# Dimension Reduction

Alex Faassen

2025-03-18

**To Do:** - Determine effective dimension reduction methods, likely PCA and/or variants. - Sparse PCA? - Perform on images and analyze results - Attach output to stats dataset and store

## Setup

```
## Load Data
load("Data/pokemon.RData")

## Libraries
library(ggplot2)
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 4.4.2
```

```
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.4.2
```

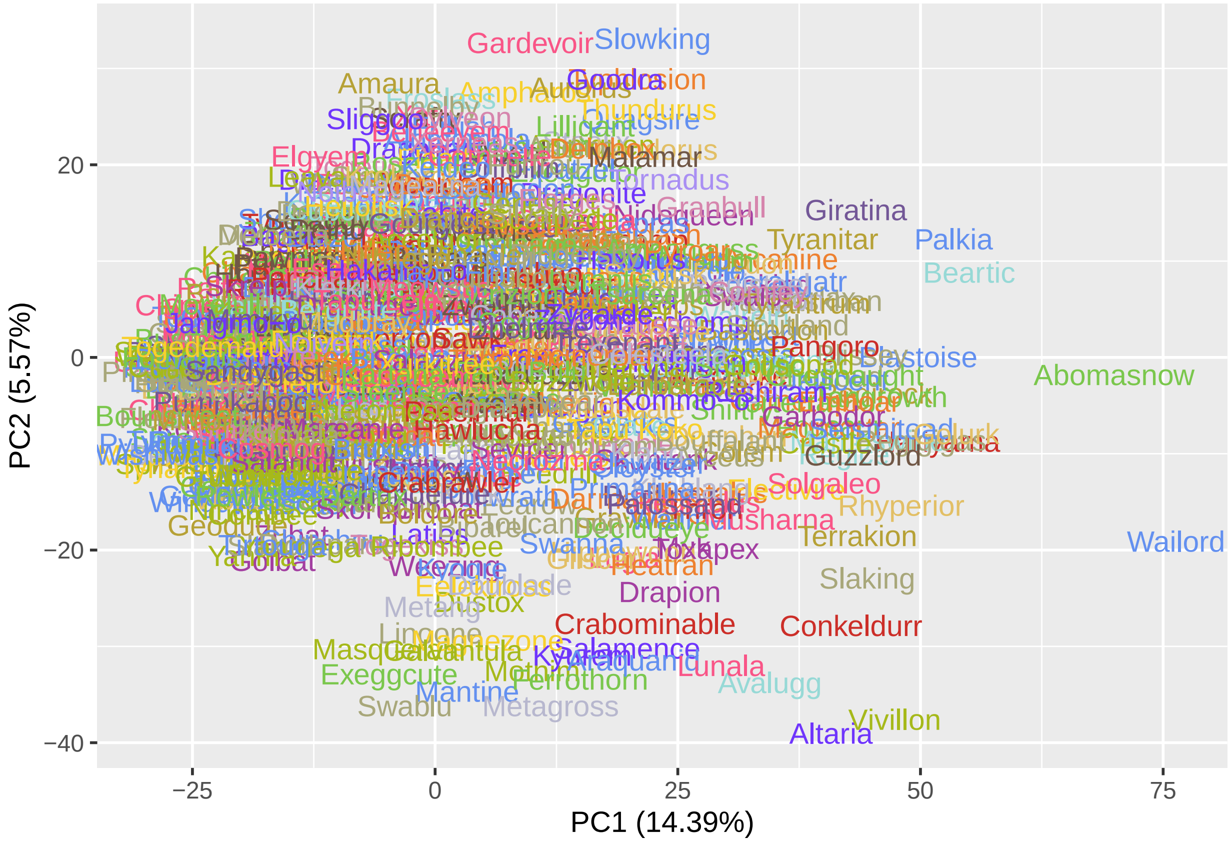
```
## Helper functions
barplot = function(values){
  n = length(values)
  df = data.frame(value = values,  index = 1:n)
  ggplot(df, aes(index, value, fill = value)) +
    geom_bar(color = "black", stat = "identity")+
    scale_fill_gradient2(low="#619CFF", mid="white", high="#F8766D")+
    theme_bw()
}

# heatmap = function(A){
#   n = nrow(A)
#   p = ncol(A)
#   df = data.frame(value = c(A),  i = 1:n, j = rep(1:p, rep(n, p)))
#   ggplot(df, aes(j, i, fill = value)) +
#     geom_tile(color = "black")+
#     scale_fill_gradient2(low="#619CFF", mid="white", high="#F8766D")+
#     scale_y_reverse()+
#     theme_void()
# }

getImg = function(flat_img){
  # matrix(unlist(gsvalues), 120, 120, 4, byrow = T)
  # rasterGrob(array(flat_img, dim = c(120, 120, 4)))
  array(as.numeric(flat_img), dim = c(120, 120, 4))
```

```r
}

plotImg <- function(img_raster) {
  # Plot using ggplot2
  ggplot() +
    annotation_raster(img_raster, xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = Inf) +
    theme_void()  # Remove axes for clean visualization
}
```

