# MA5810: Introduction to Data Mining

## Week 5; Collaborate Session 1: Principal Components Analysis

Martha Cooper, PhD

JCU Masters of Data Science

2021-08-03

# Housekeeping

- Collaborate 1 = Tuesday 6.45-8pm (Martha)
- Collaborate 2 = Thursday 6.45-8pm (Martina)

For my Collaborate Sessions, you can get the slides & R code for each week here:

https://github.com/MarthaCooper/MA5810

# Assessments

- Assessment 2 is due on Sunday 08/08/21

# Subject: MA5810 Intro to Data Mining

MA5810 Learning Outcomes

1. Overview of Data Mining and Examples
2. Unsupervised data mining methods e.g. clustering and outlier detection;
3. Unsupervised and supervised techniques for dimensionality reduction (Today = PCA);
4. Supervised data mining methods for pattern classification;
5. Apply these concepts to real data sets using R (Today).

# Today's Goals

- Understand the background behind Principal Components Analysis (PCA)
- Understand the pros and cons of PCA
- Apply PCA to real datasets using R

# Unsupervised Learning

A set of statistical tools to understand a set of features, $X_1, \ldots Xp$, without having an associated response variable, $Y$, to predict.

Data visualization, identification of subgroups & dimensionality reduction

# Principal Components Analysis (PCA)

A technique for summarizing a large set of variables into a smaller number of representative variables that collectively explain most of the variation in the original set.

*PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance*

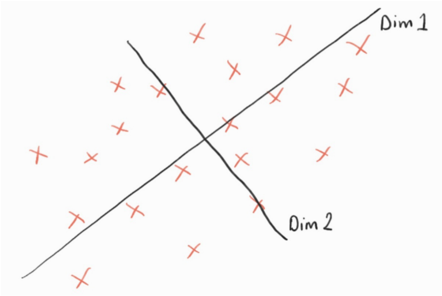
# Why reduce dimensions?

Problems:

- Correlated variables
- A large number of variables


Solutions:

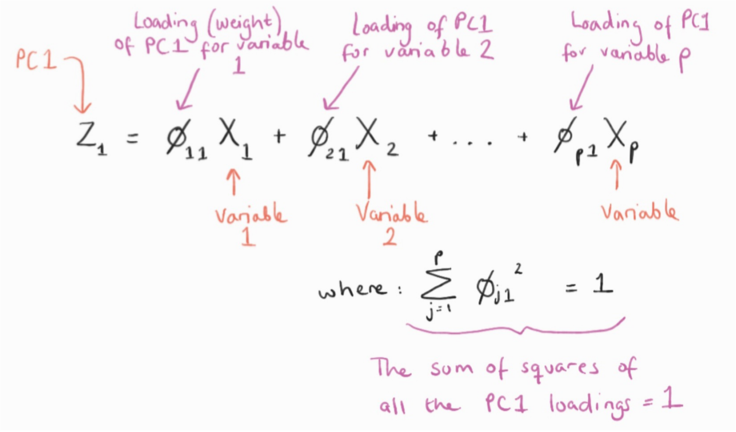
- A new set of variables that are *uncorrelated* and *explain as much variance as possible*
- *The best combination* of all the variables that *explains the original data set with less variables*



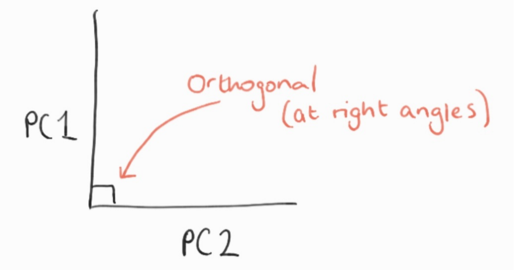
# Finding the Prinicpal Components

- Transform the data to a small number of interesting dimensions
- Interesting = Highest Variance
- These dimensions (Principal Components) are:

1. (Normalised) Linear combinations of the original variables

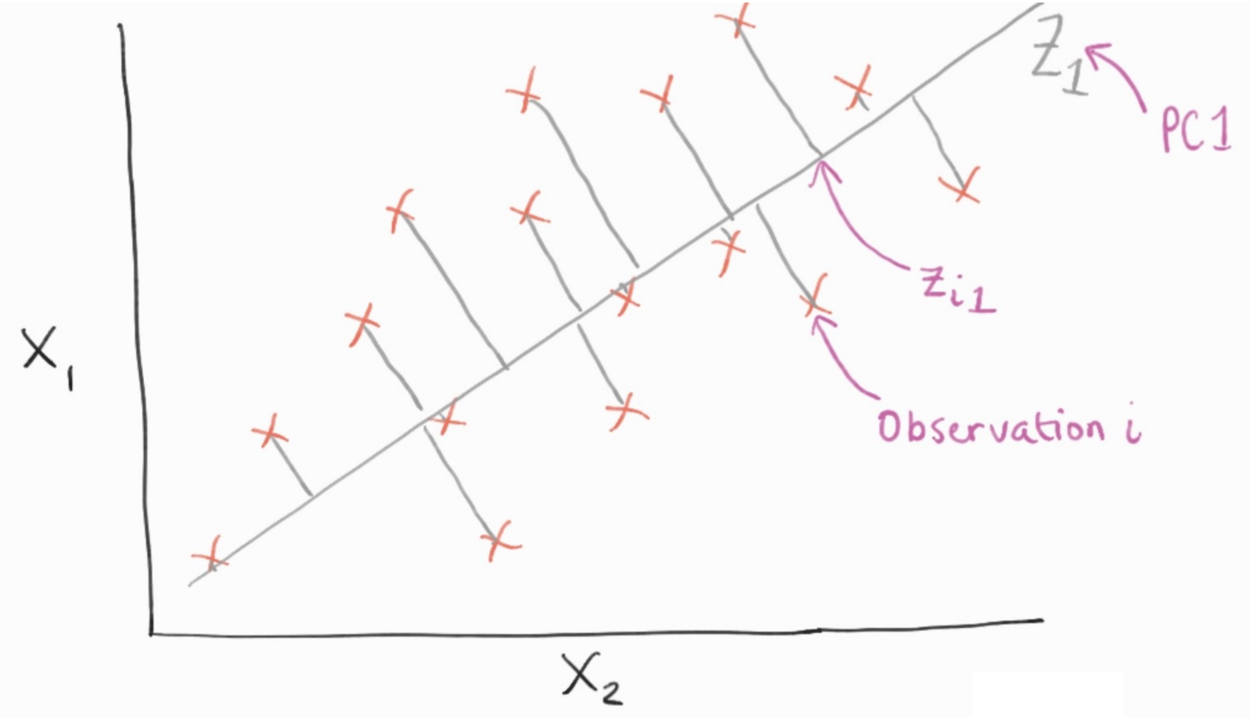



2. Uncorrelated



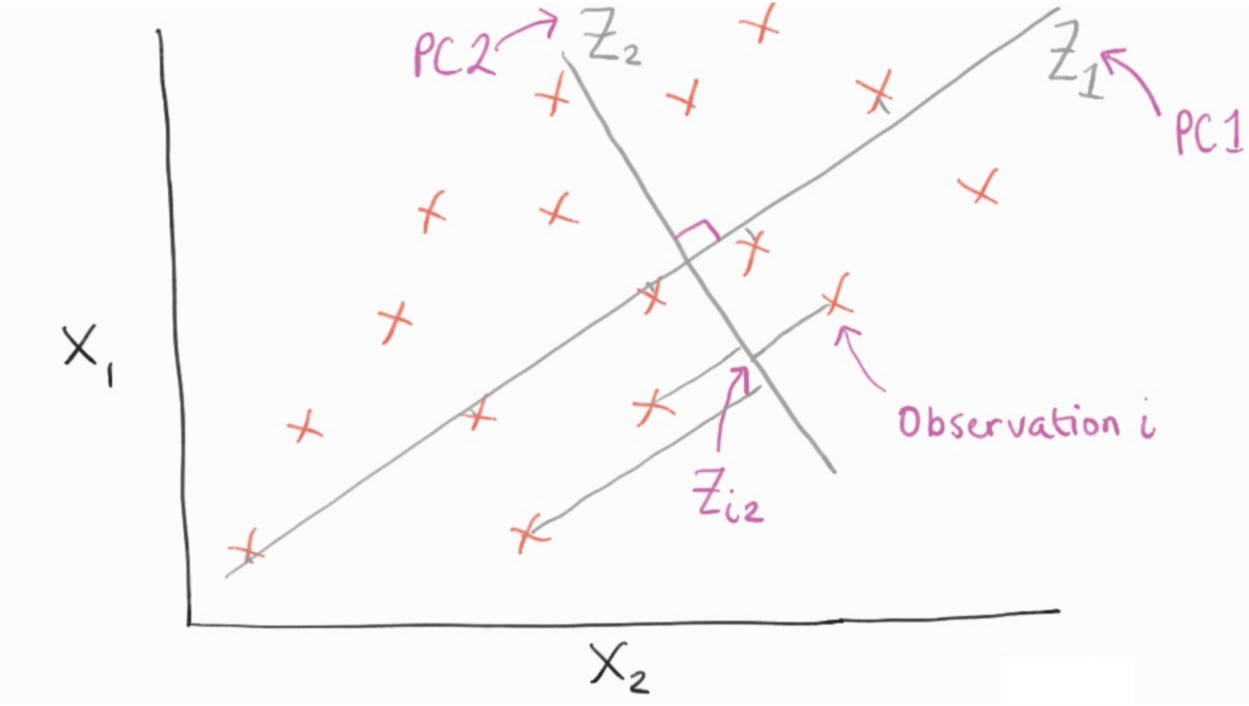
# Finding the Prinicipal Components

- How do we choose the loadings that cause PC1 to explain the most variance in the data?



# Finding the Prinicipal Components

- How do we choose PC2 - the second biggest source of variation & uncorrelated?



# Pros and Cons

Pros

- Reduce number of predictors
- Reduce number of correlated predictors
- Identify subgroups in our dataset
- Identify outliers

Cons

- Subjective
