# Exploring the MNIST dataset

ADVANCED DIMENSIONALITY REDUCTION IN R

**Federico Castanedo**
Data Scientist at DataRobot

# Why do we need dimensionality reduction techniques?

- t-Distributed Stochastic Neighbor Embedding (**t-SNE**)
- Generalized Low Rank Models (**GLRM**)

Advantages of dimensionality reduction techniques:

- Feature selection
- Data compressed into a few important features
- Memory-saving and speeding up of machine learning models
- Visualisation of high dimensional datasets
- Imputing missing data (**GLRM**)

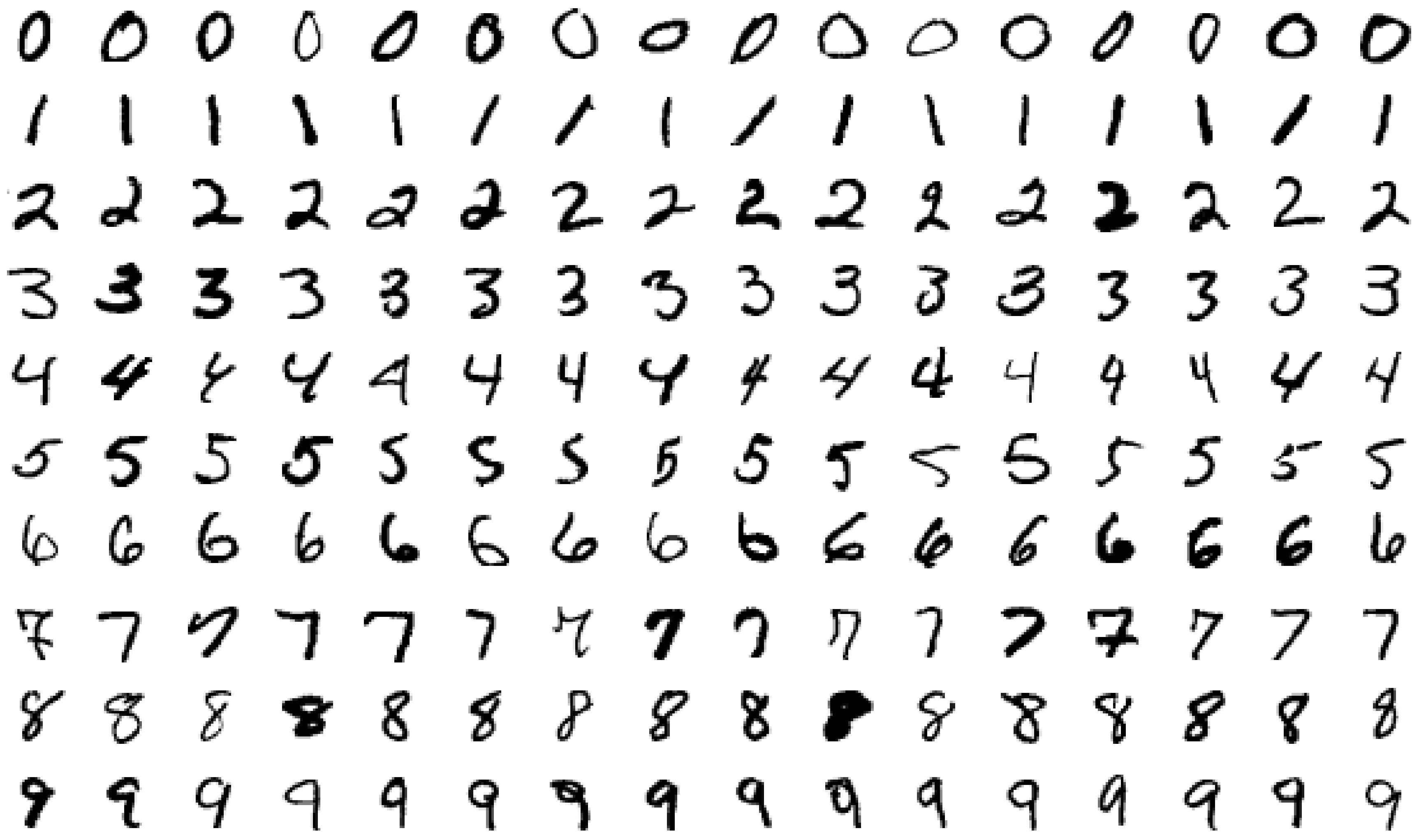
# MNIST dataset

- 70.000 images of handwritten digits (0-9)
- 28x28 pixels

mnistInput

# Several digits

Samples of handwritten digits



# Pixels values

First values

```
head(mnist[, 1:6])
```

```
  label pixel0 pixel1 pixel2 pixel3 pixel4
1     1      0      0      0      0      0
2     0      0      0      0      0      0
3     1      0      0      0      0      0
4     4      0      0      0      0      0
5     0      0      0      0      0      0
6     0      0      0      0      0      0
```