# PCA

```r
## PCA
# Defaults: center = TRUE, scale. = FALSE
PCA = prcomp(images[,-1], center = TRUE, scale = FALSE) #scale = TRUE)

sum_PCA = summary(PCA)
```

```r
## Biplot
# Colours
stats$type1 = factor(stats$type1)
# unique(stats$type1)
# type_colours = c("#A6B91A", "#705746", "#6F35FC", "#")
type_colours = c("#7AC74C", "#EE8130", "#6390F0", "#A6B91A", "#A8A77A", "#A33EA1",
             "#F7D02C", "#E2BF65", "#D685AD", "#CC2E28", "#F95587", "#B6A136", "#735797",
             "#96D9D6", "#6F35FC", "#705746", "#B7B7CE", "#A98FF3")
names(type_colours) = unique(stats$type1)

# library(ggfortify)
autoplot(PCA, data = cbind(stats$name, stats$type1, images[,-1]), shape = FALSE, color = "stats$type1",
  scale_colour_manual(values = type_colours) +
  theme(legend.position = "none")
```



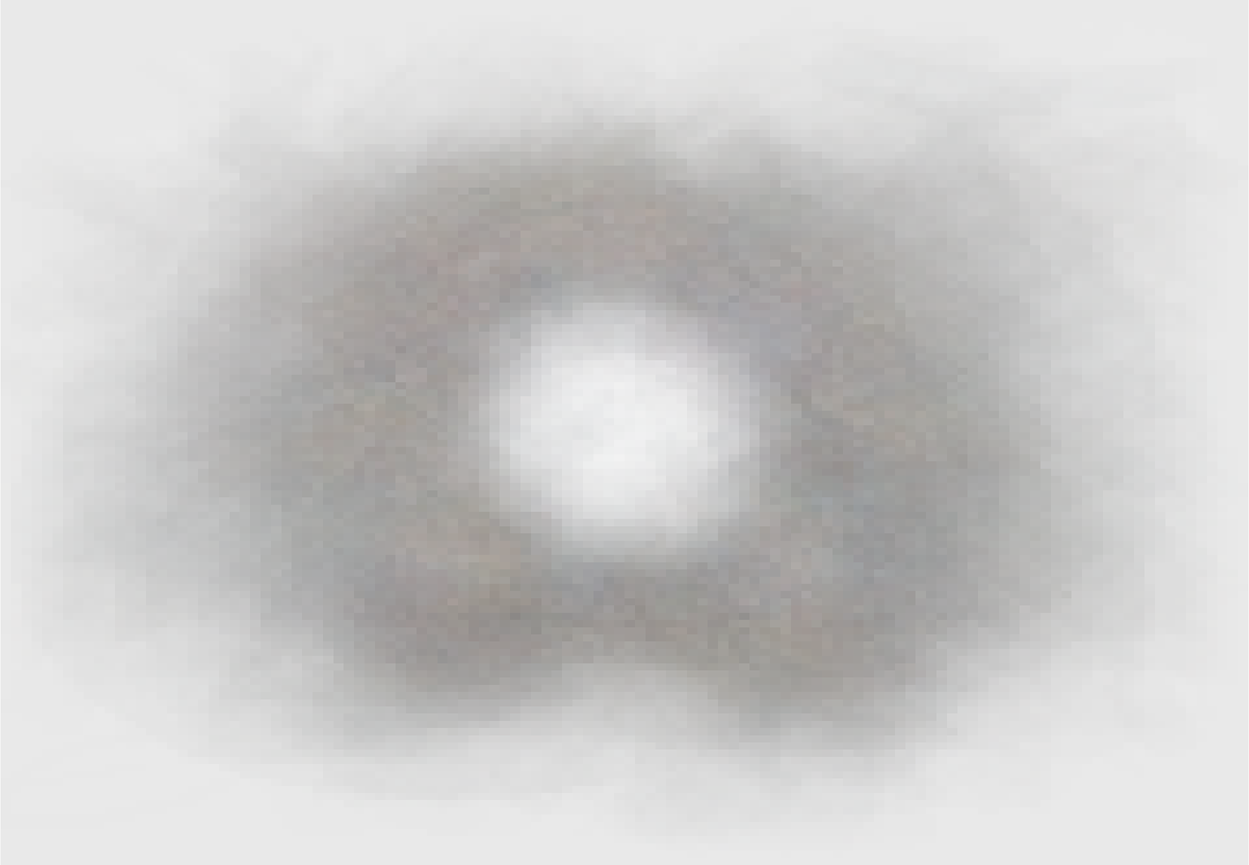
## Loading Vectors

```
## Smallest PC1 Pokemon
# Extract PCA-transformed data
pca_data <- as.data.frame(PCA$x)  # Convert PCA results to a data frame

# Add Pokémon names and types for reference
pca_data$name <- stats$name  # Ensure stats$name contains Pokémon names
pca_data$type1 <- stats$type1  # Primary type for reference

# Sort by PC1 in ascending order and select the two smallest
smallest_pc1_pokemon <- pca_data[order(pca_data$PC1), ][1:2, ]

# Print result
print(rownames(smallest_pc1_pokemon))
```

```
## [1] "images/pikipek.png" "images/unown.png"
```

