Principal Component Analysis

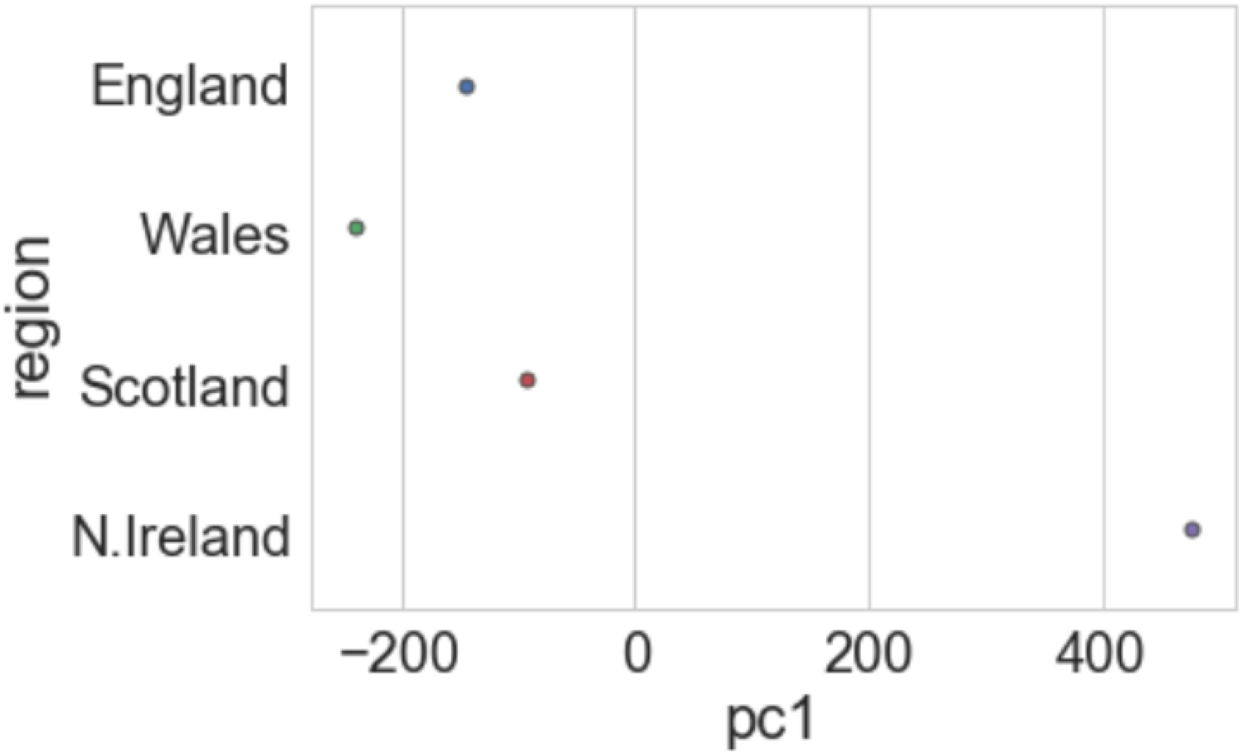
Food consumption in the UK

http://setosa.io/ev/principal-component-analysis/

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

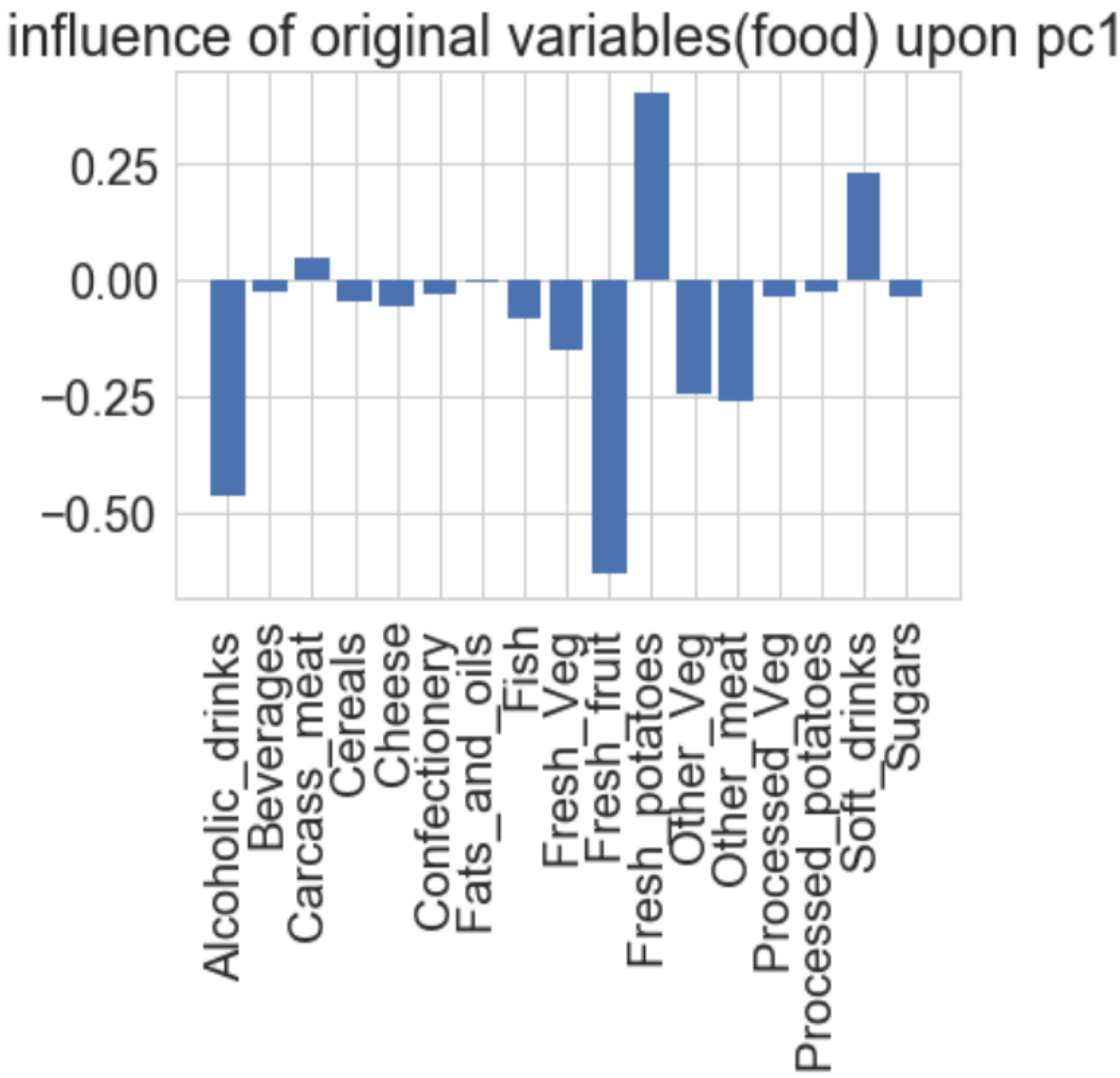
How can we focus in just a few of the variables?

We want to reduce the dimension of the feature space,
Let's try to reduce to one dimension:

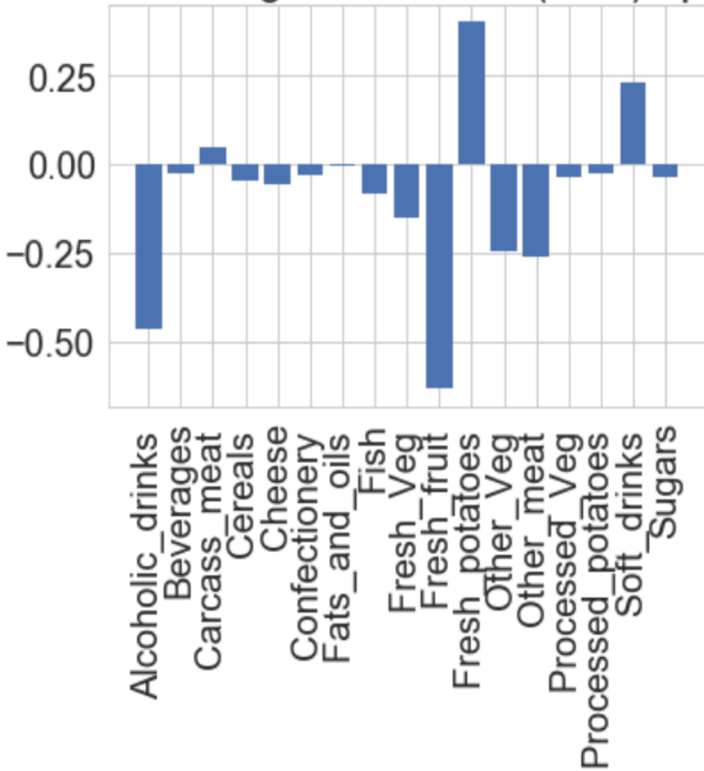


pc1: Principal component 1 - linear combination of the other 17 variables

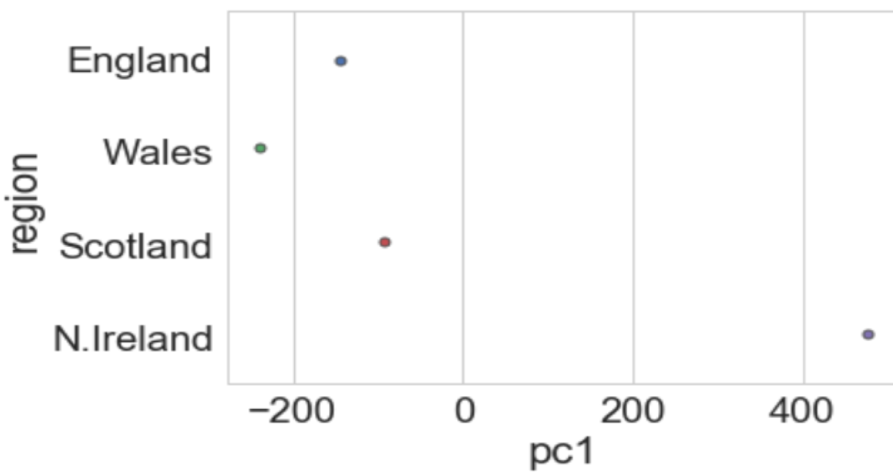
$$pc1 = x1\ Alcoholic\ Drinks + x2\ Beverages + x3\ Carcase\ meat + \ \ldots + x17\ Sugars$$



influence of original variables(food) upon pc1

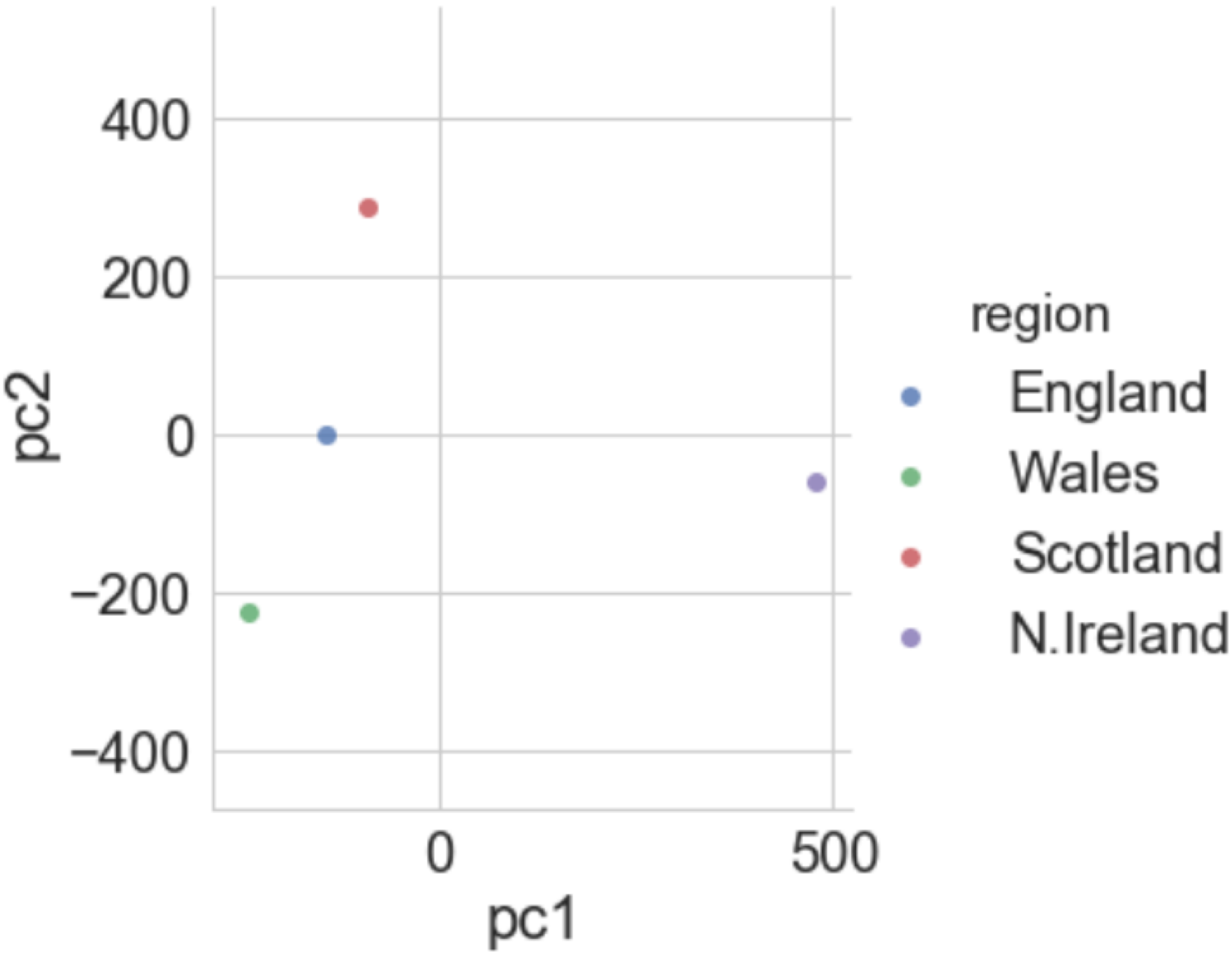


	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

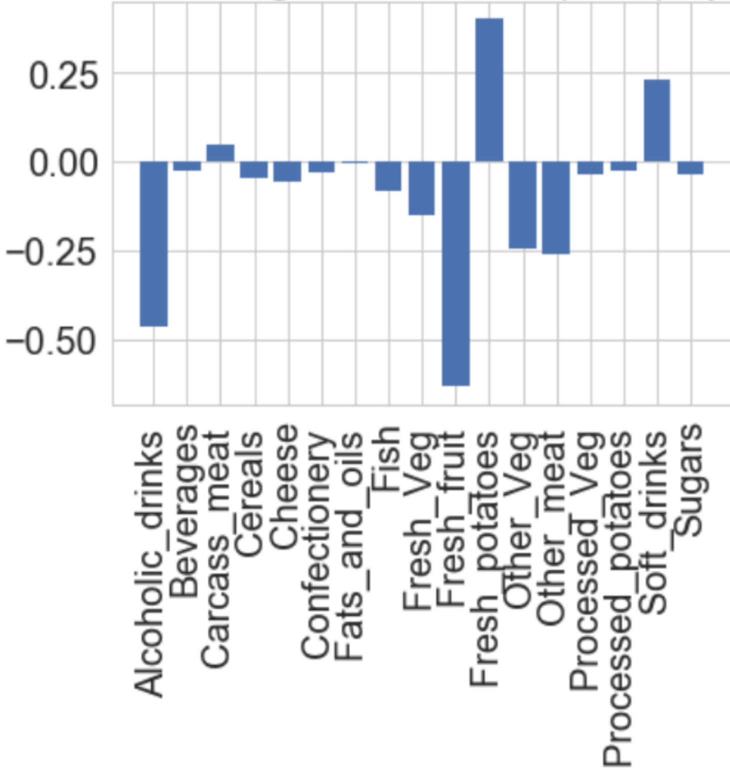


How can we focus in just a few of the variables?

