

# Principal Components Analysis

AU STAT627

Emil Hvitfeldt

2021-03-16

# More Unsupervised Learning

Another branch of unsupervised learning

- Dimensionality Reduction

- take many dimensions and create fewer ones that represent as much of the original data as possible

# Dimensionality Reduction

Why would you want to do this?

- Allows for visualization of many dimensions in 2 dimensions
- Can be used as a preprocessing step for models that can't handle many dimensions

# What is Principal Components Analysis?

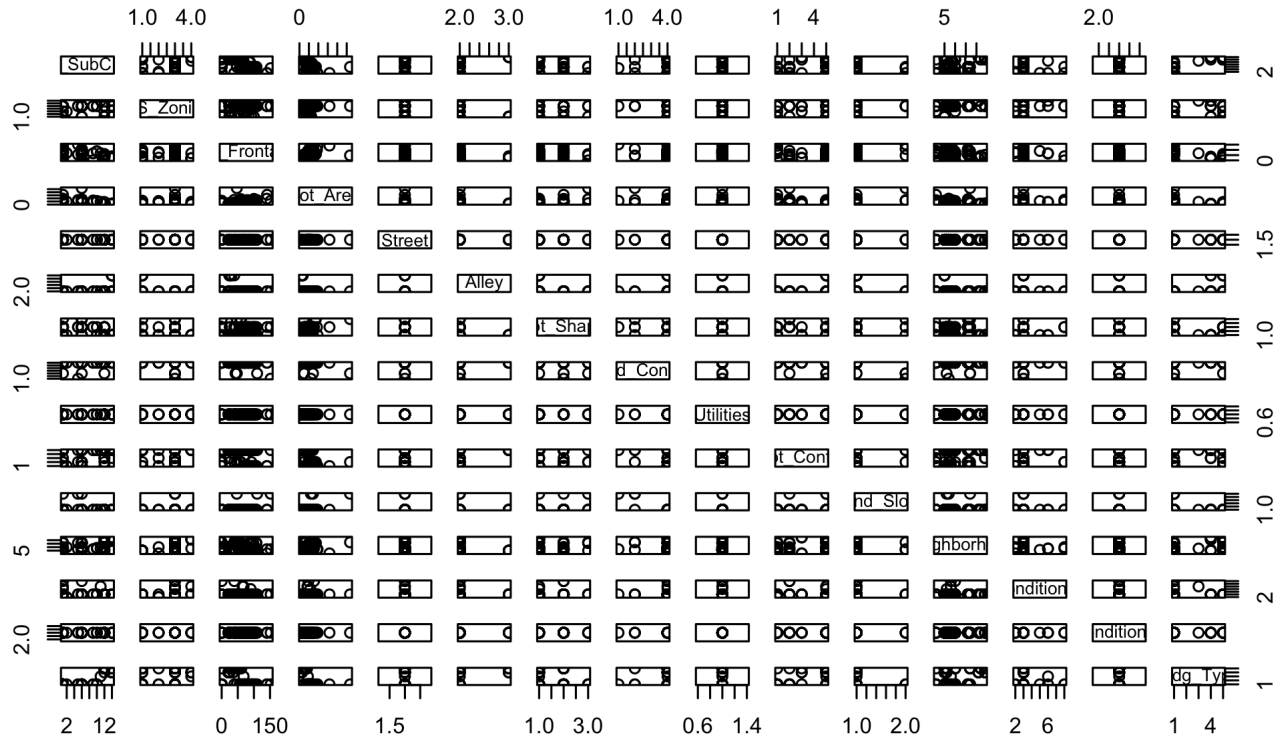
- Motivation

Suppose you have many dimension and want to visualize the relationship between them

If you wanted to do them pairwise then you have  $\binom{p}{2} = \frac{p(p-1)}{2}$  plots to do

This adds up very fast!

# Many plots



# What is Principal Components Analysis?

we want to find a low dimensional representation of the high dimensional data

Specifically, we would want 2 dimensional for plotting purposes

PCA is one such technique that does just that

PCA finds a low-dimensional representation of the data set that contains as much of the variation as possible in as few columns as possible

# Re-formulation

PCA is a linear combination of the original data such that most of the variation is captured in the first variable, then second, then third and so on

# PCA Construction

The **first principal component** of a set of features  $X_1, X_2, \dots, X_p$  is the normalized linear combination of the features

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

that has the largest variance

We mean that normalized that  $\sum_{j=1}^p \phi_{j1}^2 = 1$ .

we refer to  $\phi_{11}, \dots, \phi_{p1}$  as the loadings of the first principal component.

And think of them as the loading vector  $\phi_1$



# PCA Construction

These loadings are constrained, otherwise, we don't get any solutions since arbitrarily large values of the loadings would increase the variance

# How do we get these?

Assuming we have a  $n \times p$  data set  $\mathbf{X}$

since we are only interested in the variance we assume that the variables have been centered

$$\text{maximize}_{\phi_{11}, \dots, \phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^n \left( \sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p \phi_{j1}^2 = 1$$

# How do we get these?

since we have  $z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \dots + \phi_{p1}x_{ip}$ , then we can write

$$\underset{\phi_{11}, \dots, \phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^n z_{i1}^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p \phi_{j1}^2 = 1$$

We are in essence maximizing the sample variance of the  $n$  values of  $z_{i1}$ .

We refer to  $z_{11}, \dots, z_{n1}$  as the scores of the first principal component.

# How do we solve that problem?

Luckily this can be solved using techniques from Linear Algebra

more specifically, it can be solved using a **eigen decomposition**

One of the main strengths of PCA is that you don't need to use optimization to get the results without approximations!!!!

