DImension Reduction

David Wen

8/3/2021

Dimension Reduction

Dimension reduction is a super-useful technique to visualize high-dimensional data. For example, it would be nice to visualize how all the features in the `mtcars` data set interact, except we can't plot them all on one plot since we have at most 3 axes (and we humans are horrible at visualizing more than 3). Luckily, we have some algorithms that are super useful for condensing data down to a workable number of features (typically 2, sometimes 3).

PCA

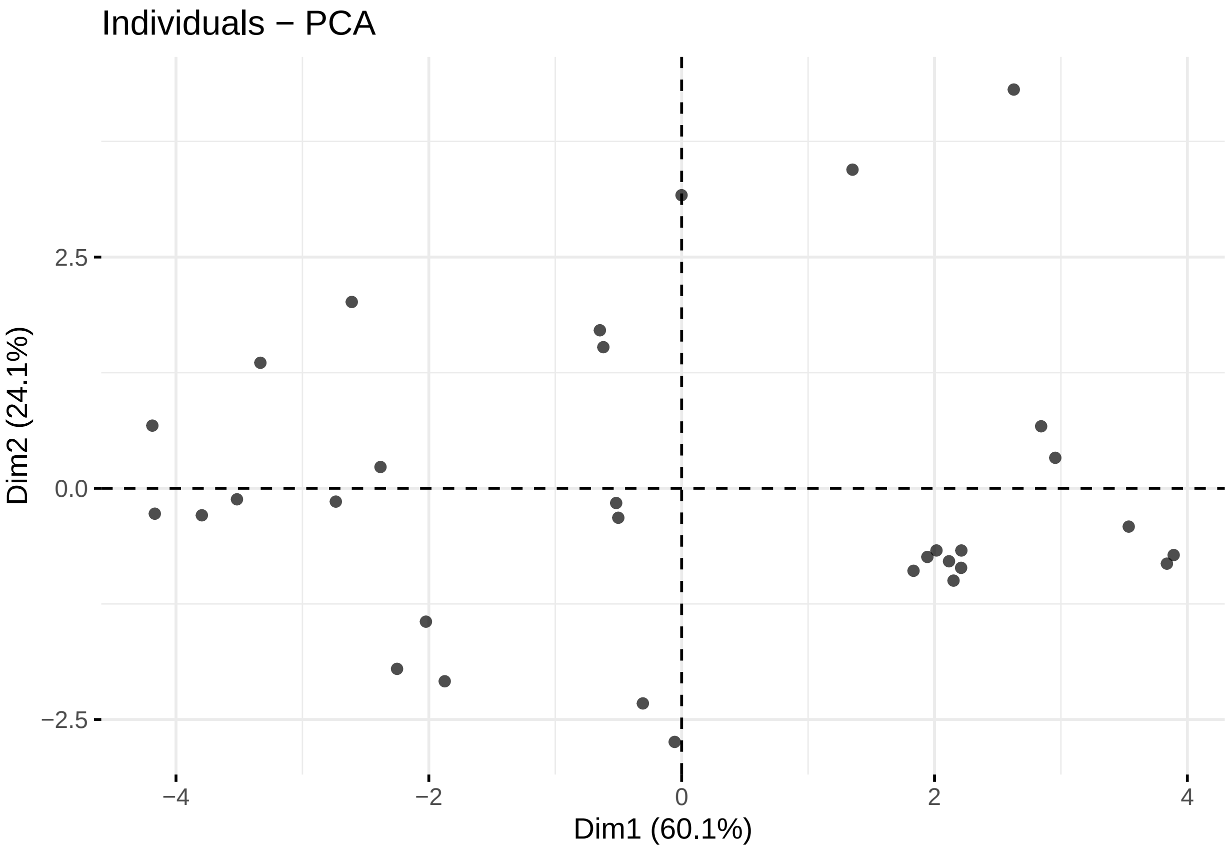
PCA stands for *principal component analysis*, which basically tries to preserve the shape of the data and removes the least important features. In `R`, you would use the`factoextra` package to visualize the data (although you can plot it yourself; it's just a bit more work). It's a relatively straightforward procedure:

```
library(factoextra)  # be sure it's already downloaded!
```

