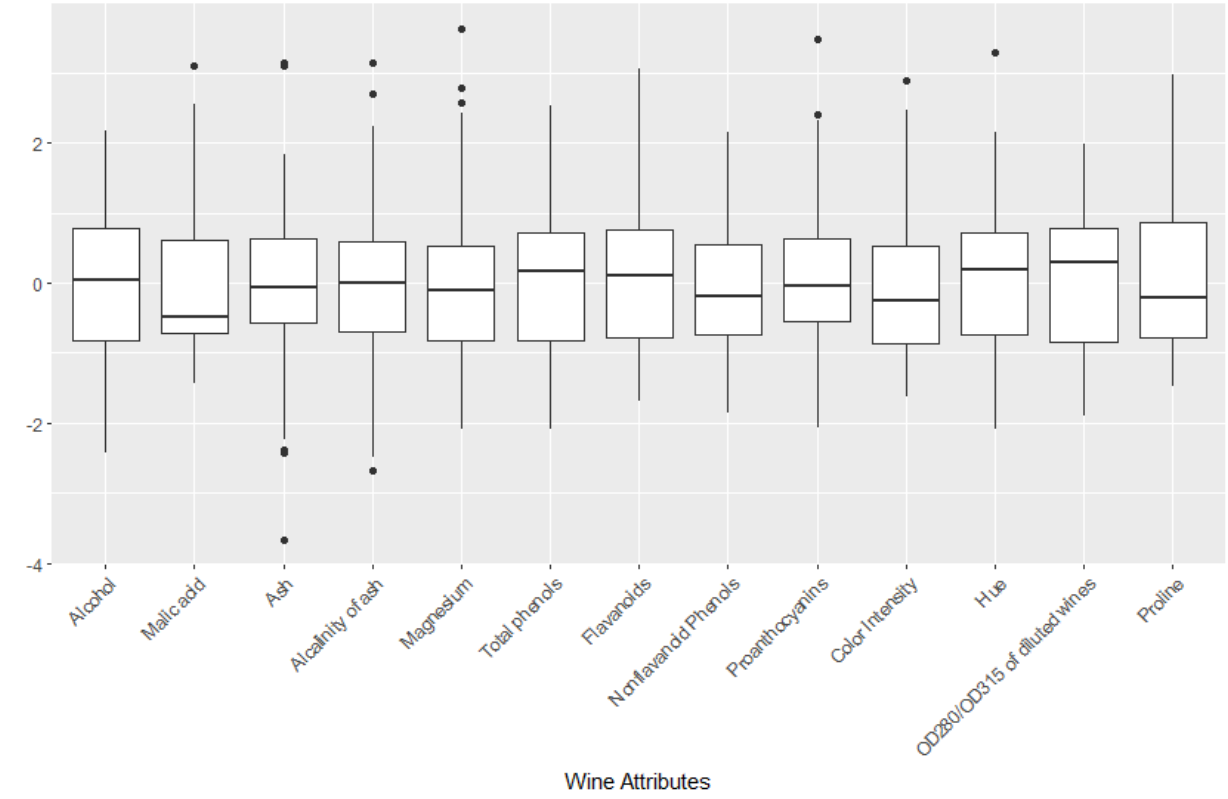
Line 43: Calculates the mean of every column and applies it to scaling

```
> apply(wine_scaled, 2, mean)
                     Alcohol                    Malic acid                          Ash           Alcalinity of ash
                -5.232102e-16                 -1.201584e-16                 -3.760675e-16                 -2.310037e-16
                   Magnesium                 Total phenols                   Flavanoids         Nonflavanoid Phenols
                -5.647851e-17                 -2.276264e-16                  1.675772e-16                 -1.755305e-16
               Proanthocyanins            Color Intensity                           Hue OD280/OD315 of diluted wines
                -3.610577e-17                  1.807604e-16                  1.112955e-16                  6.150060e-17
                      Proline
                 1.407291e-16
> |
```

