Introduction to dimension reduction with PCA

Stephanie J. Spielman

Data Science for Biologists, Spring 2020

Imagine: You have 500 columns in your data. How do you...

- Visualize 500 columns?
- Determine which 500 columns contribute variation? (without variation, there is nothing to study)

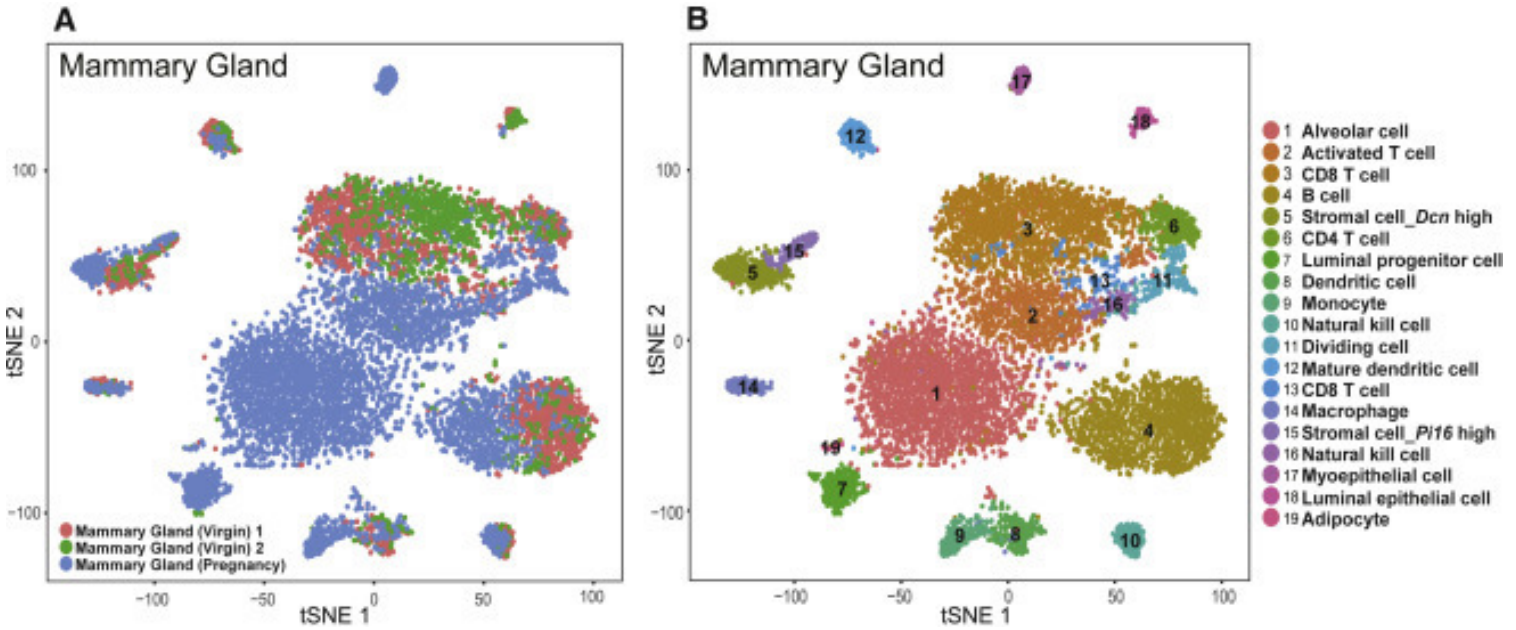
Imagine: You have 500 columns in your data. How do you...

- Visualize 500 columns?
- Determine which 500 columns contribute variation? (without variation, there is nothing to study)

Solution: *Reduce your dimensions!* 500 is just too many! Can we boil it down to a few *representative* columns?

