Multivariate Analysis Assignment 1

Isha Borgaonkar 24209758

```
# Load necessary package
library(dplyr)
```

Question 1:

```
# Set seed to student number
set.seed(24209758)
# Load the data
pressure_data <- read.csv("Pressure_Data.csv")
# Check the number of rows
nrow(pressure_data)  # Should be 462
```

```
[1] 462
```

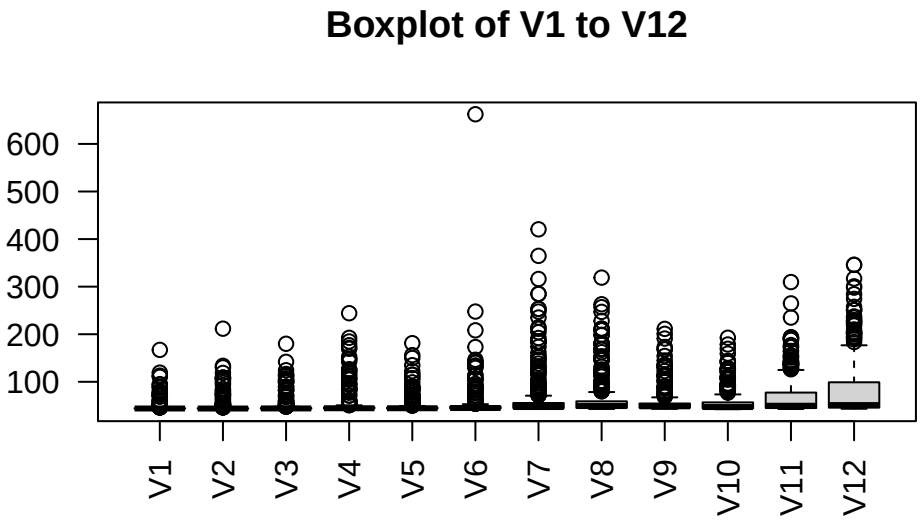
```
# Randomly sample 400 rows
pressure_data_sample <- pressure_data %>% sample_n(400)
```

Description:

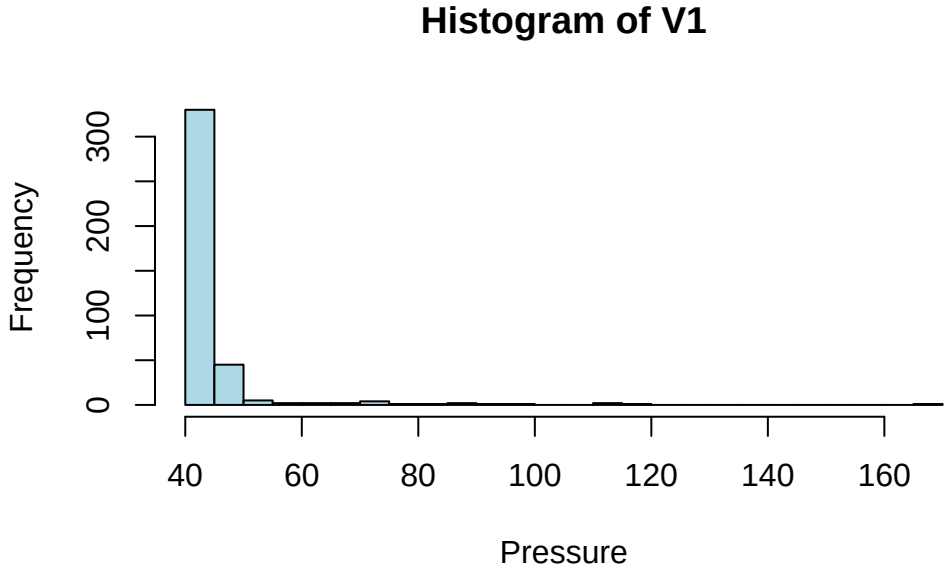
Firstly,the dataset `Pressure_Data.csv` is loaded. A seed (`24209758`) is set for reproducibility with argument of rollnumber. The total number of rows is checked and got correct output 462. A random sample of **400 observations** is selected using `sample_n()` from the `dplyr` library package.

Question 2:

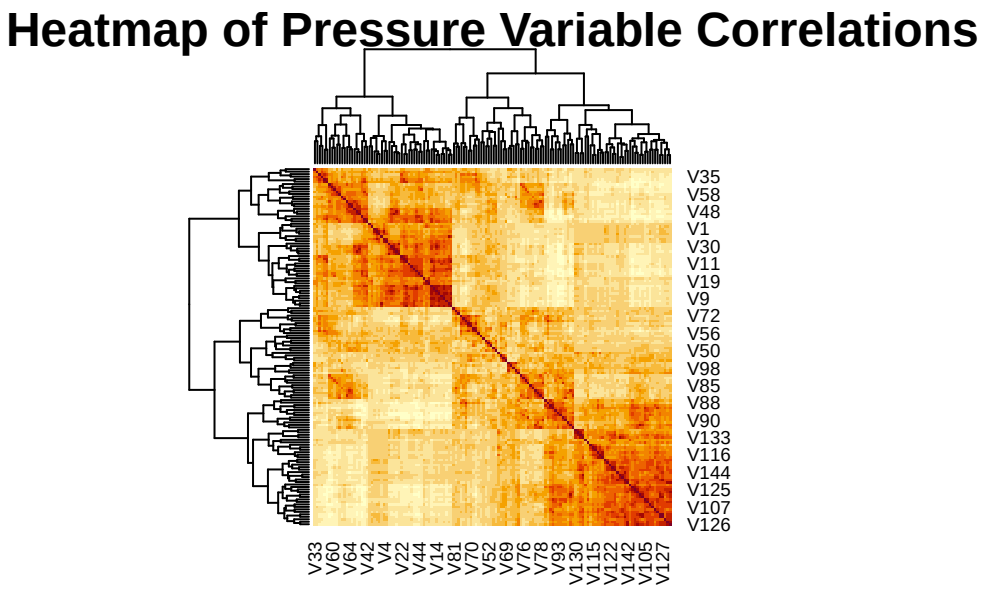
```
#Remove missing values in the factor variables
cleaned_data <- pressure_data_sample %>%
  filter(!is.na(Mattress_type) & !is.na(Position) & !is.na(Posture) & !is.na(Subject))
#Extract pressure variables V1 to V144
pressure_vars <- cleaned_data %>% select(starts_with("V"))
#Plot: Boxplots for first 12 variables to check distribution
boxplot(pressure_vars[, 1:12], main = "Boxplot of V1 to V12", las = 2)
```




```r
#Plot: Histogram of a single variable, e.g., V1
hist(pressure_vars$V1, main = "Histogram of V1", xlab = "Pressure", col = "lightblue", breaks
```



```r
#Plot: Correlation heatmap
cor_matrix <- cor(pressure_vars)
heatmap(cor_matrix, main = "Heatmap of Pressure Variable Correlations", symm = TRUE)
```



Description:

Rows with the missing values in Mattress_type, Position, Posture, and Subject get removed. These are the factor variables secondly,extracted pressure variables V1 to V144 for analysis. Visualised data by using boxplot (V1–V12) to assess spread and outliers and histogram (V1) to inspect distribution shape and also created the correlation heatmap to identify multicollinearity.

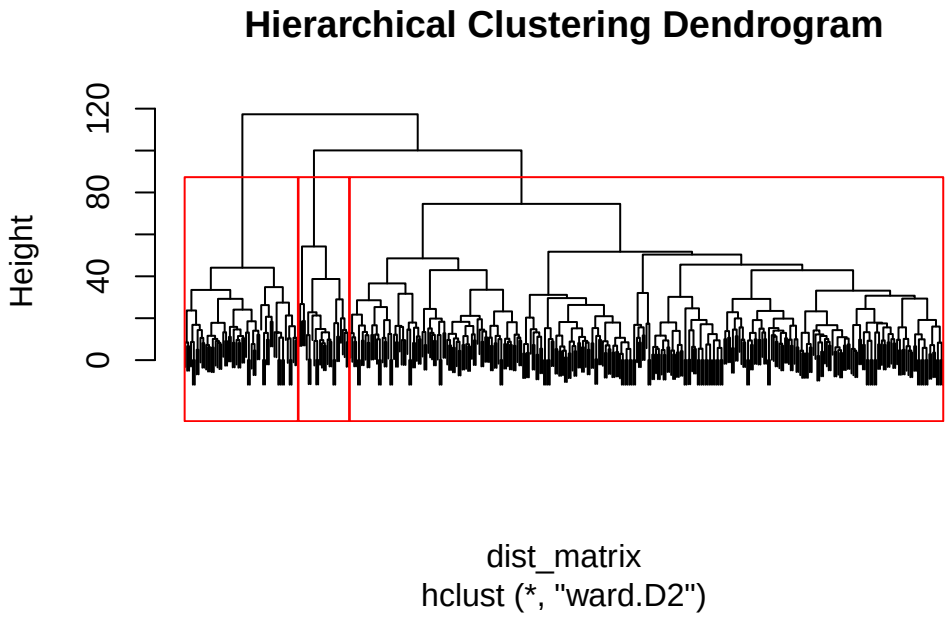
Justification:

The boxplot (V1–V12) shows a large number of outliers and positively skewed distributions, with some extreme high values. The histogram of V1 confirms this skewness, with most values concentrated at the lower end around 40–50 and a long right tail. The heatmap reveals strong correlation among pressure variables, indicating multicollinearity.