# Remaining principal components

Once the first principal component is calculated, we can calculate the second principal component

We find the second principal component  $Z_2$  as a linear combination of  $X_1, \dots, X_p$  that has the maximal variance out of the linear combinations that are uncorrelated with  $Z_1$

this is the same as saying that  $\phi_2$  should be orthogonal to the direction  $\phi_1$

# Remaining principal components

We can do this to calculate all the principal components

since we are working iteratively through the principal components, we can calculate only as many as we want

# The proportion of variance explained

the proportion of variance explained of the  $m$ th principal component is given by

$$\frac{\sum_{i=1}^n \left( \sum_{j=1}^p \phi_{jm} x_{ij} \right)^2}{\sum_{j=1}^p \sum_{i=1}^n x_{ij}^2}$$

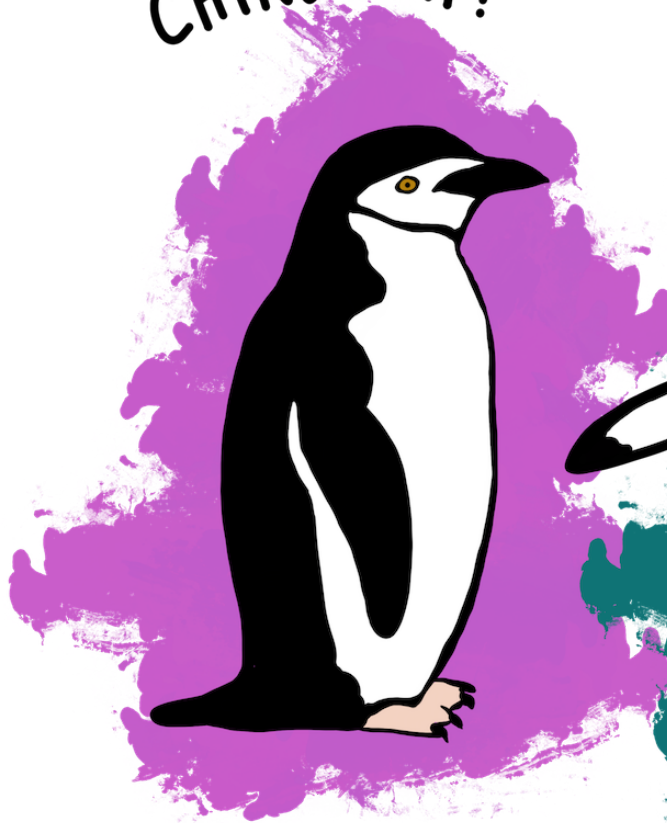
Don't worry, this is already calculated by the software you use to get PCA

# Visualizing PCA

Once we have the principal components there are a couple of things we can visualize



CHINSTRAP!



GENTOO!



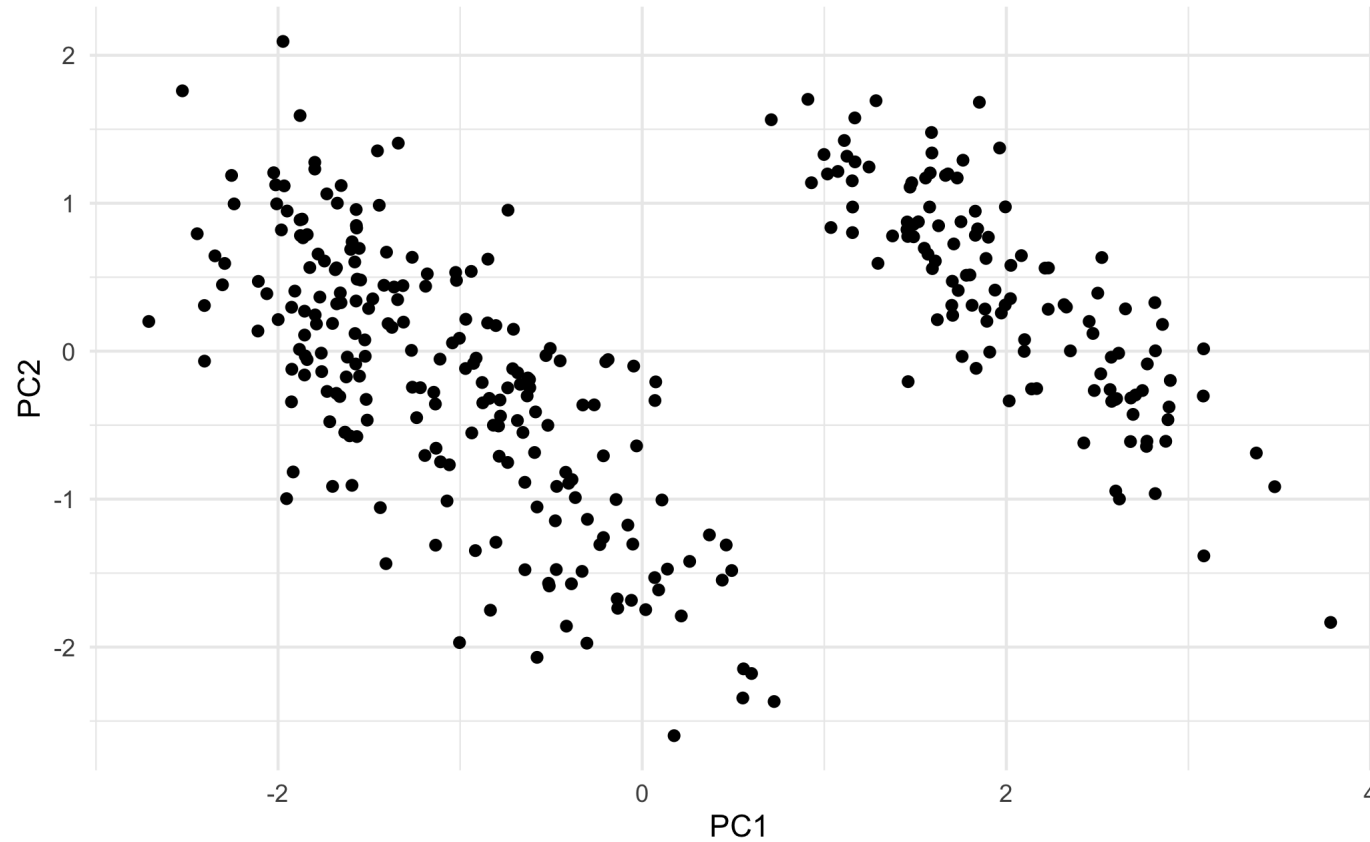
ADÉLIE!