- Exploratory data analysis
- Difficult to interpret & assess results

# PCA in R

```
head(iris) #data
dat <- iris[ ,1:4] # remove Species column

pc <- prcomp(dat, center = T, scale = T) #why center? why scale?

pc$rotation #loadings - matrix of variable loadings
pc$x #scores - the coordinates of the observations on each PC
```

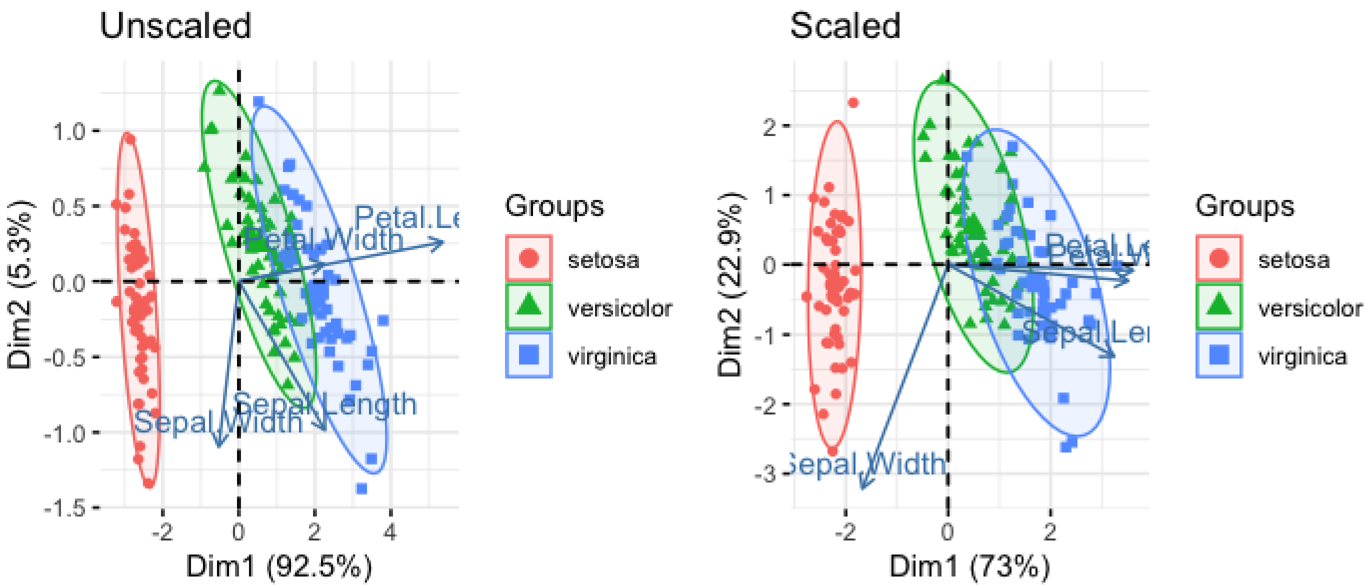
# Visualising PCA in R

```r
library(factoextra)

fviz_screeplot(pc) #scree plot - proportion of variance explained
# How might we choose how many PCs to keep?

fviz_pca_ind(pc) #PC1 vs PC2

fviz_pca_biplot(pc) #biplot
```

