Principal component analysis: examples

Introduction to Statistical Modelling

Prof. Joris Vankerschaver

Examples

1. Adulteration of olive oil

- Malavi, Derick, Amin Nikkhah, Katleen Raes, and Sam Van Haute. 2023. "Hyperspectral Imaging and Chemometrics for Authentication of Extra Virgin Olive Oil: A Comparative Approach with FTIR, UV-VIS, Raman, and GC-MS." Foods 12 (3): 429. https://doi.org/10.3390/foods12030429

2. Human faces dataset

- https://scikit-learn.org/0.19/datasets/olivetti_faces.html

Adulteration of olive oil

Problem setting

Extra virgin olive oil (EVOO):

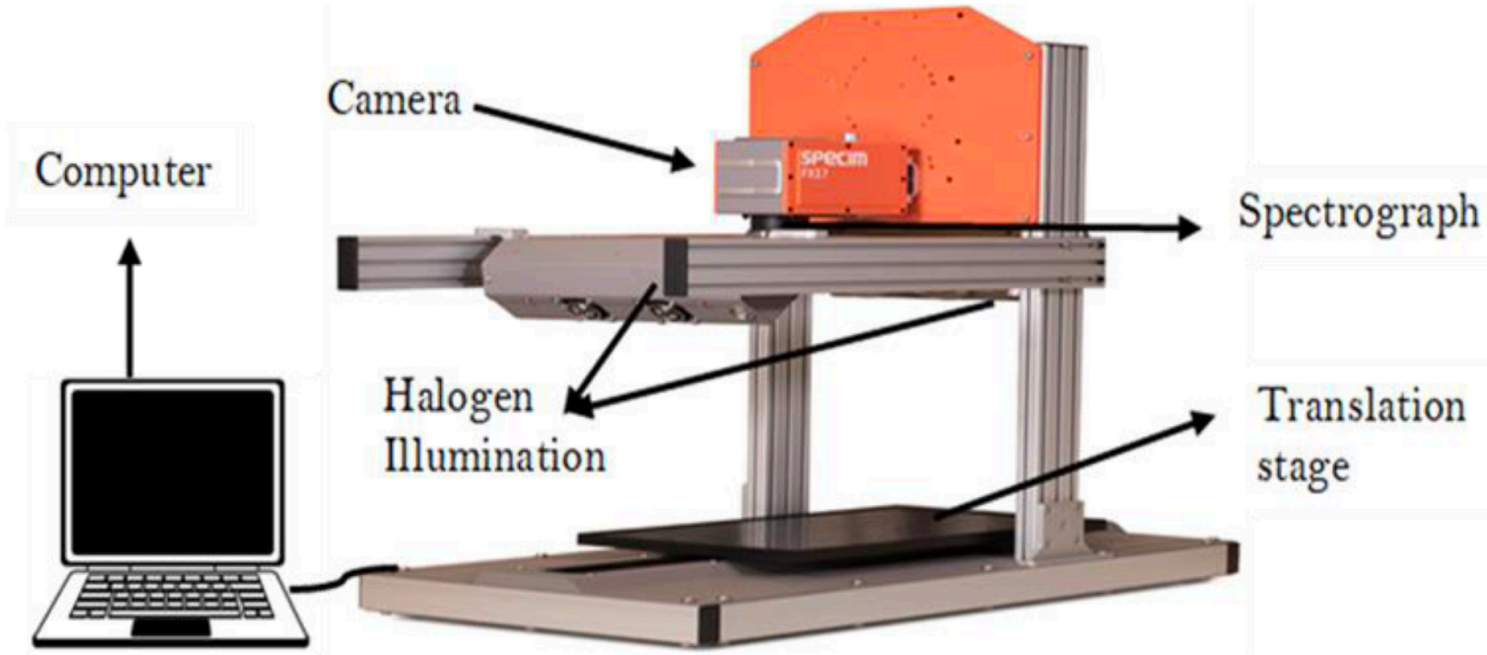
- High quality
- Flavorful
- Health benefits
- **More expensive** (than regular oil)

To reduce cost, EVOO is often **adulterated** with other, cheaper food oils.