What about reducing to two dimensions?

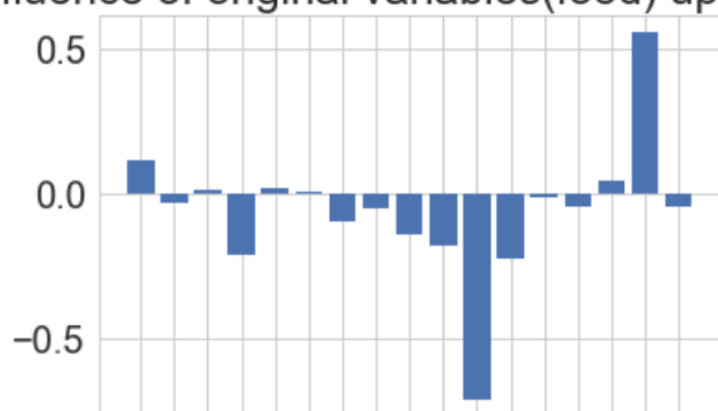


influence of original variables(food) upon pc1



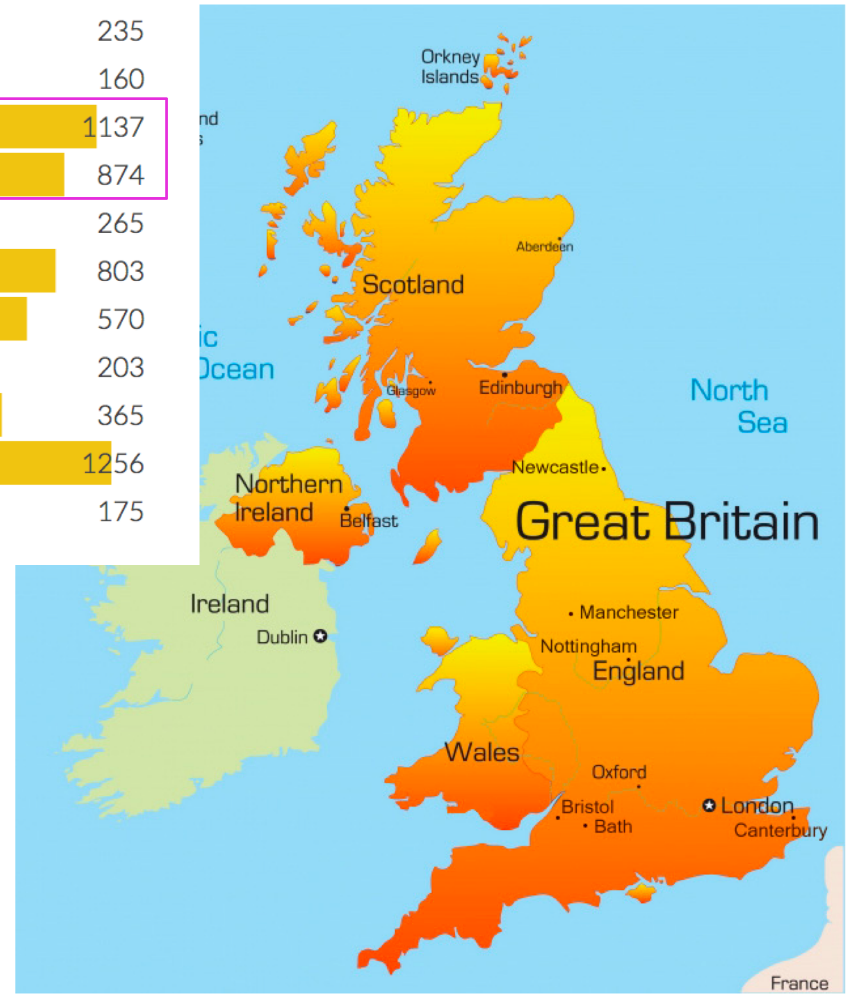
	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

influence of original variables(food) upon pc2



The three variables, Fresh potatoes, Alcoholic drinks and Fresh fruit, there is a noticeable difference between the values for England, Wales and Scotland, which are roughly similar, and Northern Ireland, which is usually significantly higher or lower.

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175



Predicting breast cancer

https://www.kaggle.com/shravank/predicting-breast-cancer-using-pca-lda-in-r

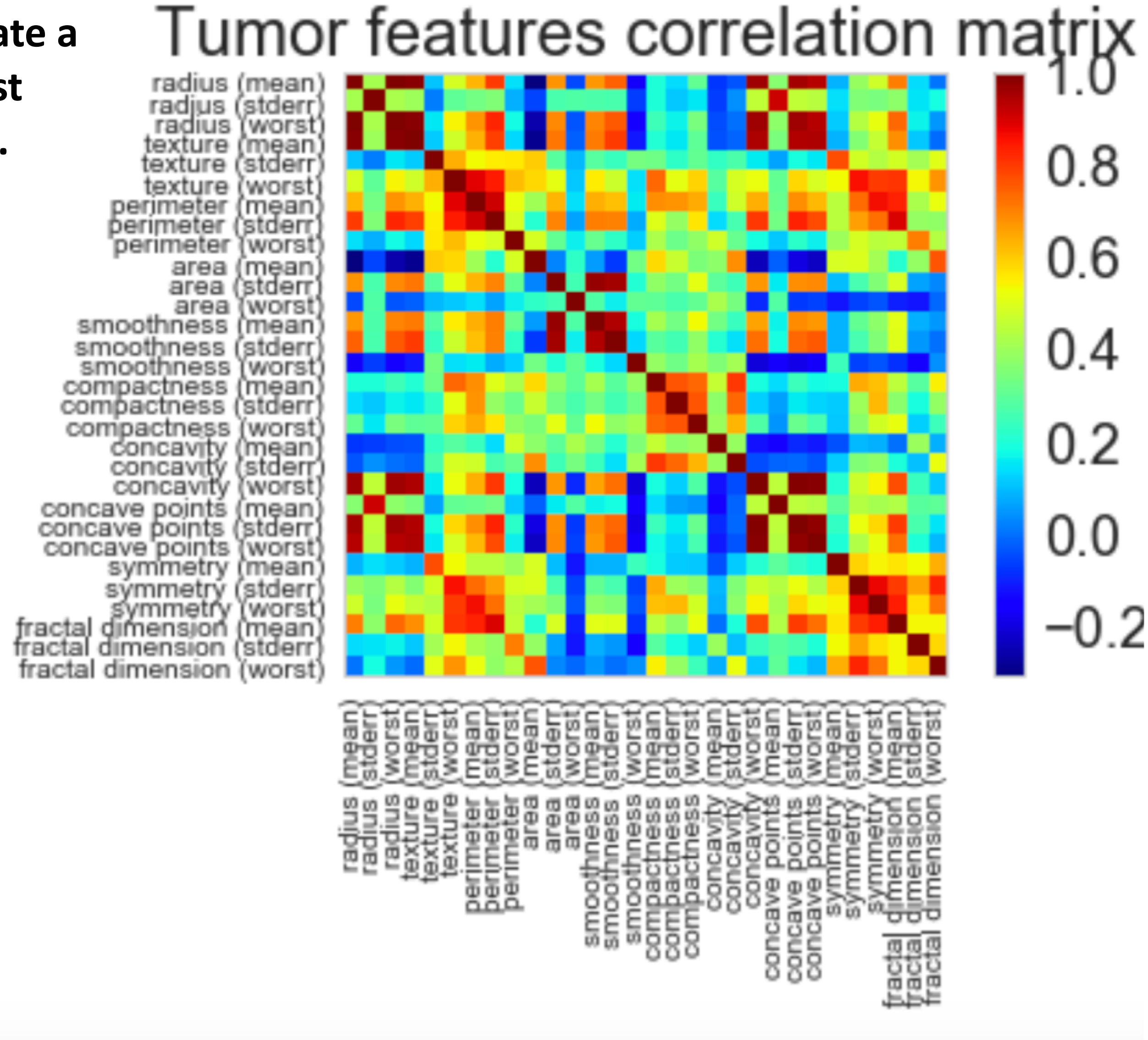
Goal (MP): Use data about tumor cell features to create a model to predict if a breast tumor is malign or benign.

The data includes 30 different cell features.

There are many variables that are highly correlated with each other.

Reduce the feature space:

<u>Approach 1</u>: remove some of the feature variables.



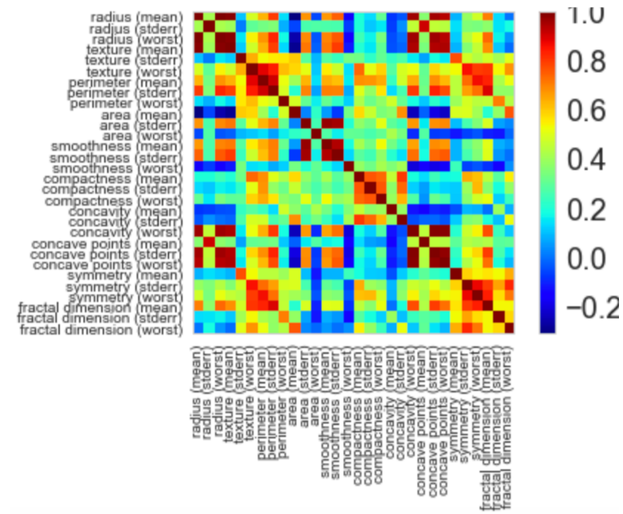
Example: Reduce the feature space by including only the features regarding the mean

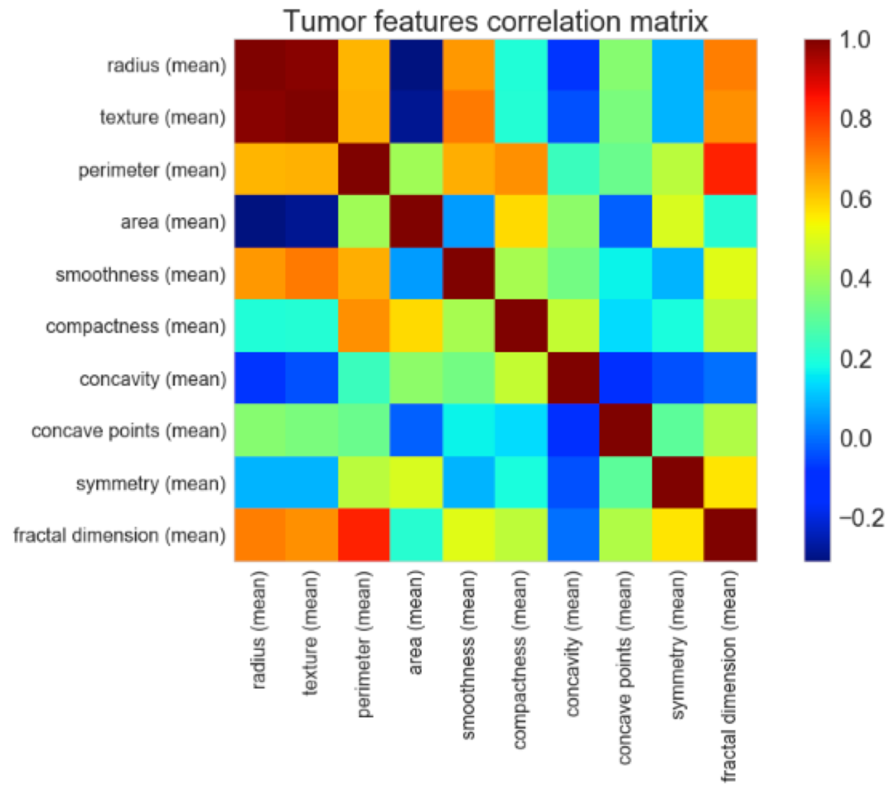
$$A = \begin{bmatrix} \vdots & & \vdots \\ F_1 & \cdots & F_{30} \\ \vdots & & \vdots \end{bmatrix}$$

$$A^* = \begin{bmatrix} \vdots & & \vdots \\ F_1 & \cdots & F_{10} \\ \vdots & & \vdots \end{bmatrix}$$

PROS: simple and maintain interpretation of the feature variables

CONS: lose information from the variables that were dropped





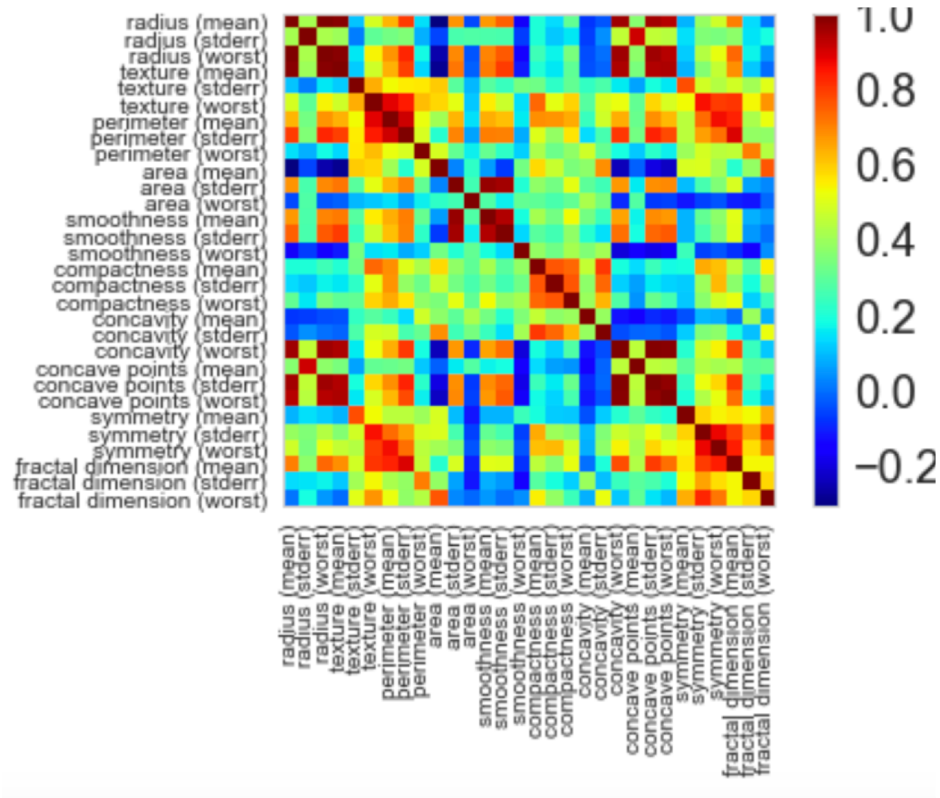
Get a new data set, resulting from a linear combination of the original dataset

$$A = \begin{bmatrix} \vdots & & \vdots \\ F_1 & \cdots & F_{30} \\ \vdots & & \vdots \end{bmatrix}$$

$$\Downarrow$$

$$A^* = \begin{bmatrix} \vdots & \vdots & \vdots \\ F_1^* & F_2^* & F_3^* \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$F_1^* = \sum_{i=1}^{n} a_i \, F_i$$