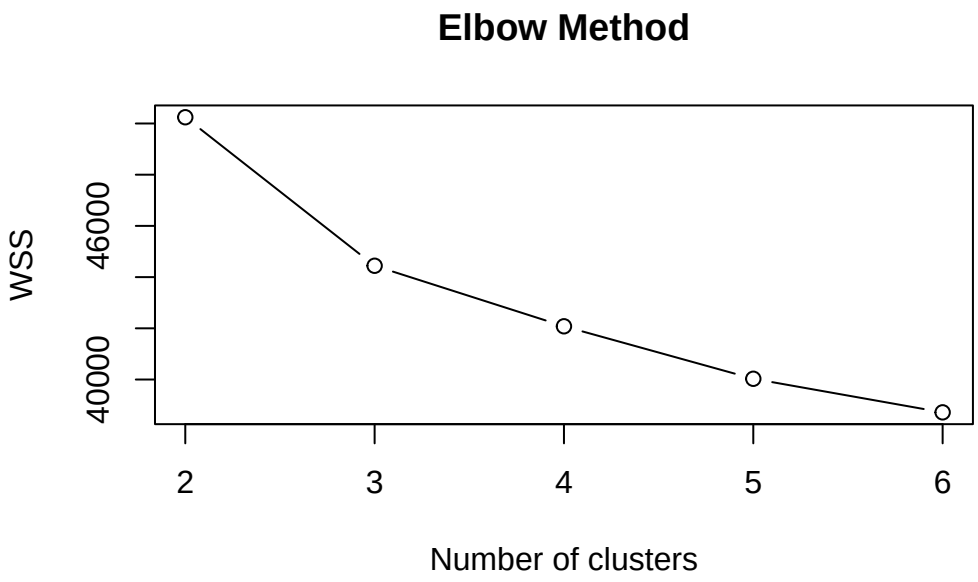
Question 3:

```r
#Select and scale pressure data
pressure_vars_scaled <- scale(cleaned_data %>% select(starts_with("V")))
#Hierarchical clustering
```

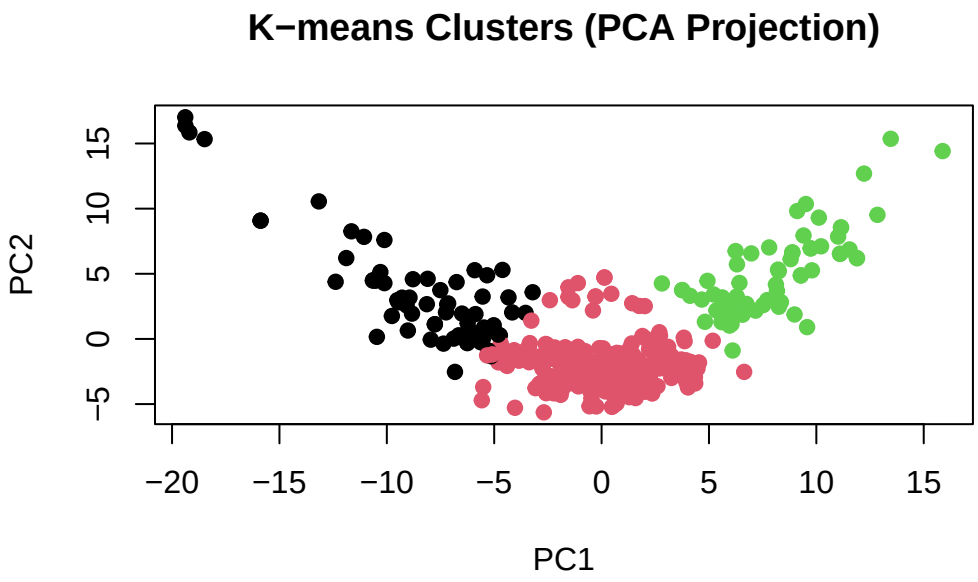
```r
dist_matrix <- dist(pressure_vars_scaled)
hclust_result <- hclust(dist_matrix, method = "ward.D2")
#Plot dendrogram
plot(hclust_result, labels = FALSE, main = "Hierarchical Clustering Dendrogram")
rect.hclust(hclust_result, k = 3, border = "red")  # Example with 3 clusters
```




```r
#K-means clustering (try different k, e.g., 2 to 6)
wss <- sapply(2:6, function(k){
  kmeans(pressure_vars_scaled, centers = k, nstart = 20)$tot.withinss
})
plot(2:6, wss, type = "b", main = "Elbow Method", xlab = "Number of clusters", ylab = "WSS")
```




```r
#Choose k = 3 (say), then apply k-means
kmeans_result <- kmeans(pressure_vars_scaled, centers = 3, nstart = 25)
# Add cluster labels to data
clustered_data <- cleaned_data
clustered_data$HierCluster <- cutree(hclust_result, k = 3)
clustered_data$KMeansCluster <- kmeans_result$cluster
# Step 4: PCA for visualization
pca <- prcomp(pressure_vars_scaled)
plot(pca$x[,1:2], col = clustered_data$KMeansCluster, pch = 19,
     main = "K-means Clusters (PCA Projection)", xlab = "PC1", ylab = "PC2")
```



Description:

Here, I scaled pressure data to standardize measurements also applied the hierarchical clustering (Ward's method) and plotted a dendrogram. Secondly, applied k-means clustering and used the elbow method to select optimal k = 3. Lastly, visualized k-means clusters in PCA space.