Research questions

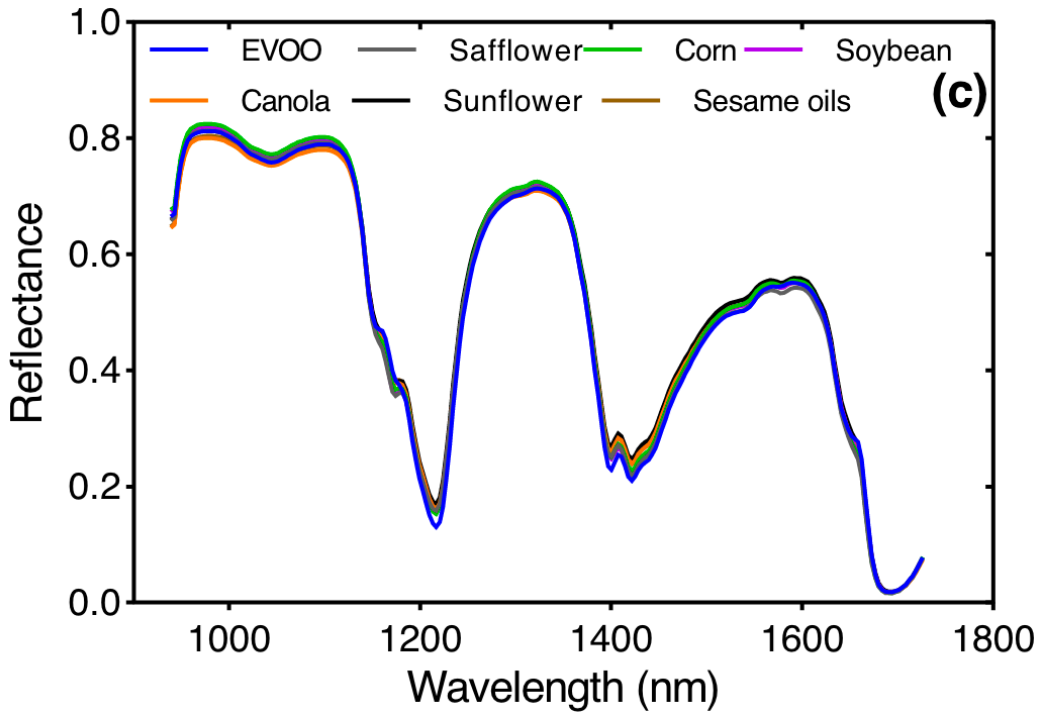
1. **Classification:** Can we detect whether a given EVOO sample has been adulterated?
   - Yes/no answer (categorical)
2. **Regression:** Can we detect the degree of adulteration?
   - Continuous answer, from 0% (no adulteration) to 100%

Hyperspectral imaging (HSI)



- Measures reflected infrared light (700-1800 nm) off sample
- Provides a non-destructive way of testing sample

Hyperspectral "images" (spectra)



- HSI measures reflectance at 224 wavelengths from 700 to 1800 nm
- Reflectance at given wavelength is determined by molecular features of sample

Experimental setup

Samples to test (61 total):

- 13 different kinds of unadulterated EVOO
- 6 vegetable oils
- 42 adulterated mixtures
  - EVOO + one of 6 vegetable oils at one of 7 different percentages (from 1% to 20%)

Each sample is imaged 3 times: **183 samples**

Each sample produces a HSI spectrum of **length 224**

Data matrix

Data matrix has 183 rows (samples) and 224 columns (spectra).