# Pixels values

Values of pixels 400 to 405 for the first record

```
mnist[1, 402:407]
```

```
pixel400 pixel401 pixel402 . pixel403 pixel404 pixel405
1        0        0        0        20       206      254
```

# Pixels statistics

Basic statistics of pixel 408 for digits of label 1

```
summary(mnist[mnist$label==1, 408])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0   253.0   253.0   246.5   254.0   255.0
```

Basic statistics of pixel 408 for digits of label 0

```
summary(mnist[mnist$label==0, 408])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   4.517   0.000 255.000
```

# Let's practice!

ADVANCED DIMENSIONALITY REDUCTION IN R

# Distance metrics

ADVANCED DIMENSIONALITY REDUCTION IN R

**Federico Castanedo**
Data Scientist at DataRobot

# Distance metrics to compute similarity

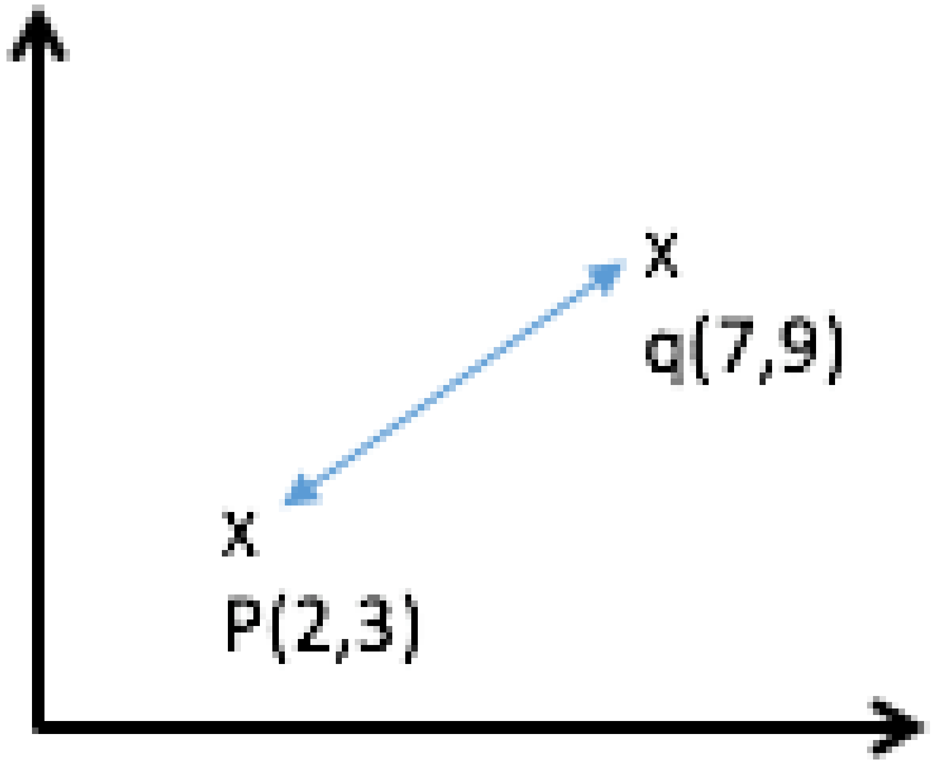
The **similarity** between MNIST digits can be computed using a distance metric.

