

Principal Component Analysis

McMaster University

Caitlin Simopoulos

2017-05-26

Principal Component Analysis:

- ▶ describes patterns in data
- ▶ is a way to reduce high dimensionality into fewer, linear components
- ▶ principal components = uncorrelated variables
- ▶ helpful when looking at data with a lot of features
- ▶ “goal is to explain the maximum amount of variance with the fewest number of principal components”
- ▶ commonly used in RNASeq QC, or for finding influential genes
- ▶ This is a great visual explanation of PCA.

If this still isn't making any sense...

Read [this](#) if you like math.

Read [this](#) if you hate math.

An example of PCA with gene expression data...

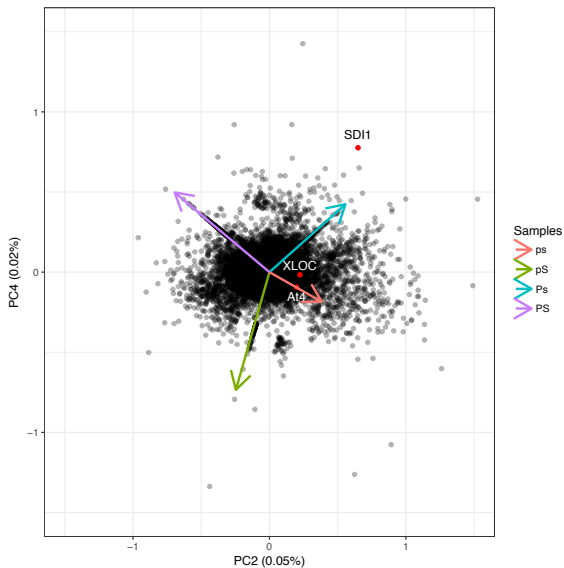


Figure 1:

Lots of features = difficult to interpret data

- ▶ visualizing data is one of the best ways to share and interpret data
- ▶ it's easy to plot and interpret 2D data...
- ▶ 3D is possible, but harder...
- ▶ 4D+ is very difficult, and will take a lot of time
- ▶ let's use PCA on 4D data

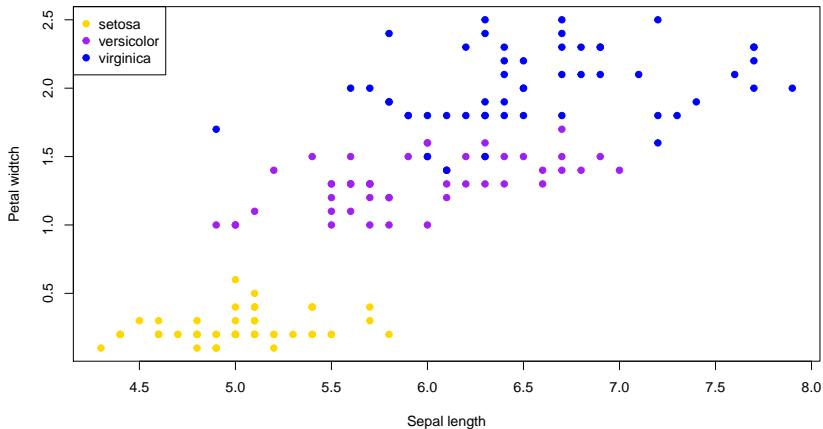
An example: iris data set

```
data("iris")  
table(iris$Species)
```

```
##  
##      setosa versicolor  virginica  
##          50          50          50
```

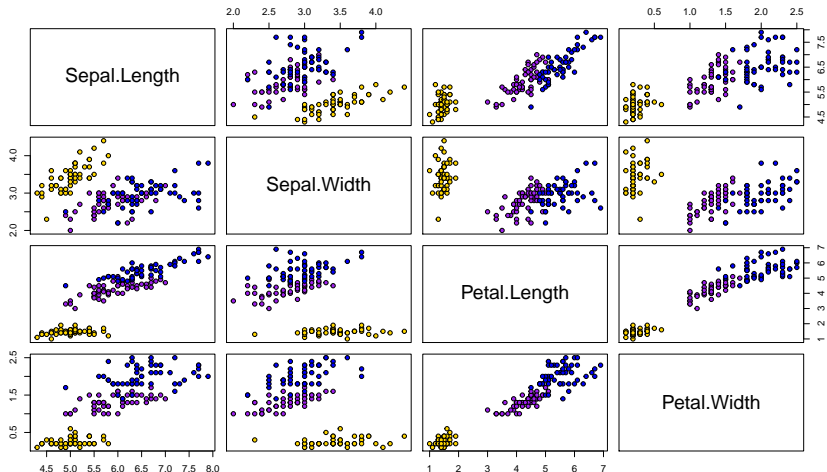
Plotting all features against each other would take a lot of time...

```
plot.colors <- c("gold", "purple", "blue")  
plot(iris$Sepal.Length, iris$Petal.Width, col=plot.colors[unclass(iris$Species)],  
     ylab = "Petal width", xlab= "Sepal length", pch=19)  
legend("topleft", pch=19, col=plot.colors, legend=unique(iris$Species))
```