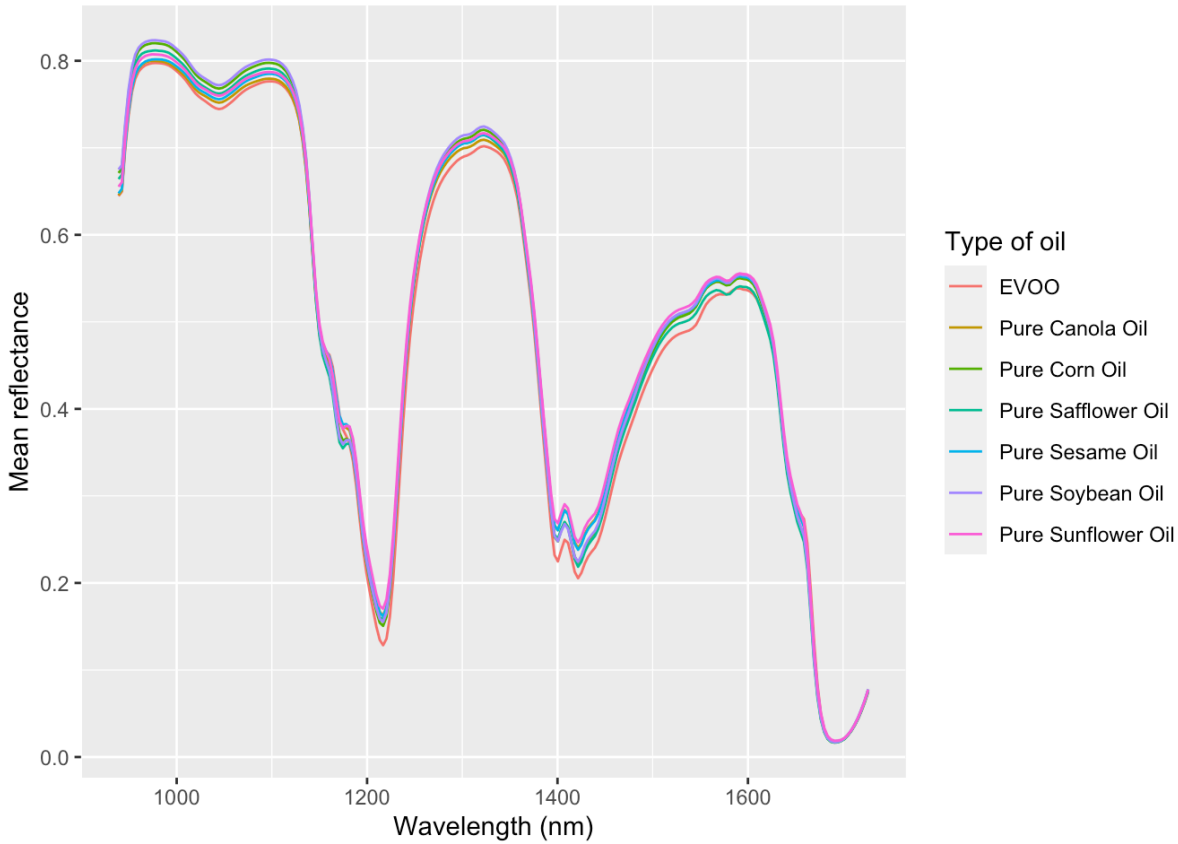
In addition, we have some metadata:

- Name of sample
- Degree of adulteration

	Sample ID/Wavelength	Sample	Classification	% Adulteration	938.94000200000005	942.45001200000002	945.96002199999998
1	Monini Classico EVOO	1	Olive	0	0.650031	0.655155	0.704436
2	Monini Classico EVOO	2	Olive	0	0.646796	0.651895	0.701250
3	Monini Classico EVOO	3	Olive	0	0.651539	0.656589	0.704596
4	Fontana EVOO	4	Olive	0	0.649832	0.654923	0.703678
5	Fontana EVOO	5	Olive	0	0.645579	0.650628	0.698899
6	Fontana EVOO	6	Olive	0	0.647227	0.652270	0.700465
7	Divella EVOO	7	Olive	0	0.646414	0.651584	0.700632
8	Divella EVOO	8	Olive	0	0.649089	0.653915	0.701284
9	Divella EVOO	9	Olive	0	0.639494	0.645490	0.701185
10	EVOO from Spain	10	Olive	0	0.643378	0.648587	0.699279
11	EVOO from Spain	11	Olive	0	0.646907	0.651400	0.696273
12	EVOO from Spain	12	Olive	0	0.640076	0.645553	0.697743
13	Borges EVOO	13	Olive	0	0.645270	0.650284	0.698843
14	Borges EVOO	14	Olive	0	0.641859	0.646935	0.695553
15	Borges EVOO	15	Olive	0	0.639936	0.645475	0.698057
16	Premium Oil EVOO	16	Olive	0	0.640139	0.645473	0.696361
17	Premium Oil EVOO	17	Olive	0	0.639872	0.645166	0.695145
18	Premium Oil EVOO	18	Olive	0	0.645821	0.650525	0.695868

