Homework 5

Collin Stewart

https://github.com/collings512/BIOS512_Collin_Stewart

This homework requires `wine.csv`, and the `tidyverse` and `Rtsne` packages. Install them if you haven't already!
See the following link for how to add new packages to Binder:
https://github.com/rjenki/BIOS512?tab=readme-ov-file#adding-packages-to-installr-later.
For readability and easier processing, please make each question part a different code chunk.

In [2]:
```
library(tidyverse)
library(Rtsne)
library(dplyr)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.2     ✓ readr     2.1.4
✓ forcats   1.0.0     ✓ stringr   1.5.0
✓ ggplot2   3.4.2     ✓ tibble    3.2.1
✓ lubridate 1.9.2     ✓ tidyr     1.3.0
✓ purrr     1.0.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

Question 1

a) Import your data.

In [3]:
```
wine <- read_csv("wine.csv")
```

```
Rows: 178 Columns: 14
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
dbl (14): Alcohol, Malicacid, Ash, Alcalinity_of_ash, Magnesium, Total_pheno...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

b) Check out the columns present using one of R's data frame summary.

In [4]:
```
glimpse(wine)
```

```
Rows: 178
Columns: 14
$ Alcohol                          <dbl> 14.23, 13.20, 13.16, 14.37, 13.24, 14.2…
$ Malicacid                        <dbl> 1.71, 1.78, 2.36, 1.95, 2.59, 1.76, 1.8…
$ Ash                              <dbl> 2.43, 2.14, 2.67, 2.50, 2.87, 2.45, 2.4…
$ Alcalinity_of_ash                <dbl> 15.6, 11.2, 18.6, 16.8, 21.0, 15.2, 14.…
$ Magnesium                        <dbl> 127, 100, 101, 113, 118, 112, 96, 121, …
$ Total_phenols                    <dbl> 2.80, 2.65, 2.80, 3.85, 2.80, 3.27, 2.5…
$ Flavanoids                       <dbl> 3.06, 2.76, 3.24, 3.49, 2.69, 3.39, 2.5…
$ Nonflavanoid_phenols             <dbl> 0.28, 0.26, 0.30, 0.24, 0.39, 0.34, 0.3…
$ Proanthocyanins                  <dbl> 2.29, 1.28, 2.81, 2.18, 1.82, 1.97, 1.9…
$ Color_intensity                  <dbl> 5.64, 4.38, 5.68, 7.80, 4.32, 6.75, 5.2…
$ Hue                              <dbl> 1.04, 1.05, 1.03, 0.86, 1.04, 1.05, 1.0…
$ `0D280_0D315_of_diluted_wines`   <dbl> 3.92, 3.40, 3.17, 3.45, 2.93, 2.85, 3.5…
$ Proline                          <dbl> 1065, 1050, 1185, 1480, 735, 1450, 1290…
$ class                            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
```

c) Get summary statistics on the numeric variables.

In [5]:
```
summary(wine)
```

```
   Alcohol        Malicacid          Ash        Alcalinity_of_ash
Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60
1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20
Median :13.05   Median :1.865   Median :2.360   Median :19.50
Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49
3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50
Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00
  Magnesium      Total_phenols     Flavanoids    Nonflavanoid_phenols
Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300
1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700
Median : 98.00   Median :2.355   Median :2.135   Median :0.3400
Mean   : 99.74   Mean   :2.295   Mean   :2.029   Mean   :0.3619
3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375
Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600
Proanthocyanins Color_intensity       Hue         0D280_0D315_of_diluted_wines
Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270
1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938
Median :1.555   Median : 4.690   Median :0.9650   Median :2.780
Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612
3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170
Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000
   Proline           class
Min.   : 278.0   Min.   :1.000
1st Qu.: 500.5   1st Qu.:1.000
Median : 673.5   Median :2.000
Mean   : 746.9   Mean   :1.938
3rd Qu.: 985.0   3rd Qu.:3.000
Max.   :1680.0   Max.   :3.000
```

Question 2

a) Scale and center your data

Hint: Use a `mutate()` statement across all columns **except class** with `function(x) as.numeric(scale(x))`.

