Exercise 3

AUTHOR
Collin Real (yhi267)

a.

```
library(caret)
```

```
Loading required package: ggplot2

Warning: package 'ggplot2' was built under R version 4.3.2

Loading required package: lattice
```

```
data(tecator)

# Use ?tecator to see more details
?tecator

str(absorp)
```

```
 num [1:215, 1:100] 2.62 2.83 2.58 2.82 2.79 ...
```

```
str(endpoints)
```

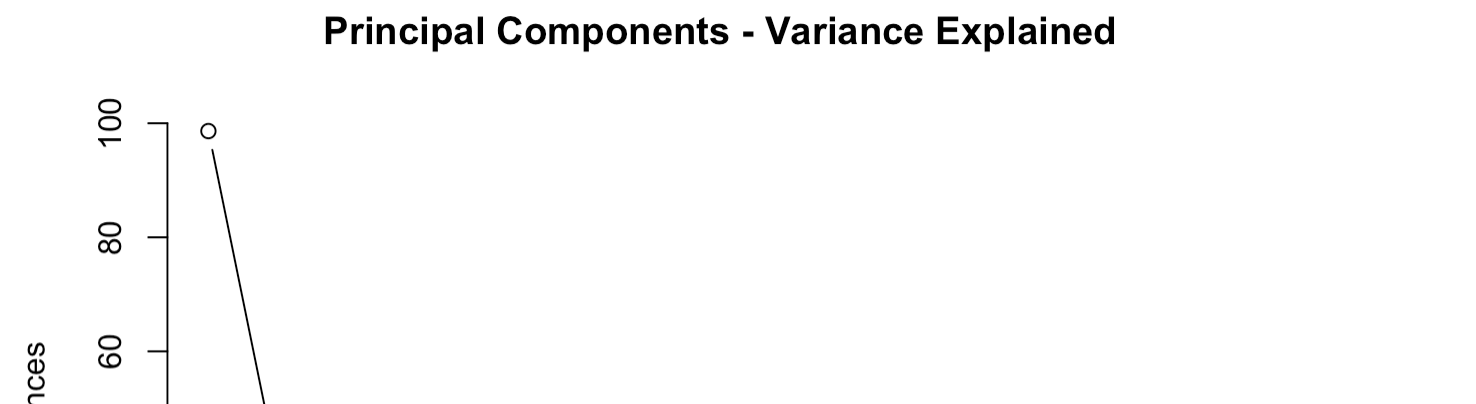
```
 num [1:215, 1:3] 60.5 46 71 72.8 58.3 44 44 69.3 61.4 61.4 ...
```

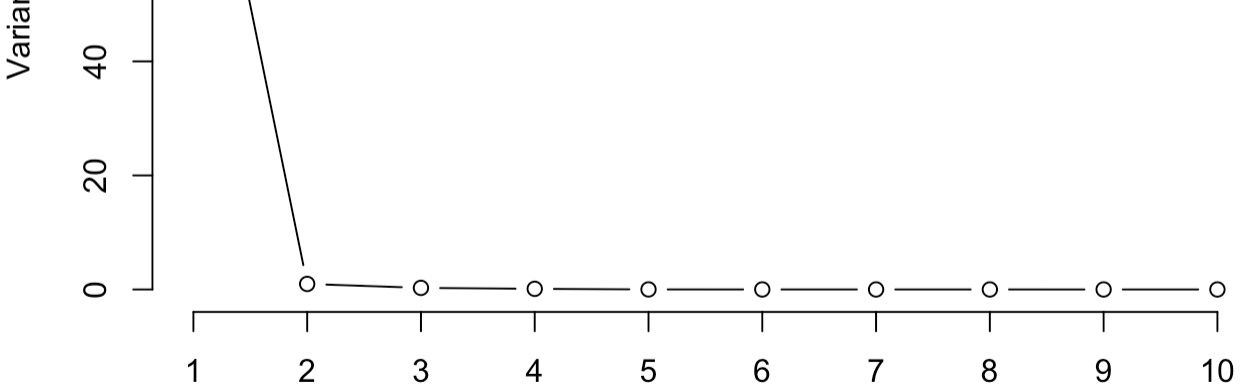
```
# Extract individual components
moisture <- endpoints[, 1]
fat <- endpoints[, 2]
protein <- endpoints[, 3]
```

b.

```
pca <- prcomp(absorp, scale. = TRUE)

# Plot the variance
plot(pca, type = "l", main = "Principal Components - Variance Explained")
```






```
# Calculate the cumulative variance
cum_variance <- cumsum(pca$sdev^2) / sum(pca$sdev^2)

# Determine the number of principal components
eff_dim <- which(cum_variance >= 0.95)[1]
eff_dim
```

```
[1] 1
```