A metric is a function that for any given points, $x, y, z$ the output satisfies:

1. **Triangle inequality**: $d(x, z) \leq d(x, y) + d(y, z)$
2. **Symmetric property**: $d(x, y) = d(y, x)$
3. **Non-negativity and identity**: $d(x, y) \geq 0$ and $d(x, y) = 0$ only if $x = y$

# Euclidean distance

- Euclidean distance in two dimensions




$$d(p, q) = \sqrt{(2-7)^2 + (3-9)^2} = 7.81025$$

- Can be generalized to *n* dimensions

# Euclidean distance in R

Euclidean distance between the last 6 digits of `mnist_sample`

```
distances <- dist(mnist_sample[195:200 ,-1])
distances
```

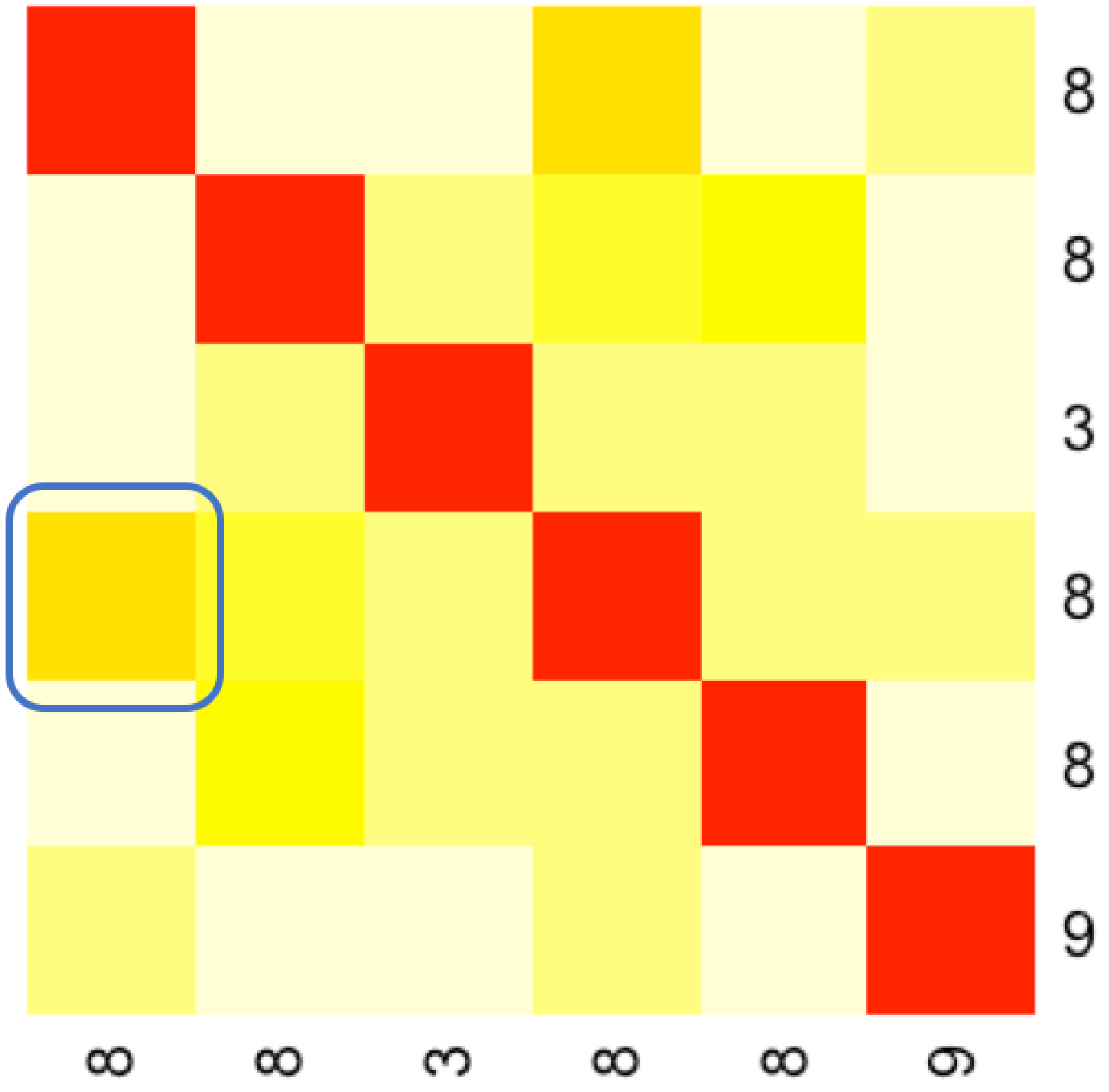
```
         195      196      197      198      199
196 2582.812
197 2549.652 2520.634
198 1823.275 2286.126 2498.119
199 2537.907 2064.515 2317.869 2304.517
200 2362.112 2539.937 2756.149 2379.478 2593.528
```