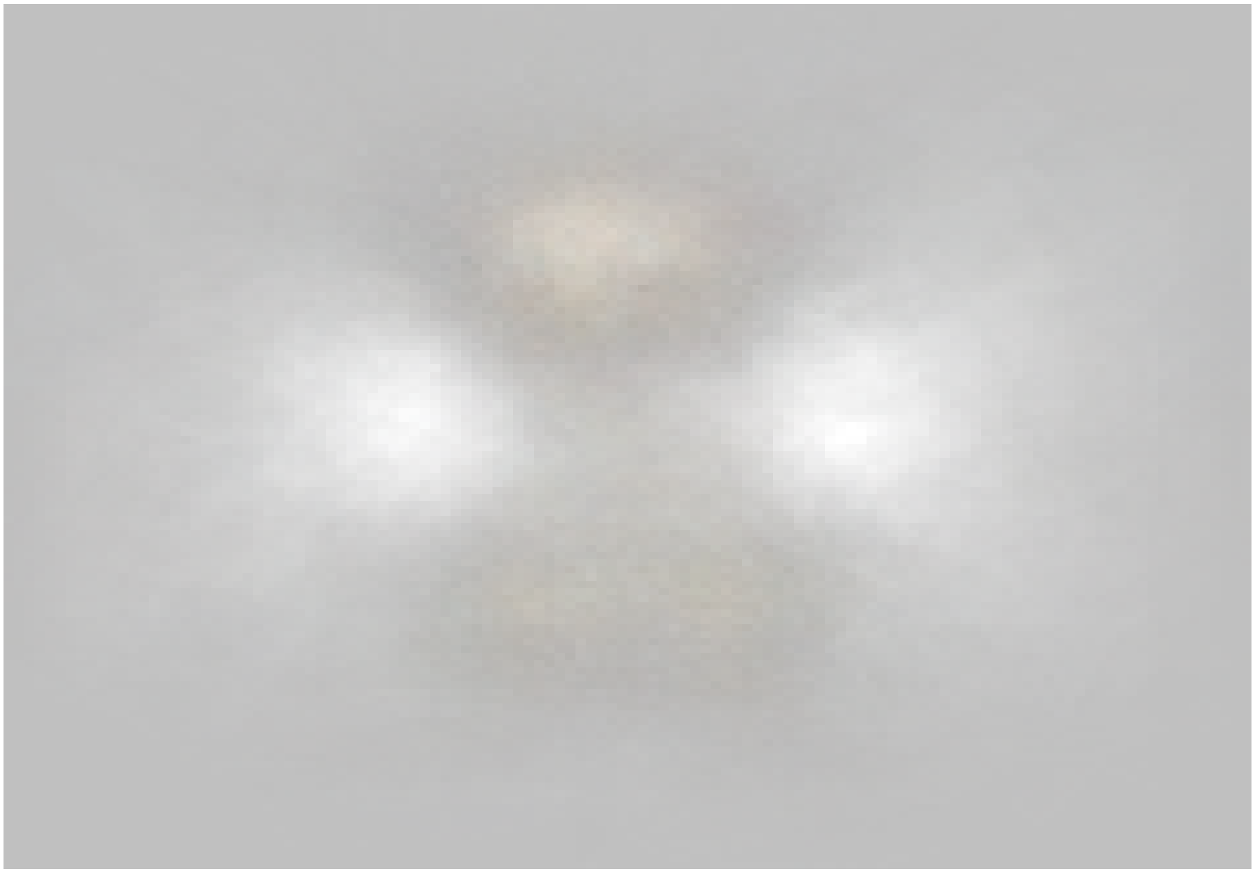
```
## PC1
# Extremes
pos1 = plotImg(getImg(images[stats$name == "Wailord", -1]))
pos2 = plotImg(getImg(images[stats$name == "Abomasnow", -1]))
neg1 = plotImg(getImg(images[stats$name == "Elgyem", -1])) # Pikipek
neg2 = plotImg(getImg(images[stats$name == "Geodude", -1])) # Unown
# exPlots = sapply(c(pos1, pos2, neg1, neg2), function(rgba) plotImg(getImg(rgba)))
(pos1 + pos2) / (neg1 + neg2)
```

```r
# Loading Vector
pca = PCA$rotation[,1]
pca_norm = (pca - min(pca)) / (max(pca) - min(pca))
plotImg(getImg(pca_norm))
```