- Principle components analysis (PCA)
- Linear discriminant analysis (LDA)
- uMAP or tSNE are commonly used for **visualization** in genomics
  - uMAP: "uniform manifold approximation and projection"
  - tSNE: "t-distributed stochastic neighbor embedding"

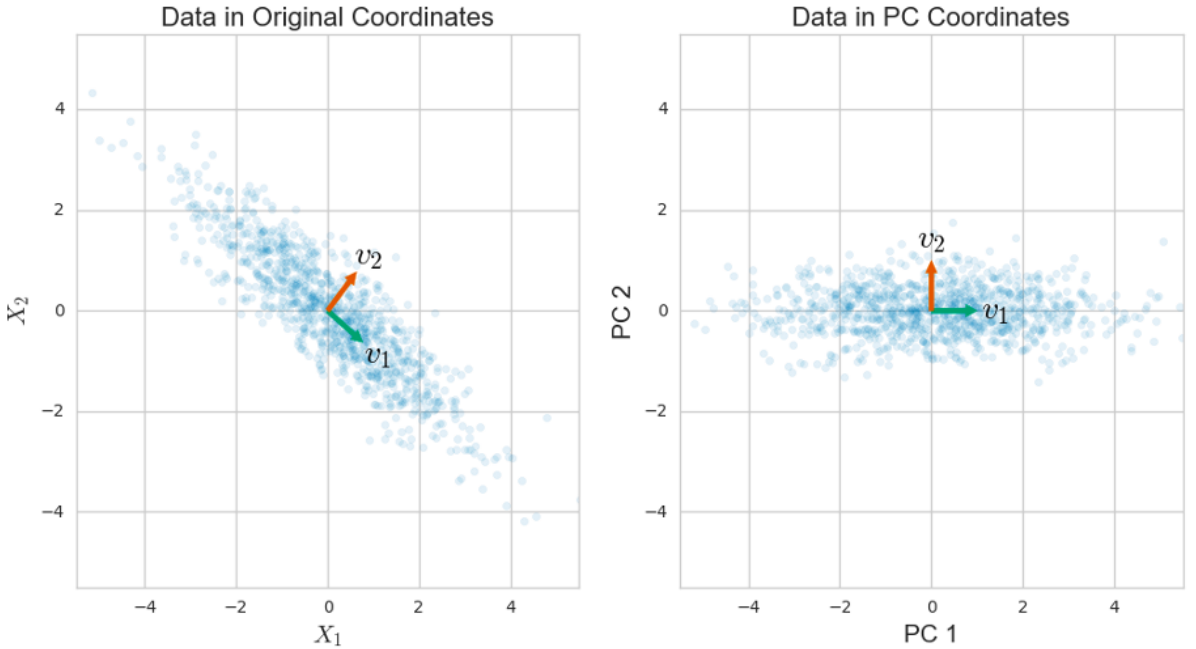
tSNE for funsies: The "Mouse Cell Atlas" from scRNA-seq