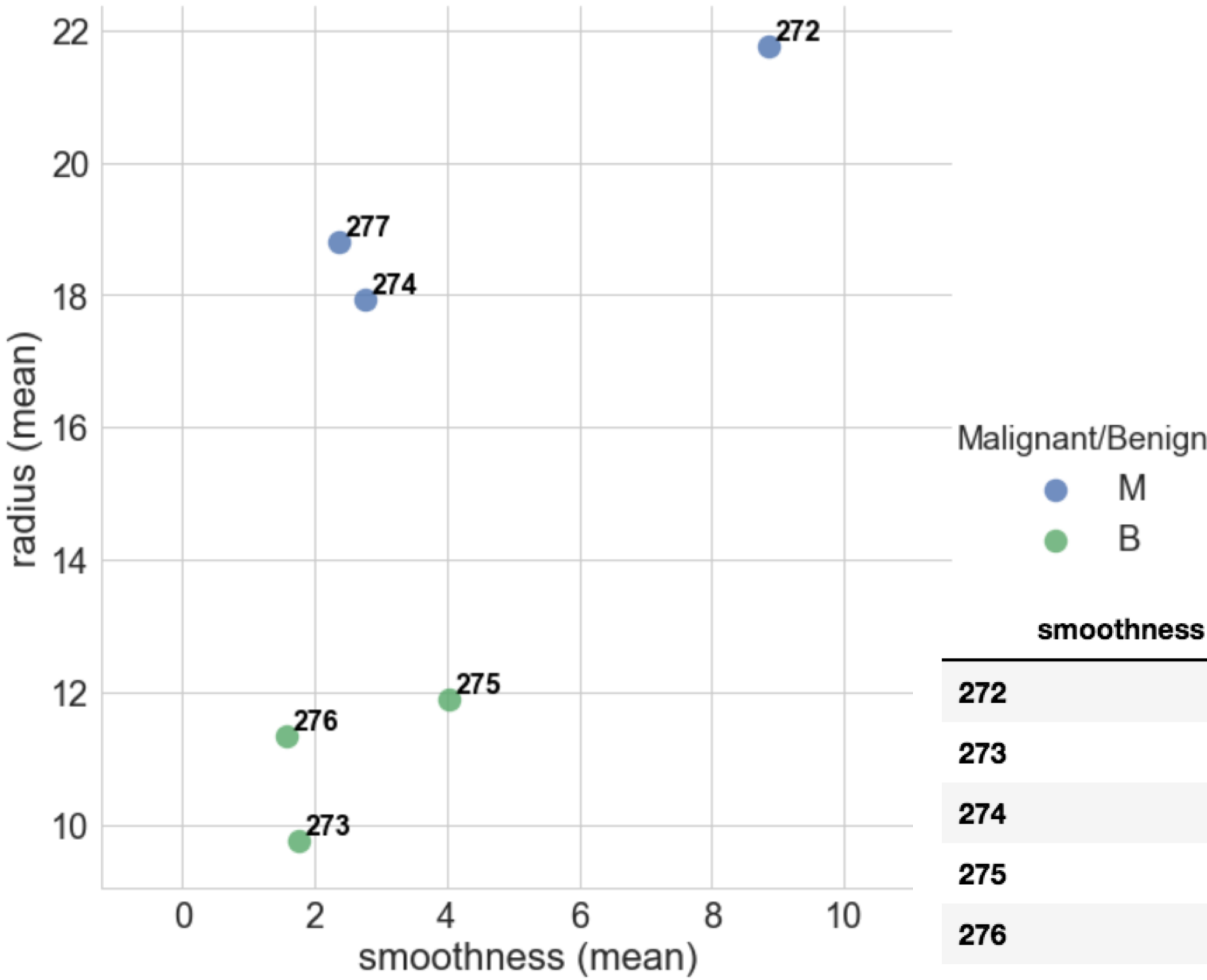
PROS: less variables containing information of all features

CONS: the new features no longer have a “meaningful” interpretation (here a characteristic of a tumor cell)

Principal component analysis

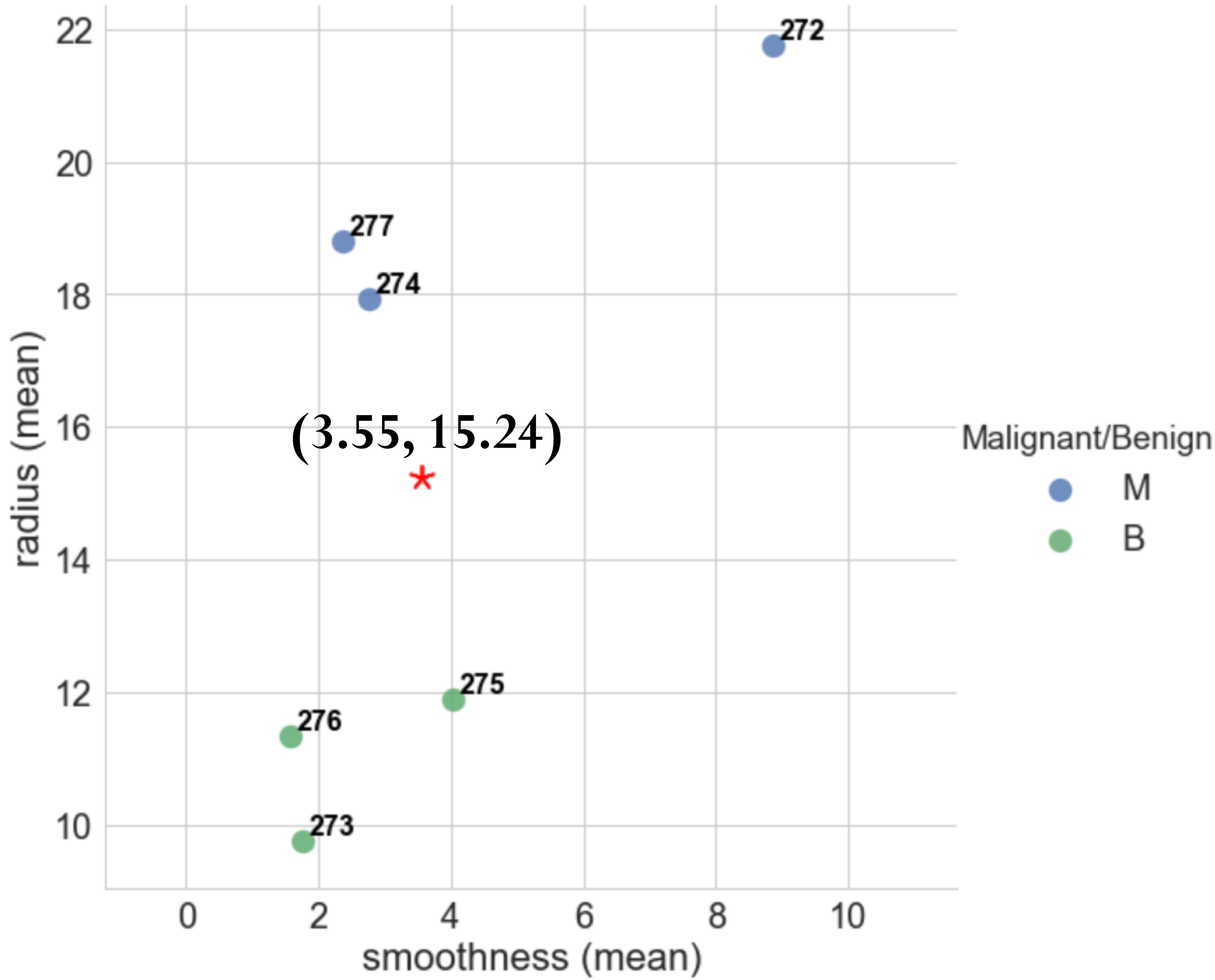
- PCA will combine the feature variables in a specific way, creating “new variables”.
- We can now drop the “least important” new variables while still retaining the most valuable parts of all of the feature variables!
- As an added benefit, each of the “new variables” after PCA are all independent of one another (important requirement for linear models).
- Cons: the new variables don’t have the same meaning as the feature variables (loss of interpretability)

Let’s start with a subset of 6 patients, and take a look at only two of the features: smoothness and radius

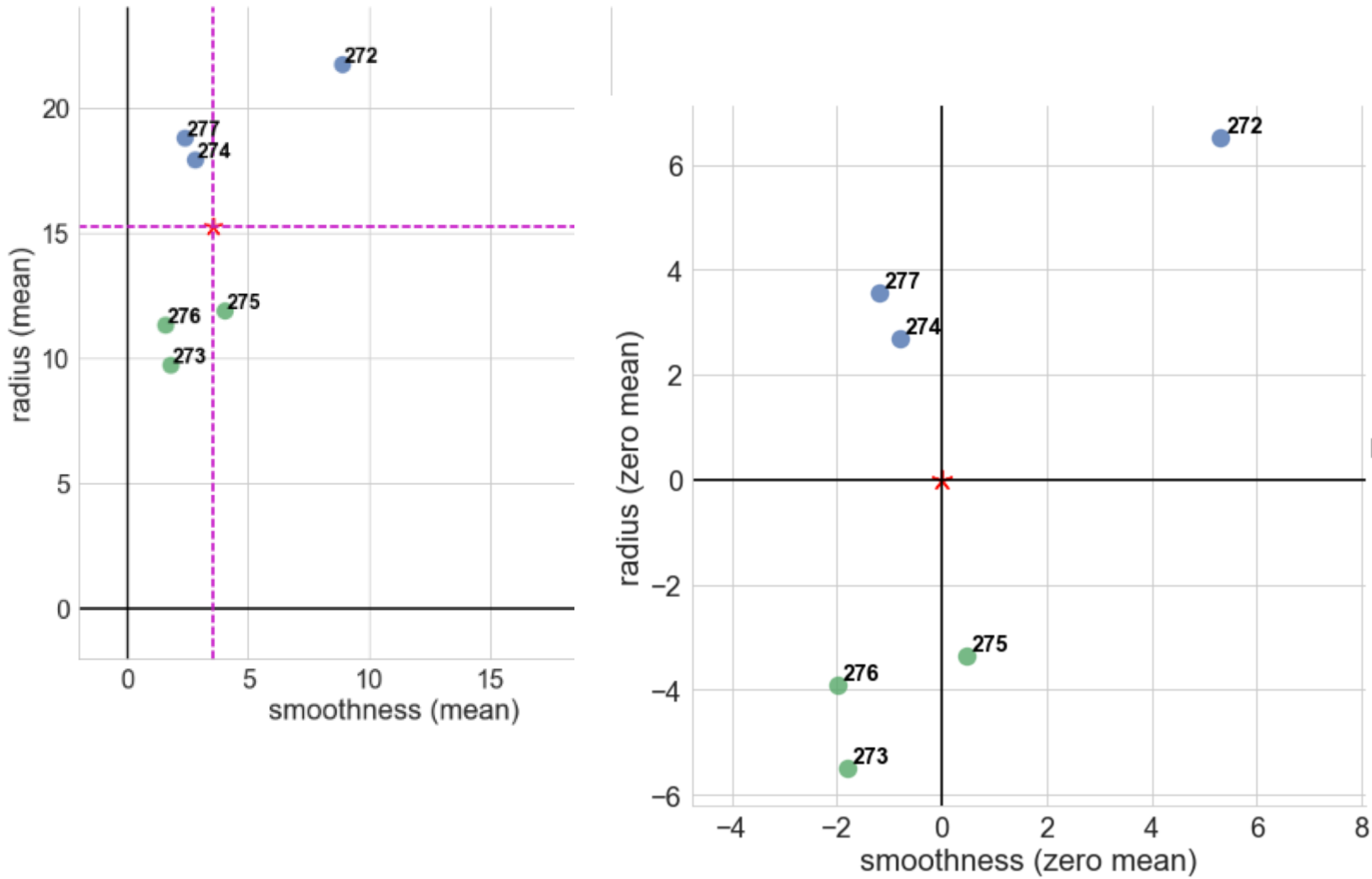


	smoothness (mean)	radius (mean)
272	8.867	21.750
273	1.750	9.742
274	2.765	17.930
275	4.021	11.890
276	1.565	11.330
277	2.363	18.810

