Principal Component Analysis

Christopher J. Fiscus

updated 1/22/2018

What is Principal Component Analysis (PCA)?

PCA a clustering method to reduce dimensionality in datasets and to explore patterns in data. The details of how this works can be found elsewhere.

Let's use the iris dataset

This dataset includes sepal length, sepal width, petal length, and petal width for 150 iris plants from three different species.

```
## read in data
library(datasets)
df = iris

## Format of data (Name, lat, long)
head(df)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Transform data

When doing PCA it is a good idea to transform the data so it is normalized. It's common to use a log transformation to do this.

```
## log transformation on numerical data
normalized<-log(iris[,1:4])
species<-iris[,5]
```

Do PCA

```
## pca using single value decomposition, centered and scaled
df.pca<-prcomp(normalized, scale.=TRUE, center=TRUE)
```

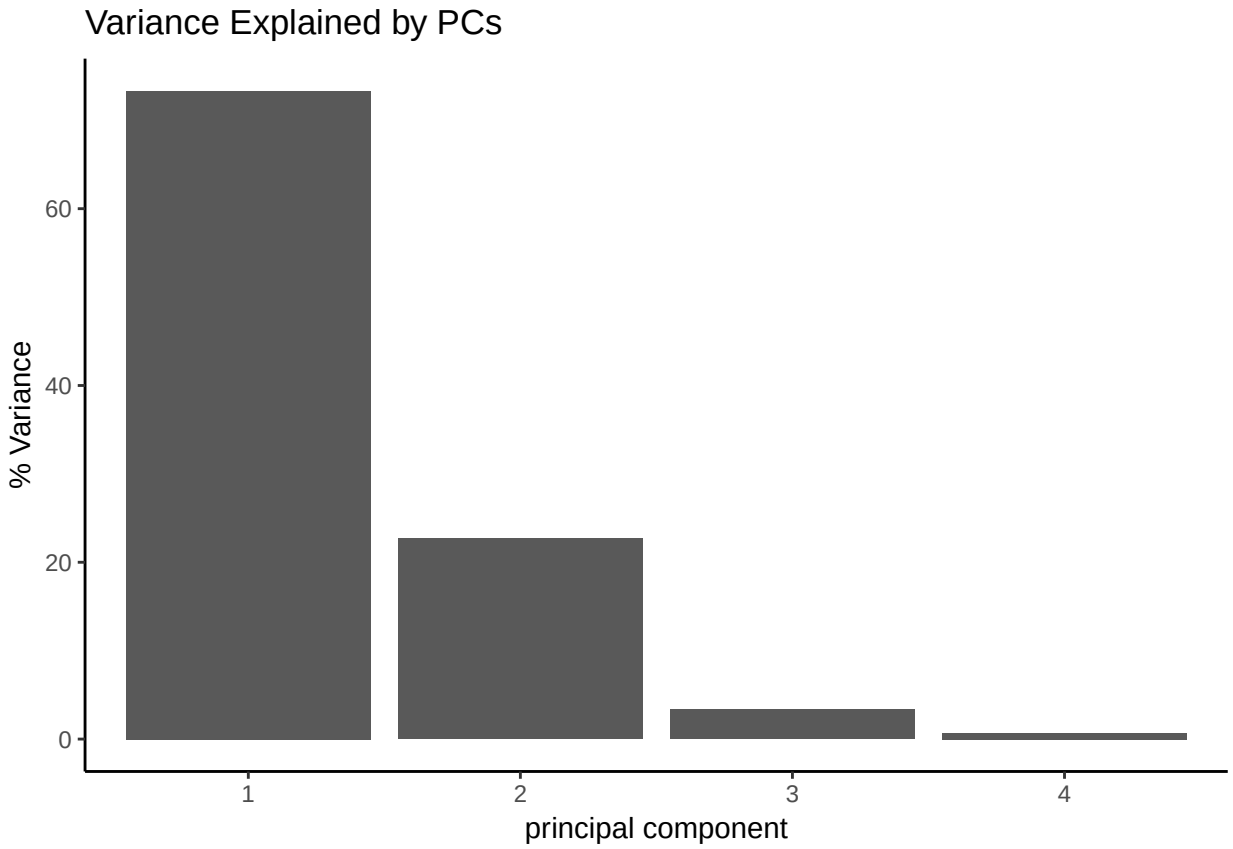
Plot the Percent Variance Explained by each PCA

1. Format the data

```
## calculate percent variance
per.var<- (df.pca$sdev^2/sum(df.pca$sdev^2))*100
per.var<-as.data.frame(per.var)

## format data nicely
per.var<-cbind(1:nrow(per.var), per.var)
colnames(per.var)<-c("PC", "PercentVariance")
```

2. plot with ggplot2