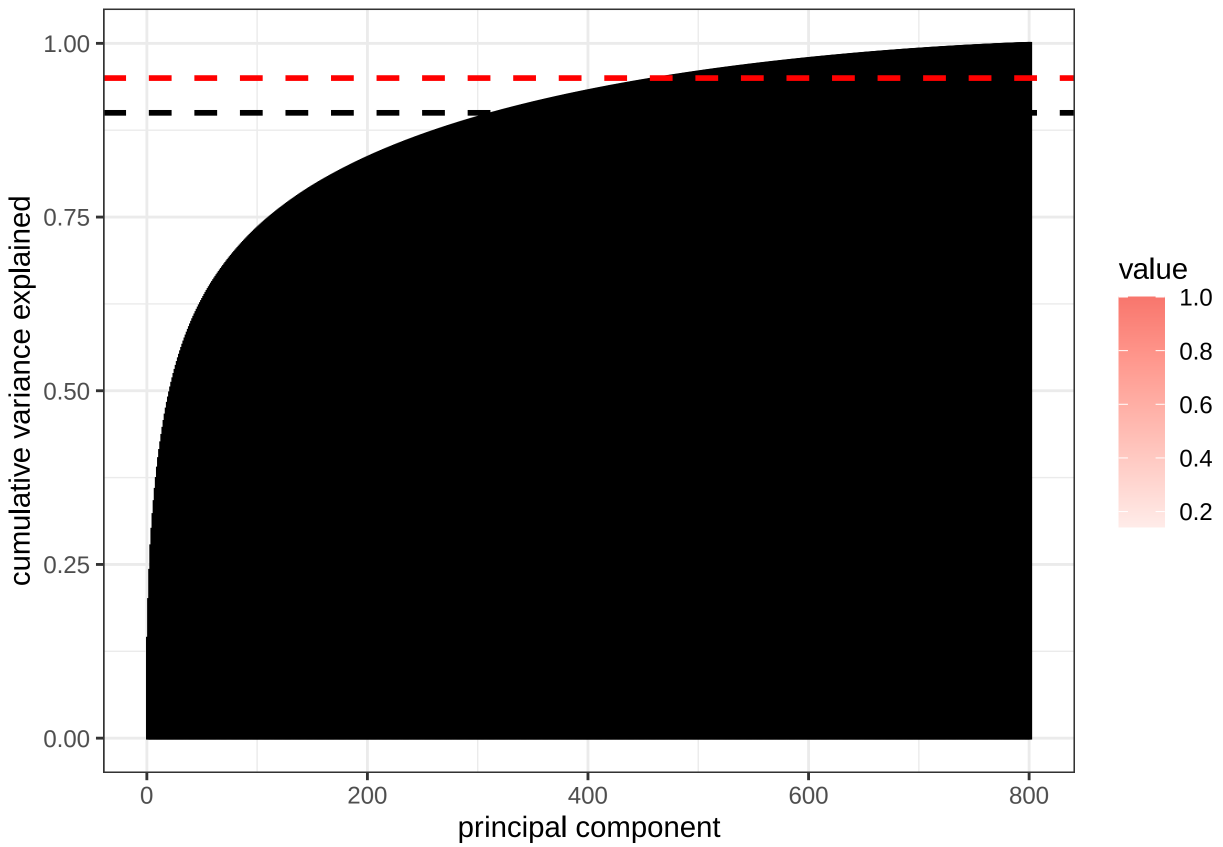
```
## PC1
# Extremes
pos1 = plotImg(getImg(images[stats$name == "Slowking", -1]))
pos2 = plotImg(getImg(images[stats$name == "Gardevoir", -1]))
neg1 = plotImg(getImg(images[stats$name == "Altaria", -1]))
neg2 = plotImg(getImg(images[stats$name == "Swablu", -1]))
# exPlots = sapply(c(pos1, pos2, neg1, neg2), function(rgba) plotImg(getImg(rgba)))
(pos1 + pos2) / (neg1 + neg2)
```

```
# Loading Vector
pca = PCA$rotation[,2]
pca_norm = (pca - min(pca)) / (max(pca) - min(pca))
plotImg(getImg(pca_norm))
```



## Variance Explained

```
## Variance Explained
ve = summary(PCA)$importance[3,]

# Plot
barplot(ve)+
  xlab("principal component")+
  ylab("cumulative variance explained")+
  ylim(0, 1)+
  geom_hline(aes(yintercept = 0.9), linewidth = 1, linetype = "dashed") +
  geom_hline(aes(yintercept = 0.95), linewidth = 1, linetype = "dashed", color = "red")
```



```r
## Retain PCs
var_retain = 0.95 # 95% of VE
dr_images = data.frame(
  image_path = images$image_path,
  PCA$x[,which(ve <= var_retain)]
)
```

## Local Dimension Reduction

```r
#
```

## Save Dataset

```r
## Save Dataset
save(stats, dr_images, file = "Data/dr_pokemon.RData")
```