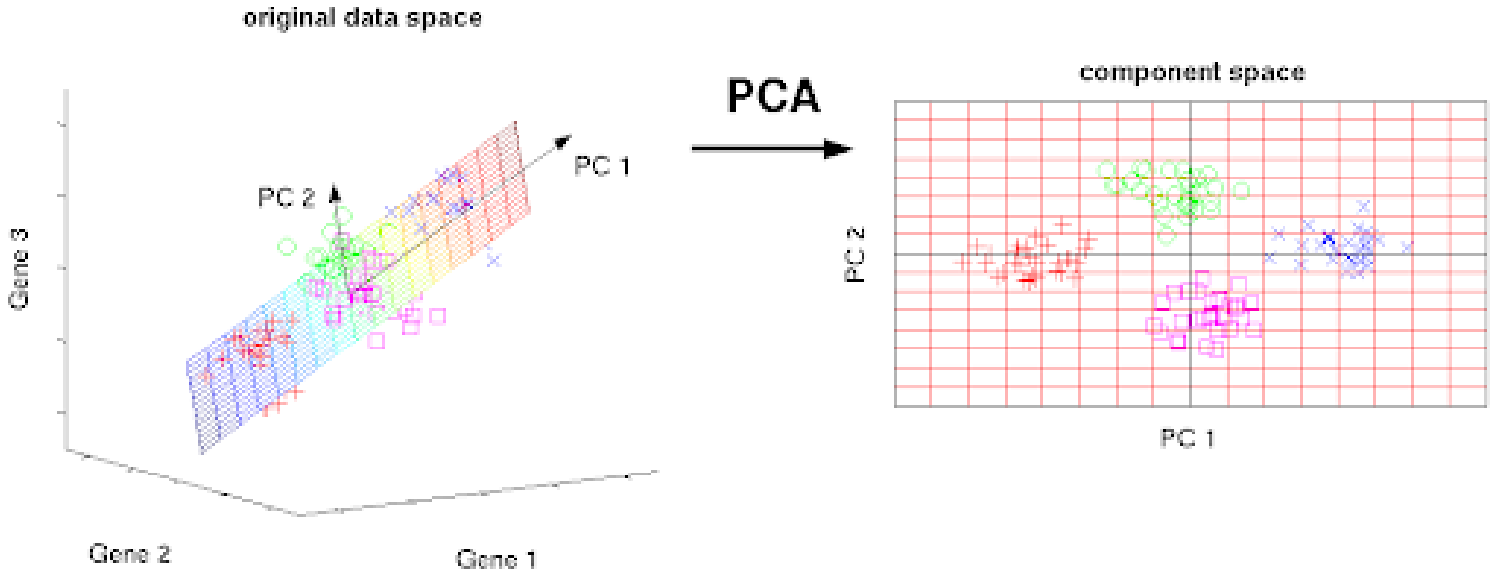


Source: https://doi.org/10.1016/j.cell.2018.02.001

Principle Components Analysis (PCA)

- A linear algebra technique to emphasize *axes of variation* the data.
- Principle component (PC) = a new *axis* constructed from the covariance matrix of the data
  - https://towardsdatascience.com/pca-eigenvectors-and-eigenvalues-1f968bc6777a
- *Not* a stochastic method - it is *deterministic* (same result every time)




original data space
PC 1
PC 2
Gene 3
Gene 2
Gene 1
PCA
component space
PC 2
PC 1

Conducting a PCA

```
## just making the URL fit..
wine_url <- paste0("https://raw.githubusercontent.com/sjspielman/",
                   "datascience_for_biologists/master/data/wine.csv")
wine <- read_csv(wine_url)
dplyr::glimpse(wine)
## Rows: 178
## Columns: 9
## $ Cultivar       <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A",…
## $ Alcohol        <dbl> 14.23, 13.20, 13.16, 14.37, 13.24, 14.20, 14.39, 14.06…
## $ MalicAcid      <dbl> 1.71, 1.78, 2.36, 1.95, 2.59, 1.76, 1.87, 2.15, 1.64, …
## $ Ash            <dbl> 2.43, 2.14, 2.67, 2.50, 2.87, 2.45, 2.45, 2.61, 2.17, …
## $ Magnesium      <dbl> 127, 100, 101, 113, 118, 112, 96, 121, 97, 98, 105, 95…
## $ TotalPhenol    <dbl> 2.80, 2.65, 2.80, 3.85, 2.80, 3.27, 2.50, 2.60, 2.80, …
## $ Flavanoids     <dbl> 3.06, 2.76, 3.24, 3.49, 2.69, 3.39, 2.52, 2.51, 2.98, …
## $ NonflavPhenols <dbl> 0.28, 0.26, 0.30, 0.24, 0.39, 0.34, 0.30, 0.31, 0.29, …
## $ Color          <dbl> 5.64, 4.38, 5.68, 7.80, 4.32, 6.75, 5.25, 5.05, 5.20, …
```

Conducting a PCA

1. Remove categorical columns
2. Scale all numeric columns to be centered at 0 with `scale()`. **Converts to a matrix**
3. PCA it up with `prcomp()`

