

Principal Component
Analysis (PCA)
(explained without
the math)

PC2

PC1

WHAT IS PCA?


Principal Component Analysis - or what is often just shortened to **PCA** is an unsupervised learning technique often used in Data Science for **dimensionality reduction**...

...this means it can help us reduce a large set of variables or features down to a smaller set that still contains much of the original information or variance!



REDUCING DIMENSIONS...

For examples sake, let's say our original dataset contained **ten** numeric columns (features)...



Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10
0.83	0.07	0.52	0.63	0.54	0.12	0.57	0.26	0.02	0.43
0.54	0.59	0.16	0.39	0.06	0.30	0.22	0.87	0.34	0.21
0.28	0.56	0.07	0.49	0.51	0.20	0.04	0.59	0.77	0.10
...

PCA could reduce this set of **ten** features down to a smaller number of features, in the example to the right we have reduced this down to **three** new features, each of which is a "**principal component**"



PC 1	PC 2	PC 3
0.71	-0.57	-0.14
0.34	-0.07	0.14
0.01	0.12	-0.06
...

THE NEW COMPONENTS

These newly created features or **principal components** are somewhat abstract...

They are **blend** of some of the original features, where the algorithm found they were **correlated**.

By blending the original variables rather than simply removing them (like we might with feature selection techniques) we hope to **keep much of the key information** that is held within our original feature set.

PC 1	PC 2	PC 3
0.71	-0.57	-0.14
0.34	-0.07	0.14
0.01	0.12	-0.06
...



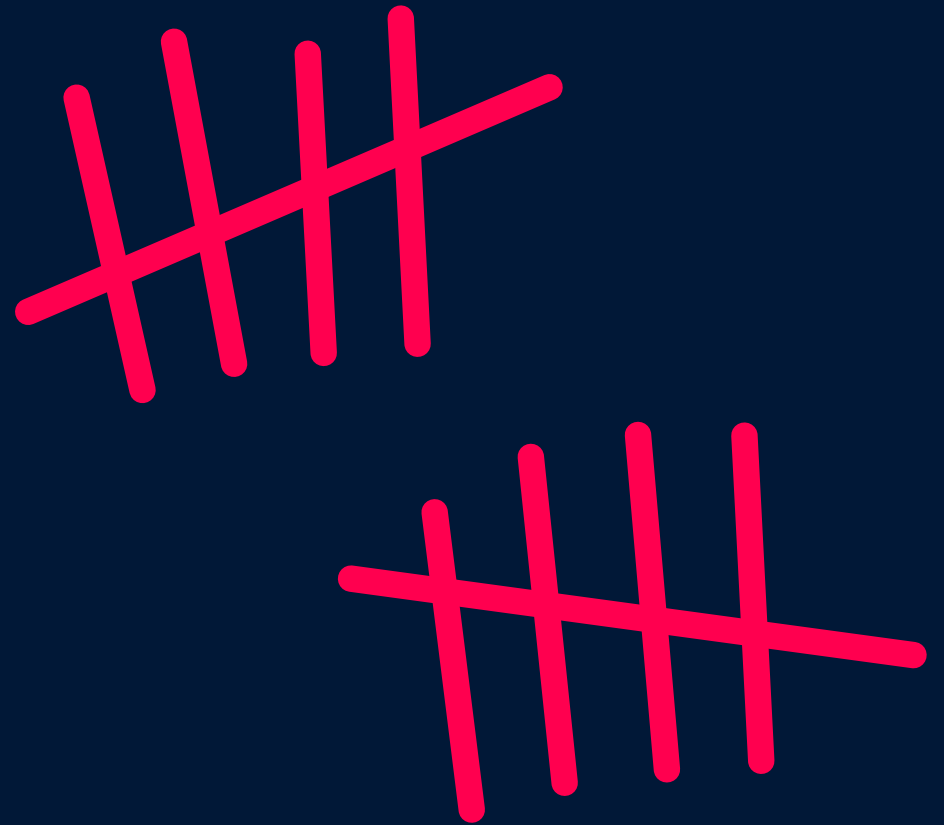
HOW MANY COMPONENTS? [1]

To be completely clear - in our example so far, the PCA algorithm itself **did not** choose to create three, we, the Data Scientist actually pre-specified this number...

Similar to algorithms like k-means, we have to tell the algorithm how many components we want to end up with - otherwise it will just construct a component for every original feature!

...