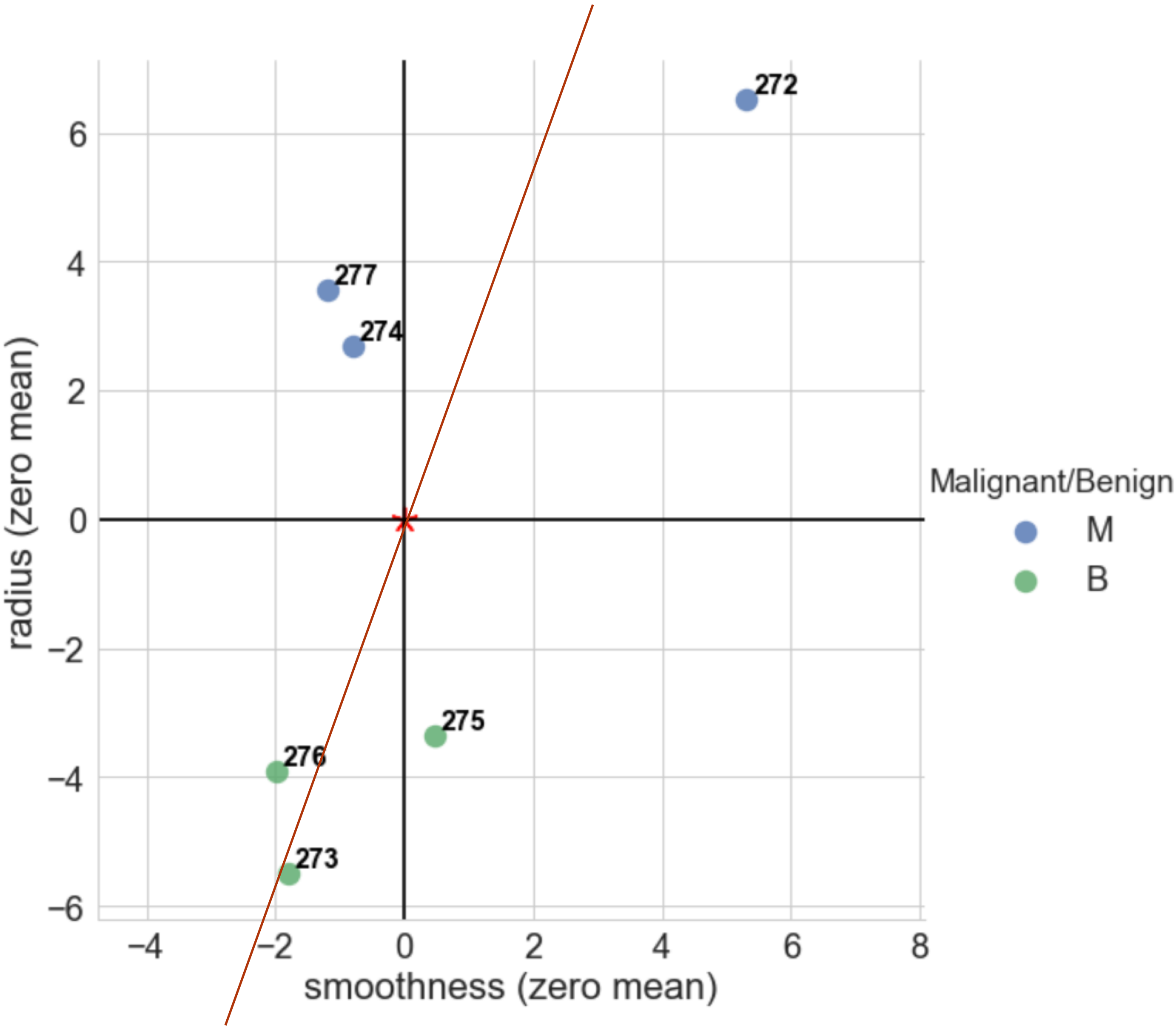
Determine the “center” of the dataset – the mean value of each feature



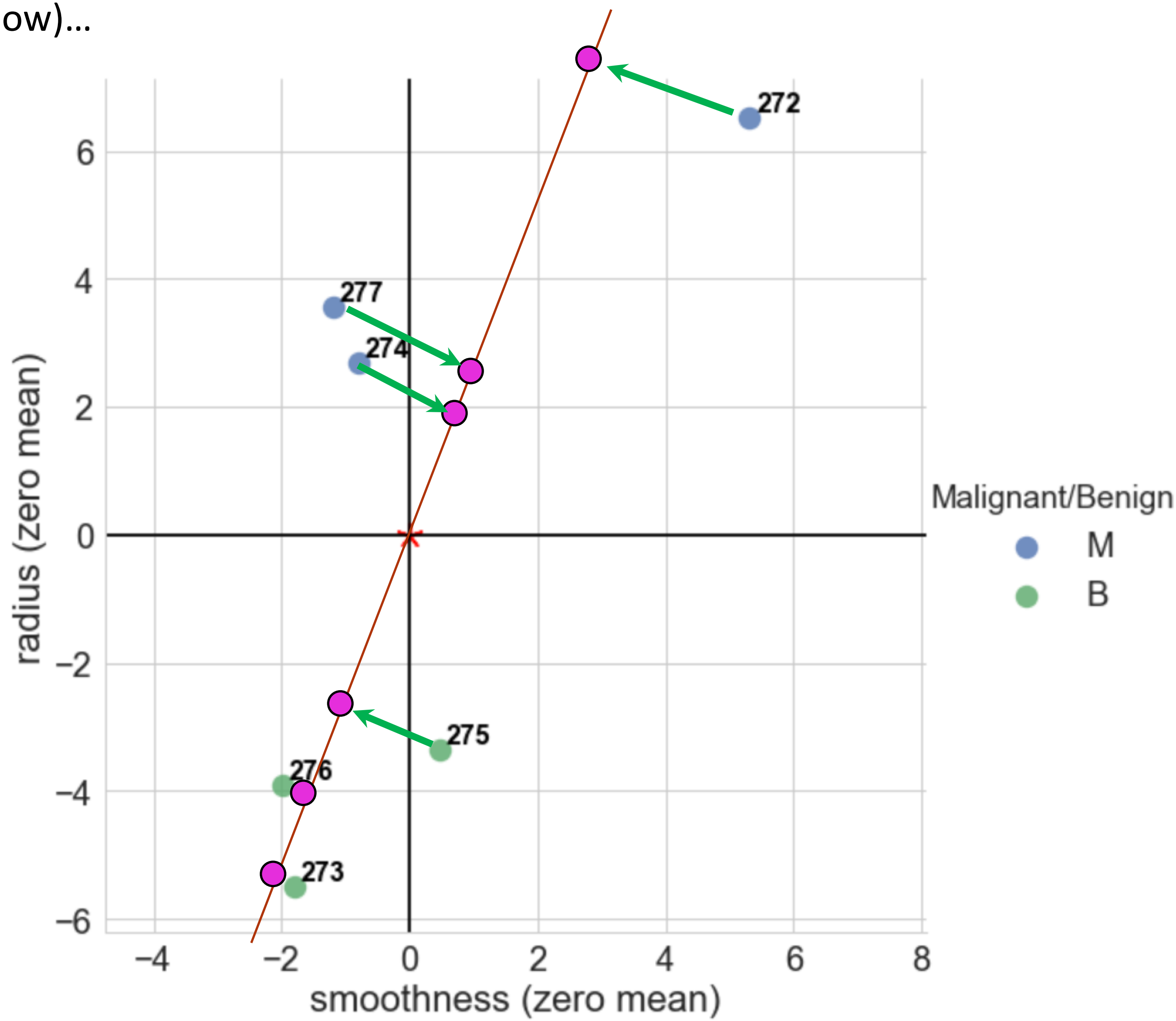
We will shift the dataset such that the “center” of the dataset (mean value) is at the origin (0,0) – the new dataset has zero mean value.



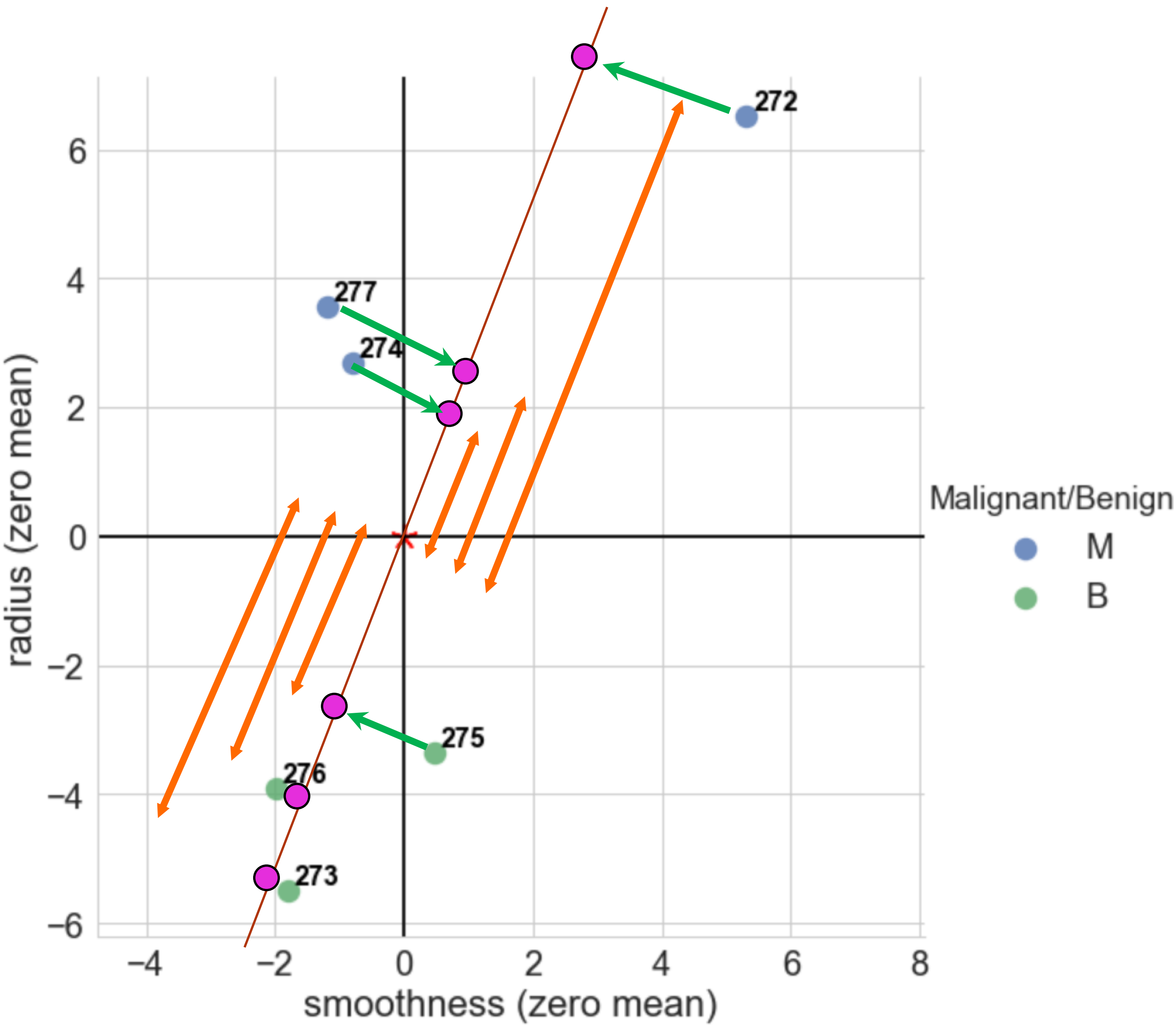
We want to find a straight line that fits the dataset.



Let's propose the red line below. To quantify how good the fit is, PCA projects the data onto the line. The best fit minimizes the distances from the points to the line (indicated in green below)…

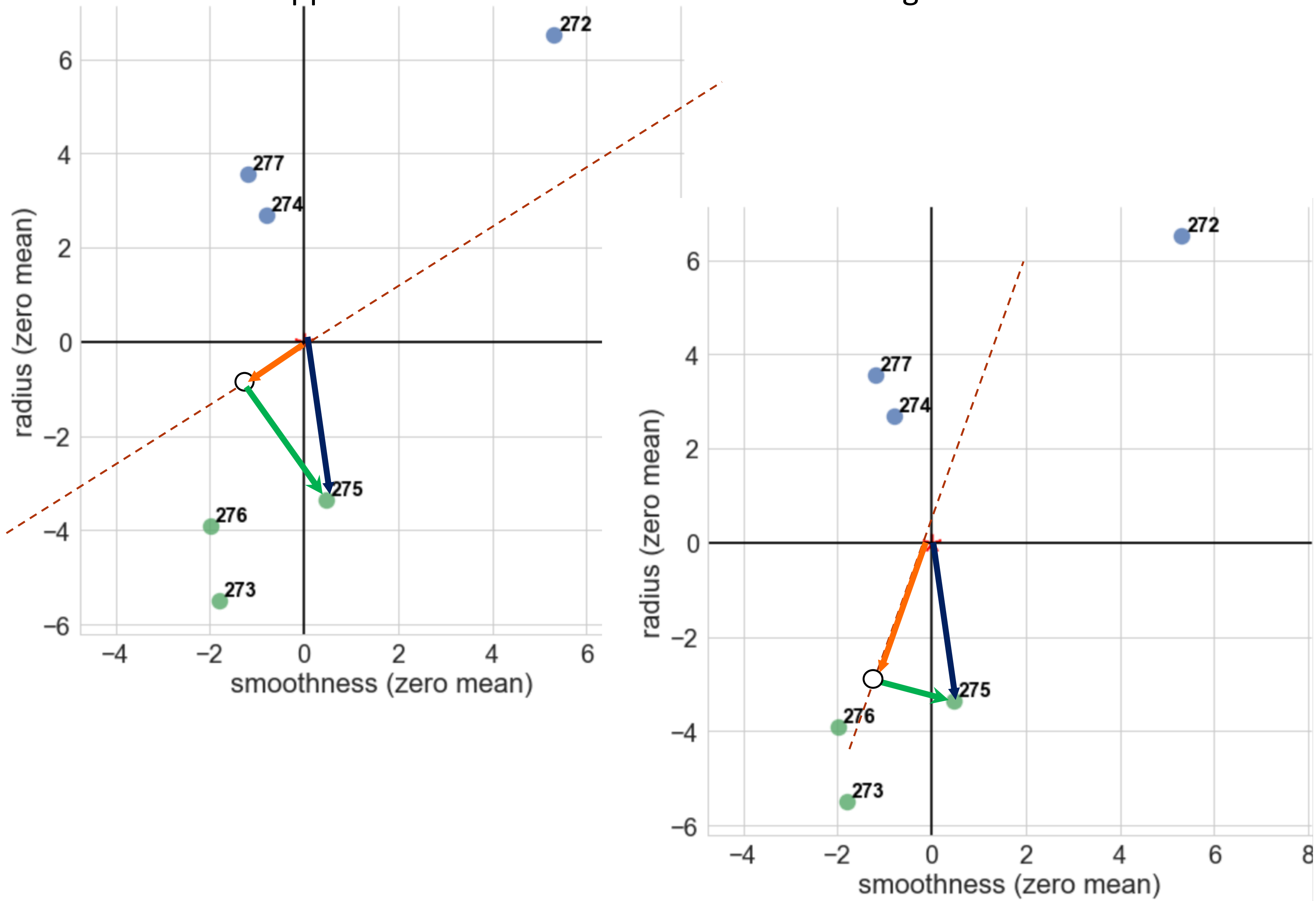


Or maximizes the distances from the projected points to the origin (indicated in orange)