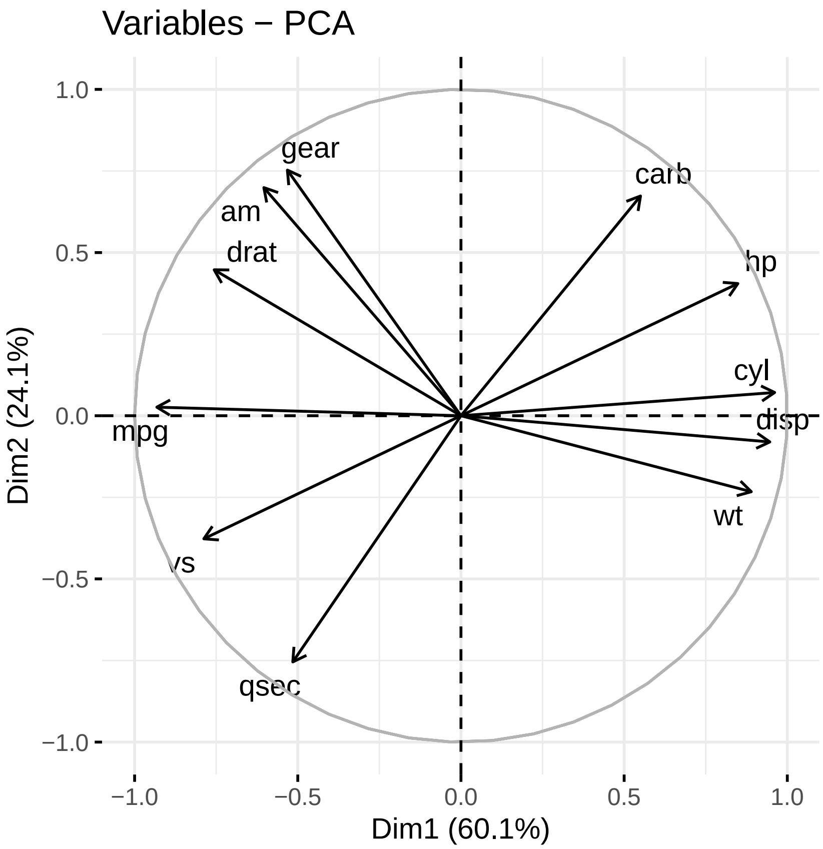
```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

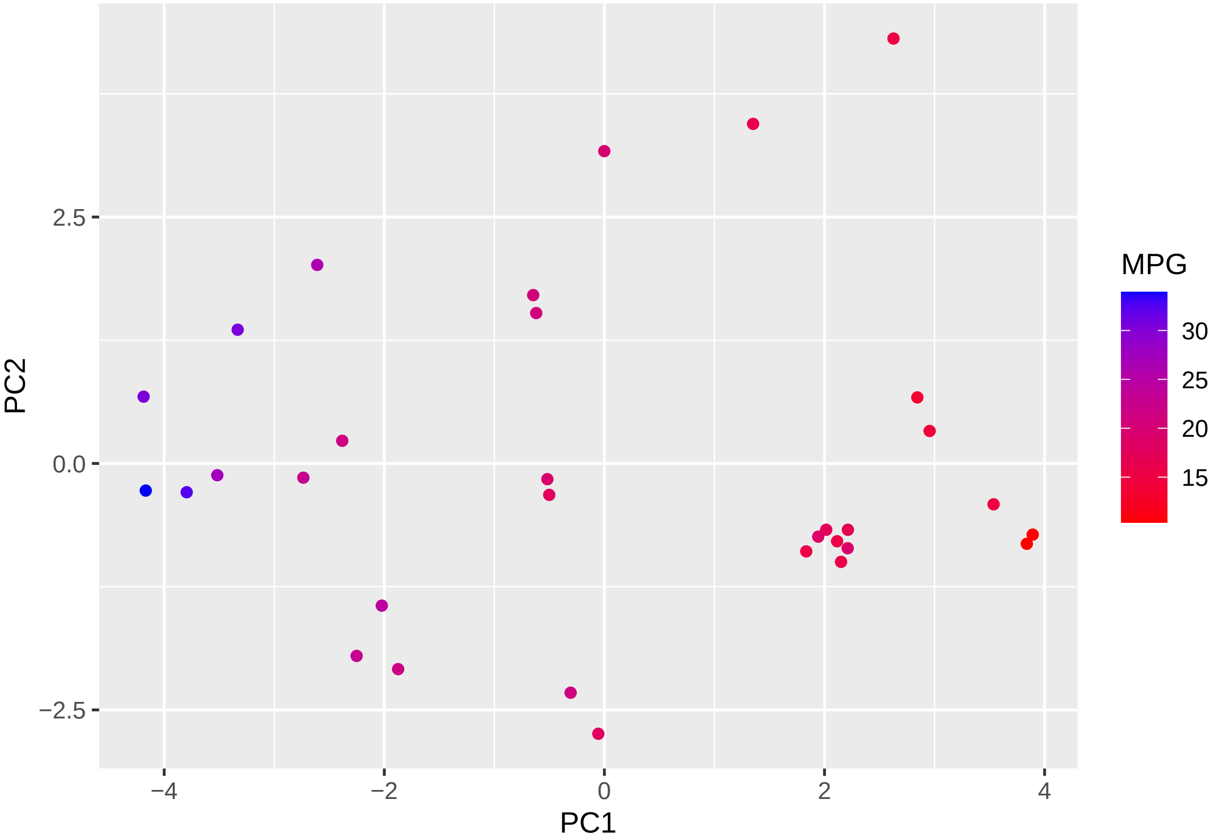
```
