn-in samples
  n.thin=10, #Thinning rate
  DIC=TRUE,
  model.file = Simpler_Bayesian_Model
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 15
##    Unobserved stochastic nodes: 6
##    Total graph size: 34
##
## Initializing model
```

```r
# Summarising the results
print(Simpler_Bayesian_inference, intervals = c(0.025, 0.5, 0.975))
```

```
## Inference for Bugs model at "/var/folders/ls/d_6cnrhs3yz5fw356d6dgpqr0000gn/T//Rtmpk537vd/model23cc6
##  3 chains, each with 1e+05 iterations (first 5000 discarded), n.thin = 10
##  n.sims = 28500 iterations saved
##          mu.vect sd.vect    2.5%     50%   97.5%  Rhat n.eff
## beta[1]    0.000   0.000   0.000   0.000   0.000 1.000     1
## beta[2]   12.788   5.651   1.405  12.821  23.988 1.001 28000
## beta[3]   21.523   5.665  10.242  21.549  32.779 1.001 28000
## beta[4]   30.471   5.699  19.136  30.439  41.903 1.001 28000
## beta[5]   17.834   5.710   6.453  17.836  29.145 1.001 28000
## m        200.140   4.003 192.065 200.170 208.027 1.001 28000
## sigma      6.745   1.703   4.344   6.449  10.855 1.001 28000
## tau        0.026   0.012   0.008   0.024   0.053 1.001 28000
## deviance  99.044   4.411  92.995  98.171 109.853 1.001 28000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 9.7 and DIC = 108.8
## DIC is an estimate of expected predictive error (lower deviance is better).
```

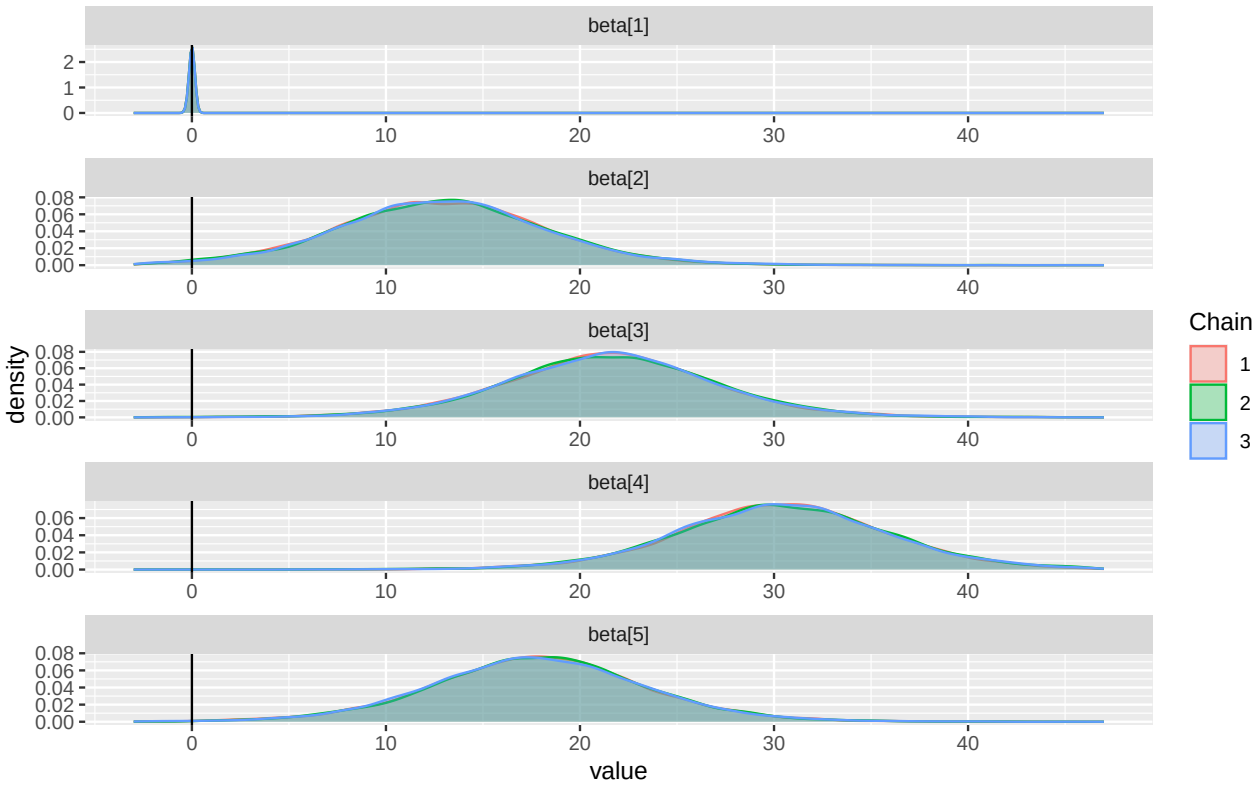
Based on the simpler Bayesian model's JAGS output, here is a concise interpretation of the results:

- **Overall Mean (μ)**: The estimated average carbon sequestration is about 200 units, with a tight 95% credible interval suggesting precise estimation.
- **Treatment Effects (βj)**:
  - **$\beta 2$, $\beta 3$, and $\beta 5$** have positive effects on carbon sequestration compared to the baseline ($\beta 1$), with their credible intervals lying above zero. $\beta 2$ shows a moderate effect, $\beta 3$ shows a larger effect, and 4 5$'s effect is between $\beta 2$ and $\beta 3$.
  - $\beta 4$ exhibits the largest effect among treatments with its credible interval entirely above zero, highlighting a significant increase in carbon sequestration compared to the baseline.
- **Precision (τ) and Standard Deviation (σ)**: The precision is fairly high, indicating that the model accounts for a low level of variability in carbon sequestration measurements around the mean. The corresponding standard deviation provides a sense of the variability, with a mean of approximately 6.77 units.

The Rhat values near 1 and large effective sample sizes across parameters suggest that the MCMC sampling process is robust, and the posterior estimates are reliable. The DIC value indicates a good fit of the model to the data.

Bayesian Statistics Part (k)**:

```r
# Graphical summaries
library(ggmcmc)
library(dplyr)
#
# Prepare the jags samples
Simpler_Bayesian_inference.mcmc <- as.mcmc(Simpler_Bayesian_inference)
Simpler_Bayesian_inference.ggs <- ggs(Simpler_Bayesian_inference.mcmc)
ggs_density(Simpler_Bayesian_inference.ggs,family = "beta")+xlim(-3,47)+ geom_vline(xintercept = 0)
```



The posterior density plots for the β parameters of the simpler Bayesian model provide insights into the distribution of treatment effects:

- $\beta 1$: The density plot confirms that $\beta 1$ is fixed at 0; it serves as the baseline reference for comparison with other treatments.
- $\beta 2$, $\beta 3$, $\beta 5$: Each shows a unimodal distribution centered above zero, indicating a positive treatment effect relative to 1. Their 95% credible intervals do not include zero, signifying these treatments have a significant positive effect on carbon sequestration.
- $\beta 4$: Exhibits the highest mean effect, and its density plot peaks well above zero. The entire credible interval is above zero, highlighting its strong positive impact on carbon sequestration, the most substantial among all treatments.
- **Overlapping Chains**: All chains (red, green, blue) appear to overlap and converge well in these plots, which supports the reliability of the posterior distributions.

We can conclude that treatments $\beta 2$, $\beta 3$, $\beta 4$, and $\beta 5$ all significantly enhance carbon sequestration compared to the baseline ($\beta 1$), with $\beta 4$ being the most effective.

Bayesian Statistics Part (l)**:

In choosing between the full Bayesian two-ways Analysis of Variance model (part f) and the simpler Bayesian model (part j), several aspects including model complexity, interpretability, and fit to data, should be considered and addressed.

- **Complexity vs. Simplicity**: The full model (part f) includes both field effects (αi) and treatment effects (βj), which makes it more complex. It could be preferable if the research question specifically involves understanding the variability due to field locations. - The simpler model (part j) omits the field effects and focuses only on treatment effects, which might be preferable for its simplicity and direct focus on treatment comparisons.

- **Model Fit**: DIC (Deviance Information Criterion) provides a measure of model fit and complexity. A lower DIC is preferred as it shows a better trade-off between model fit and complexity. For the full model, DIC is 119.0, and for the simpler model, it is 109.0. The lower DIC of the simpler model suggests a better balance of fit and parsimony.
- **Interpretability**: The simpler model is easier to interpret because it only examines treatment effects, which could be sufficient if the primary interest is in comparing treatments rather than exploring field variations.
- **Parameter Estimation**: Both models show good parameter convergence (Rhat $\sim$ 1) and effective sample size (n.eff is large), so either model seems reliable for inference based on the posterior distributions.
- **Preference**: If the goal is to understand the effects of treatments while controlling for field variability, the full model is more appropriate. If the research aims only to compare the effectiveness of treatments regardless of field location, then the simpler model is preferred due to its lower DIC (better model fit), sufficient complexity for the task, and ease of interpretation.

In summary, without specific context, the simpler model could be preferred for its parsimony and better model fit as indicated by the lower DIC value, provided that field effects are not of primary interest in the analysis.