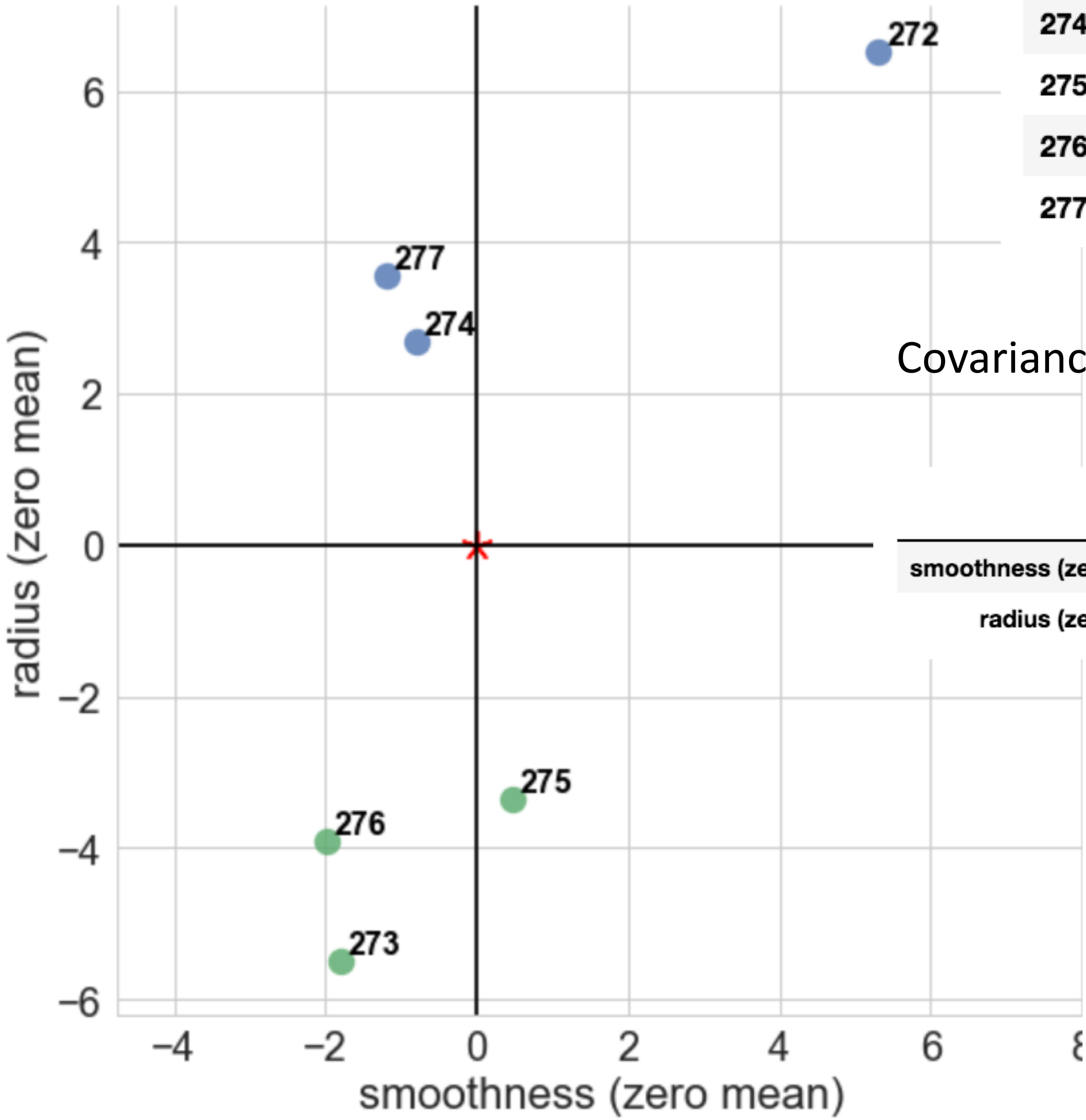


Why are they the same?

Take a look at what happens to the vectors below when we change the fit curve.



Let's talk about the variance of the dataset



$A =$

	smoothness (zero mean)	radius (zero mean)
272	5.311833	6.508
273	-1.805167	-5.500
274	-0.790167	2.688
275	0.465833	-3.352
276	-1.990167	-3.912
277	-1.192167	3.568

Covariance matrix: $\frac{1}{(n-1)} A^T A$

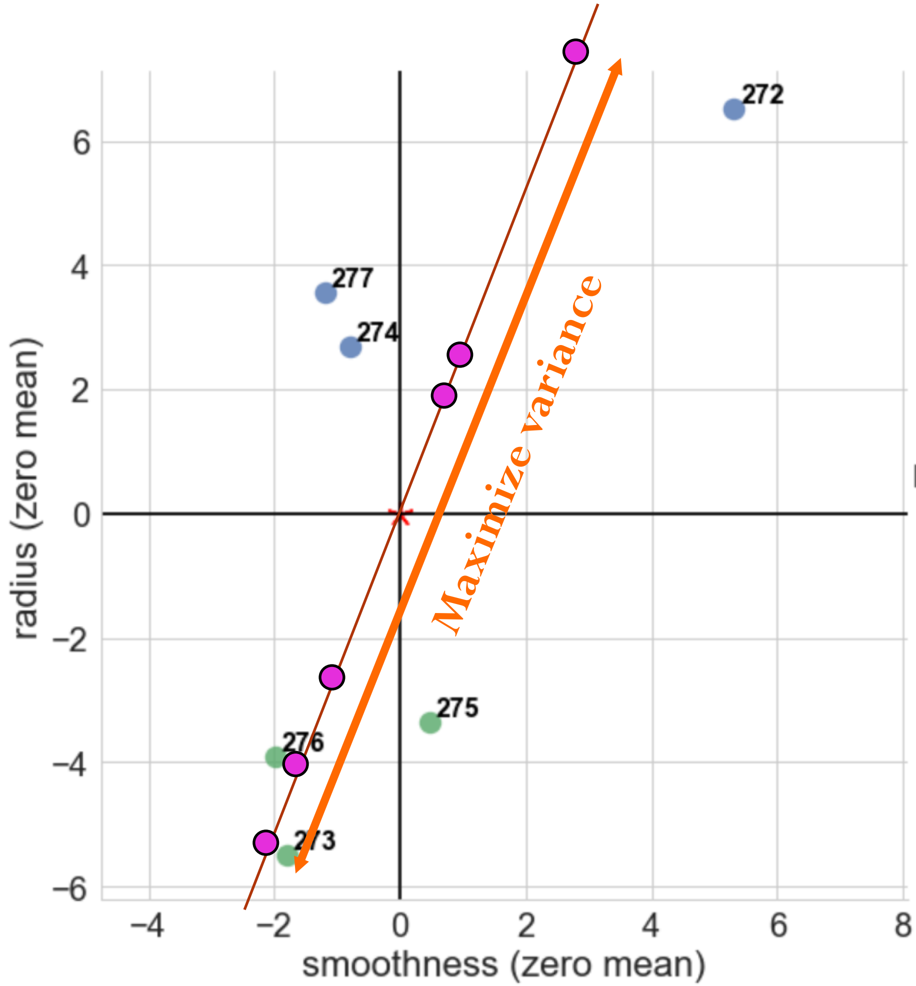
	smoothness (zero mean)	radius (zero mean)
smoothness (zero mean)	7.539518	8.868854
radius (zero mean)	8.868854	23.819936

$A =$

	smoothness (zero mean)	radius (zero mean)
272	5.311833	6.508
273	-1.805167	-5.500
274	-0.790167	2.688
275	0.465833	-3.352
276	-1.990167	-3.912
277	-1.192167	3.568

Covariance matrix: $\frac{1}{(n-1)} A^T A$

	smoothness (zero mean)	radius (zero mean)
smoothness (zero mean)	7.539518	8.868854
radius (zero mean)	8.868854	23.819936



Diagonalization of covariance matrix:

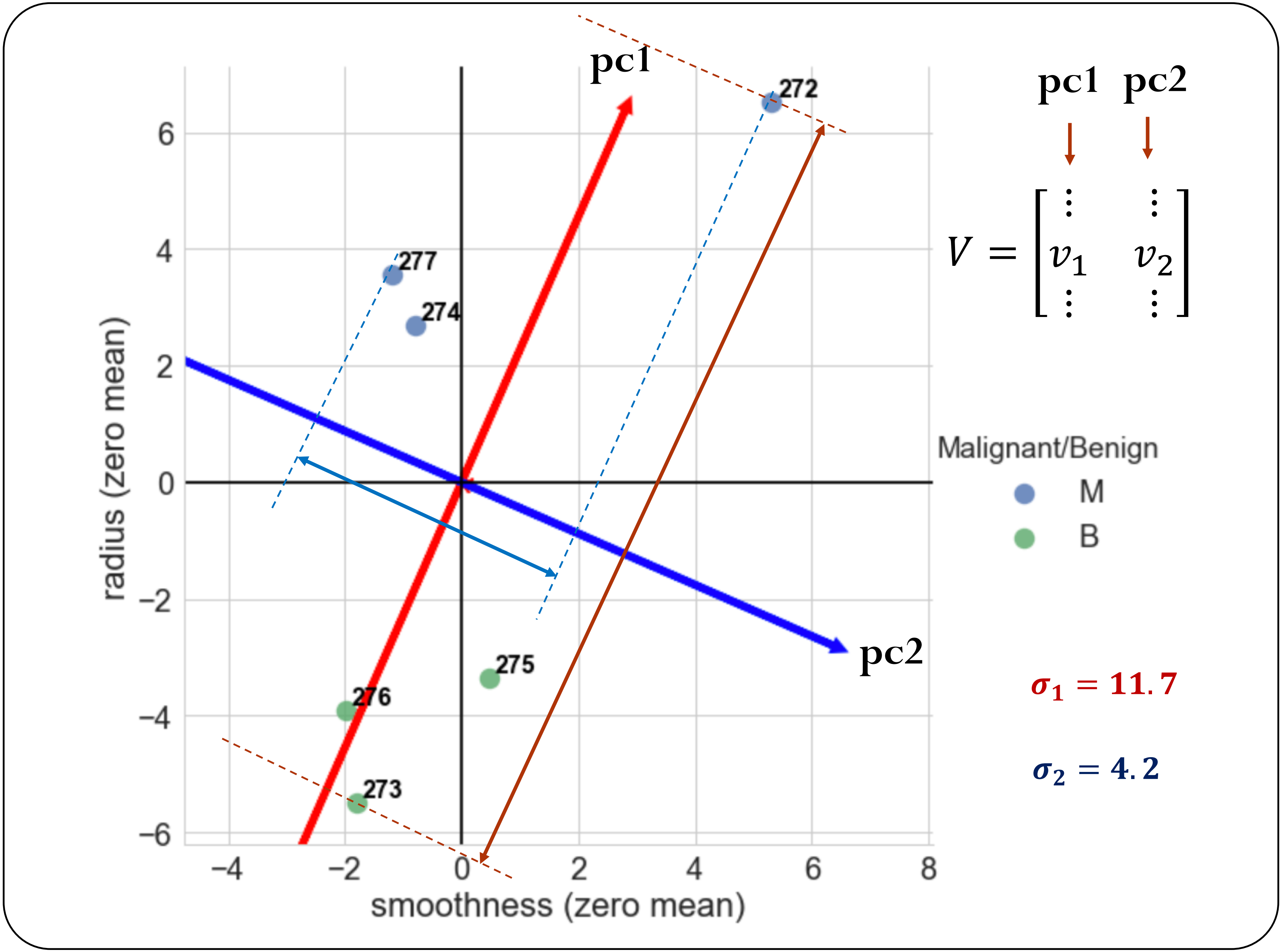
$$A^T A = X D X^T$$

X: eigenvectors of $A^T A$
D: eigenvalues of $A^T A$

From SVD: $A = U \Sigma V^T$

Maximum variance:
largest singular value of Σ

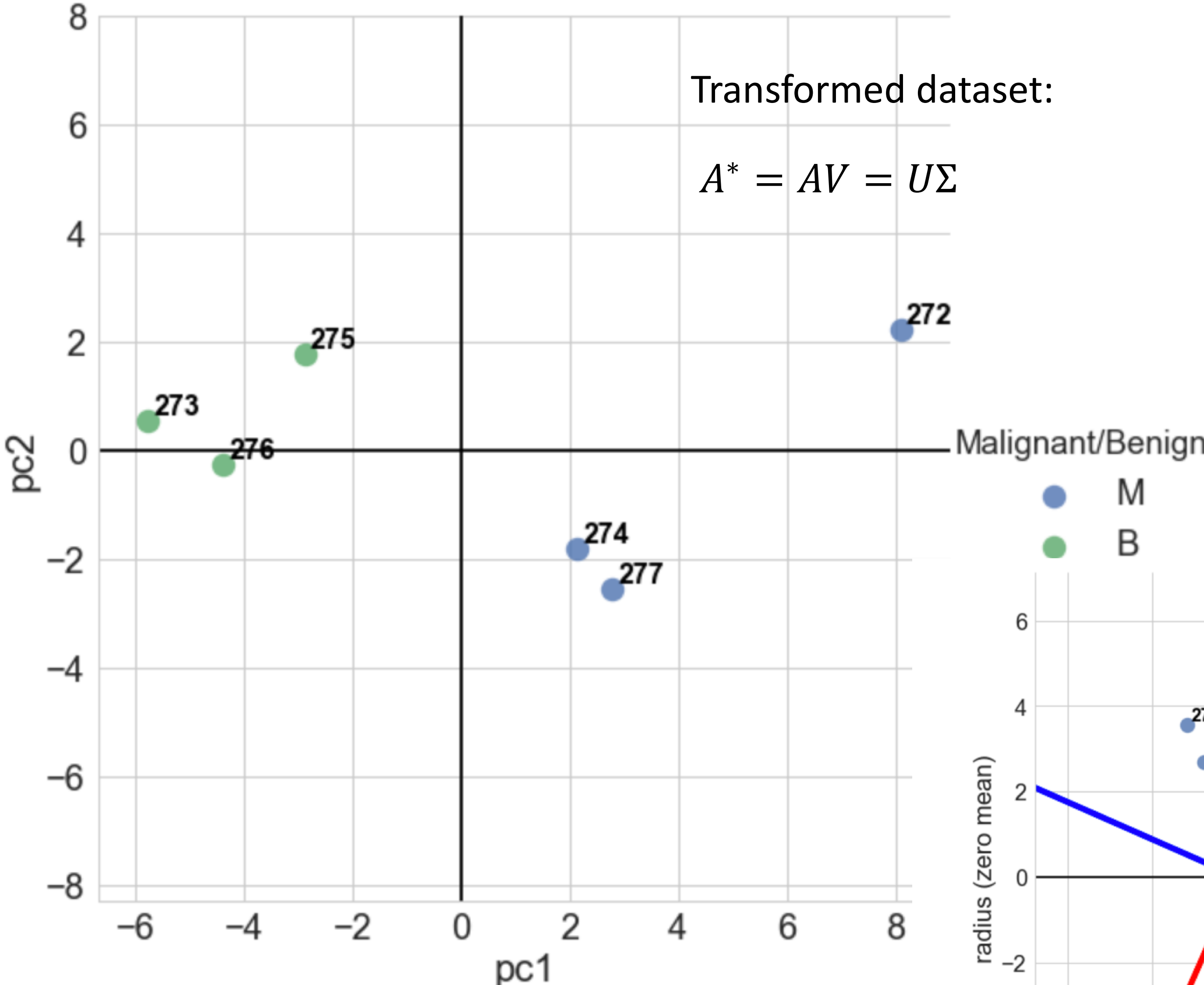
Direction of maximum variance:
Corresponding column of V