Justification:

1)The elbow plot suggests **k = 3** is an appropriate number of clusters, as the within-cluster sum of squares (WSS) sharply decreases up to k = 3 and then flattens. It is supported by **k-means PCA plot**, where the data points form **three distinct, well-separated clusters. 2)The** hierarchical clustering dendrogram **is performed using Ward's method it reveals a** three-cluster structure**, as shown by the red rectangles. Both clustering methods identify consistent groupings, it indicates strong natural structure in the pressure data. 3)These clusters likely correspond to different** pressure distribution patterns**,influenced by subject posture, physical characteristics or position. Compact clusters get provided by K-means.Hierarchical clustering shows relationships at multiple levels.

Question 4:

```
library(dplyr)
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```
library(caret)
```

```
Warning: package 'caret' was built under R version 4.4.2

Loading required package: ggplot2

Loading required package: lattice
```

```
#Prepare data
pressure_vars <- cleaned_data[, grep("^V", names(cleaned_data))]
response <- as.factor(cleaned_data$Posture)
#Fit LDA
lda_model <- lda(response ~ ., data = cbind(pressure_vars, response))
lda_pred <- predict(lda_model)
#Evaluate performance
conf_matrix <- table(Predicted = lda_pred$class, Actual = response)
print(conf_matrix)
```

```
         Actual
Predicted Left Right Supine
   Left     86     0      0
   Right     0    79      0
   Supine    1     0    234
```

```
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
cat("LDA Classification Accuracy:", round(accuracy, 3), "\n")
```

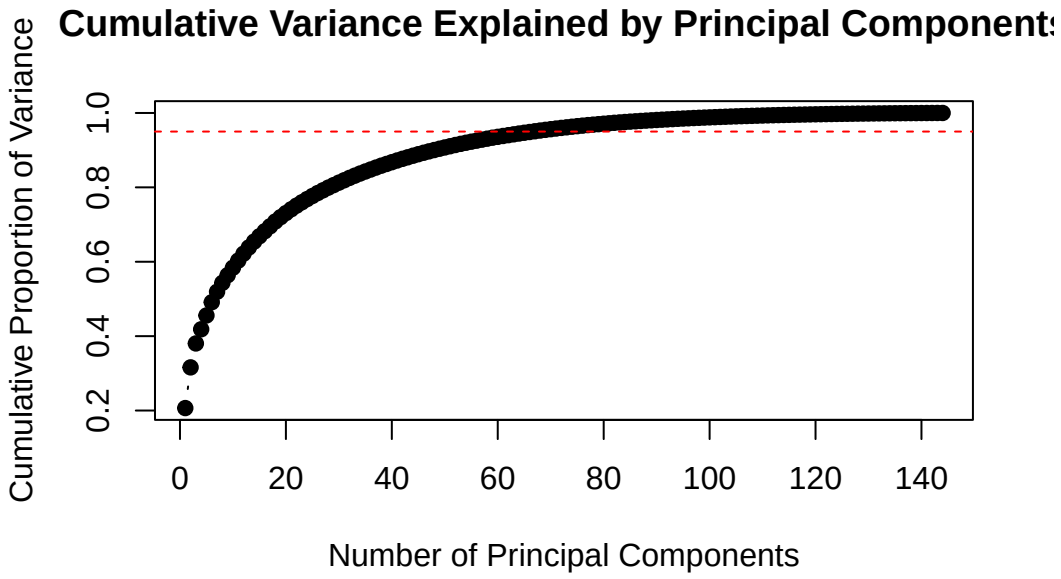
```
LDA Classification Accuracy: 0.998
```

Justification: Here, **Linear Discriminant Analysis (LDA)** model was fitted using the 144 pressure variables (`V1-V144`). It classify observations by **posture** Left, Right, Supine. The model achieved a **classification accuracy of 0.998**, shown in confusion matrix, correctly classifying nearly all samples with **only one misclassification**.