A first look at the data

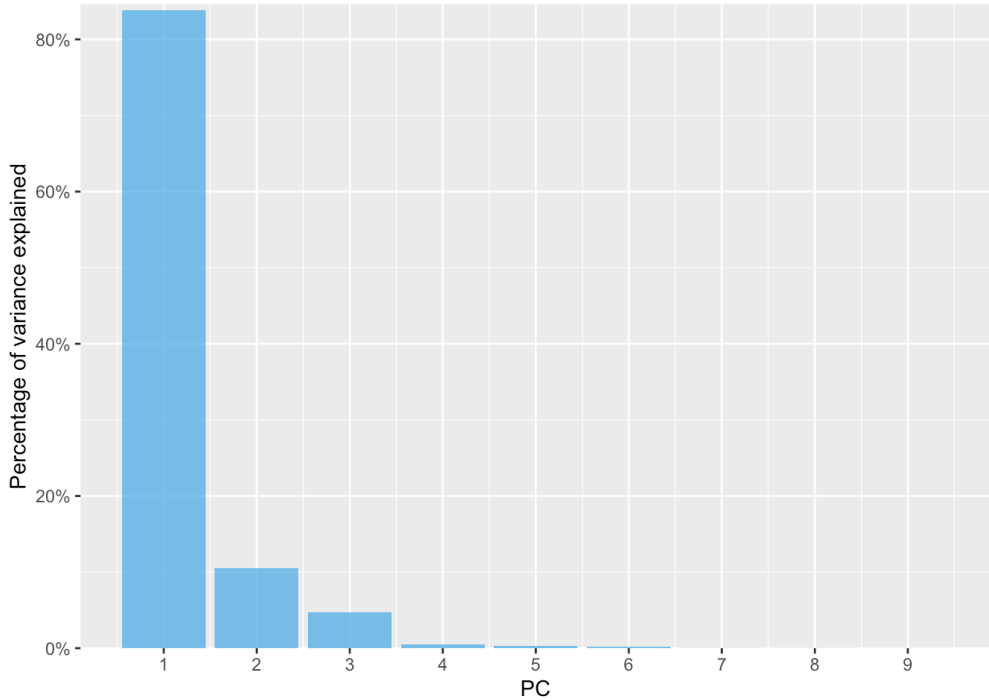
Averaged spectra for each kind of oil (EVOO + 6 others)



Plot shows small differences between spectra: **promising sign** that we will be able to address the research questions.

Principal component analysis: scree plot

Not all 224 wavelengths are equally informative. Much of our dataset is redundant.

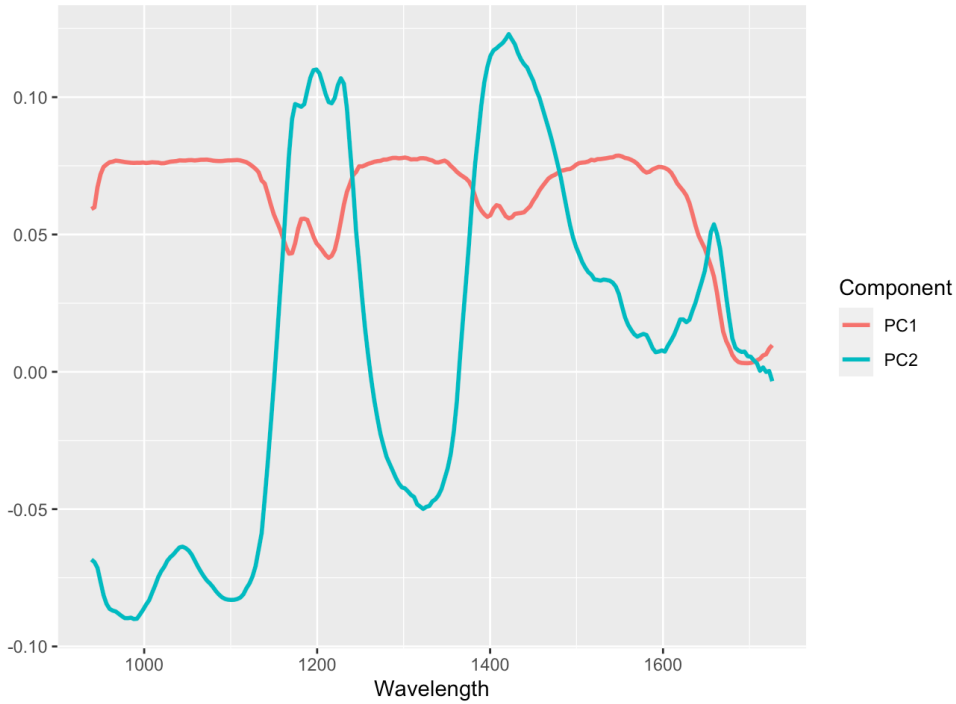


This is confirmed by the scree plot:

- First 2 PCs explain **94% of variance** in the data
- First 3 PCs: almost 100%

Principal component analysis: loadings vectors

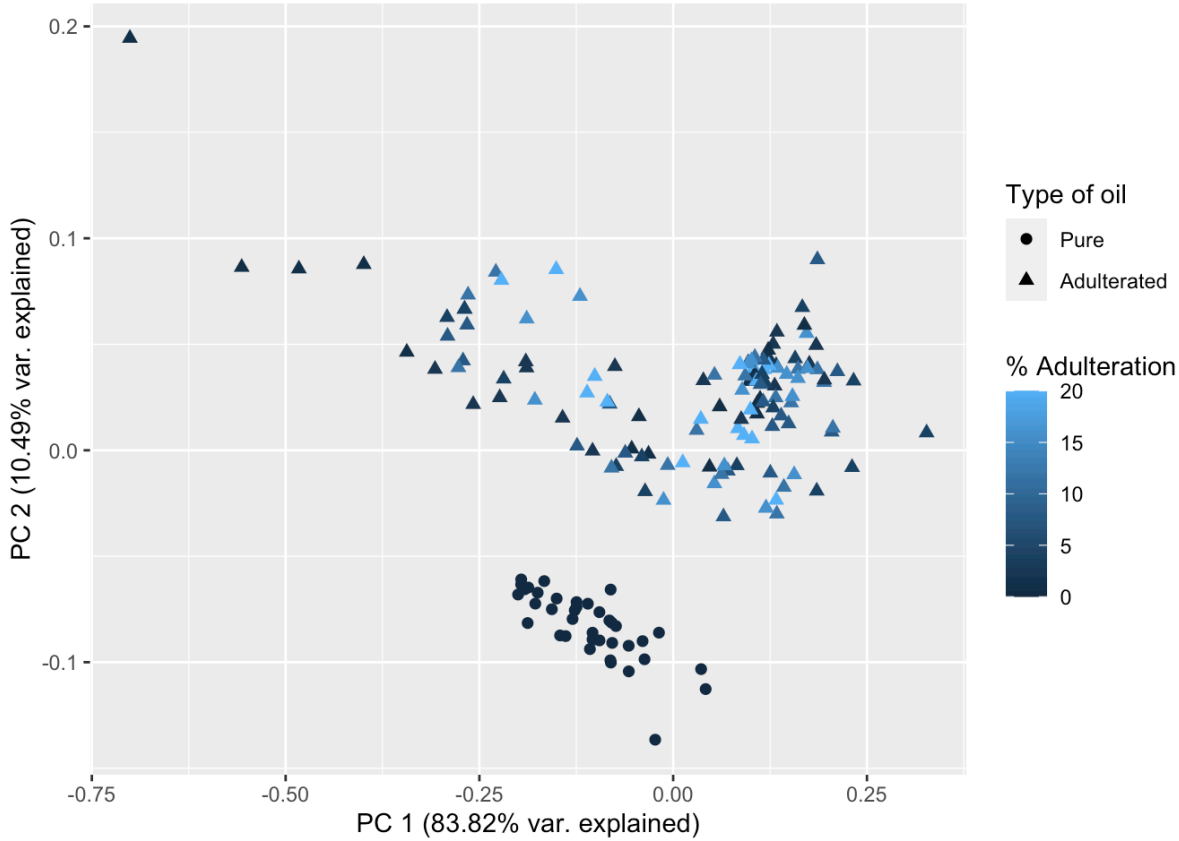
Loadings vectors are linear combinations of features, tell us how features contribute to variability in dataset.



For our example:

- Loadings vector 1: where do spectra differ the most?
- Loadings vector 2: where is next source of variability located?

Principal component analysis: scores



Can we tell pure and adulterated samples apart?

- **Yes**: clearly different on score plot.

Can we predict the percentage of adulteration?

- **No**: hard to distinguish from first 2 PCs alone.

Predicting the percentage of adulteration

We will need more than 2 PCs to correctly predict percentage of adulteration.

Two different approaches:

- **Principal component regression**:
  1. Compute PCs
  2. Do a regression on PCs
- **Partial least squares regression**:
  1. Compute factors that are most variable and **most correlated with outcome**
  2. Do a regression on resulting factors

Both models can be built using the `pls` package in R.