# Interpretting PCA with domain knowledge

```
fviz_pca_ind(pc,
             habillage = iris$Species,
             addEllipses = TRUE)
```

# Scaling




```
##              unscaled_variance scaled_variance
## Sepal.Length         0.6856935               1
## Sepal.Width          0.1899794               1
## Petal.Length         3.1162779               1
## Petal.Width          0.5810063               1
```

# PCA in R

Different functions have different methods for calculating the principal components:

- prcomp() - Singular Value Decomposition (SVD)
- factoextra::PCA() - Singular Value Decomposition (SVD)
- princomp() - Spectral Decomposition

# Principal Components Regression

Use *principal components as predictors in a regression model* instead of using the original larger set of variables.

1. Transform $n \times p$ into $n \times M$ dimensions using Principal Components Analysis, where $M >> p$
2. Fit regression model using the $M$ predictors.

# Principal Components Regression in R

Use ISLR::Hitters data set to predict baseball players salaries using 19 different variables

```
library(ISLR) # data
library(pls) # https://cran.r-project.org/web/packages/pls/index.html
library(caret) # test/training split

data("Hitters")
hit_dat <- na.omit(Hitters) #remove missing values
set.seed(6)
split_train<- createDataPartition(Default$default, p = 0.8, list = F)

# perform PCR on training data with standardisation of predictors
# perform 10 fold cross validation to compute error for each possible va
set.seed(6)
pcr_fit <- pcr(Salary ~., data = hit_dat, scale = TRUE, subset = split_t
summary(pcr_fit) #view fit
validationplot(pcr_fit, val.type="MSEP", legendpos = "top") #plot mean s

# make predictions on test data
pcr_pred <- predict(pcr_fit, hit_dat[-split_train, ], ncomp=6) # predict
mean((pcr_pred-hit_dat[-split_train, "Salary"])^2) # assess MSE
```

# Preprocessing with Principal Components Analysis - not just for regression

Other classification and clustering methods can be adapted to use $n \times M$ matrix where there columns are the first $M << p$ principal component score vectors, rather than the full $n \times p$ data set.

- caret::preProcess()
  - Caret Book
  - Jeff Leak's lecture
- Do it manually yourself using prcomp() and then your classifier of choice *e.g.* glm(family="binomial")
- Notes
  - Interpretation is harder as PCs are complex linear combinations of original variables
  - Outliers can have large effects on PCA - visualise first
  - Need to transform data? *e.g.* log scale

# Extra reading

- Chapter 10.2 of James *et al.*, ISLR
  - PCA for unsupervised learning
- Chapter 6.3 of James *et al.*, ISLR
  - Principal Components Regression

# References

- James *et al.*, ISLR

## Slides

- xaringhan, xaringanthemer, remark.js, knitr, R Markdown