# Plotting distances

Plot of the distances using `heatmap()`

```
heatmap(as.matrix(distances), Rowv = NA, symm = T,
        labRow = mnist_sample$label[195:200],
        labCol = mnist_sample$label[195:200])
```

# Heatmap of the Euclidean distance



# Minkowski family of distances

- Minkowski: $d = (\sum |P_i - Q_i|^p)^{1/p}$
- Example: Minkowski distance of order 3

```
distances <- dist(mnist_sample[195:200, -1,
                  method = "minkowski", p = 3])
```

# Manhattan distance

- Manhattan distance (Minkowski distance of order 1)

```
distances <- dist(mnist_sample[195:200 ,-1],
                  method = "manhattan")
```

# Kullback-Leibler (KL) divergence

- Not a metric since it does not satisfy the symmetric and triangle inequality properties
- Measures differences in probability distributions
- A divergence of 0 indicates that the two distributions are identical
- A common distance metric in Machine Learning (t-SNE). For example, in decision trees it is called *Information Gain*

# Kullback-Leibler (KL) divergence in R

Load the `philentropy` package and get the last 6 MNIST records

```
library(philentropy)
mnist_6 <- mnist_sample[195:200, -1]
```

Add `1` to all records to avoid `NaN` and compute the totals per row

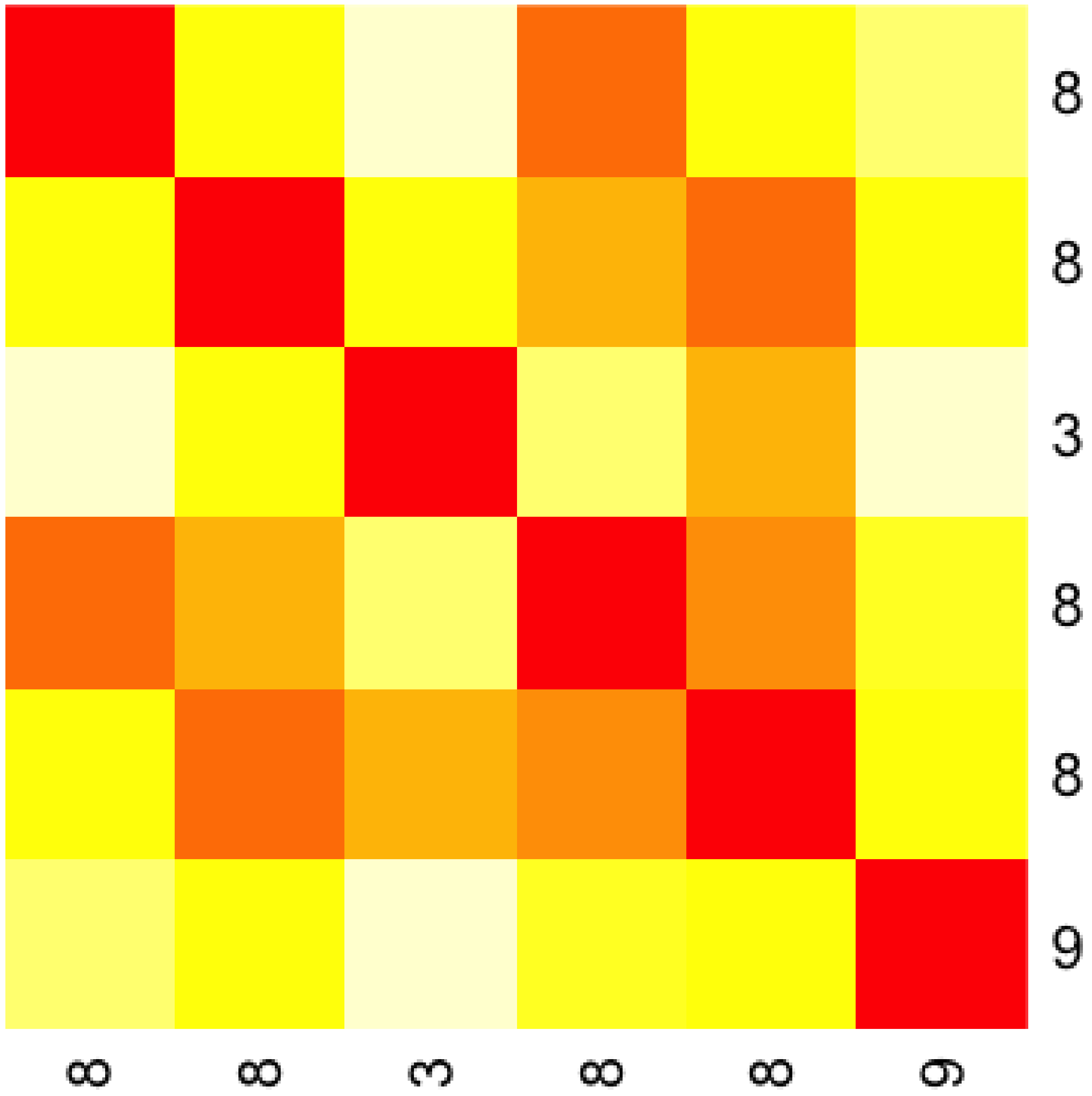
```
mnist_6 <- mnist_6 + 1
sums <- rowSums(mnist_6)
```