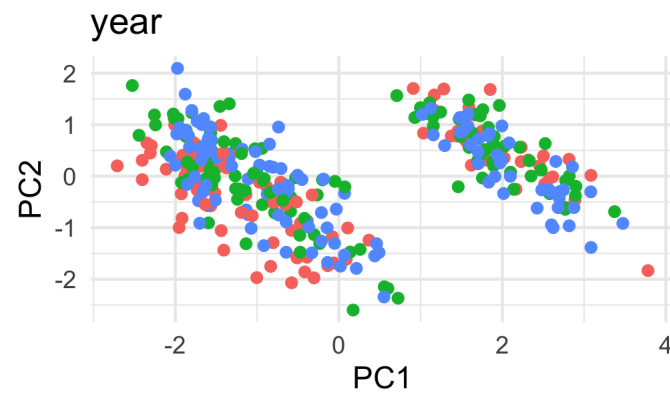
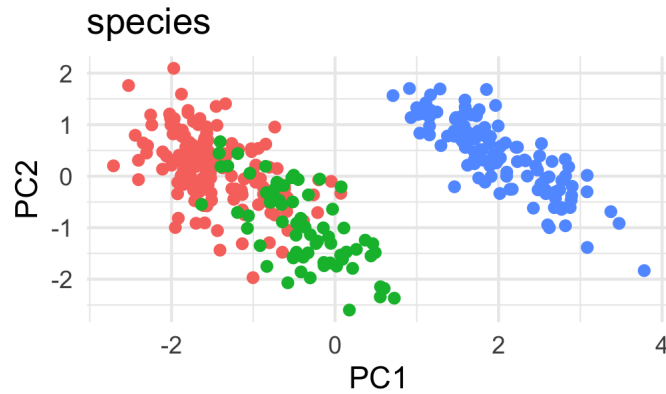
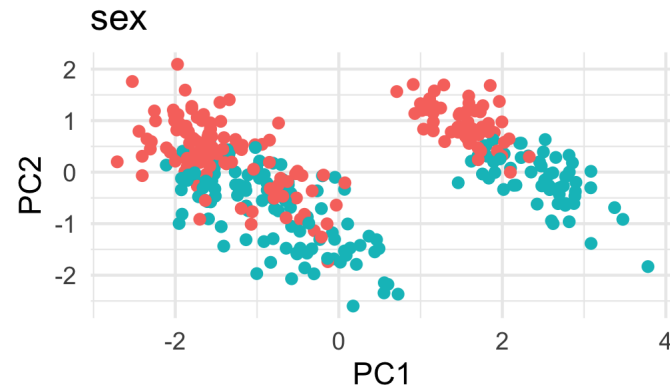
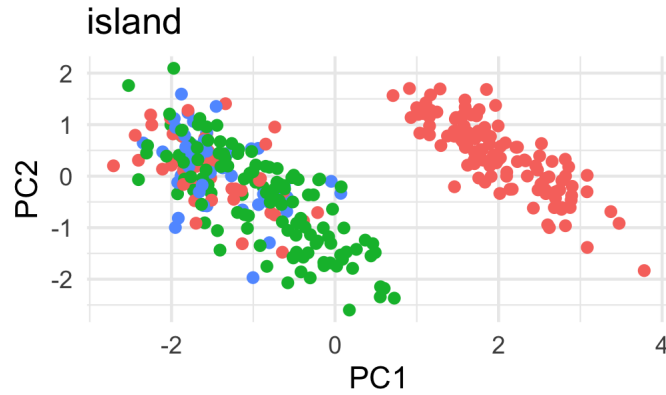
Art by Allison Horst