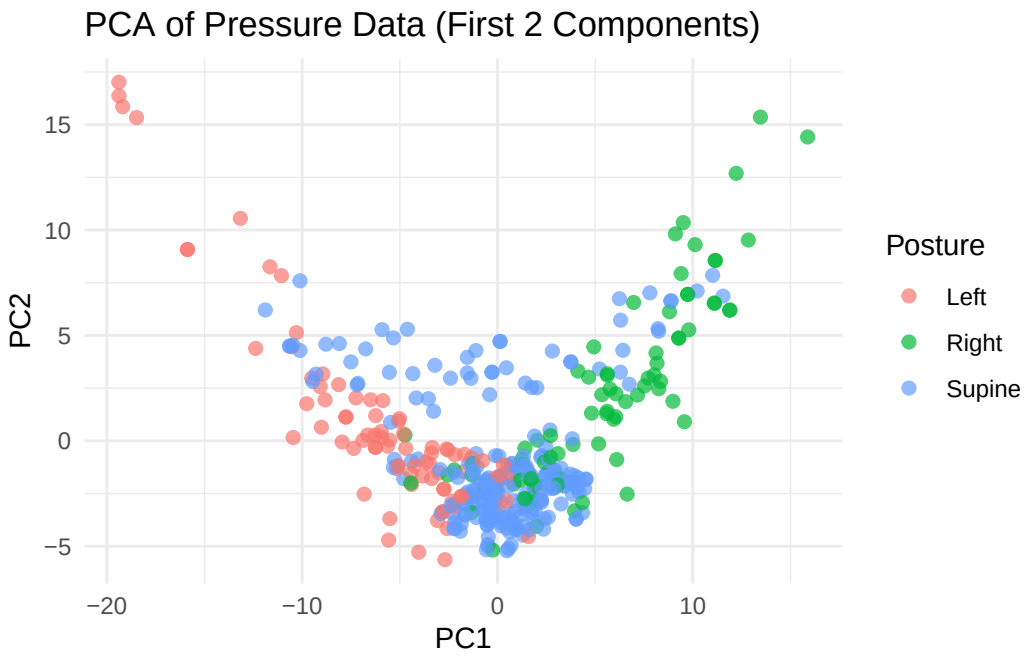
LDA and QDA: 1)**LDA** assumes equal covariance matrices across classes. It performs well when the number of predictors is large relative to sample size. 2)**QDA** estimates a separate covariance matrix for each class. This requires more observations per class, which is not feasible here beacause of **High dimensionality (144 predictors)**, **Limited number of observations per posture group** , **Risk of singular non-invertible covariance matrices**. Thus, **QDA is not appropriate** in this case, and LDA is preferred for its stability and excellent performance on this dataset.

Question5:

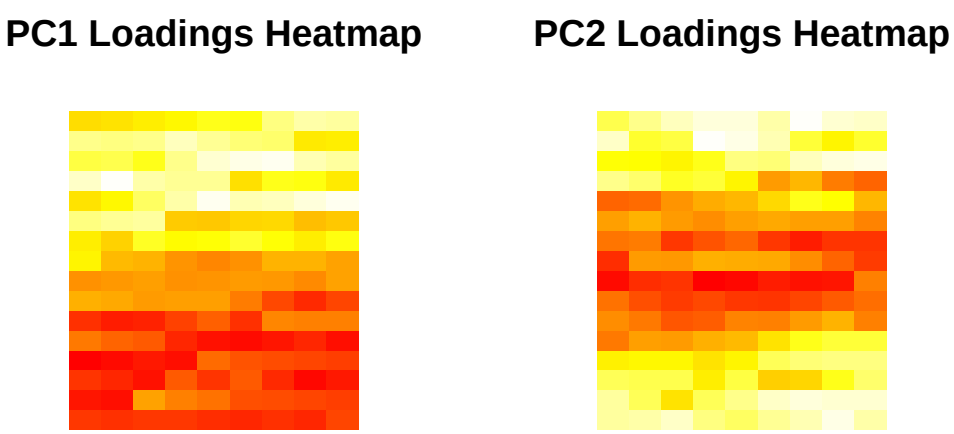
```
# Load necessary packages
library(ggplot2)
library(RColorBrewer)
#Extract and standardize pressure variables
pressure_vars <- cleaned_data %>% dplyr::select(starts_with("V"))
pressure_scaled <- scale(pressure_vars)
#Run PCA
pca_result <- prcomp(pressure_scaled)
#Plot cumulative variance explained
explained_variance <- summary(pca_result)$importance[3, ]  # cumulative proportion
plot(explained_variance, type = "b", pch = 19,
     main = "Cumulative Variance Explained by Principal Components",
     xlab = "Number of Principal Components", ylab = "Cumulative Proportion of Variance")
abline(h = 0.95, col = "red", lty = 2)  # 95% threshold
```



```r
#Get PCA scores (first 2 PCs)
pc_scores <- as.data.frame(pca_result$x[, 1:2])
pc_scores$Posture <- cleaned_data$Posture
#Plot first two principal components colored by posture
ggplot(pc_scores, aes(x = PC1, y = PC2, color = Posture)) +
  geom_point(alpha = 0.7, size = 2) +
  theme_minimal() +
  labs(title = "PCA of Pressure Data (First 2 Components)", x = "PC1", y = "PC2")
```




```r
#Visualize loadings as heatmaps for interpretation
#For this, reshape PC1 and PC2 loadings into a 16x9 matrix
pc1_loading <- matrix(pca_result$rotation[, 1], nrow = 16, ncol = 9, byrow = TRUE)
pc2_loading <- matrix(pca_result$rotation[, 2], nrow = 16, ncol = 9, byrow = TRUE)
# Optional heatmap visualisation (base R)
par(mfrow = c(1,2))
image(t(apply(pc1_loading, 2, rev)), main = "PC1 Loadings Heatmap", col = heat.colors(100), a
image(t(apply(pc2_loading, 2, rev)), main = "PC2 Loadings Heatmap", col = heat.colors(100), a
```




```
par(mfrow = c(1,1))
```