Dataset

For this example we will use only the 42 adulterated mixtures.

Each mixture is imaged 3 times: $42 \times 3 = 126$ samples

Predictors: 224 wavelengths

Outcome: percentage of adulteration (1%-20%)

Performing a fair assessment: train/test split

Evaluating the model using the same data used to train it leads to an **optimistic** estimate of the model's performance.

To avoid this bias, randomly select and set aside some data for testing, and use the remaining data to develop the model.

Test data (20%)	Train data (80%)

Adulteration prediction:

- Train dataset: 101 samples
- Test dataset: 25 samples

Can you spot an issue with this?

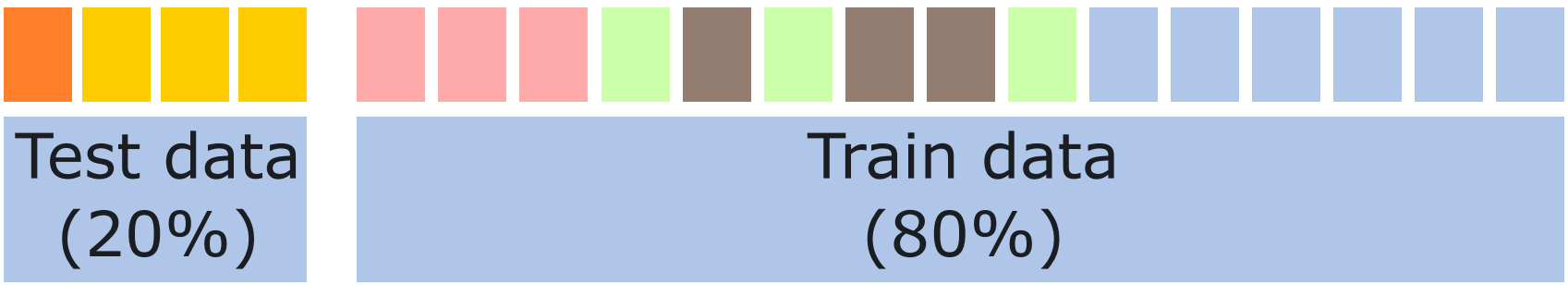
Performing a fair assessment: data leakage

- Each of the 42 mixtures is imaged 3 times.
- Presumably these replicates are very similar
- If some replicates end up in the test dataset and some in the train dataset: model gains unfair advantage.

Avoiding data leakage: stratified train/test split

Main idea: develop model with some of the mixtures, test performance on different mixtures:

1. Randomly select 80% of **mixtures**
2. Put all 3 replicates for those 80% in the training set
3. Put the remainder in the test set.



Building the PCR/PLS models

PCR model:

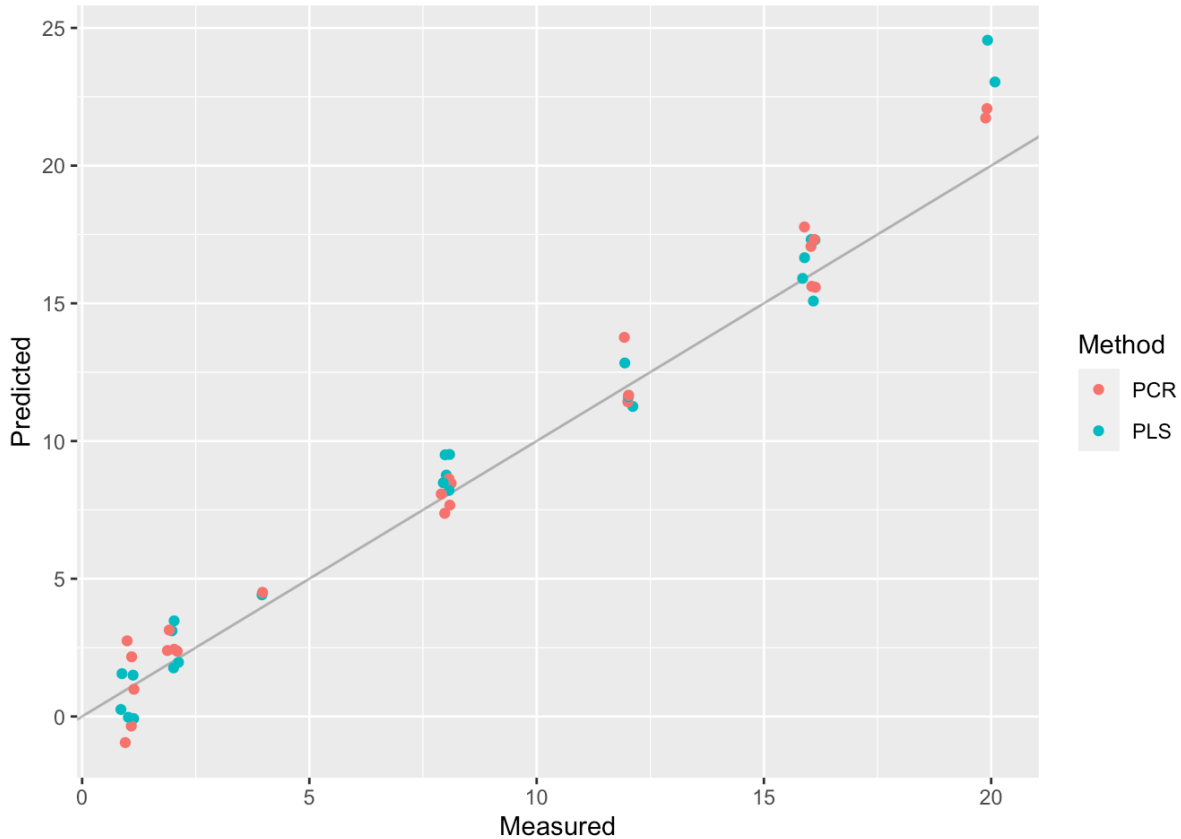
```
pcr_model <- pcr(
  `% Adulteration` ~ ., data = adulterated_train,
  scale = FALSE, validation = "CV", ncomp = 10
)
```

PLS model: replace `pcr` by `plsr`.

Arguments:

- `scale = FALSE`: Don't scale spectra (same units)
- `ncomp = 10`: Build model with up to 10 components
- `validation = "CV"`: Assess performance of model with i components using cross-validation