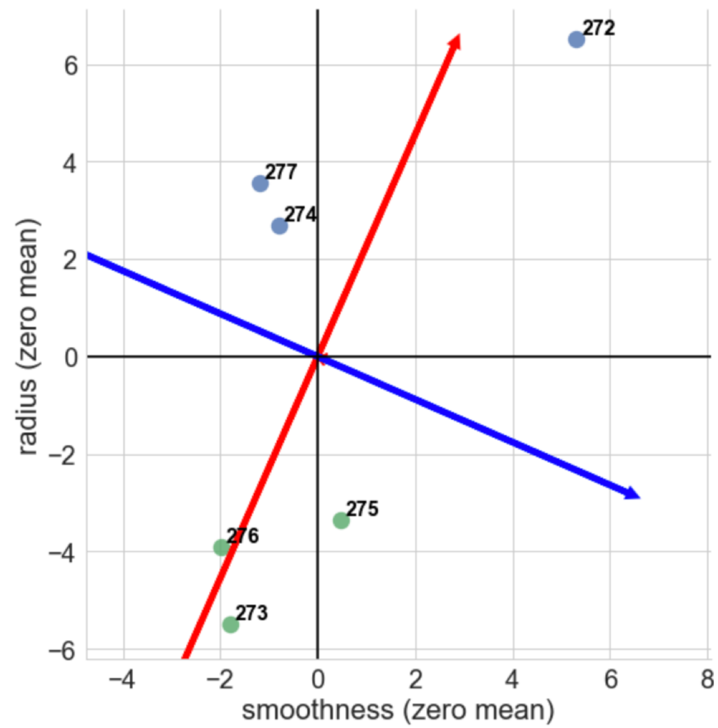


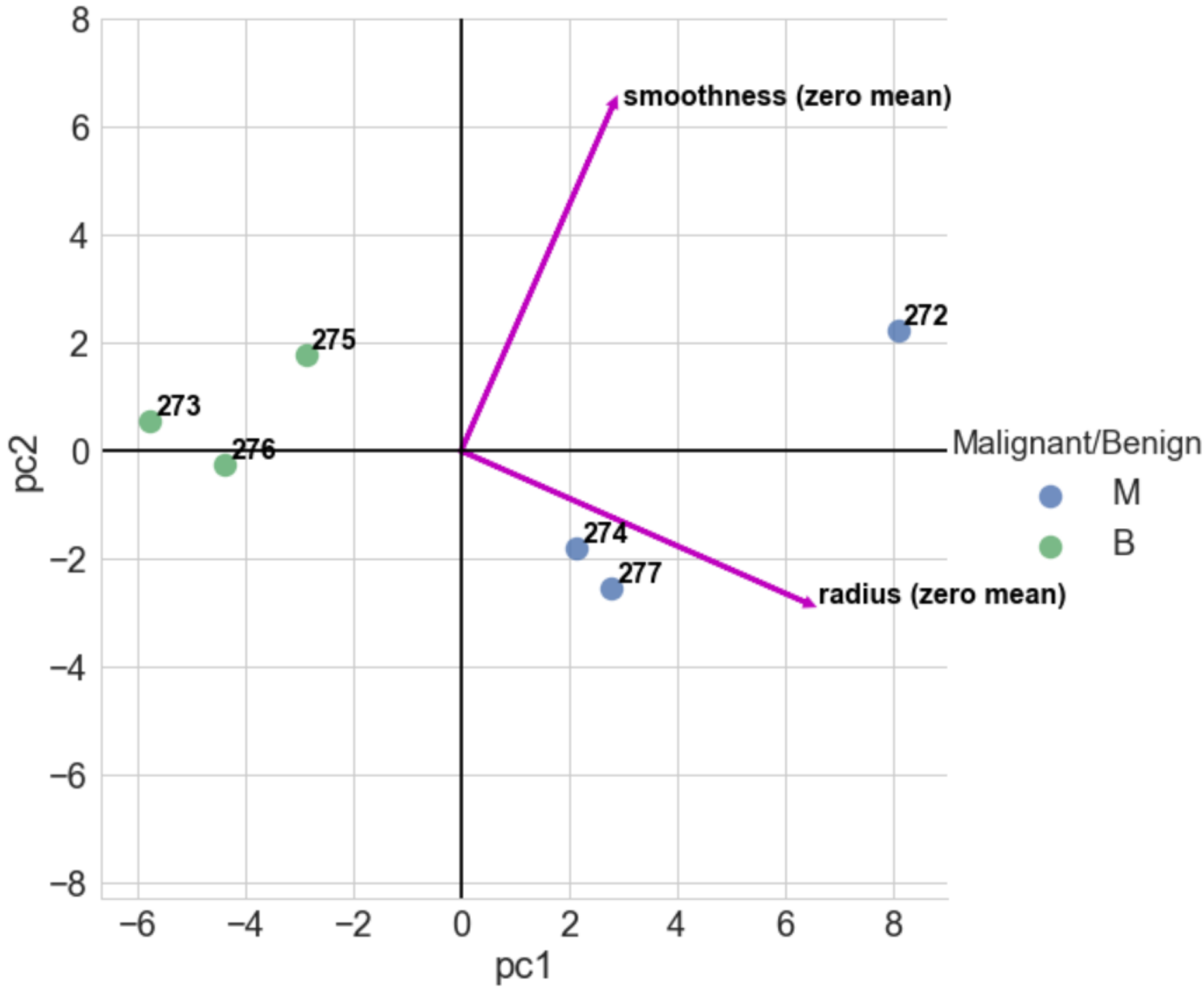
pc1
pc2
pc1 pc2
$V = \begin{bmatrix} \vdots & \vdots \\ v_1 & v_2 \\ \vdots & \vdots \end{bmatrix}$
272
277
274
275
276
273
Malignant/Benign
M
B
$\sigma_1 = 11.7$
$\sigma_2 = 4.2$
radius (zero mean)
smoothness (zero mean)
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6
8

Transformed dataset:

$$A^* = AV = U\Sigma$$





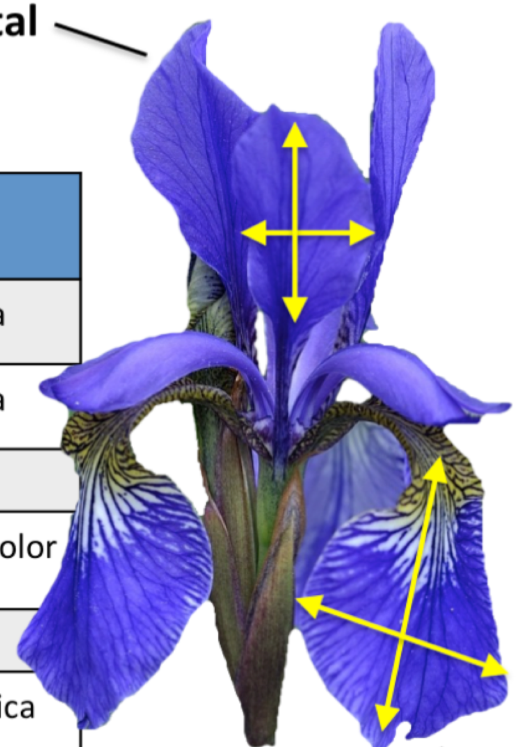

smoothness (zero mean)
radius (zero mean)
272
273
274
275
276
277
Malignant/Benign
M
B
pc1
pc2
8
6
4
2
0
−2
−4
−6
−8

Let's add more features! Flower classification

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Petal

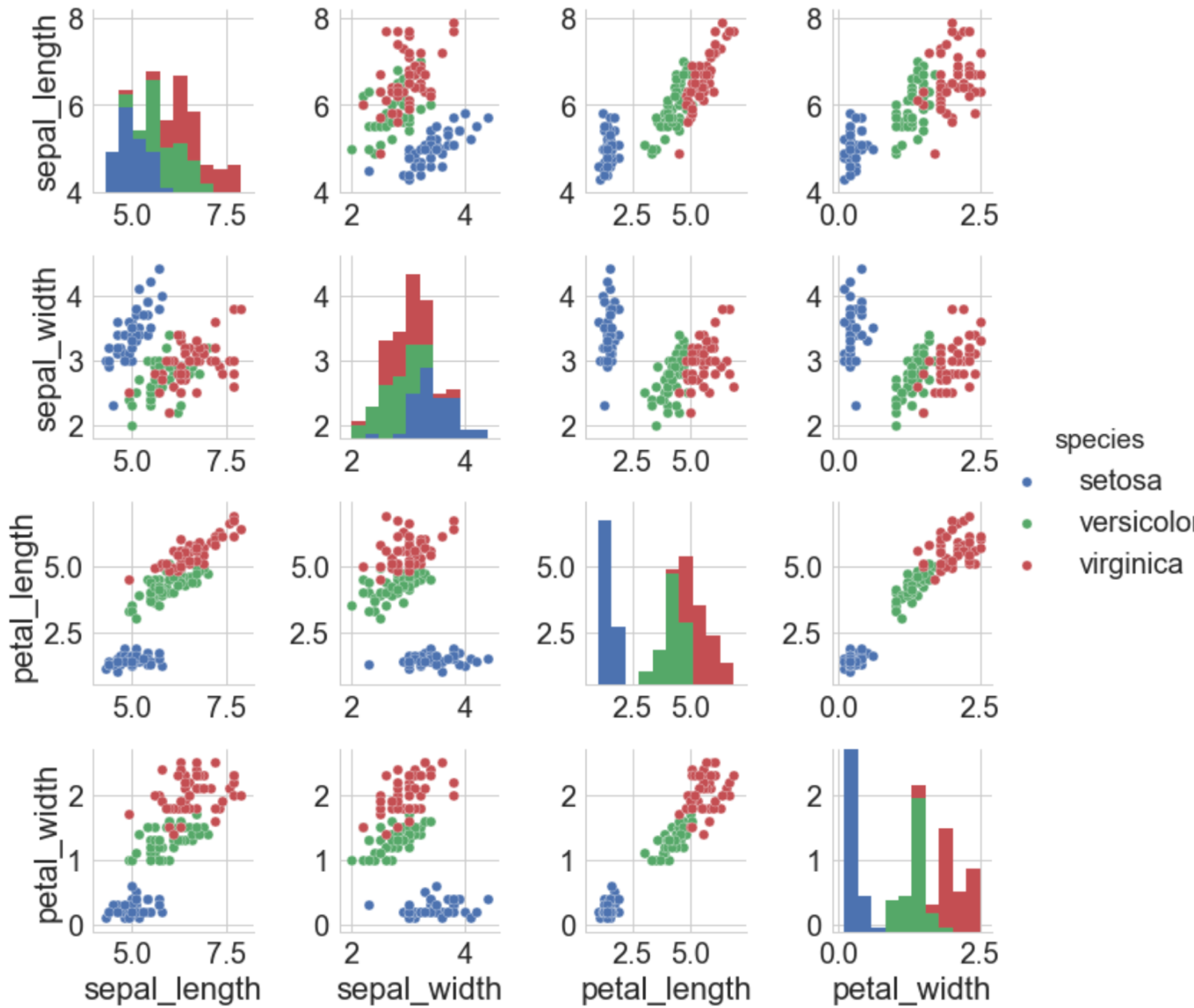


Sepal

Features
(attributes, measurements, dimensions)

Class labels
(targets)

http://sebastianraschka.com/Articles/2015_pca_in_3_steps.html


sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

Principal component analysis

How can we reduce the dimension of a dataset without missing important information?

Detect correlation between variables, if a strong correlation exists, then reducing the dimension of the dataset makes sense.

Overall idea: Find the directions of maximum variance in high-dimensional dataset (n dimension) and project it onto a subspace with smaller dimension (k dimension, with $k < n$), while retaining most of the information.

What is the adequate value for k?

Demo "Features and the SVD"

1) Shift the dataset to zero mean: $A = A - A.mean(\,)$

2) Compute SVD: $A = U\Sigma V^T$

3) Principal components: variances = singular values squared

4) Principal directions: columns of V

5) New dataset: $A^* = A\,V$

Note how the variances of the new dataset correspond to the singular values squared of the original dataset:

$$(A^*)^T A = V^T A^T A\,V = V^T (U\Sigma V^T)^T U\Sigma V^T V = \Sigma^T \Sigma$$

6) In general: $A^* = A\,V$

$m \times n$ $\quad$ $m \times n$ $\quad$ $n \times n$

7) But since we want to reduce the dimension of the dataset, we only use the first k columns of V

$A^* = A\,V$

$m \times k$ $\quad$ $m \times n$ $\quad$ $n \times k$

Iris dataset

```
iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

1) Shift the dataset to zero mean:

```
X = iris.iloc[:,:4] - iris.iloc[:,:4].mean()
```

Optional (modeling choice!): decide whether or not to standardize. If you want to standardize, divide each observation in a column by that column's standard deviation.