© allison\_horst

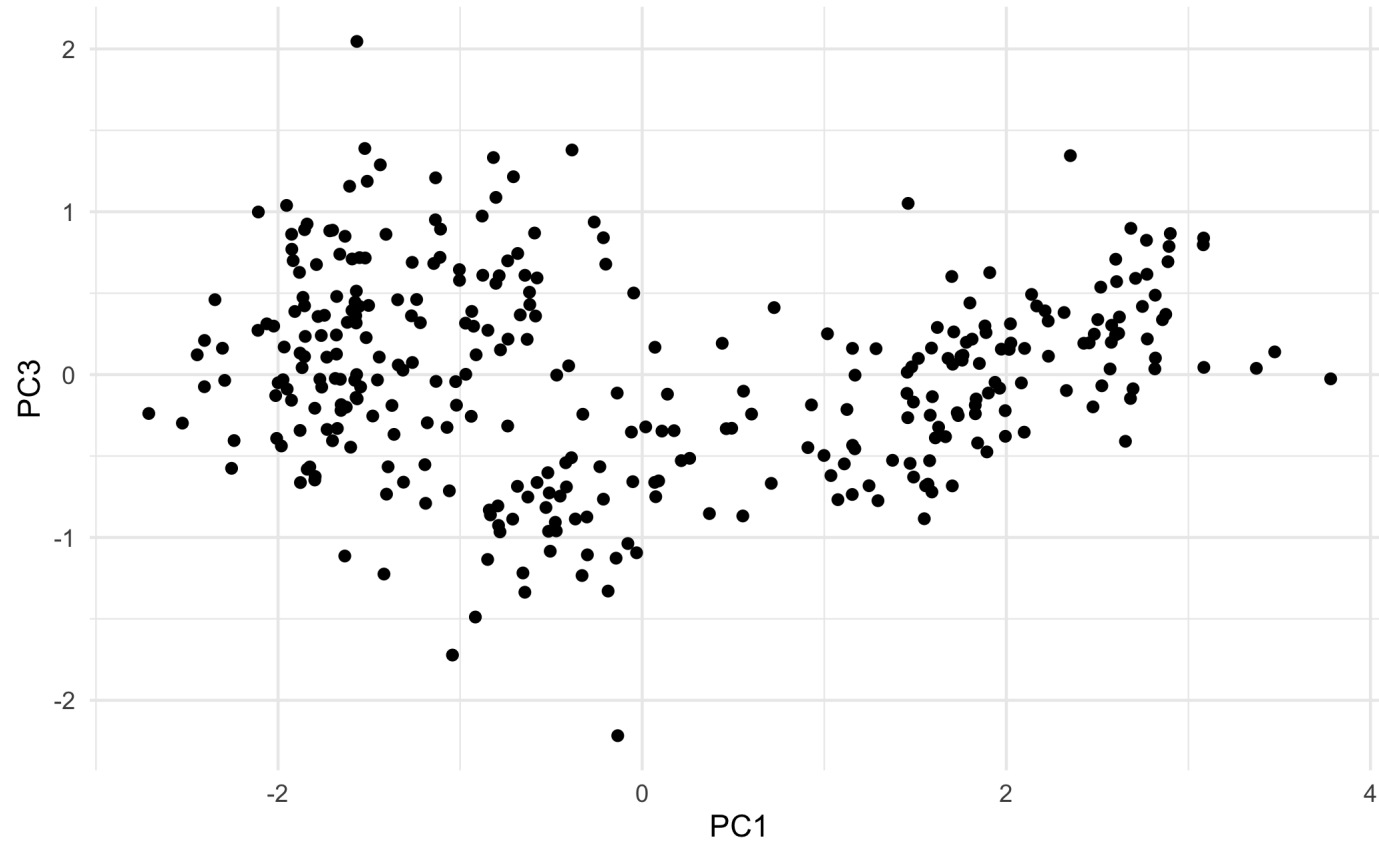
# Plotting PC1 against PC2



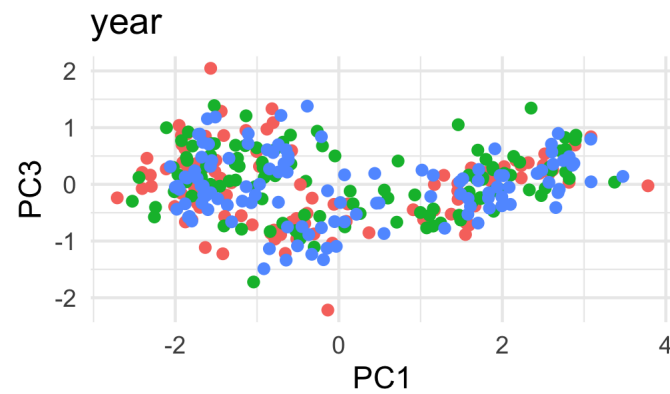
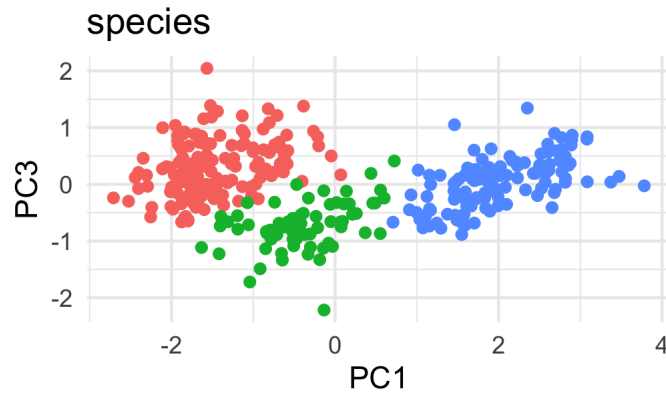
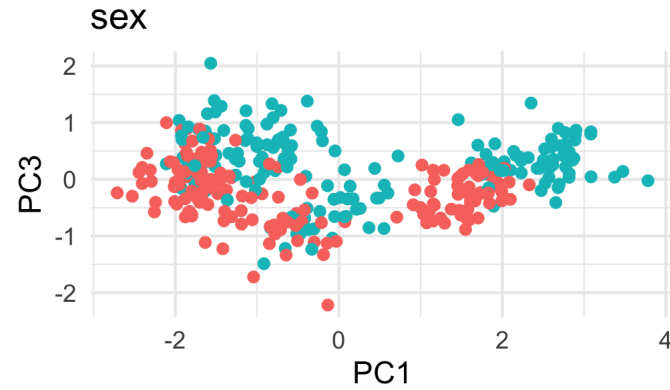
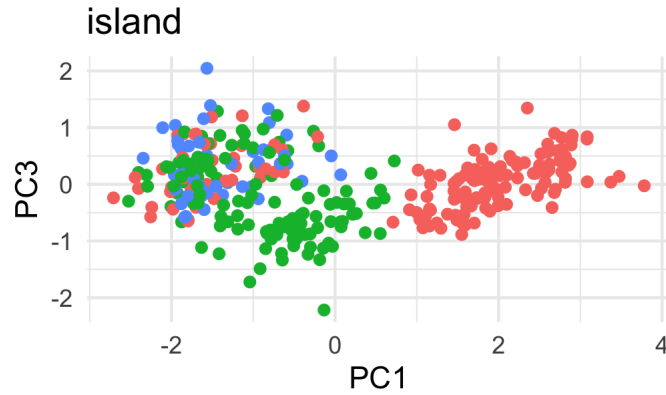
# Plotting PC1 against PC2