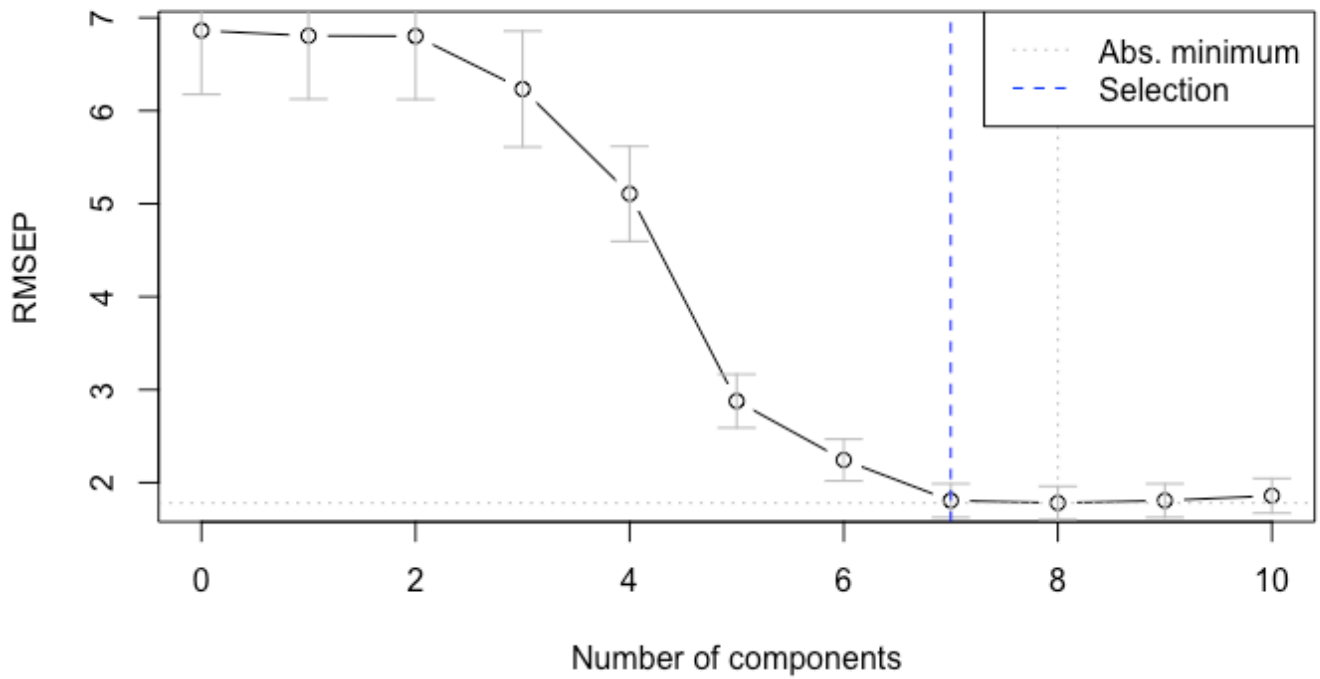
Performance of PCR/PLS models



Both models do well on the test data.

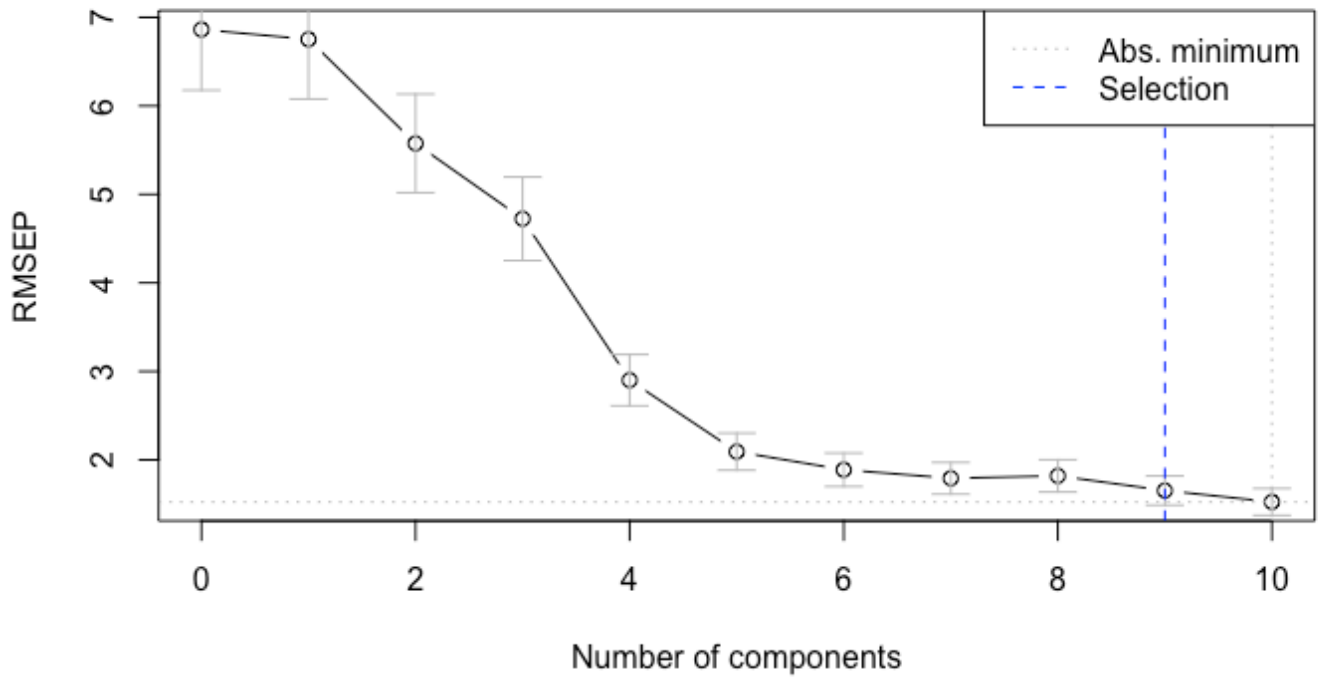
Optimal number of components: PCR

(obtained via `selectNcomp(method = "onesigma")`)



- Optimal number of components: 7
- RMSEP for 7 components: 1.796

Optimal number of components: PLS



- Optimal number of components: 9
- RMSEP for 9 components: 1.627

Conclusions

Can we detect whether a given EVOO sample has been adulterated?

- **Yes**: Look at score plot
- More conclusive answer next lecture

Can we detect the degree of adulteration?

- **Yes**: Build PCR or PLS model

Human faces dataset

There are no slides for this part of the lecture. Instead, the lecture will follow the discussion in the following book chapter: https://jvkersch.github.io/ISM/pca-applications.html#sec-eigenfaces