Conducting a PCA

1. Remove categorical columns
2. Scale all numeric columns to be centered at 0 with `scale()`. **Converts to a matrix**
3. PCA it up with `prcomp()`

```
wine %>%
  select(-Cultivar) %>%
  scale() -> scaled_wine

# Demonstrating how scale() works first, and a lesson on "what is 0"?
head(scaled_wine, 3)
##        Alcohol   MalicAcid        Ash  Magnesium TotalPhenol Flavanoids
## [1,] 1.5143408 -0.56066822  0.2313998 1.90852151   0.8067217  1.0319081
## [2,] 0.2455968 -0.49800856 -0.8256672 0.01809398   0.5670481  0.7315653
## [3,] 0.1963252  0.02117152  1.1062139 0.08810981   0.8067217  1.2121137
##      NonflavPhenols      Color
## [1,]     -0.6577078  0.2510088
## [2,]     -0.8184106 -0.2924962
## [3,]     -0.4970050  0.2682629
mean(scaled_wine[,1]) ## Mean of the first column in the matrix
## [1] -8.591766e-16
mean(scaled_wine[,4]) ## Mean of the fourth column in the matrix
## [1] -4.073935e-17
```

Conducting a PCA

```
# And here it is in full!!
wine %>%
  select(-Cultivar) %>%
  scale() %>%
  prcomp() -> wine_pca
```

PCA output: The principle components (PCs)

```
wine_pca$x %>% as_tibble() -> wine_pca_components
wine_pca_components
## # A tibble: 178 x 8
##       PC1    PC2    PC3      PC4       PC5    PC6     PC7     PC8
##     <dbl>  <dbl>  <dbl>    <dbl>     <dbl>  <dbl>   <dbl>   <dbl>
##  1 -2.29  1.27  -0.205 -0.904   -0.272    -0.620 -0.558  -0.171
##  2 -1.24 -0.718 -0.722 -0.00793  0.0920   -0.137 -0.0389 -0.0863
##  3 -1.44  0.612  0.725  0.515    0.323     0.468  0.237  -0.286
##  4 -3.27  1.70  -0.451  0.598    0.000541 -0.341  0.416   0.543
##  5 -1.15  1.22   1.80  -0.297    0.308    -0.216 -0.294   0.112
##  6 -2.37  1.29  -0.183  0.477   -0.390    -0.665 -0.0161 -0.0184
##  7 -1.13  0.717 -0.608  0.653   -0.215     0.282 -1.07    0.0211
##  8 -1.49  1.37   0.345 -0.704   -0.0202   -0.223 -0.794   0.0756
##  9 -1.82  0.451 -1.47   0.840   -0.274    -0.425 -1.13   -0.0568
## 10 -2.12  0.363 -1.14   0.578   -0.447     0.350  0.252  -0.0474
## # … with 168 more rows
```

```
nrow(wine_pca_components)
## [1] 178
nrow(wine %>% select(-Cultivar))
## [1] 178
ncol(wine_pca_components)
## [1] 8
ncol(wine %>% select(-Cultivar))
## [1] 8
```

PCA output: The *loadings*

- The percent of variation in each variable explained by the given PC.
- How much does each variable in the data *load* on each PC? Range [-1,1]
  - NOT Pearson's correlation coefficient but similar in spirit
  - Sign indicates direction of the relationship

```
wine_pca$rotation
##                       PC1          PC2         PC3         PC4          PC5
## Alcohol        -0.22654303  0.501465867 -0.40546435  0.21466476 -0.061544336
## MalicAcid       0.31157625  0.322362963 -0.02402787  0.07622524  0.865553459
## Ash            -0.06559042  0.430407129  0.73311784  0.09876505 -0.009604591
## Magnesium      -0.25009429  0.318535452  0.16787273 -0.80749990 -0.032273948
## TotalPhenol    -0.53747068  0.004786501  0.10823030  0.32034021  0.111965977
## Flavanoids     -0.55663487 -0.072794005  0.13105601  0.25795708  0.121640862
## NonflavPhenols  0.42920190  0.136227208  0.35263552  0.32878655 -0.352644293
## Color           0.04263628  0.577943979 -0.34078880  0.09506746 -0.306927958
##                       PC6          PC7         PC8
## Alcohol        -0.1282794 -0.681546057  0.04449257
## MalicAcid      -0.1591111  0.119692299 -0.06155780
## Ash             0.4838802 -0.151710216  0.07735021
## Magnesium      -0.3897976  0.037144643 -0.03724463
## TotalPhenol    -0.3082298  0.307596024  0.62818264
## Flavanoids     -0.1094416  0.095275330 -0.75185755
## NonflavPhenols -0.6528980 -0.001695339 -0.11890648
## Color           0.1971978  0.626958878 -0.11322052
```