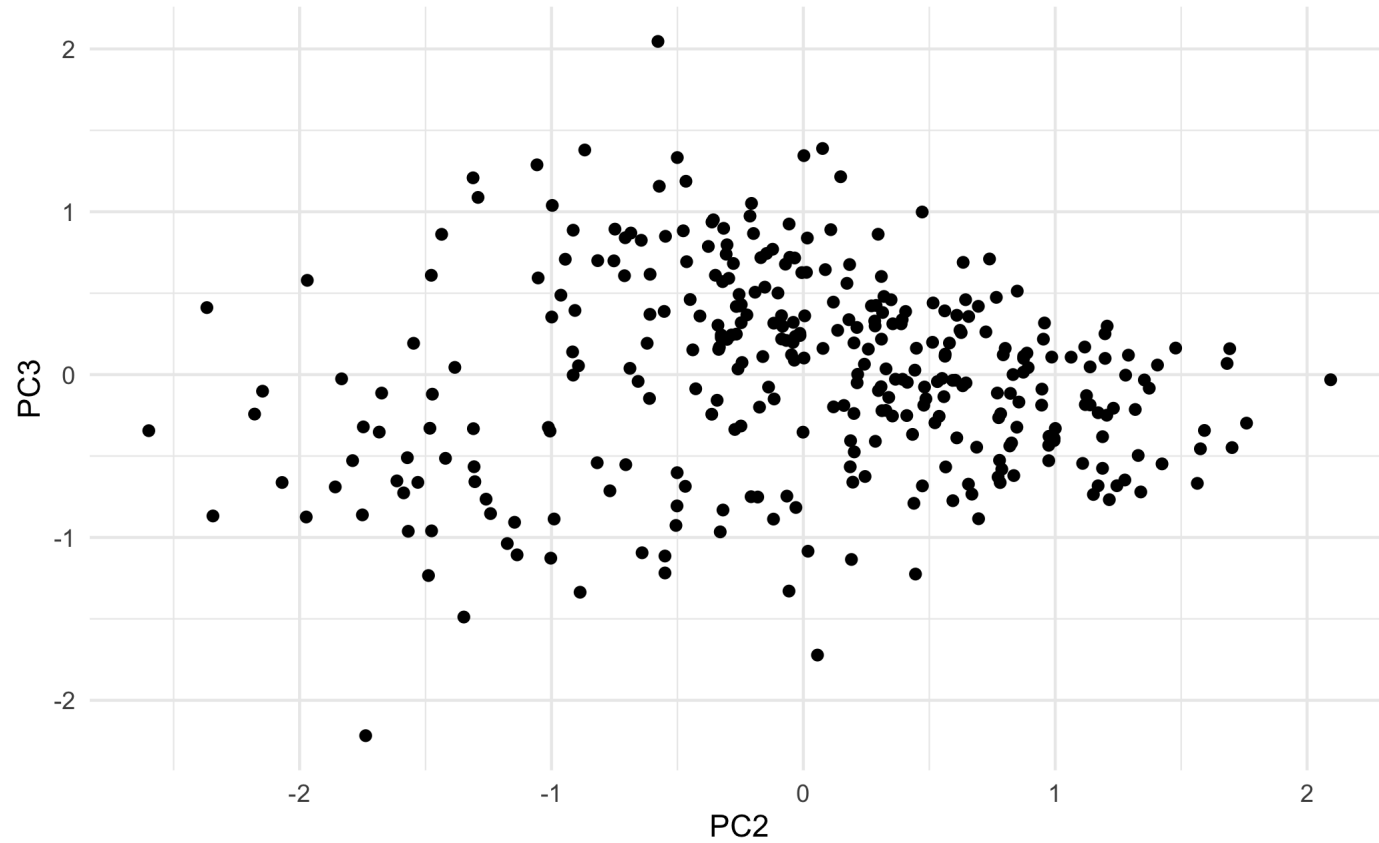
# Plotting PC1 against PC3



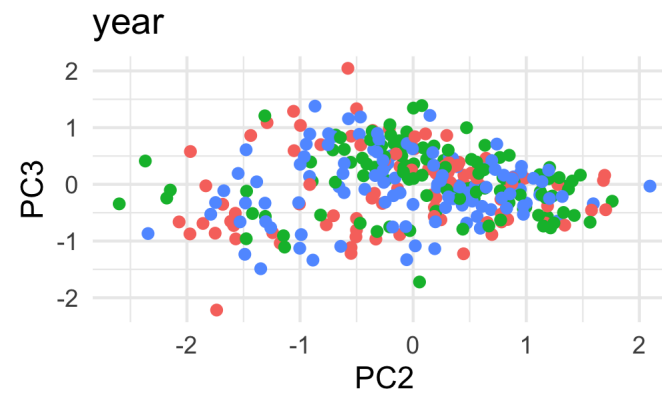
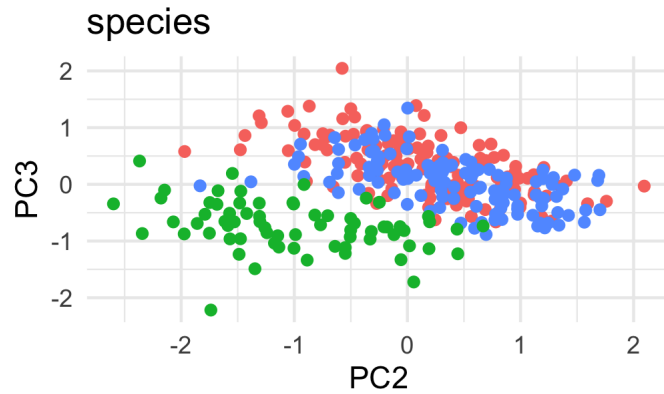
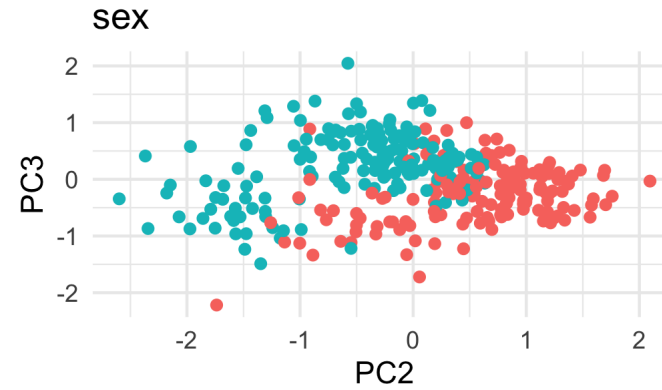
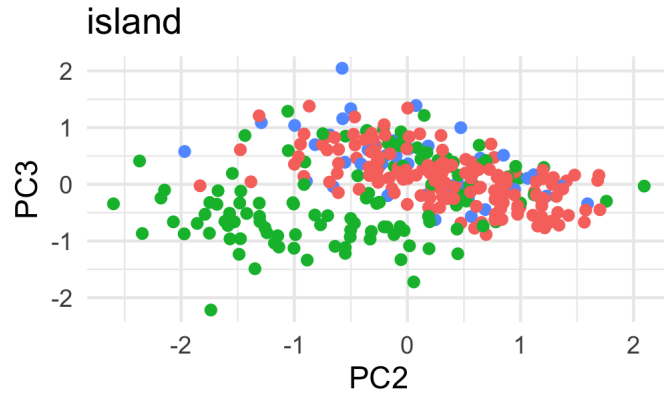
# Plotting PC1 against PC3