pca = prcomp(mtcars, scale. = T)
fviz_pca_ind(pca, labels = F, alpha.ind = 0.7)  # plot the data
```

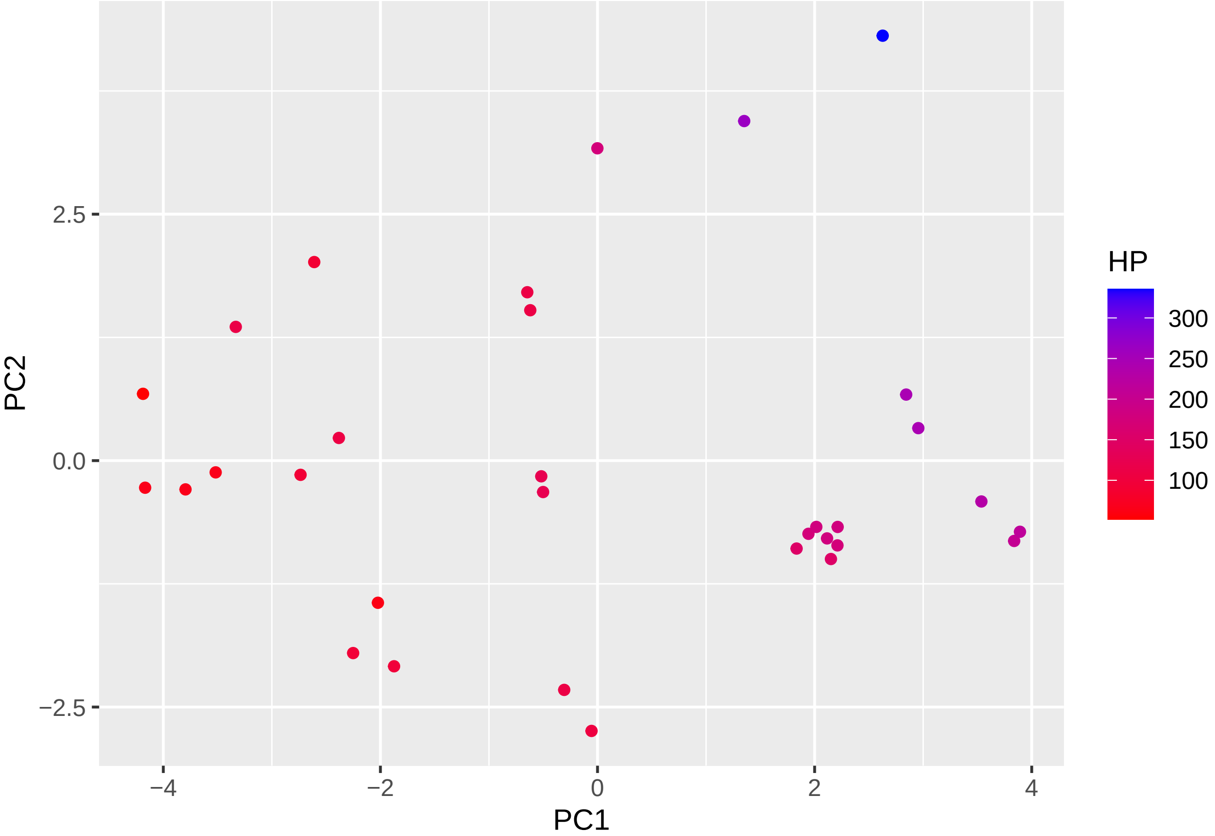