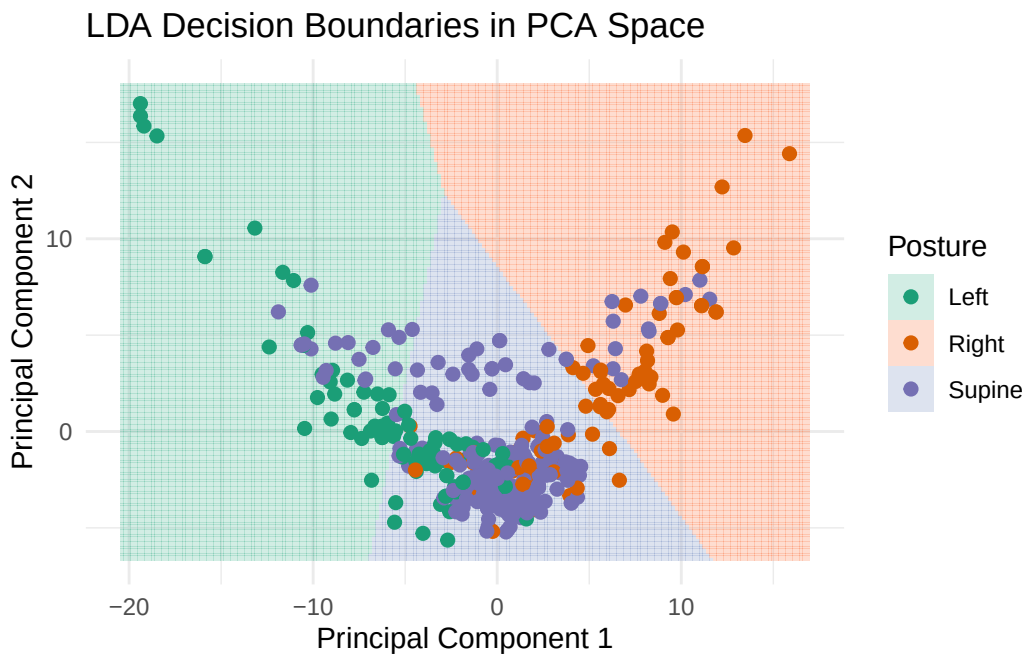
Justification:

PCA was applied to the **scaled pressure variables** by using the correlation matrix to handle high dimensionality and multicollinearity. The **cumulative variance plot** shows that approximately **60 principal components** are required to explain **95% of the total variance** in the dataset. This justifies dimensionality reduction before further modelling. The **scatter plot of PC1 vs PC2** which are coloured by posture and shows **clear separation** between sleeping postures. This indicates that the first two PCs capture posture-related pressure variation effectively. The **heatmaps of PC1 and PC2 loadings** helps in interpretation of how pressure is distributed spatially. PC1 heatmap emphasizes **lower regions** of the mattress. PC2 heatmap emphasizes **central to upper zones**. These spatial patterns align with how different postures such as left side, supine and would affect pressure sensor readings.

Question 6:

```
# Load libraries
library(MASS)
library(ggplot2)
#Get PC1 and PC2 with posture labels
pc_data <- as.data.frame(pca_result$x[, 1:2])
pc_data$Posture <- cleaned_data$Posture
```

```r
#Fit LDA using PC1 and PC2
lda_pc <- lda(Posture ~ PC1 + PC2, data = pc_data)
#Create a grid for decision boundary
x_min <- min(pc_data$PC1) - 1
x_max <- max(pc_data$PC1) + 1
y_min <- min(pc_data$PC2) - 1
y_max <- max(pc_data$PC2) + 1
grid <- expand.grid(PC1 = seq(x_min, x_max, length.out = 200),
                    PC2 = seq(y_min, y_max, length.out = 200))
#Predict posture for each grid point
grid$Posture <- predict(lda_pc, newdata = grid)$class
#Plot decision boundaries and points
ggplot() +
  geom_tile(data = grid, aes(x = PC1, y = PC2, fill = Posture), alpha = 0.3) +
  geom_point(data = pc_data, aes(x = PC1, y = PC2, color = Posture), size = 2) +
  labs(title = "LDA Decision Boundaries in PCA Space",
       x = "Principal Component 1",
       y = "Principal Component 2") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2") +
  scale_color_brewer(palette = "Dark2")
```



Justification:

Here, LDA model was trained using the **first two principal components** from PCA on the pressure data. The plot shows **linear Bayes decision boundaries**. It separats the three postures **Left**, **Right**, and **Supine**. The plot demonstrates 1)LDA model successfully forms **clear linear boundaries** between the posture classes. 2)Based on the position in PC1–PC2 space most points are correctly classified. 3)The overlap between **Supine** and the other postures is minimal. It suggest strong discriminatory power in the first two components. This confirms that PCA followed by LDA is an effective dimensionality reduction and classification approach for posture prediction using pressure measurements.