In [6]:
```
wine_scaled <- wine %>%
    mutate(across(-class, ~ as.numeric(scale(.))))
```

b) Based on what you saw in the summary statistic table from the imported data, why would scaling and centering this data be helpful before we perform PCA?

Scaling and centering the data will be important for PCA because most columns (except class) have a wide distribution of possible values compared to each other. For example, values for "Alcohol" are distributed between 11.03 and 14.83, but values for "Proline" are distributed between 278.0 and 1680.0. Since we don't necessarily know which variables are most important, we want to assess their variance equally. Otherwise, some fields may weigh the PCA more than other columns due to the magnitude of their values.

Question 3

a) Perform PCA

In [7]:
```
wine_pca <- prcomp(wine_scaled %>% select(-class), center = FALSE, scale. = FALSE)

summary(wine_pca)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     2.169 1.5802 1.2025 0.95863 0.92370 0.80103 0.74231
Proportion of Variance 0.362 0.1921 0.1112 0.07069 0.06563 0.04936 0.04239
Cumulative Proportion  0.362 0.5541 0.6653 0.73599 0.80162 0.85098 0.89337
                           PC8     PC9   PC10    PC11    PC12    PC13
Standard deviation     0.59034 0.53748 0.5009 0.47517 0.41082 0.32152
Proportion of Variance 0.02681 0.02222 0.0193 0.01737 0.01298 0.00795
Cumulative Proportion  0.92018 0.94240 0.9617 0.97907 0.99205 1.00000
```

b) How much of the total variance is explained by PC1? PC2? What function do we use to see that information?

Approximately 36.2% of the variance is explained by PC1, and approximately 19.21% of the variance is explained by PC2. To view this, use the function summary().

c) Why are we doing PCA first?

We are doing PCA first because there is such a relatively large datasets, 178 observations with 14 columns. This is a lot of data, and reducing the dimensions makes it easier to visualize the data and possible reveal underlying patterns. We can identify the direction of maximum variation while also preserving most of the structure.

d) What is the rotation matrix? Print it explicitly.¶

Hint: Check the notes for a simple way to do this!