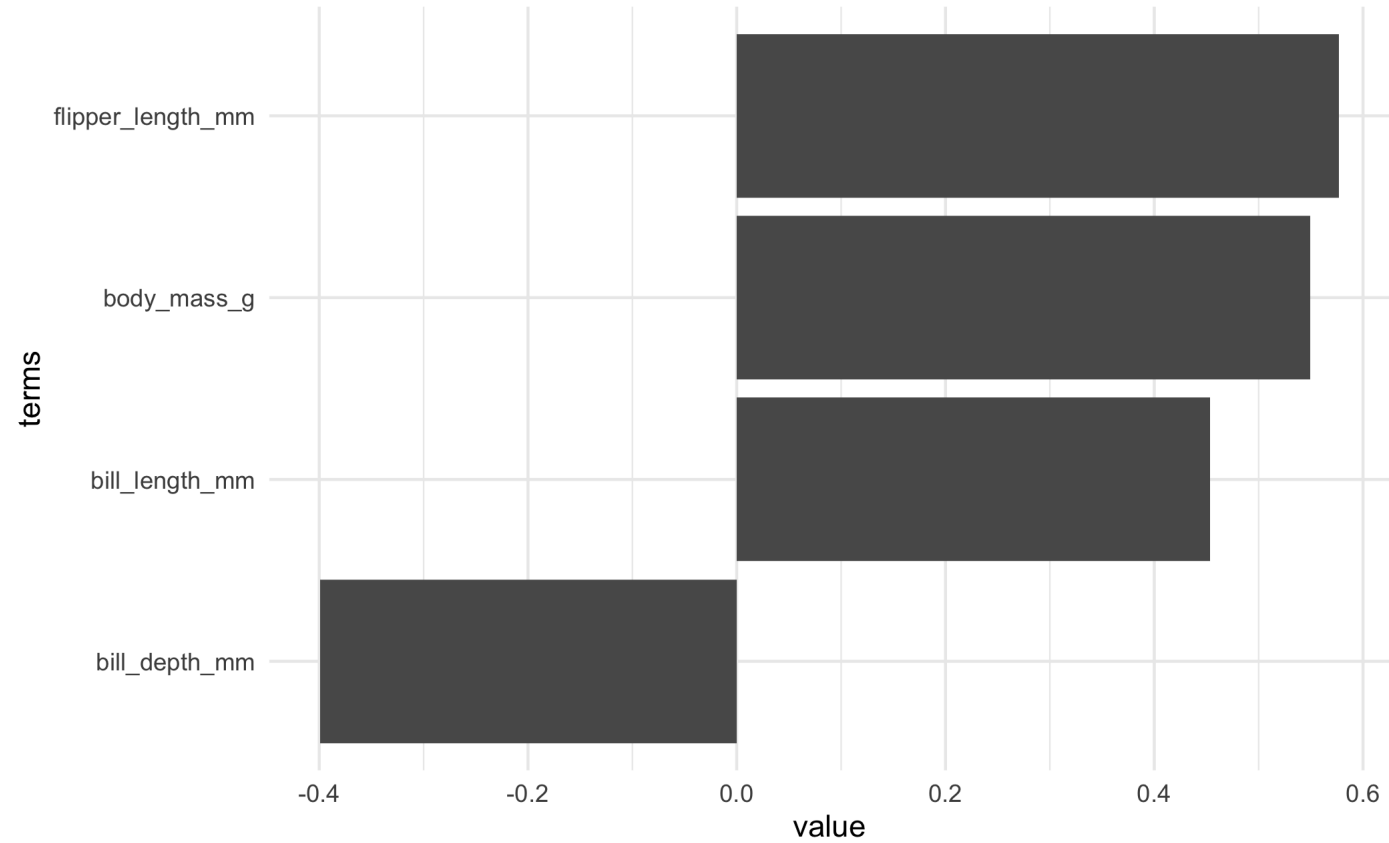
# Plotting PC2 against PC3



# Plotting PC2 against PC3

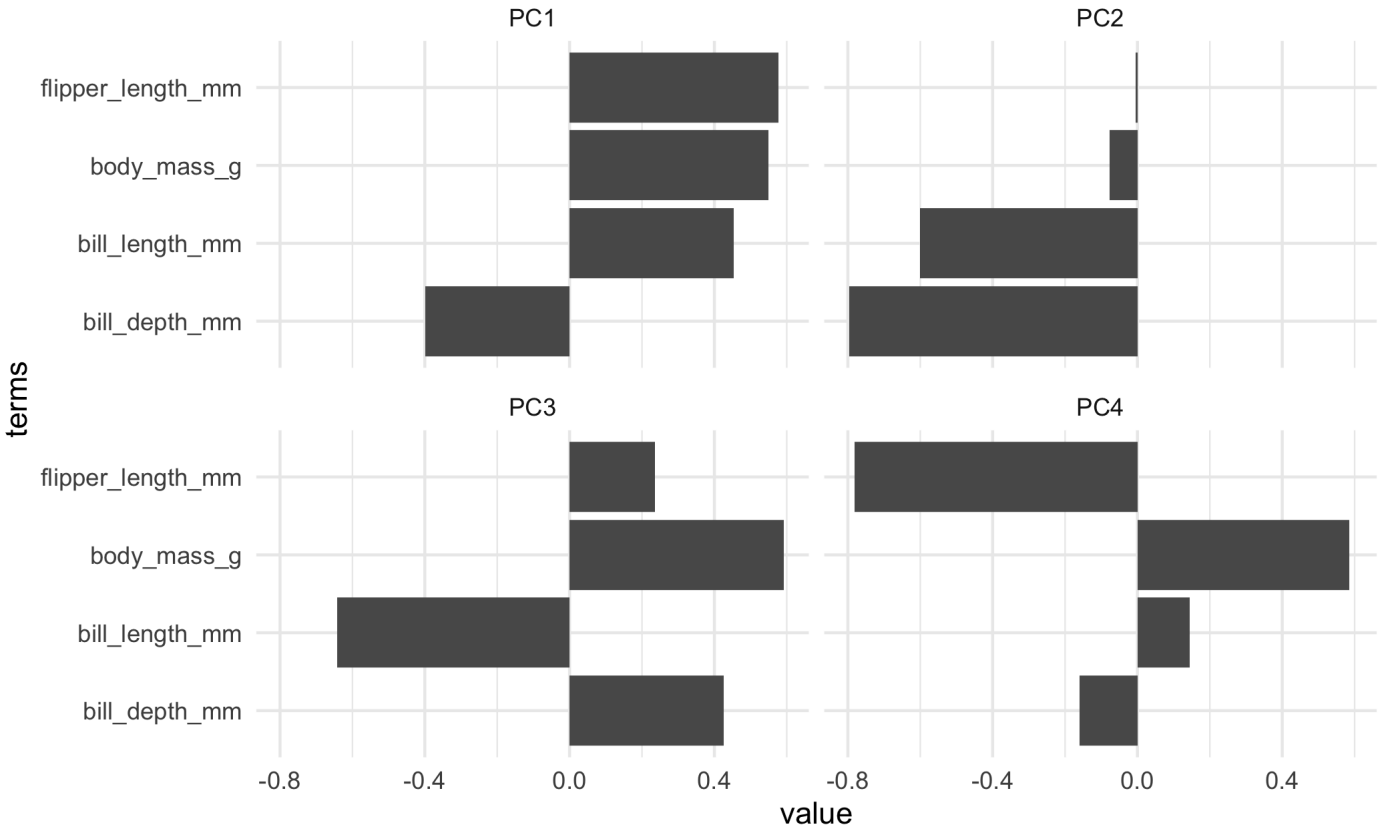


# Loadings for PC1

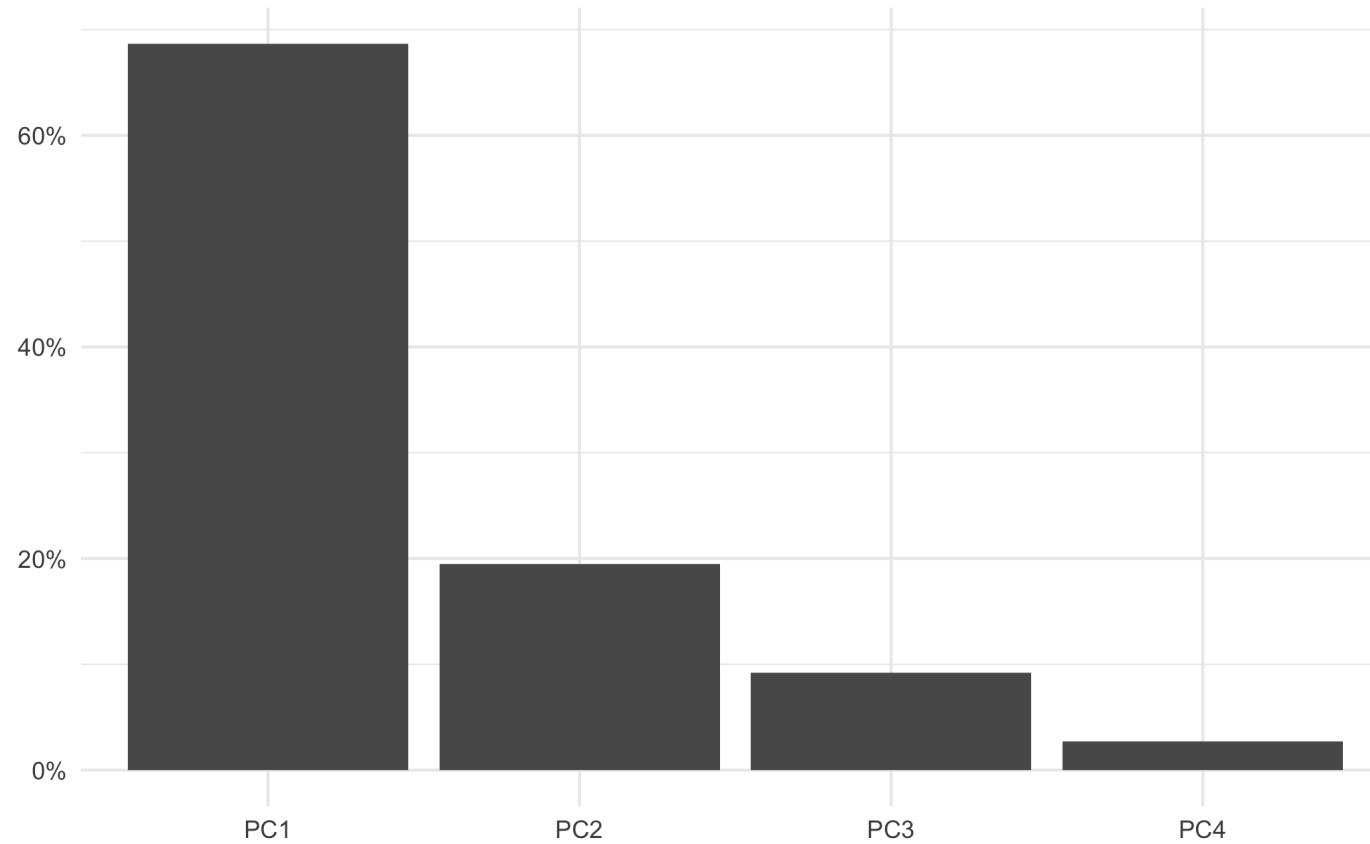




# Loadings for all Principal components



# Percent variance explained by each PCA component



# Alternative interpretation

Interpretation as a rotation of the space

# Scaling of variables

You must think about scaling the variables

Since we are maximizing some value then the magnitude of the variables will matter

If you don't have any prior knowledge of the data, it is advisable to set all the variables to the same scale

# Uniqueness of Principal Components

The principal components you generate should be unique up to a sign-flip of the loadings