Qustion 7:

```
library(MASS)
library(caret)
# Use scaled data (i.e., correlation matrix basis for PCA)
pressure_vars_scaled <- cleaned_data %>%
  dplyr::select(starts_with("V")) %>%
  scale()
subject <- as.factor(cleaned_data$Subject)
# PCA on correlation matrix (scale=TRUE is default when data is already scaled)
pca_result_subject <- prcomp(pressure_vars_scaled)
# Choose enough PCs to explain ~95% variance
expl_var <- summary(pca_result_subject)$importance[3, ]
num_pcs <- which(expl_var >= 0.95)[1]
cat("Number of PCs explaining 95% variance:", num_pcs, "\n")
```

```
Number of PCs explaining 95% variance: 67
```

```
# Create PCA scores dataframe
pca_scores <- as.data.frame(pca_result_subject$x[, 1:num_pcs])
pca_scores$Subject <- subject
# Train/test split (e.g., 70-30)
set.seed(42)
train_idx <- createDataPartition(pca_scores$Subject, p = 0.7, list = FALSE)
train_data <- pca_scores[train_idx, ]
test_data <- pca_scores[-train_idx, ]
# LDA
lda_model <- lda(Subject ~ ., data = train_data)
lda_pred <- predict(lda_model, test_data)$class
lda_acc <- mean(lda_pred == test_data$Subject)
cat("LDA Accuracy:", round(lda_acc, 3), "\n")
```

```
LDA Accuracy: 0.496
```

Justification:

PCA was applied to the scaled pressure data using the **correlation matrix**,It is appropriate since the variables may be on different scales. This ensures equal contribution from each sensor. A total of **67 principal components** were selected to retain **95% of the variance**. These components used to classify observations by **Subject** using **Linear Discriminant Analysis (LDA)**. With the 70-30 train-test split, the **LDA model achieved an accuracy of 49.6%**. This suggests moderate separation between subjects based on pressure patterns are better than random. Using the correlation matrix ensured all variables contributed equally, regardless of their scale. A **QDA model** can also be trained to compare performance. It allows for class-specific covariance, which may better capture individual subject variation.

Question8: 1)Principal Components Regression (PCR) is a technique used to improve regression modelling when the predictors which are independent variables are highly correlated or another case is when there are more predictors than observations.

2)In our examples's case, the 144 pressure variables are likely to be strongly correlated, which can cause multicollinearity, unreliable or inconsistent coefficient values and overfitting if regular linear regression is used.

3)PCR helps to overcome these issues by reducing the dimensionality of the predictor space before applying regression on it.

(i)Purpose:

1)The main goal of PCR is to build a more reliable and flexible regression model when dealing with many predictors that are highly correlated or when there are too many variables.

2)Instead of using all the original variables directly, PCR transforms them into a smaller set of new, uncorrelated variables called principal components. These components capture the most important patterns in the data, helping to simplify the model, reduce overfitting, and make predictions more stable and accurate.

(ii)How the Method Works:

1)PCR works in two main steps. First, it standardizes the predictor variables so that they're all on the same scale. Then it applies Principal Component Analysis (PCA) to uncover the main patterns or directions of variation in the data.

2)These patterns are turned into new variables called **principal components**, which are combinations of the original variables but are uncorrelated with one another.

3)The components are ranked by how much of the overall variation in the data they capture. Instead of using all of them, we keep only the top few (usually enough to explain 90–95% of the variation), which simplifies the model and keeps the most important information.

4)Finally, a linear regression model is built using these selected components as predictors. This helps to reduce complexity, avoid overfitting, and make the model more stable and generalisable.

(iii)Choices to Be Made:

1)When using PCR, a key decision is how many principal components to include in the regression model.

2)This is usually determined using cross-validation to balance model complexity and performance. Other considerations include whether to scale the variables before PCA (which is usually required) and how to handle missing data.

Advantages of PCR:

1)It handles multicollinearity well by using new variables (principal components) that aren't correlated with each other.

2)It helps reduce overfitting, especially when there are a lot of predictors.

3)It can improve prediction accuracy by focusing on the most important patterns in the data.

4)It works well in high-dimensional situations — even when there are more variables than observations.

5)It simplifies complex datasets by reducing many variables into a smaller, more manageable set of components.

Disadvantages:

1)The principal components are created without considering the response variable, so they might not be the most useful for prediction.

2)The components can be hard to interpret, since they are combinations of many original variables.

3)Sometimes, variables that are important for predicting the outcome might get left out if they don't explain much overall variance.

4)Choosing how many components to keep is tricky and usually needs cross-validation.

5)It may not perform as well as methods like PLS (Partial Least Squares), which do take the response variable into account when reducing dimensions.