```
library(ggplot2)
g<-ggplot(data=per.var, aes(x=factor(PC), y=PercentVariance, axis.ticks=1)) + geom_bar(stat="identity")
print(g)
```




```
# write graph out
# ggsave("varexplained.png", plot=g)
```

Plot PCs 1-3

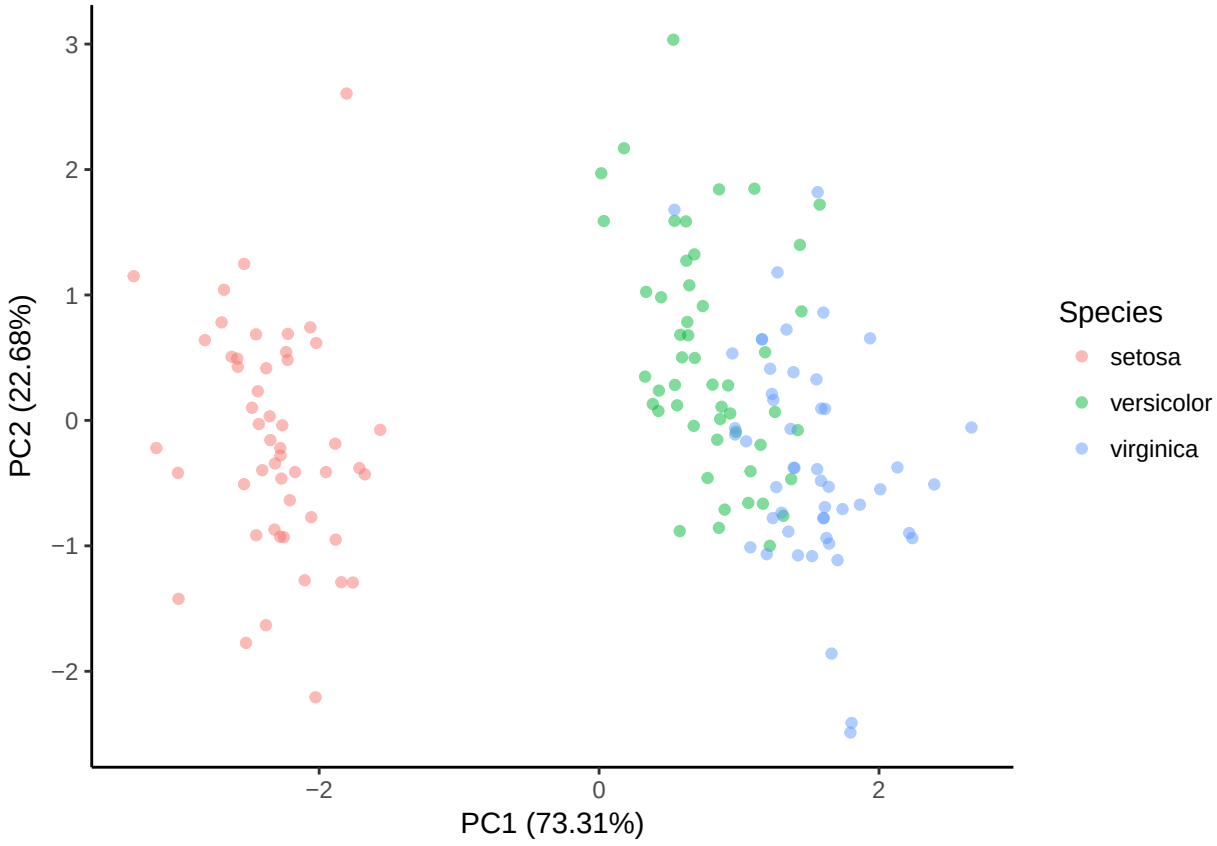
1. Plot PC 1 vs. PC 2 Note: The github version of ggfortify will add the % variance explained to the axis. The CRAN version does not have this yet.