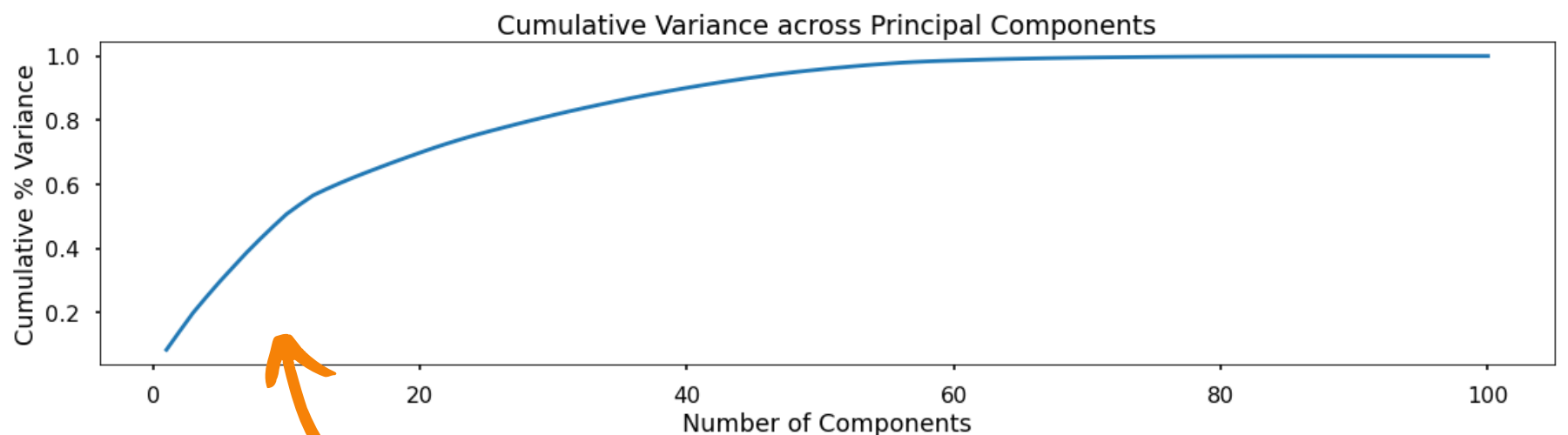
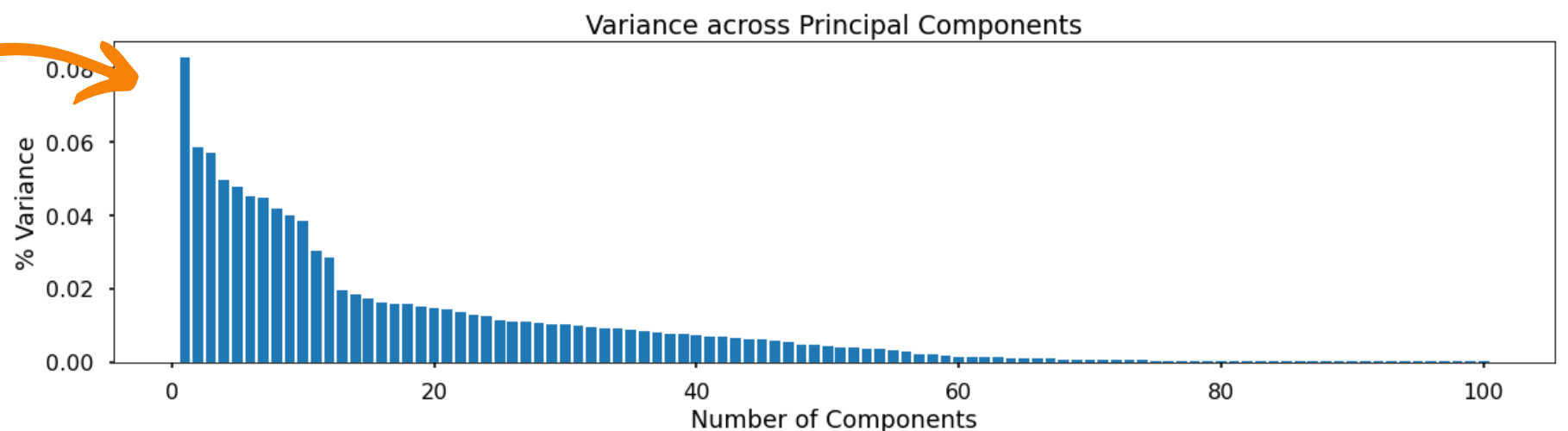
So how do we decide how many components we want or need?



HOW MANY COMPONENTS? [2]

There is no right or wrong answer to this question - we have a **trade-off** on our hands! We need to understand **how much variance from the original feature set** is captured by each additional principal component. Based upon this, we must decide what is best for our task!

Variance explained by each component



Cumulative variance explained by each number of components created (up to 100% with all components)

HOW MANY COMPONENTS? [3]

Perhaps we started with 100 features. If we found that 10 principal components contained 80% of the original variance, this could be a huge benefit in terms of simplifying and reducing our efforts!

It always depends on the particular outcome that we want - and this often requires some trial & error!

If your goal was to reduce your feature set down to two dimensions so it can be easily visualised - then you could just ask for two components to be created. We must be mindful however - how much of the original variance is held within these two new components? We need to ask ourselves if this is fit for purpose!



ALWAYS KEEP IN MIND...

Before applying PCA...



Standardise your original features to ensure they all exist on a comparable scale



Accept that you will **lose** some of the information/variance contained in your original data



Accept that it may become more **difficult to interpret** the outputs of a model using components as inputs vs. the original features