```
Z = X / iris.iloc[:,:4].std()
```

In this new dataset Z each feature has mean zero and standard deviation 1.

This decision depends on the problem you are solving. If some variables have a large variance and some small, since PCA maximizes the variance, it will weight more the features with large variance. If you want your PCA to be independent of the variance, standardizing the features will do that.

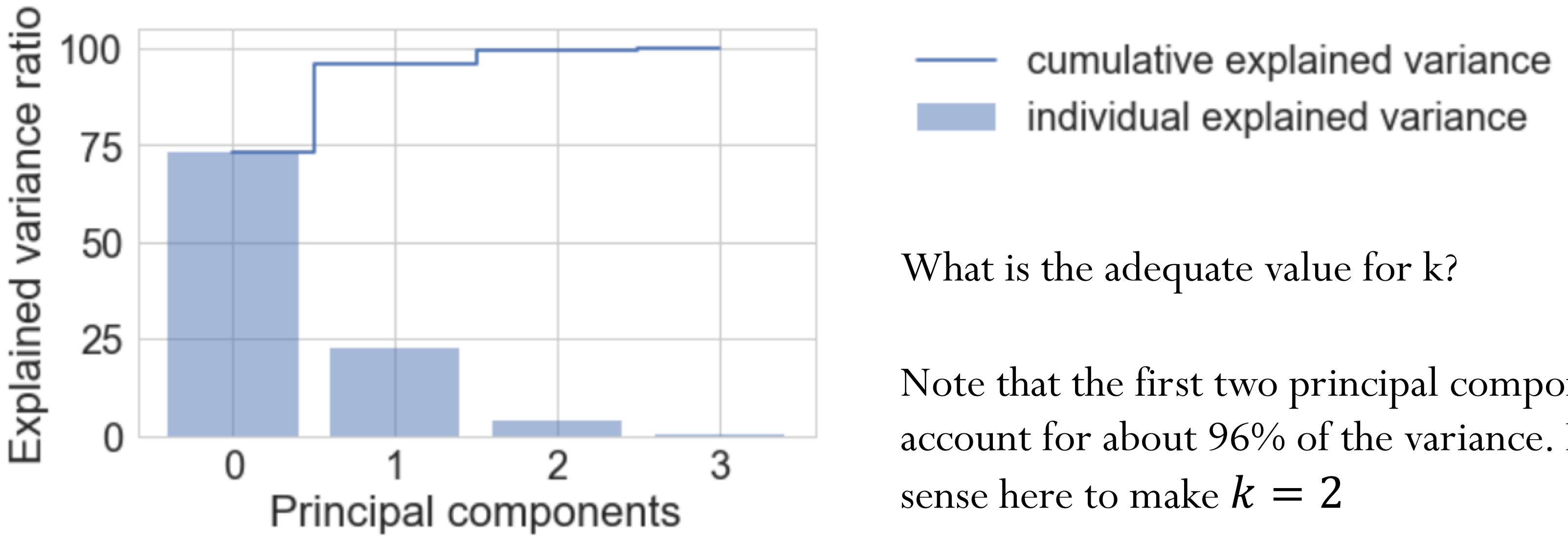
Explained variance

2) Compute SVD: $A = U\Sigma V^T$
3) Principal components: variances = singular values squared

```
U, S, Vt = np.linalg.svd(Z, full_matrices=False)
variances = S**2
print(variances)
```

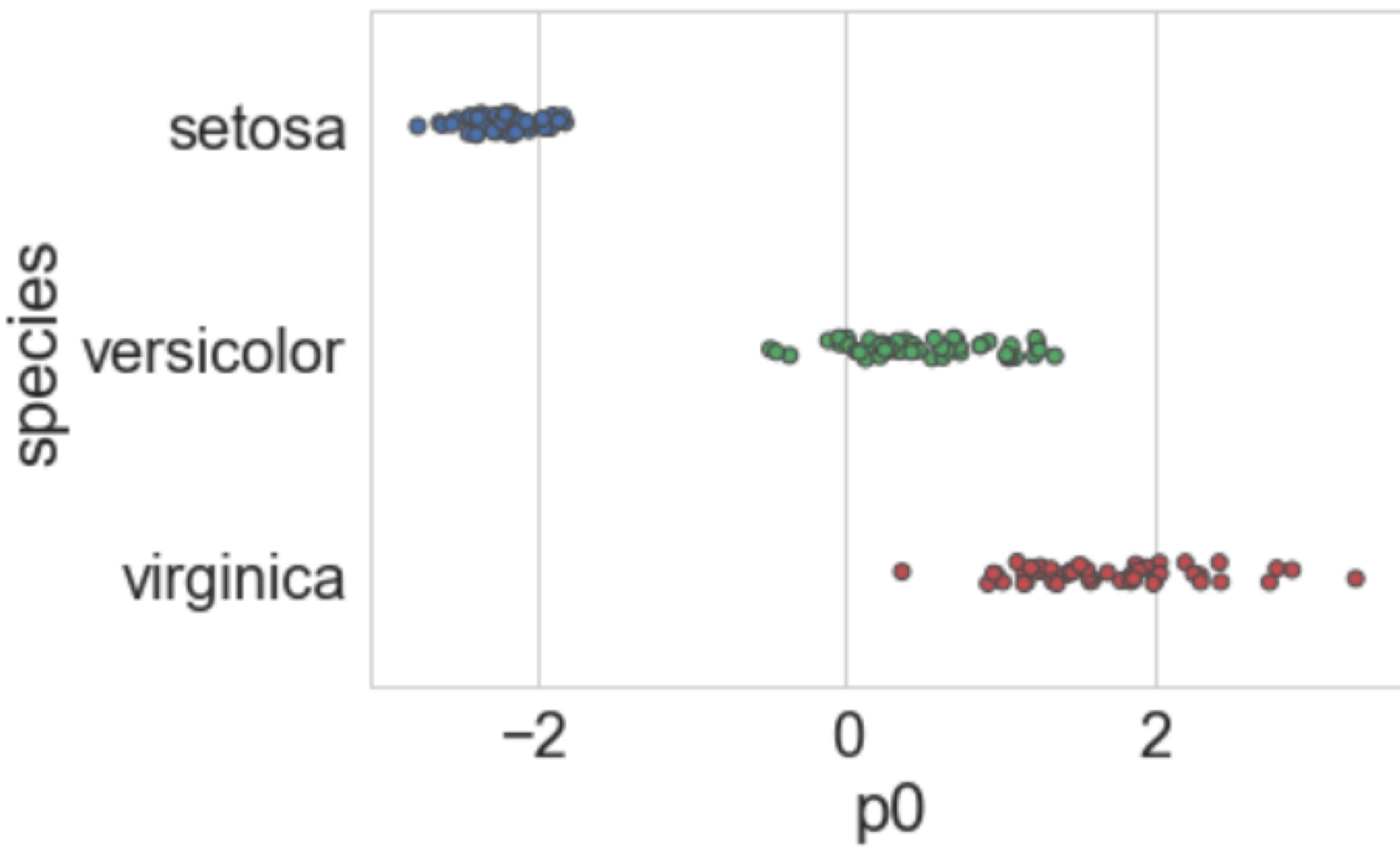
```
[ 434.85617466  136.19054025   21.86677446    3.08651063]
```

Explained variance: $\exp var_i = \frac{variance_i}{sum(variance)}$



What is the adequate value for k?

Note that the first two principal components account for about 96% of the variance. It makes sense here to make $k = 2$

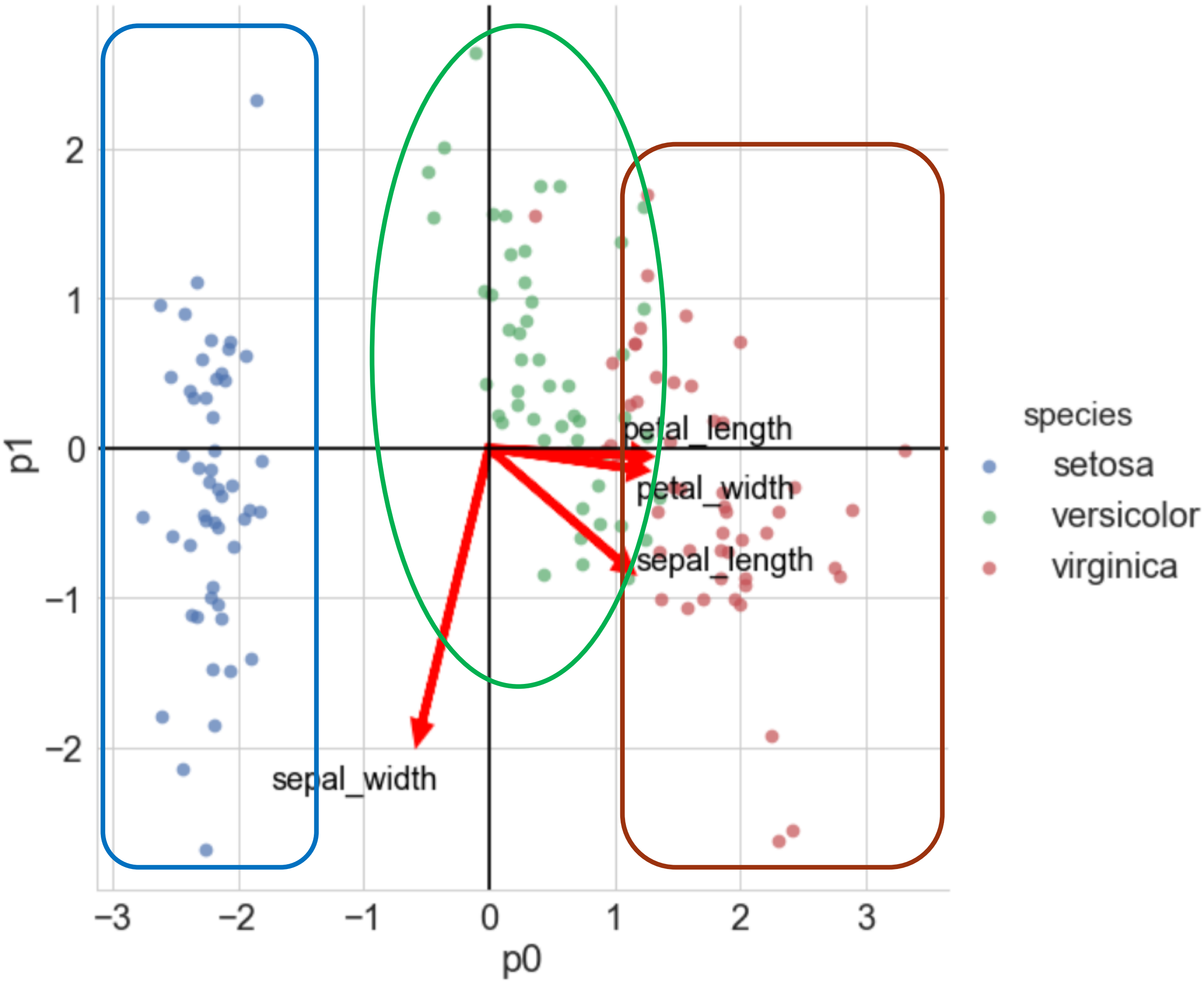


setosa
versicolor
virginica
species
−2
0
2
p0

5) New REDUCED dataset:

```
1  Zstar = Z@V[:,:2]
```

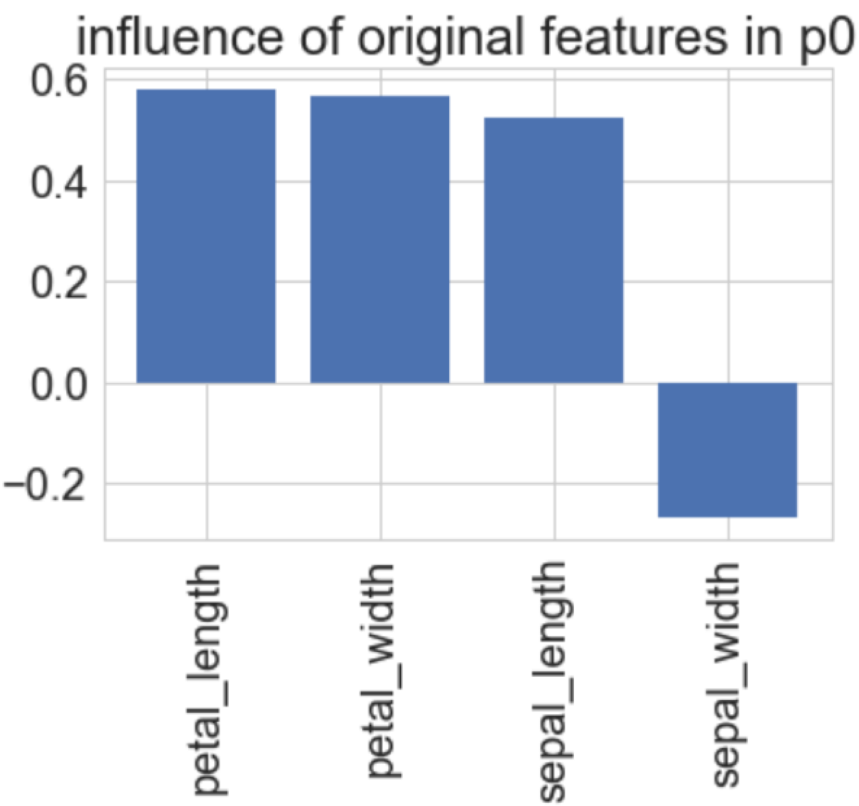
$$Z^* = \begin{bmatrix} \vdots & \vdots \\ p0 & p1 \\ \vdots & \vdots \end{bmatrix}$$

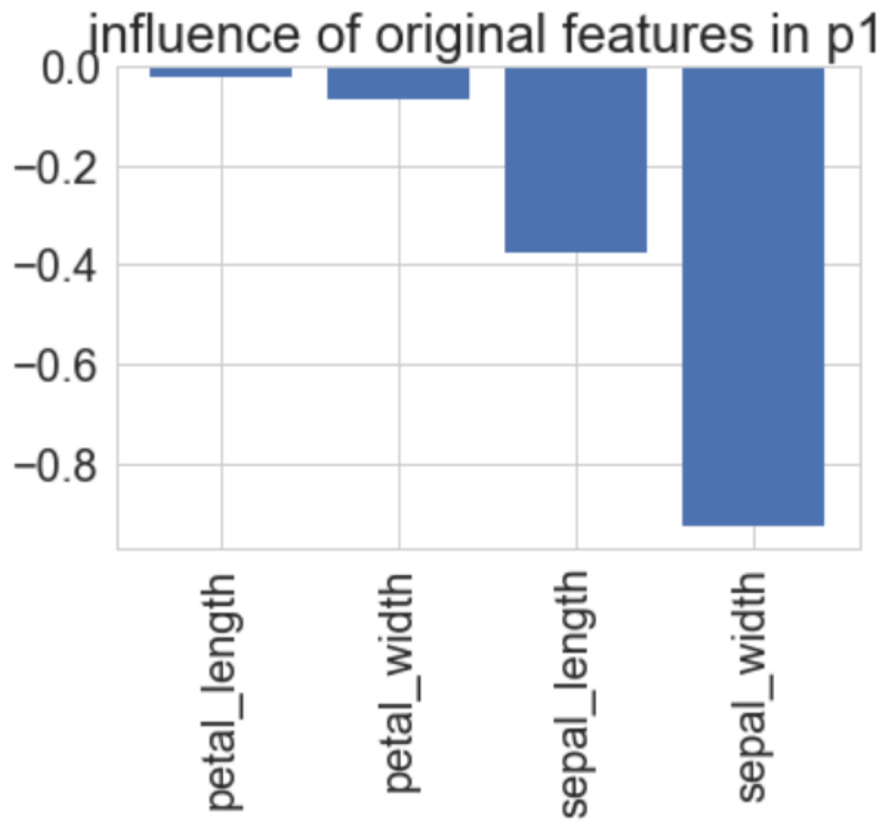


Weight (importance) of each feature in the principal components

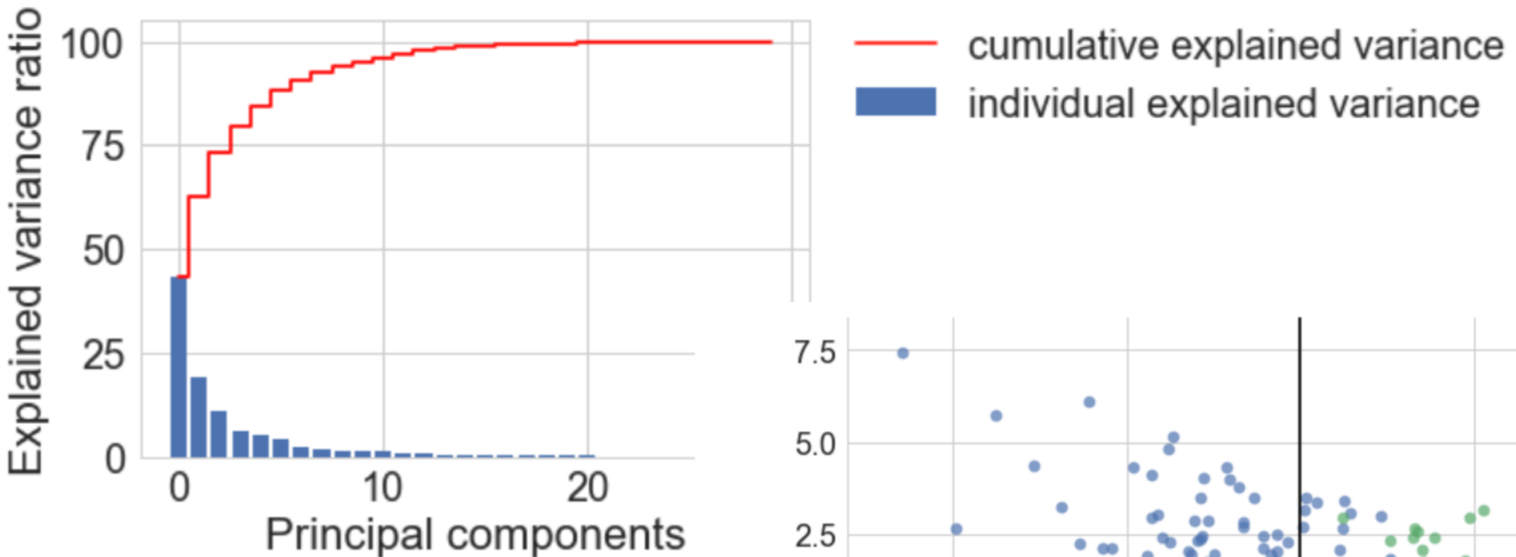
```
plt.figure()
plt.bar(headers[:4],V[:,0])
plt.xticks(rotation=90)
plt.title('influence of original features in p0')
```

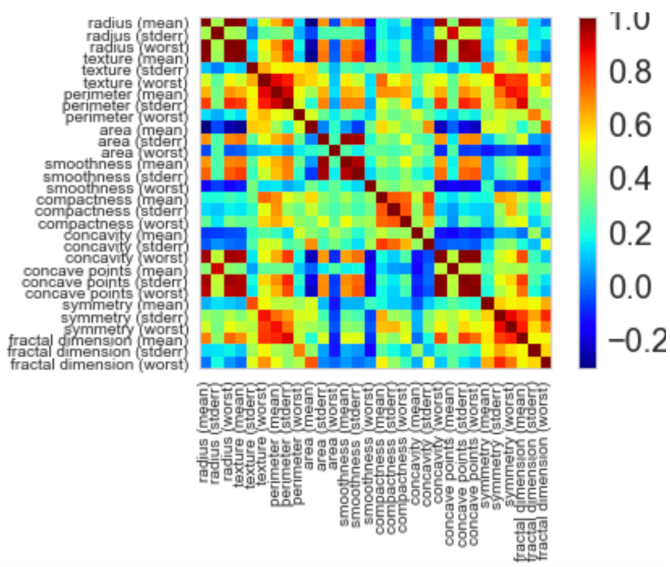
```
plt.figure()
plt.bar(headers[:4],V[:,1])
plt.xticks(rotation=90)
plt.title('influence of original features in p1')
```

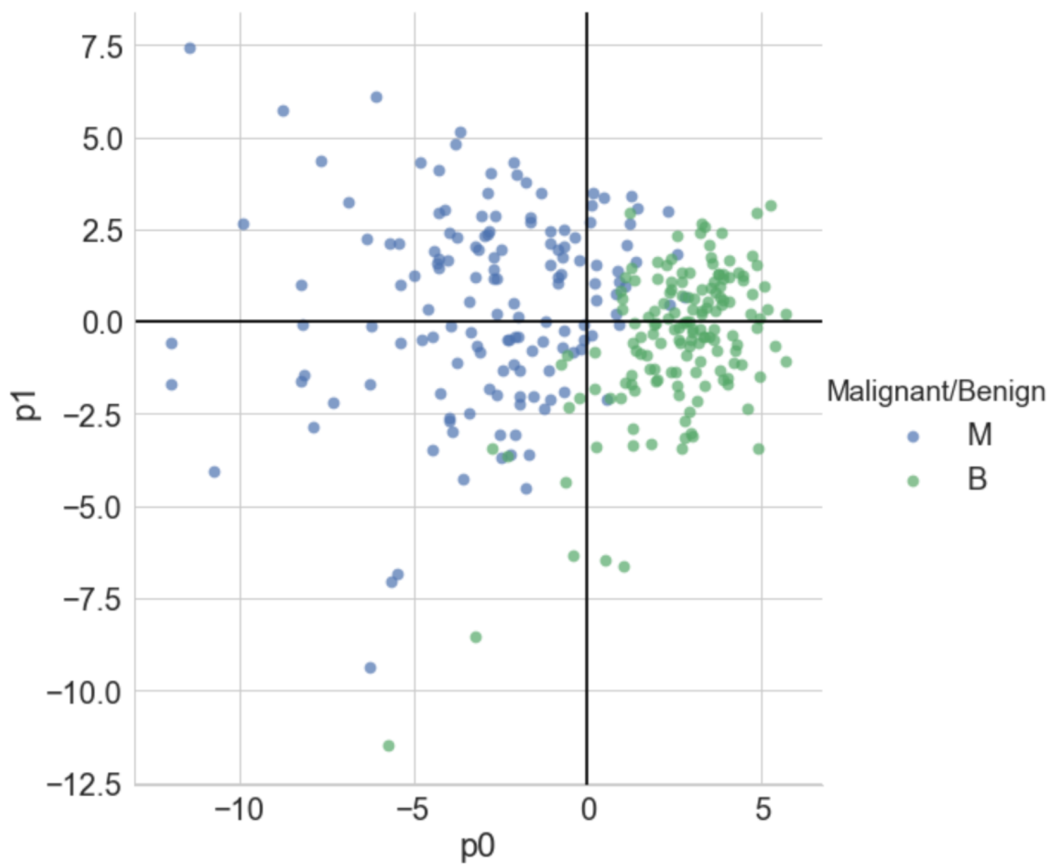


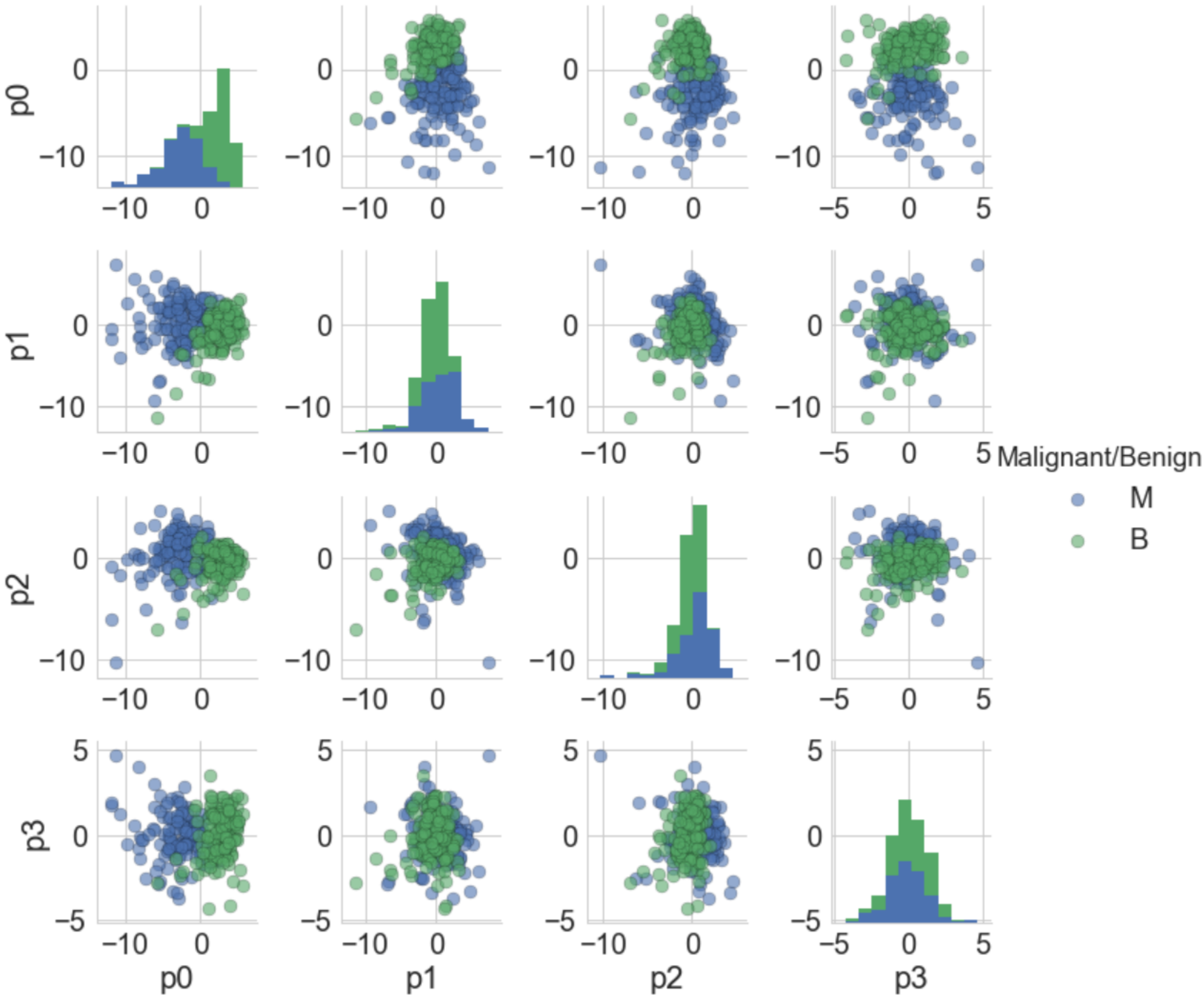


Let's go back to a dataset with many features!

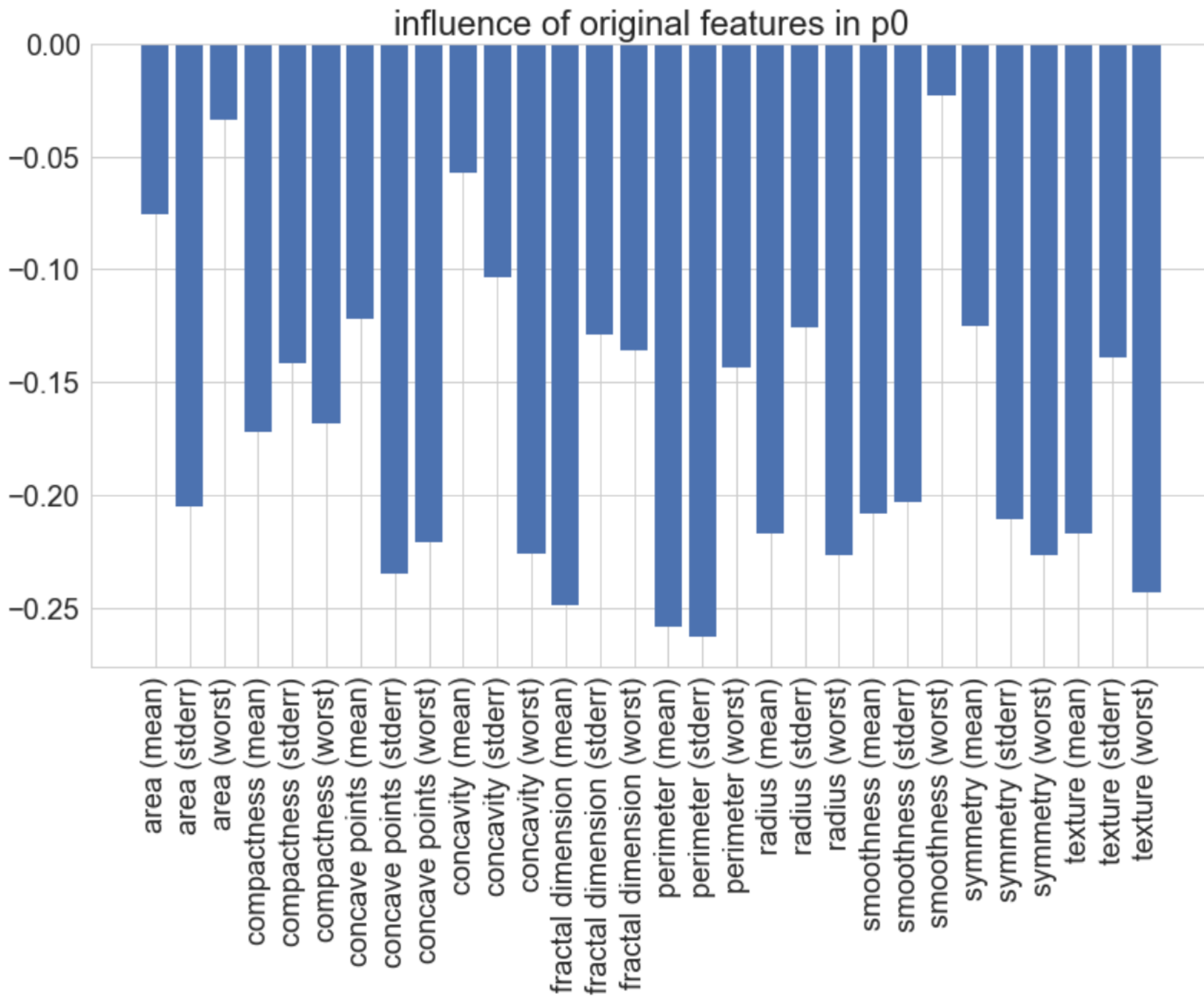








p0
p1
p2
p3
Malignant/Benign
M
B



influence of original features in p0
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
area (mean)
area (stderr)
area (worst)
compactness (mean)
compactness (stderr)
compactness (worst)
concave points (mean)
concave points (stderr)
concave points (worst)
concavity (mean)
concavity (stderr)
concavity (worst)
fractal dimension (mean)
fractal dimension (stderr)
fractal dimension (worst)
perimeter (mean)
perimeter (stderr)
perimeter (worst)
radius (mean)
radius (stderr)
radius (worst)
smoothness (mean)
smoothness (stderr)
smoothness (worst)
symmetry (mean)
symmetry (stderr)
symmetry (worst)
texture (mean)
texture (stderr)
texture (worst)