Line 44: Calculates the standard deviation of every column and applies it to scaling

```
> apply(wine_scaled, 1, sd)
  [1] 0.9727858 0.7435431 1.1376523 0.5557609 0.9451675 0.9167995 0.8198611 0.9955096 0.9017866 0.9702920 0.8075282
 [12] 0.7994999 1.2902038 1.3410812 0.7963271 0.6944472 0.5957341 1.1065656 0.8484084 0.9742810 0.6444981 0.8104487
 [23] 0.6735878 0.6784847 0.9575878 0.7396011 0.6959366 0.6279918 0.7982750 0.7628353 0.8731456 0.5757438 0.8294749
 [34] 0.5674619 0.5502159 0.7320312 0.5469166 0.6937057 1.1453858 0.7961347 0.6882637 1.0328573 0.5895686 0.7364033
 [45] 0.7390835 0.8856126 0.9409444 0.5858646 0.8254480 1.2674864 0.8952858 1.0642499 0.7096851 0.8079725 0.7601082
 [56] 0.8676854 0.7161920 1.0160040 1.0354484 1.1080541 0.8169227 0.7739069 1.0261892 0.9465510 0.6313028 1.2007797
 [67] 0.8235678 0.9491479 1.5977185 0.6646794 1.0873936 0.6856968 1.2872865 0.9532897 0.7290656 0.8315357 0.7093582
 [78] 1.2337904 0.7479783 0.9773745 0.5863502 0.9549004 0.9321170 0.8520047 0.6734195 0.8221230 1.0989209 0.7263419
 [89] 0.9329040 0.7402557 0.7852461 0.8304657 0.8292832 0.8764098 1.5639771 1.1737148 0.8034603 1.0028004 1.1346599
[100] 0.8866184 0.5700625 0.6319091 0.6521448 0.7022019 1.0114268 0.6166726 0.6838886 0.7482259 0.9768125 1.4975333
[111] 0.6267857 1.0647401 0.8222766 0.8157462 1.4252517 0.7204607 0.7031905 0.7381383 0.7855579 0.8175100 1.4979557
[122] 1.0493407 1.1873657 1.3110714 0.7314486 0.7728168 1.1764454 0.7895635 0.8540388 0.8221015 0.7623959 0.9006042
[133] 0.8678354 0.8652829 0.9944835 1.2107893 1.3724757 0.9868819 0.8840946 0.9148605 0.7608020 1.0530124 0.9502061
[144] 0.9901022 0.8756251 1.3342636 1.1503862 0.9601366 1.1491489 1.2576281 1.2197320 1.1255205 1.2331295 0.9661510
[155] 1.3056337 1.1295744 1.2537178 1.4402862 1.2693189 1.0221198 0.8807909 0.9569368 0.7892602 1.0602509 1.1381092
[166] 1.0911262 1.0320931 0.9867378 1.2238015 0.7704305 1.1241133 1.0697593 1.2880355 0.9366848 1.2450173 1.1073400
[177] 1.2912621
```

Line 56-61: Print boxplot of all Wine Attributes



Line 66-68: SVM Model 1 - Linear Kernel creation

```
Call:
svm(formula = wine.class ~ Alcohol + Ash + `Alcalinity of ash` + Magnesium + `Total phenols` + Flavanoids,
    data = trainingX, kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  34
```

Line 70-71: Confusion Matrix of SVM Model 1

```
       Predicted
Actual  1  2  3
     1 48  0  0
     2  2 53  3
     3  0  1 34
> |
```

Line 72-73: Table of precision, recall, and F1 values for SVM Model 1

```
  precision    recall        f1
1 0.9600000 1.0000000 0.9795918
2 0.9814815 0.9137931 0.9464286
3 0.9189189 0.9714286 0.9444444
>
```

Line 77-79: SVM Model 2 - Radial Kernel creation

```
Call:
svm(formula = wine.class ~ Alcohol + Ash + `Alcalinity of ash` + Magnesium + `Total phenols` + Flavanoids,
    data = trainingX, kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  71
```

Line 80-81: Confusion Matrix of SVM Model 2

```
       Predicted
Actual  1  2  3
     1 48  0  0
     2  0 57  1
     3  0  0 35
> |
```

Line 82-83: Table of precision, recall, and F1 values for SVM Model 2

```
> svm_metrics
  precision    recall        f1
1 1.0000000 1.0000000 1.0000000
2 1.0000000 0.9827586 0.9913043
3 0.9722222 1.0000000 0.9859155
> |
```

Line 89-91: Tuning an SVM model to find the optimal gamma and cost values

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 gamma cost
  0.23   16

- best performance: 0.03571429
```

Line 94-97: Take best gamma and cost values and use them to create an SVM model (SVM model 2 - radial TUNED)

```
> cm
       Predicted
Actual  1  2  3
     1 10  0  0
     2  0 13  0
     3  0  1 12
```

Line 98-99: Table of precision, recall, and F1 values for SVM Model 2 TUNED

```
> svm_metrics
  precision    recall       f1
1 1.0000000 1.0000000 1.000000
2 0.9285714 1.0000000 0.962963
3 1.0000000 0.9230769 0.960000
>
```

Line 103-105: Model 3 - KNN and print out contingency table

```
knn.predicted  1  2  3
            1 48  1  0
            2  0 56  0
            3  0  1 35
```

Line 106-107: Table of precision, recall, and F1 values for Model 3 - KNN

```
> svm_metrics
  precision    recall       f1
1 1.0000000 0.9795918 0.9896907
2 0.9655172 1.0000000 0.9824561
3 1.0000000 0.9722222 0.9859155
```

Conclusion:

The following are portions of the end table that each model had the best metric on:
SVM Model 1: Recall1 (1 metric)
SVM Model 2: Precision1, Recall1, F1 1, Precision2, F1 2, Recall3, F1 3 (7 metrics)
SVM Model 2 TUNED: Precision1, Recall1, F1 1, Recall2, Precision3 (5 metrics)
Model 3 KNN: Precision1, Recall2, Precision3, F1 3 (4 metrics)
Based on the above information, Model 2 is the best because it scored the highest in the most metrics compared to the other models