```
fviz_pca_var(pca, repel = T)  # see what is contributing to each component
```



First, note that we scale, or normalize, the data when we pass it to the pca function. Otherwise, the data with the highest variances tend to be weighted more, which is a problem because a larger variance can be explained by just having larger values (for example, in the `mtcars` data, the values that `mpg`takes on are around 10-35, whereas `hp` goes from around 50-350). Then, we visualize the data in redcued dimensions, where we notice that there's not much clustering of points (they're distributed relatively randomly). The `Dim` labels are the two most important axes that account for the variance within the points – basically, it's saying that about 60% of the variance between the samples can be explain by `Dim1`, and about 24% can be explained by `Dim2`. Finally, the last plot visualizes the contribution of each feature to the two dimensions. For example, a higher `mpg` tends to place samples on the negative end of `Dim1`, while a larger `hp` tends to place samples on both the positive `Dim1` and `Dim2`. We can visualize these effects in the following two plots:



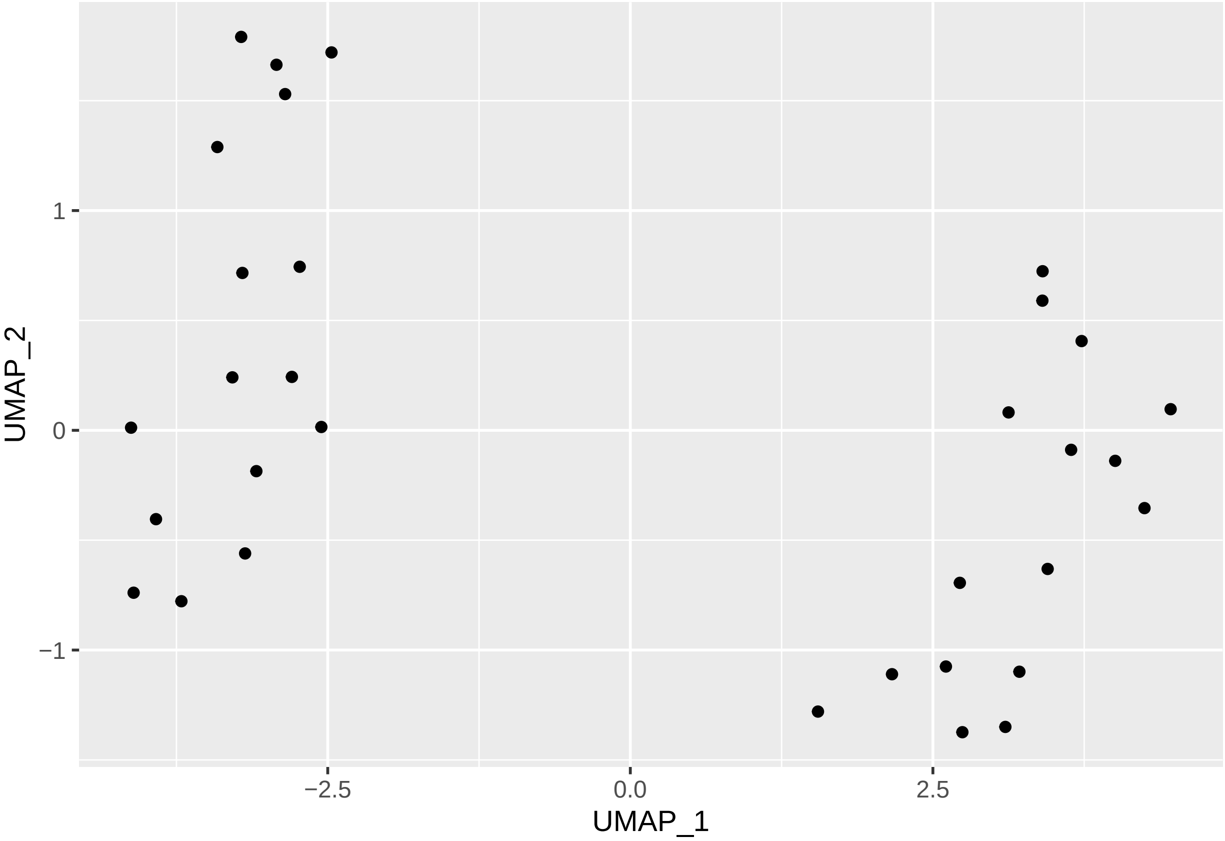


As an additional exercise, play around with the `pca` object. How is it different from all the other data types we've used before?

UMAP

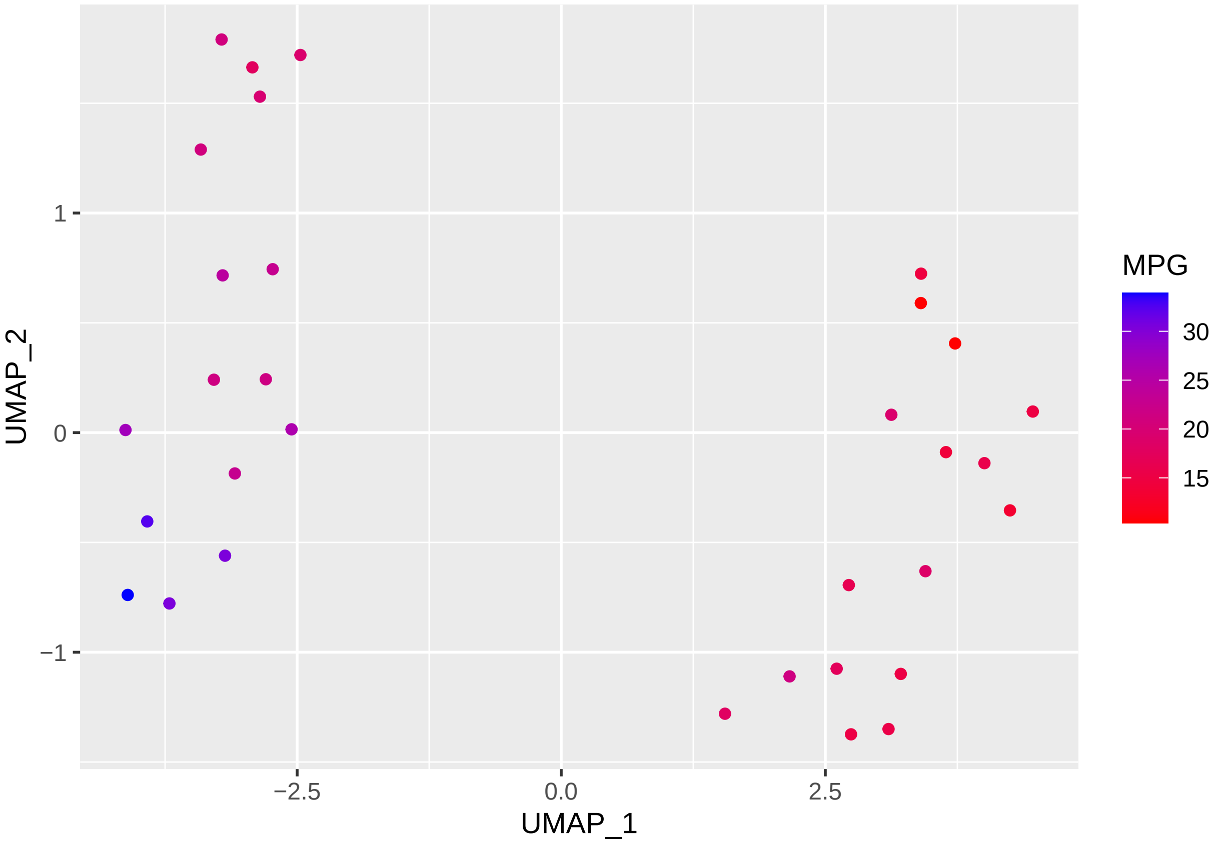
UMAP is another method used to reduce dimensionality (don't ask me about the math), and it's implemented in the `umap` package. Here's how you would use it:

```r
library(umap)

mtcars_umap = umap(mtcars)
umap_dims = data.frame(mtcars_umap$layout)  # turn into a data frame
names(umap_dims) = c("UMAP_1", "UMAP_2")  # assign names

ggplot(data = umap_dims, aes(x = UMAP_1, y = UMAP_2)) +
  geom_point()
```



The `umap` object is a little less descriptive, but you would normally overlay colors on the plot like so:

```r
ggplot(data = umap_dims, aes(x = UMAP_1, y = UMAP_2)) +
  geom_point(aes(color = mtcars$mpg)) +
  labs(color = "MPG") +
  scale_color_continuous(low = "red", high = "blue")
```



There's another, older method called *t-SNE* (both are commonly used in plotting single-cell RNAseq data), but it seems like UMAP just does a overall better job. Feel free to explore both!