Our PCA analysis indiciates that the first principal component captures almost all of the variance in our data. This conclusion is illustrated in our scree plot, which shows most of the variance explained by the first principal component with subsequent components explaining minimal additional variance.

c.

```
set.seed(123)
train_index <- createDataPartition(endpoints[,1], p = .8, list = FALSE)
train_data <- absorp[train_index, ]
test_data <- absorp[-train_index, ]
train_moisture <- endpoints[train_index, 1]
test_moisture <- endpoints[-train_index, 1]
```

```
colnames(train_data) <- paste0("V", 1:ncol(train_data))
colnames(test_data) <- paste0("V", 1:ncol(test_data))
pre_process <- preProcess(train_data, method = c("center", "scale", "pca"))
train_transformed <- predict(pre_process, train_data)
test_transformed <- predict(pre_process, test_data)
```

```
ols <- train(train_transformed, train_moisture, method = "lm")
ols
```

```
Linear Regression

175 samples
  2 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
```

```
Summary of sample sizes: 175, 175, 175, 175, 175, 175, ...
Resampling results:

  RMSE      Rsquared  MAE
  8.785449  0.266636  7.079

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
pcr <- train(train_transformed, train_moisture, method = "pcr", trControl = trainControl("cv"
pcr
```

```
Principal Component Analysis

175 samples
  2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 158, 157, 157, 157, 159, 159, ...
Resampling results:

  RMSE      Rsquared   MAE
  8.684837  0.2663912  7.324718

Tuning parameter 'ncomp' was held constant at a value of 1
```

```
pls <- train(train_transformed, train_moisture, method = "pls", trControl = trainControl("cv"
pls
```

```
Partial Least Squares

175 samples
  2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 156, 159, 158, 157, 159, 158, ...
Resampling results:

  RMSE      Rsquared   MAE
  8.714407  0.2602078  7.358483

Tuning parameter 'ncomp' was held constant at a value of 1
```

```
ols_pred <- predict(ols, test_transformed)
pcr_pred <- predict(pcr, test_transformed)
pls_pred <- predict(pls, test_transformed)

postResample(ols_pred, test_moisture)
```

```
     RMSE  Rsquared       MAE
8.3734237 0.1975907 6.5058512
```

```
postResample(pcr_pred, test_moisture)
```

```
     RMSE  Rsquared       MAE 
8.4750029 0.1790017 6.7827463 
```

```
postResample(pls_pred, test_moisture)
```

```
     RMSE  Rsquared       MAE 
8.4729782 0.1794105 6.7779390 
```

The optimal tuning parameters for PCA and PLS are the number of components where the value is held constant at 1. There are no tuning parameters for the linear regression because it's a straightfoward implementation of the ordinary least squares regression. Both PCA and PLS models were tested with only one component for these results. Experimenting with a higher number of components to improve model performance can be explored for further tuning.

d. To determine the best predictive model, we evaluate the performance metrics. **Linear Regression:** RMSE: 8.785, R-squared: 0.267; MAE: 7.079 **Principal Component Regression:** RMSE: 8.6845; R-squared: 0.266; MAE: 7.325 **Partial Lease Squares:** RMSE: 8.714; R-squared: 0.260; MAE: 7.358. These metrics indicate that the Principal Components Regression model is slightly better than the other models and has the lowest RMSE; however, the overall differences between these models is not substantial which is indicative that no model is significantly better predictive power.

e. I would choose the model with the lowest Root Mean Square Error (RMSE), so I would use the Principal Components Regression model to predict the percentage of moisture from the IR spectroscopy data. The reason I would choose the model with the lowest RMSE is because it provides the most accurate predictions.