In [8]:
```
wine_pca$rotation
```

A mat

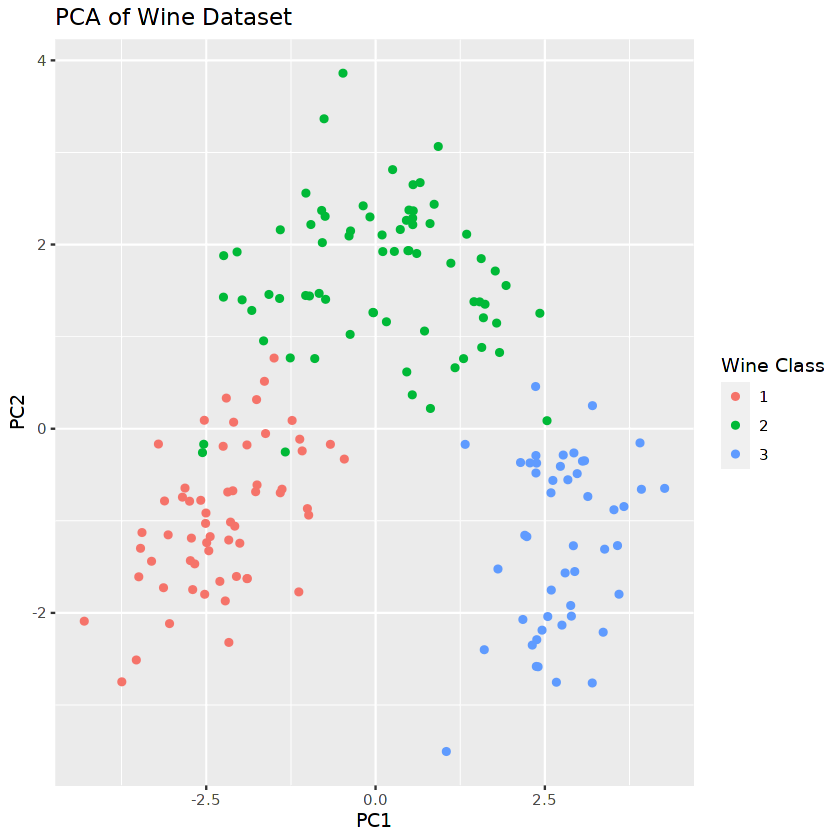
	PC1	PC2	PC3	PC4	
Alcohol	-0.144329395	-0.483651548	-0.20738262	-0.01785630	0.26566
Malicacid	0.245187580	-0.224930935	0.08901289	0.53689028	-0.03521
Ash	0.002051061	-0.316068814	0.62622390	-0.21417556	0.14302
Alcalinity_of_ash	0.239320405	0.010590502	0.61208035	0.06085941	-0.06610
Magnesium	-0.141992042	-0.299634003	0.13075693	-0.35179658	-0.72704
Total_phenols	-0.394660845	-0.065039512	0.14617896	0.19806835	0.14931
Flavanoids	-0.422934297	0.003359812	0.15068190	0.15229479	0.10902
Nonflavanoid_phenols	0.298533103	-0.028779488	0.17036816	-0.20330102	0.50070
Proanthocyanins	-0.313429488	-0.039301722	0.14945431	0.39905653	-0.13685
Color_intensity	0.088616705	-0.529995672	-0.13730621	0.06592568	0.07643
Hue	-0.296714564	0.279235148	0.08522192	-0.42777141	0.17361
0D280_0D315_of_diluted_wines	-0.376167411	0.164496193	0.16600459	0.18412074	0.10116
Proline	-0.286752227	-0.364902832	-0.12674592	-0.23207086	0.15786

e) Plot PC1 vs. PC2, using the wine class as labels for coloring.

Hint: You'll first need a data set with only PC1 and PC2, then add back the class variable from your scaled data set with a mutate() statement. Then, you can use color = factor(class) in your ggplot statement.

In [9]:
```
wine_pca_dataframe <- as.data.frame(wine_pca$x) %>%
    mutate (class = wine_scaled$class)

ggplot(wine_pca_dataframe, aes(x = PC1, y = PC2, color = factor(class))) +
    geom_point() +
    labs(title = "PCA of Wine Dataset", x = "PC1", y = "PC2", color = "Wine Class")
```



f) What do you see after plotting PC1 vs. PC2? What does this mean in context of wine classes?

I see 3 distinct clusters, with some small overlap of Wine Class 2 spilling over into the clusters for Wine Classes 1 and 3. This means that the values of PC1 and PC2 are meaningfully tied to the value of class, either 1, 2, or 3. In the context of the wine, this means that the 13 numeric variables relating to the chemical properties of the wine are most likely also clustered according to the value of "class" for the wine.

g) Give an example of data where PCA would fail. You can describe the data or do a simulation.

Hint: Our notes have a few examples!

PCA would fail if a dataset had classes that shared the same center. For example, PCA would fail with concentric circles (shaped like a target) because each ring/circle shares the same center. In this scenario, no rotation will ever be able to separate the two axes.

h) Explain the difference between vector space and manifold, and how these terms apply to what we did/will do with T-SNE.

Vector space is a linear space where data can be represented by vectors, or combinations of vectors. PCA works here, as it assumes data to be linear and that it can rotated. A manifold is a nonlinear space for higher dimension structures. T-SNE works for manifolds because it can reveal finer underlying clusters between wine types that PCA is unable to.

Question 4

a) Perform T-SNE

Set `seed = 123`.
Hint: Subset your PCA results to PC1–PC10, add the class variable back in, remove duplicates, then perform T-SNE.