```
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
## install github version of ggfortify
#library(devtools)
```

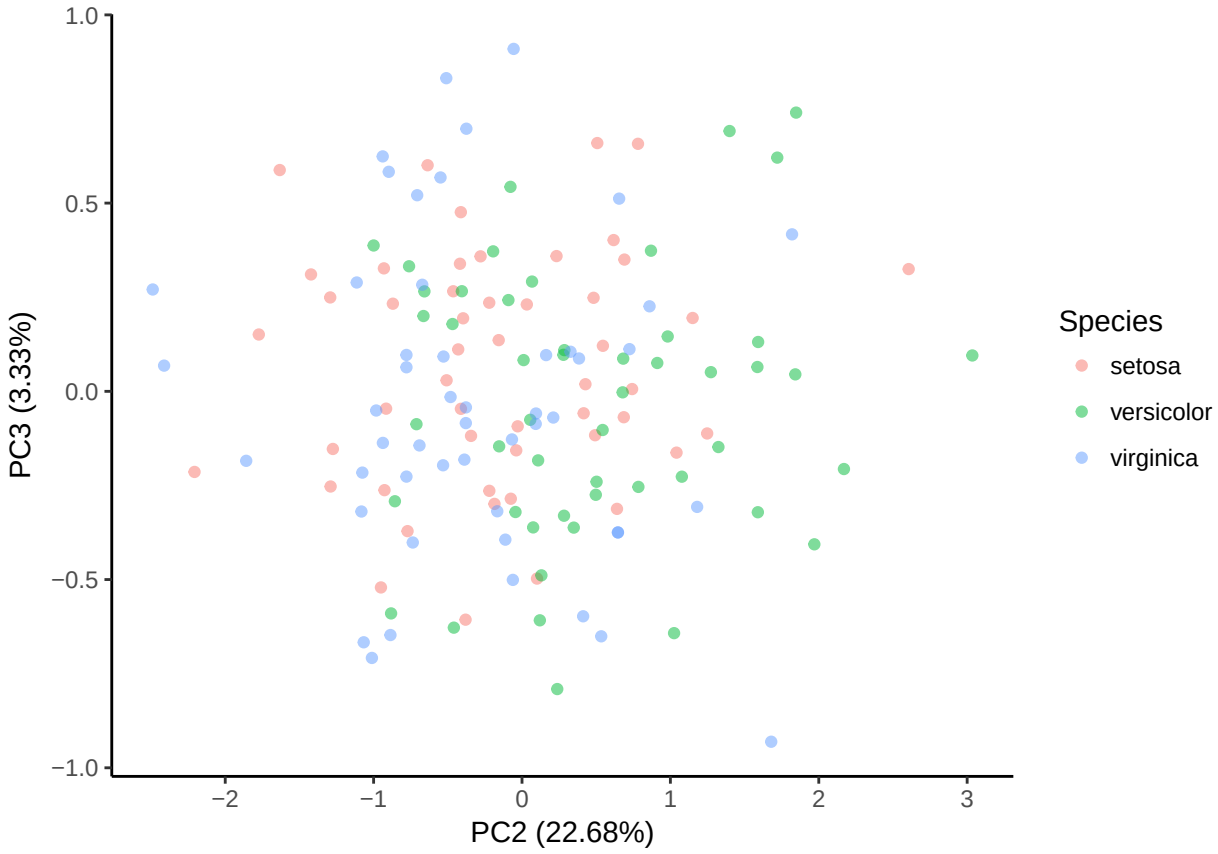
```
#install_github('sinhrks/ggfortify')

library(ggfortify) # load lib

## Plot PC1 vs. PC2
g<-autoplot(df.pca, data=df, colour='Species', alpha=0.5, variance_percentage=TRUE, loadings=FALSE, scal
print(g)
```



2.

Plot PC2 vs. PC 3