# How is this a dimensionality reduction technique?

PCA is not a dimensionality reduction method by itself in the strictest sense

You get the reduction by only keeping some of the columns

- by number of columns
- Threshold by variance explained

# Extensions

Think of this problem as

$$X \approx UV$$

where

- $X$  is a  $n \times p$  matrix
- $U$  is a  $n \times d$  matrix
- $V$  is a  $d \times p$  matrix

# Extensions

What we want to do is find

$$\sum_{i=1}^n \sum_{j=1}^p \text{loss} (X_{ij}, (UV)_{ij})$$

subject to some constraints



# PCA

$$\sum_{i=1}^n \sum_{j=1}^p (X_{ij} - (UV)_{ij})^2$$

with no constraints

# Sparse PCA

$$\sum_{i=1}^n \sum_{j=1}^p (X_{ij} - (UV)_{ij})^2$$

Under the constraint that there is at most  $k$  columns in  $U$

In other words, each principal component can contain at most  $k$  loadings

# K-Means

$$\sum_{i=1}^n \sum_{j=1}^p (X_{ij} - (UV)_{ij})^2$$

Under the constraint that there is at most 1 columns in  $U$

# Non-Negative Matrix Factorization

$$\sum_{i=1}^n \sum_{j=1}^p (X_{ij} - (UV)_{ij})^2$$

Under the constraint that all the values of  $U$  and  $V$  are non-negative

# A Bluffer's Guide to Dimension Reduction - Leland McInnes

More about this: <https://www.youtube.com/watch?v=9iol3Lk6kyU&t=6s>

# Final Project

Search for data!

Ideas:

- <https://github.com/rfordatascience/tidytuesday>
- <https://www.data-is-plural.com/>
- <https://www.kaggle.com/>