Clean up the loadings

```
# Into a usable format
wine_pca$rotation %>%
  as.data.frame() %>%
  rownames_to_column("quantity") %>%
  as_tibble() -> wine_pca_loadings

wine_pca_loadings %>% head()
## # A tibble: 6 x 9
##   quantity        PC1      PC2     PC3     PC4      PC5    PC6     PC7     PC8
##   <chr>         <dbl>    <dbl>   <dbl>   <dbl>    <dbl>  <dbl>   <dbl>   <dbl>
## 1 Alcohol     -0.227   0.501   -0.405   0.215  -0.0615  -0.128 -0.682   0.0445
## 2 MalicAcid    0.312   0.322   -0.0240  0.0762  0.866   -0.159  0.120  -0.0616
## 3 Ash         -0.0656  0.430    0.733   0.0988 -0.00960  0.484 -0.152   0.0774
## 4 Magnesium   -0.250   0.319    0.168  -0.807  -0.0323  -0.390  0.0371 -0.0372
## 5 TotalPhenol -0.537   0.00479  0.108   0.320   0.112   -0.308  0.308   0.628
## 6 Flavanoids  -0.557  -0.0728   0.131   0.258   0.122   -0.109  0.0953 -0.752
```

- **`TotalPhenol`** and **`Flavanoids`** load most strongly on PC1
- **`Color`** and **`Ash`** load most weakly on PC1

Standard deviation of components

```
wine_pca$sdev
## [1] 1.6403425 1.4035587 0.9757545 0.9117714 0.8274837 0.6365139 0.5909130
## [8] 0.3417867

# Convert to variance and normalize
wine_pca_variance <- wine_pca$sdev**2
wine_pca_variance / sum(wine_pca_variance) -> variation_explained
variation_explained
## [1] 0.33634045 0.24624712 0.11901210 0.10391587 0.08559117 0.05064374
0.04364728
## [8] 0.01460227
```

Standard deviation of components

```
wine_pca$sdev
## [1] 1.6403425 1.4035587 0.9757545 0.9117714 0.8274837 0.6365139 0.5909130
## [8] 0.3417867

# Convert to variance and normalize
wine_pca_variance <- wine_pca$sdev**2
wine_pca_variance / sum(wine_pca_variance) -> variation_explained
variation_explained
## [1] 0.33634045 0.24624712 0.11901210 0.10391587 0.08559117 0.05064374
0.04364728
## [8] 0.01460227
```

- PC1 explains ~33.6% of variation in the whole wine data set
- PC2 explains ~24.6% of variation in the whole wine data set
- **By definition, PC1 = explains most variation. PC2 = second most variation. etc.**