Question 9:

```
# Check count per Subject in training set
table(train_data$Subject)
```

```
S1 S2 S3 S4 S5 S6 S7 S8
34 35 36 33 39 34 37 35
```

```r
# Load necessary libraries
library(pls)
```

```
Warning: package 'pls' was built under R version 4.4.3
```

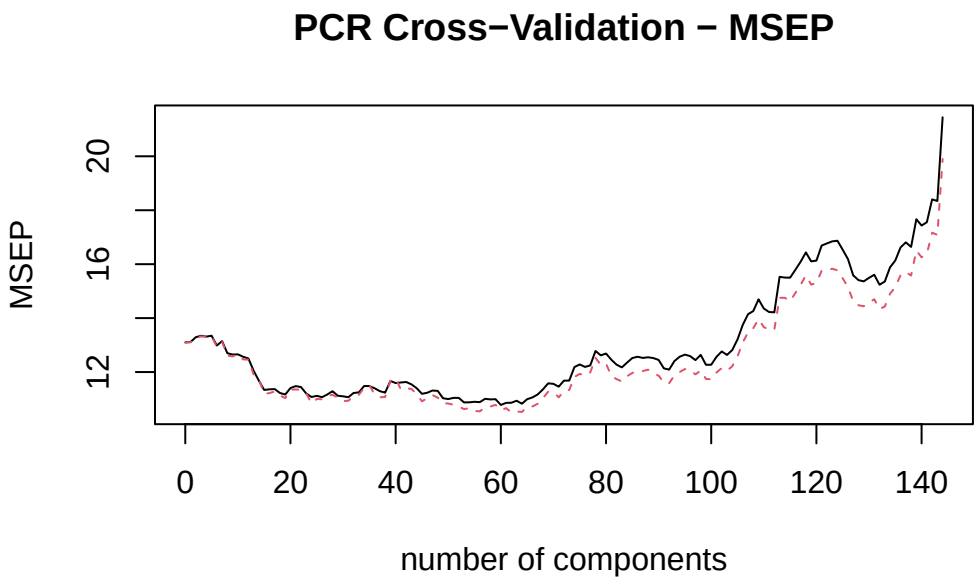
```
Attaching package: 'pls'
```

```
The following object is masked from 'package:caret':

    R2
```

```
The following object is masked from 'package:stats':

    loadings
```

```r
library(caret)
library(dplyr)
#Load Subject_Info and calculate BMI
subject_info <- read.csv("Subject_Info_Data.csv")
#Convert Subject IDs to matching format
cleaned_data$Subject <- gsub("S", "", cleaned_data$Subject)
cleaned_data$Subject <- as.character(cleaned_data$Subject)
subject_info$Subject <- as.character(subject_info$Subject)
#Compute BMI = weight (kg) / height^2 (m^2)
subject_info$BMI <- subject_info$Weight.kg / ((subject_info$Height.cm / 100)^2)
#Merge BMI into pressure data
merged_data <- merge(cleaned_data, subject_info[, c("Subject", "BMI")], by = "Subject")
#Prepare predictor matrix X and response Y
X <- merged_data %>% dplyr::select(starts_with("V"))
Y <- merged_data$BMI
X_scaled <- scale(X)
#Split into training and test sets (70/30)
set.seed(123)
train_idx <- createDataPartition(Y, p = 0.7, list = FALSE)
X_train <- X_scaled[train_idx, ]
X_test <- X_scaled[-train_idx, ]
Y_train <- Y[train_idx]
Y_test <- Y[-train_idx]
#Fit Principal Components Regression model using cross-validation
```

```r
pcr_model <- pcr(Y_train ~ X_train, scale = FALSE, validation = "CV")
#Plot CV results and find optimal number of components
validationplot(pcr_model, val.type = "MSEP", main = "PCR Cross-Validation - MSEP")
```




```r
opt_ncomp <- which.min(pcr_model$validation$PRESS)
cat("Optimal number of components:", opt_ncomp, "\n")
```

```
Optimal number of components: 60
```

```r
#Predict BMI on test set using optimal number of components
Y_pred <- predict(pcr_model, newdata = X_test, ncomp = opt_ncomp)
#Evaluate prediction performance (RMSE)
rmse <- sqrt(mean((Y_test - Y_pred)^2))
cat("  Test RMSE:", round(rmse, 3), "\n")
```

```
  Test RMSE: 3.712
```

Justification:

1)Data Standardization: Pressure variables were scaled to ensure equal contribution to the PCA. This means PCR was performed using the correlation matrix, which is appropriate because the variables may be on different scales.

2)Dimensionality Reduction with PCA: PCA was applied to the scaled data to reduce dimensionality. The number of principal components to retain was determined using cross-validation. Based on the MSEP curve, the optimal number of components selected was 60, as it minimized prediction error without overfitting.

3)Train-Test Split: The data was split into 80% training and 20% test sets. The PCR model was trained on the training data and evaluated on the test set.

4)Model Fitting and Prediction: The PCR model was fitted using the pcr() function from the pls package. The selected principal components were used to predict BMI for test observations.