Thankfully R has a built-in plotting feature for this

```
pairs(iris[,1:4], pch = 21,  
      bg = plot.colors[unclass(iris$Species)])
```



Imagine doing this with more than 4 features...

Like with an RNASeq project with data from 30,000 genes and 50 different samples...yuck

Let's do PCA instead

```
pca<- prcomp(iris[1:4], center=TRUE, scale=TRUE) # PCA with centering
pca$rotation # The loadings are here
```

```
##              PC1          PC2          PC3          PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width   -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

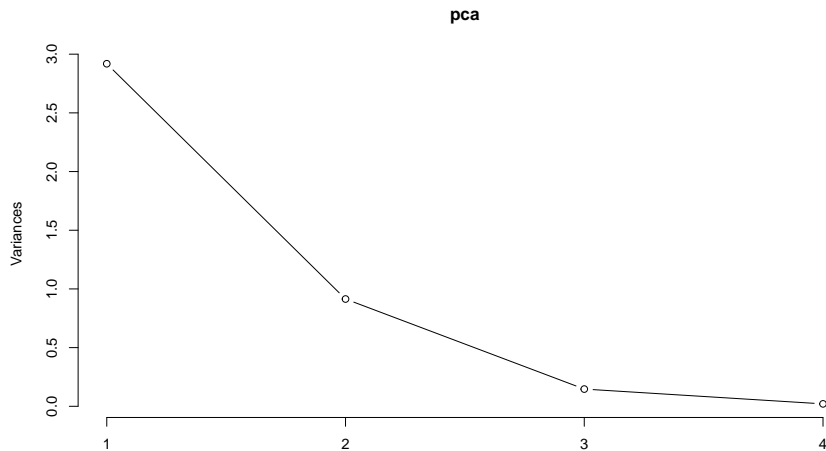
```
summary(pca)
```

```
## Importance of components:
```

```
##              PC1      PC2      PC3      PC4
## Standard deviation    1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion 0.7296 0.9581 0.99482 1.00000
```

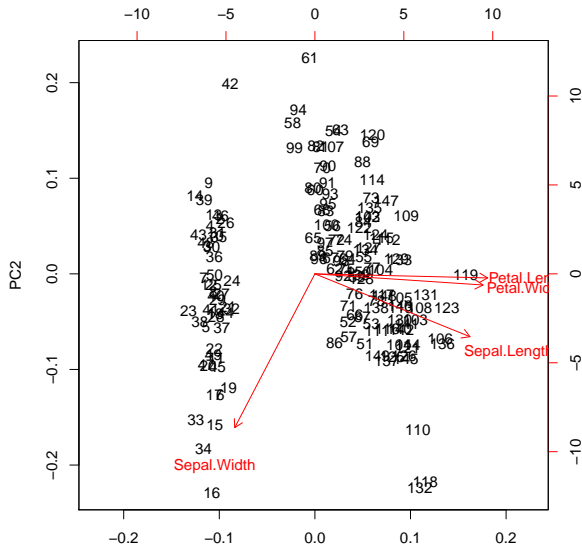
How much variance is described by each component?

```
plot(pca, type = "l")
```



Visualizing PC1 vs PC2

```
biplot(pca)
```



Hmm...

What do you think of this plot?

ggplot2 to the rescue!

- ▶ ggplot2 is a super flexible, super sleek plotting package for R
- ▶ used in combination with other packages of the “tidyverse”
- ▶ ggplot2 requires data in long format
 - ▶ 1 row per observation per feature

1. Make data.frame for PCA variables

```
#components  
indVals<-data.frame(pca$x)  
# variables  
varVals<-data.frame(pca$rotation)  
dim(indVals)
```