In [10]:
```
set.seed(123)

wine_PC110 <- wine_pca_dataframe %>%
    select(PC1:PC10) %>%
    as.matrix()

tsne_frame <- Rtsne(wine_PC110, sims = 2, check_duplicates = FALSE)
results <- as_tibble(tsne_frame$Y)
colnames(results) <- c("Dim1", "Dim2")
results$class <- wine_pca_dataframe$class
```

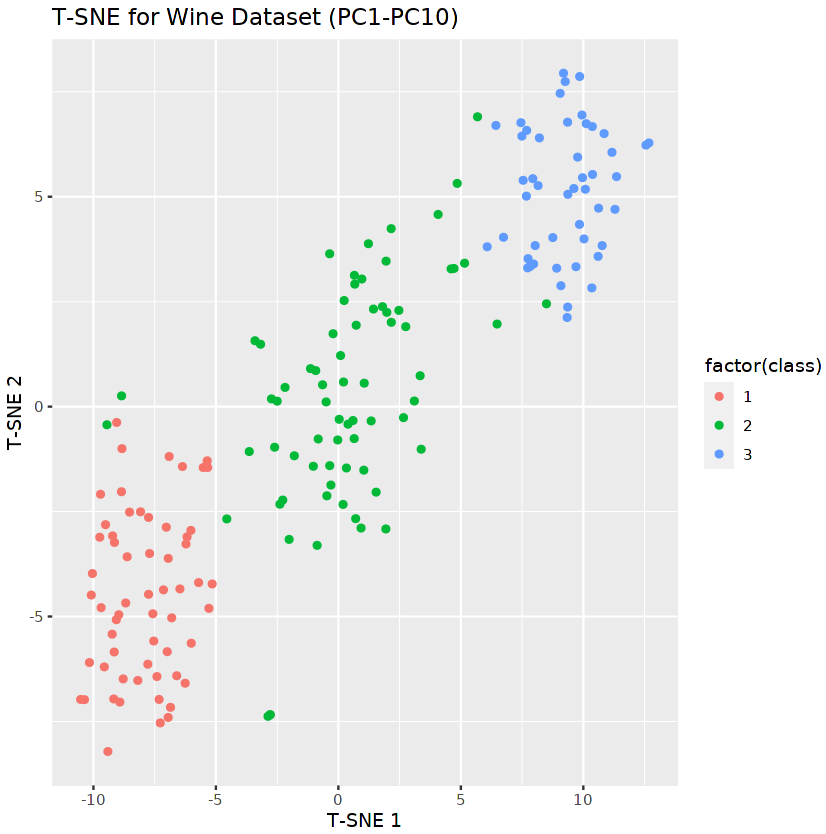
```
Warning message:
“The `x` argument of `as_tibble.matrix()` must have unique column names if
`.name_repair` is omitted as of tibble 2.0.0.
i Using compatibility `.name_repair`.”
```

b) Plot the results in 2D

Hint: Convert your T-SNE results to a tibble and add back the class variable from your scaled data set using a mutate() statement. Then, you can use color = factor(class) in your ggplot statement.

In [12]:
```
tsne_dataframe <- as_tibble(tsne_frame$Y) %>%
    mutate(class = wine_scaled$class)

ggplot(results, aes(x = Dim1, y = Dim2, color = factor(class))) +
    geom_point() +
    labs(title = "T-SNE for Wine Dataset (PC1-PC10)", x = "T-SNE 1", y = "T-SNE 2")
```



c) Why didn't we stop at PCA?

We didn't stop at PCA because it only covered the linear relationships in the data. In the previous ggplot, it identified 3 clusters along the axes of PC1 and PC2. However, it failed to address nonlinear relationships in the wine dataset, which T-SNE can identify. In this ggplot, there are still 3 distinct clusters, but now there is clear trendline along the axes of T-SNE1 and T-SNE2

d) What other types of data does this workflow make sense for?

This could be applied to physiological biometric data, such as heart rate, respiratory rate, height, weight, body temperature, blood oxygenation, ECG signals, etc. Doing an initial PCA on these variables may cluster according to classes based on someone's activity level, their age, addressing linear relationships in the data. Doing a T-SNE could later reveal new interesting patterns related to non-linear relationships in the data.