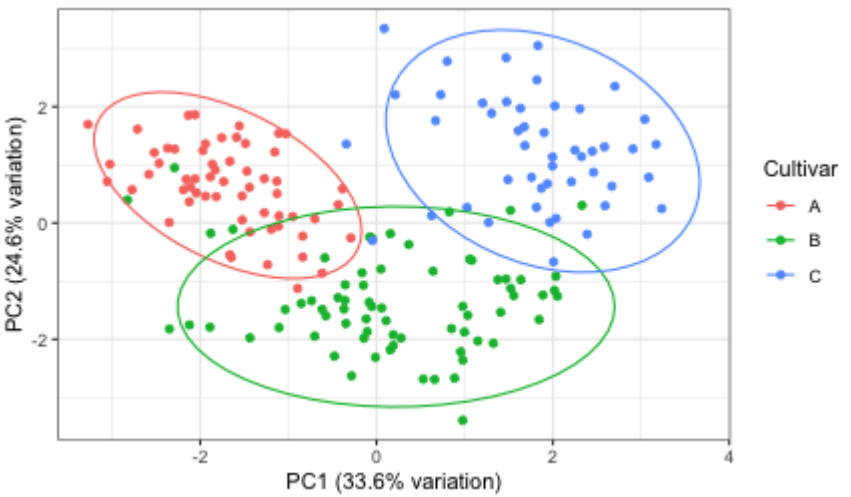
Visualizing the PCA

```
## The PCs themselves
wine_pca_components %>% head(3)
## # A tibble: 3 x 8
##     PC1    PC2    PC3      PC4     PC5    PC6     PC7     PC8
##   <dbl>  <dbl>  <dbl>    <dbl>   <dbl>  <dbl>   <dbl>   <dbl>
## 1 -2.29  1.27  -0.205 -0.904   -0.272  -0.620 -0.558  -0.171
## 2 -1.24 -0.718 -0.722 -0.00793  0.0920 -0.137 -0.0389 -0.0863
## 3 -1.44  0.612  0.725  0.515    0.323   0.468  0.237  -0.286

## how much do original variables LOAD on PCs
wine_pca_loadings %>% head(3)
## # A tibble: 3 x 9
##   quantity      PC1   PC2     PC3    PC4      PC5    PC6    PC7     PC8
##   <chr>       <dbl> <dbl>   <dbl>  <dbl>    <dbl>  <dbl>  <dbl>   <dbl>
## 1 Alcohol   -0.227  0.501 -0.405  0.215  -0.0615  -0.128 -0.682  0.0445
## 2 MalicAcid  0.312  0.322 -0.0240 0.0762  0.866   -0.159  0.120 -0.0616
## 3 Ash       -0.0656 0.430  0.733  0.0988 -0.00960  0.484 -0.152  0.0774

## what percent of variation is in each PC
variation_explained
## [1] 0.33634045 0.24624712 0.11901210 0.10391587 0.08559117 0.05064374
0.04364728
## [8] 0.01460227
```

The main plot we all want to make

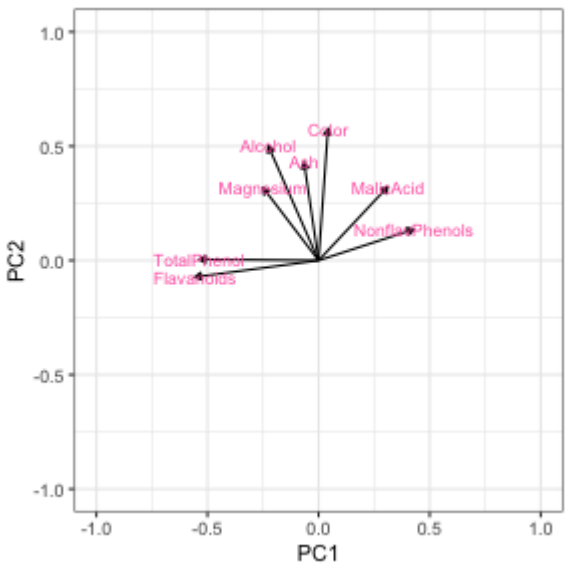
- PC1 *dicriminates* all three Cultivars
- PC2 *discriminates* Cultivar B from A, C. It does not discriminate A and C.

```
wine_pca_components %>%
  # Bring the original variables back in
  bind_cols(wine) %>%
  ggplot(aes(x = PC1, y = PC2, color = Cultivar)) +
    geom_point() +
    stat_ellipse() +
    labs(x = "PC1 (33.6% variation)",
         y = "PC2 (24.6% variation)")
```