```
## [1] 150 4
```

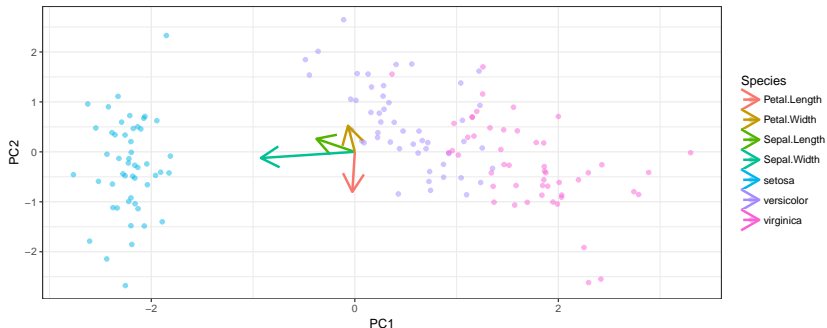
```
dim(varVals)
```

```
## [1] 4 4
```

```
## extrating all PCA data for ggplot  
coords<-data.frame(X=rep(0, 4), Y=rep(0, 4), varVals,  
                   feature = colnames(iris[1:4]))  
indVals <- cbind(indVals, Species= iris$Species)
```

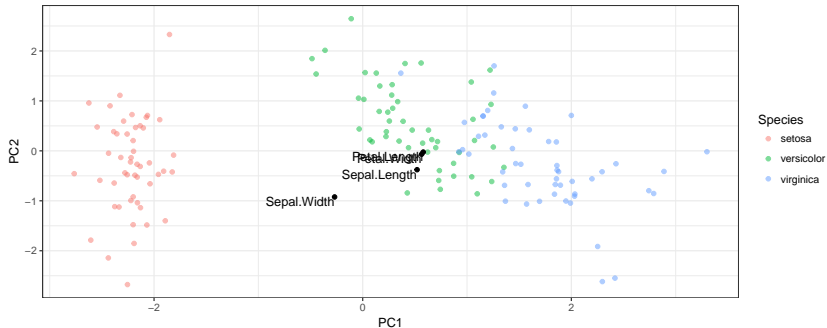
Plot PC1 vs PC2

```
library(ggplot2)
pc12plot <- ggplot(data = indVals, aes(x=PC1, y=PC2)) +
  geom_point(aes(color = Species), alpha=0.5) +
  geom_segment(data=coords, aes(x=X, y=Y, xend=PC2, yend=PC4,
    colour=colnames(iris[1:4])), arrow=arrow(), size=1) +
  theme_bw()
print(pc12plot)
```



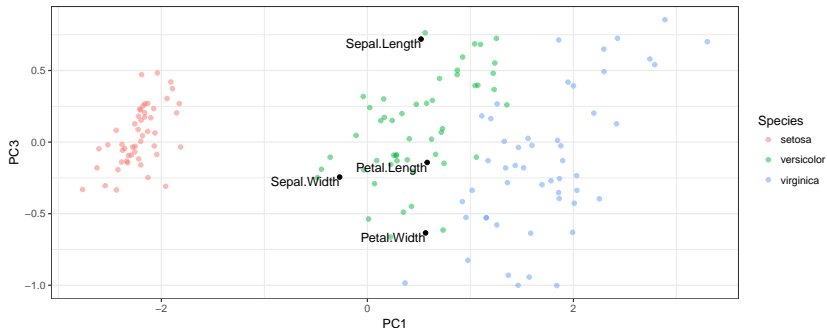
Plot PC1 vs PC2

```
pc12plot <- ggplot(data = indVals, aes(x=PC1, y=PC2)) +  
  geom_point(aes(color = Species), alpha=0.5) +  
  geom_point(data=coords, aes(x=PC1, y=PC2)) +  
  geom_text(data = coords, aes(x=PC1, y=PC2,  
    label=feature),hjust=1, vjust=1) +  
  theme_bw()  
print(pc12plot)
```



Plot PC1 vs PC3

```
pc12plot <- ggplot(data = indVals, aes(x=PC1, y=PC3)) +  
  geom_point(aes(color = Species), alpha=0.5) +  
  geom_point(data=coords, aes(x=PC1, y=PC3)) +  
  geom_text(data = coords, aes(x=PC1, y=PC3,  
    label=feature),hjust=1, vjust=1) +  
  theme_bw()  
print(pc12plot)
```



Plot PC2 vs PC4

```
pc24plot <- ggplot(data = indVals, aes(x=PC2, y=PC4)) +  
  geom_point(aes(color = Species), alpha=0.5) +  
  geom_point(data=coords, aes(x=PC2, y=PC4)) +  
  geom_text(data = coords, aes(x=PC2, y=PC4,  
    label=feature),hjust=1, vjust=1) +  
  theme_bw()  
print(pc24plot)
```