```
## Plot PC2 vs. PC3
g<-autoplot(df.pca, data=df, colour='Species', alpha=0.5, variance_percentage=TRUE, loadings=FALSE, x=2
print(g)
```



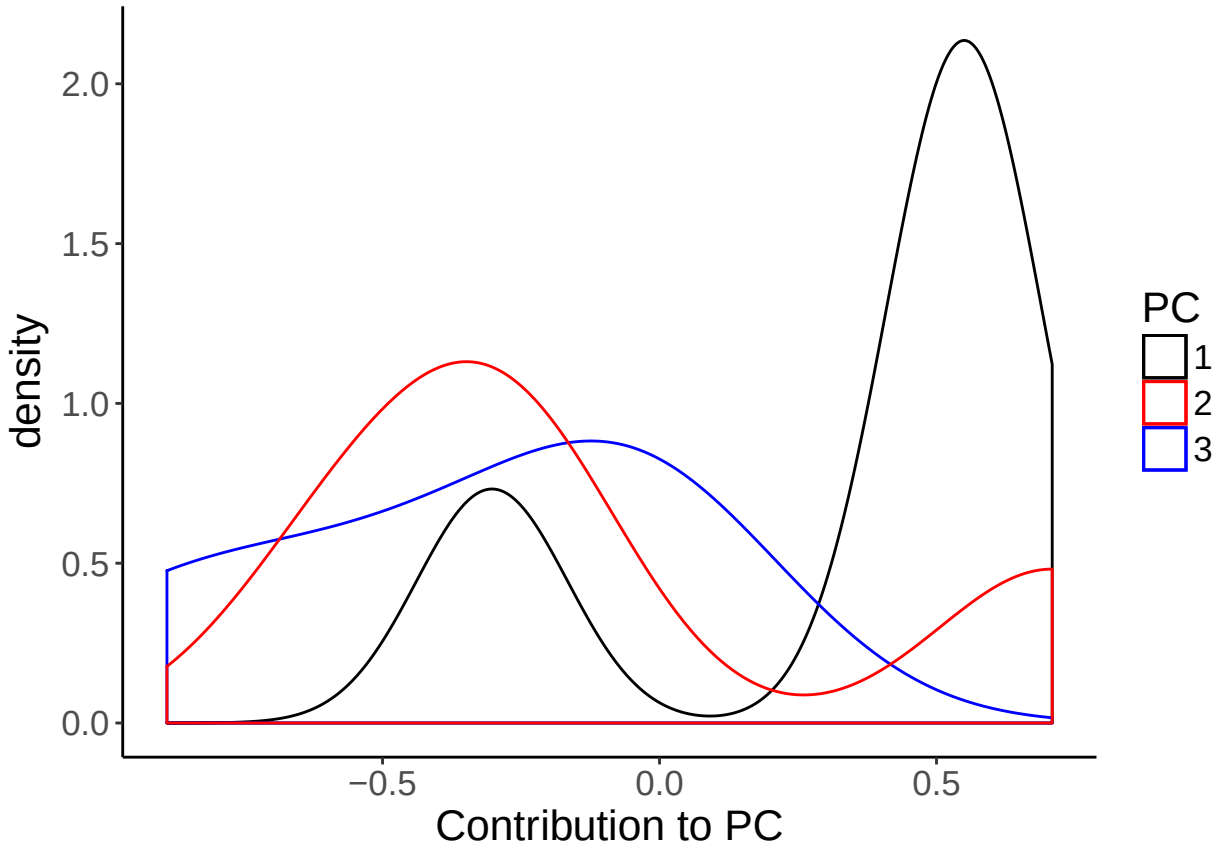
Explore loadings

```
## Loadings for PCs 1-3
Loadings<-as.data.frame(df.pca$rotation[,1:3])
head(Loadings)
```

```
##                     PC1         PC2        PC3
## Sepal.Length  0.5038236 -0.45499872  0.7088547
## Sepal.Width  -0.3023682 -0.88914419 -0.3311628
## Petal.Length  0.5767881 -0.03378802 -0.2192793
## Petal.Width   0.5674952 -0.03545628 -0.5829003
```

Plot Densities of PCs 1 - 3

```
g<-ggplot(Loadings) + geom_density(aes(PC1, color="black")) + theme_classic() + geom_density(aes(PC2, co
print(g)
```



Explore PCA

```
res.ind <- get_pca_ind(df.pca)

## coordinates
coord<-as.data.frame(res.ind$coord)
head(coord)
```

```
##       Dim.1      Dim.2       Dim.3        Dim.4
## 1 -2.406639 -0.3969554  0.19396467  0.004779476
## 2 -2.223539  0.6901804  0.35000151  0.048868378
## 3 -2.581105  0.4275418  0.01889761  0.049909545
## 4 -2.450869  0.6860074 -0.06874595 -0.149646465
## 5 -2.536853 -0.5082516  0.02932259 -0.040048202
## 6 -1.841495 -1.2899381 -0.25276831  0.163890597
```

```
## contributions
contri<-as.data.frame(res.ind$contrib)
head(contri)
```

```
##       Dim.1     Dim.2       Dim.3        Dim.4
## 1 1.3167113 0.1158169 0.188571250 0.0005547636
## 2 1.1239788 0.3501174 0.614002362 0.0579966981
## 3 1.5145376 0.1343525 0.001789963 0.0604943249
## 4 1.3655536 0.3458965 0.023687787 0.5438515179
## 5 1.4630507 0.1898655 0.004309579 0.0389505393
```

```
## 6 0.7709209 1.2229994 0.320239914 0.6523121915
```

```
## quality
qual<-as.data.frame(res.ind$cos2)
head(qual)
```

```
##       Dim.1      Dim.2        Dim.3        Dim.4
## 1 0.9673936 0.02631872 6.283862e-03 3.815416e-06
## 2 0.8915782 0.08590042 2.209072e-02 4.306514e-04
## 3 0.9728903 0.02669376 5.215146e-05 3.637640e-04
## 4 0.9234795 0.07235105 7.265767e-04 3.442866e-03
## 5 0.9610562 0.03857586 1.283994e-04 2.395103e-04
## 6 0.6590047 0.32335917 1.241631e-02 5.219820e-03
```