Visualize and interpret the loadings

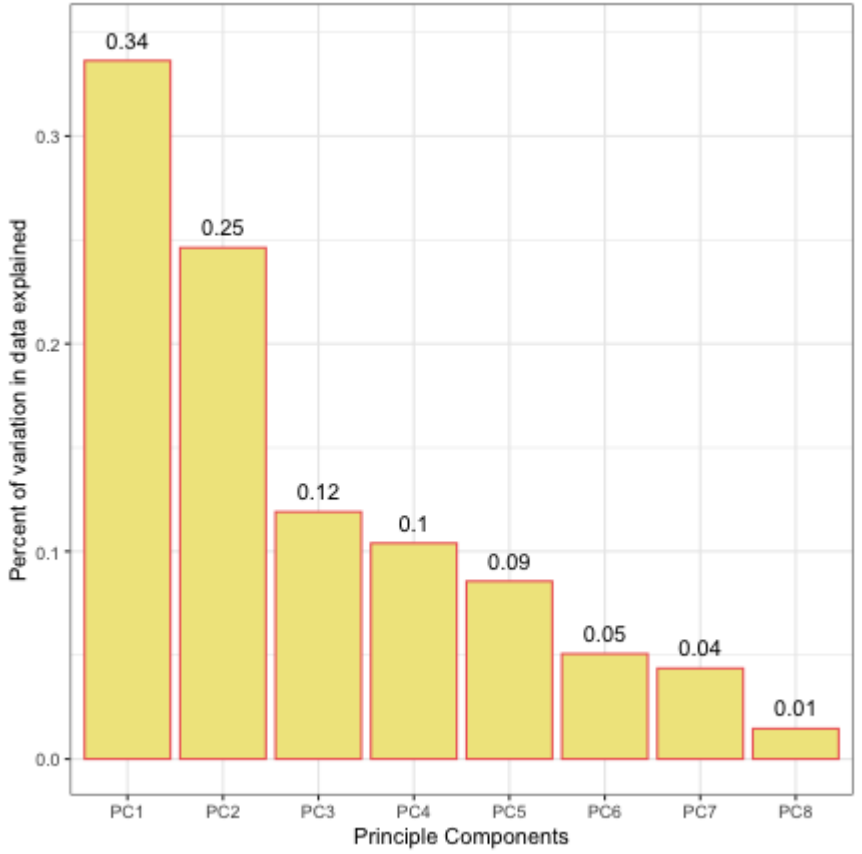
```
# uses library grid (gets loaded with ggplot2, NOT part of tidyverse!)
my_arrow <- grid::arrow(length = unit(0.05, "inches"), type = "closed")

ggplot(wine_pca_loadings, aes(x = PC1, y = PC2)) +
  geom_segment(x = 0, y = 0,
               arrow = my_arrow,
                aes(xend = PC1, yend = PC2))  +
  geom_text(aes(label = quantity),
            size=3, color = "hotpink") +   ## size trial and error
  xlim(c(-1,1)) + ylim(c(-1,1)) ## fixed axis along limits of loadings
```



Variation explained by each PC

```
tibble(variation_explained) %>%
  mutate(PC = paste0("PC", 1:8))
## # A tibble: 8 x 2
##   variation_explained PC
##                 <dbl> <chr>
## 1              0.336  PC1
## 2              0.246  PC2
## 3              0.119  PC3
## 4              0.104  PC4
## 5              0.0856 PC5
## 6              0.0506 PC6
## 7              0.0436 PC7
## 8              0.0146 PC8
```

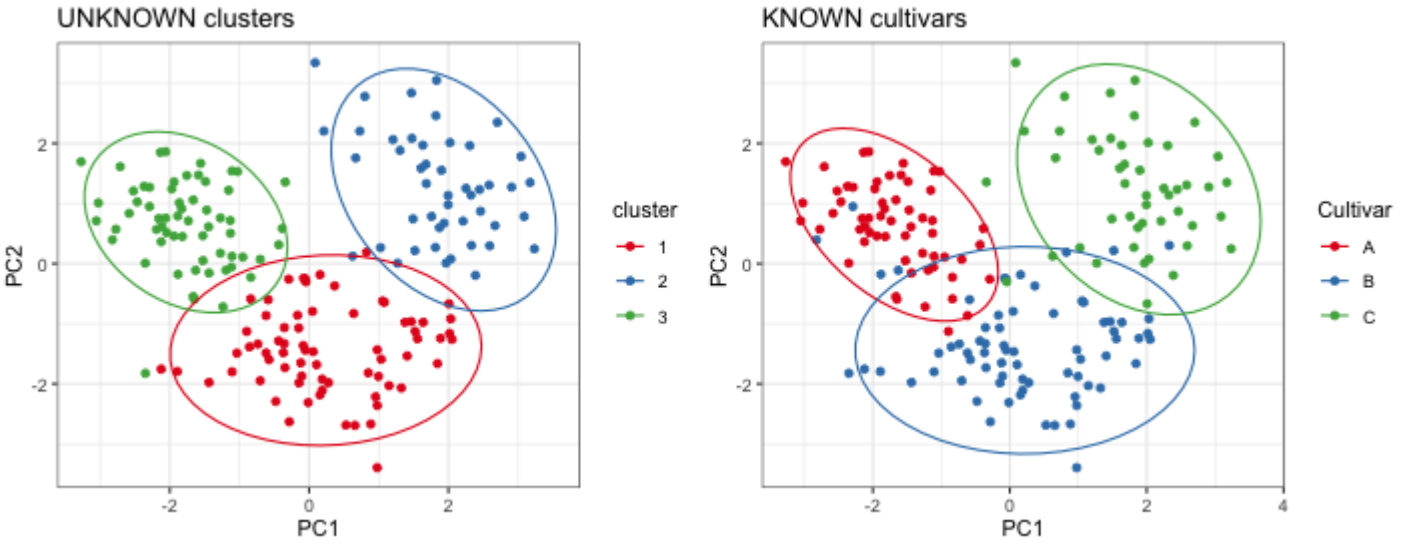
```
tibble(variation_explained) %>%
  mutate(PC = paste0("PC", 1:8)) %>%
  ggplot(aes(x = PC, y = variation_explained)) +
    geom_col(color = "indianred2", fill = "khaki") + ## ISOLATION BRAIN COLORS
    geom_text(aes(x = PC,
                  y = variation_explained + 0.01,  ## trial and error
                  label = round(variation_explained,2))) +
    xlab("Principle Components") + ylab("Percent of variation in data explained")
```



Not uncommon to cluster the PCs

```
set.seed(1011)

wine_pca_components %>% head(2)
## # A tibble: 2 x 8
##     PC1    PC2    PC3      PC4     PC5    PC6     PC7     PC8
##   <dbl>  <dbl>  <dbl>    <dbl>   <dbl>  <dbl>   <dbl>   <dbl>
## 1 -2.29  1.27  -0.205 -0.904   -0.272  -0.620 -0.558  -0.171
## 2 -1.24 -0.718 -0.722 -0.00793  0.0920 -0.137 -0.0389 -0.0863

k <- 3 ## ehhhh why not......
wine_pca_components %>%
  kmeans(k) -> wine_pca_kmeans


## Add the clusters into the PCA data
wine_pca_components %>%
  mutate(cluster = factor(wine_pca_kmeans$cluster),## factor()!!!!!!!!!
         Cultivar = wine$Cultivar) -> wine_pca_kmeans_data
```

```
wine_pca_kmeans_data %>%
  ggplot(aes(x = PC1, y = PC2, color = cluster)) +
    geom_point() + stat_ellipse() + scale_color_brewer(palette = "Set1") +
    ggtitle("UNKNOWN clusters") -> p1

wine_pca_kmeans_data %>%
  ggplot(aes(x = PC1, y = PC2, color = Cultivar)) +
    geom_point() + stat_ellipse() + scale_color_brewer(palette = "Set1") +
    ggtitle("KNOWN cultivars") -> p2

p1 +  p2
```



Takeaways from the last slide

- Imagine: What if we didn't know that there were three cultivars in this dataset?
  - By clustering the PCs, we can (mostly) identify the Cultivars