Compute the KL divergence

```
distances <- distance(mnist_6/sums, method = "kullback-leibler")

heatmap(as.matrix(distances), Rowv = NA, symm = T,
        labRow = mnist_sample$label,
```

# Heatmap of the KL divergence



# Let's practice!

ADVANCED DIMENSIONALITY REDUCTION IN R

# Dimensionality reduction: PCA and t-SNE

ADVANCED DIMENSIONALITY REDUCTION IN R

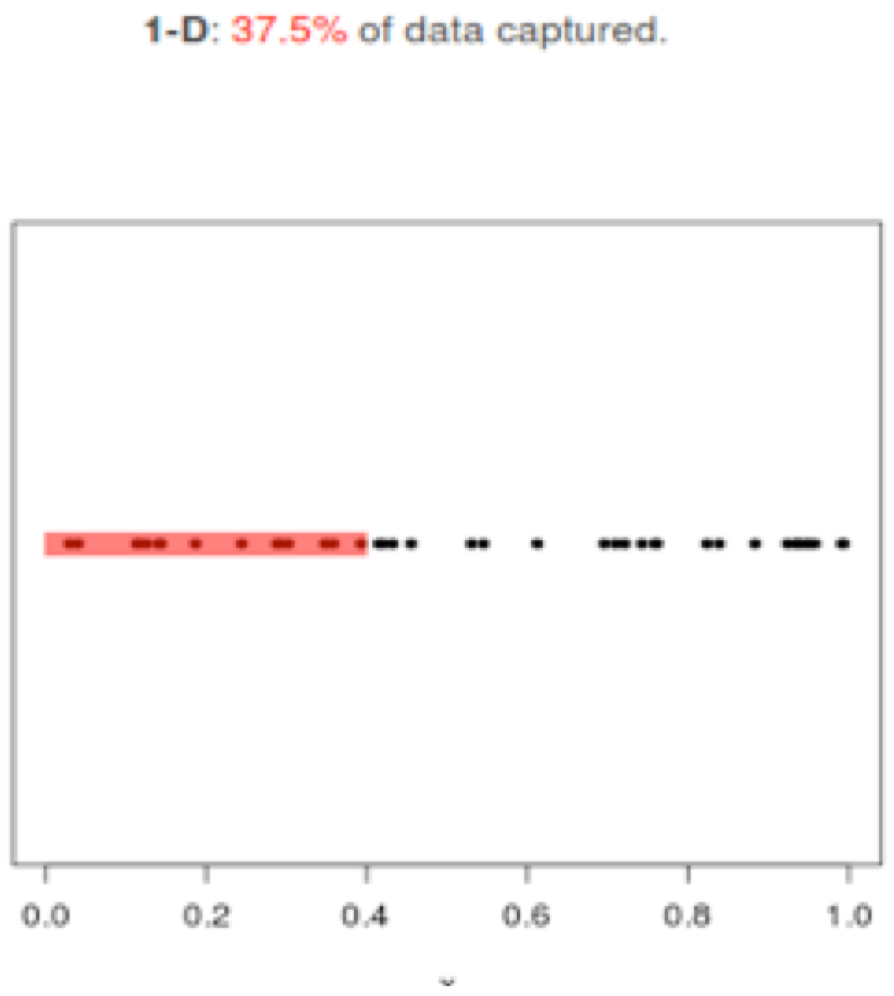
**Federico Castanedo**
Data Scientist at DataRobot

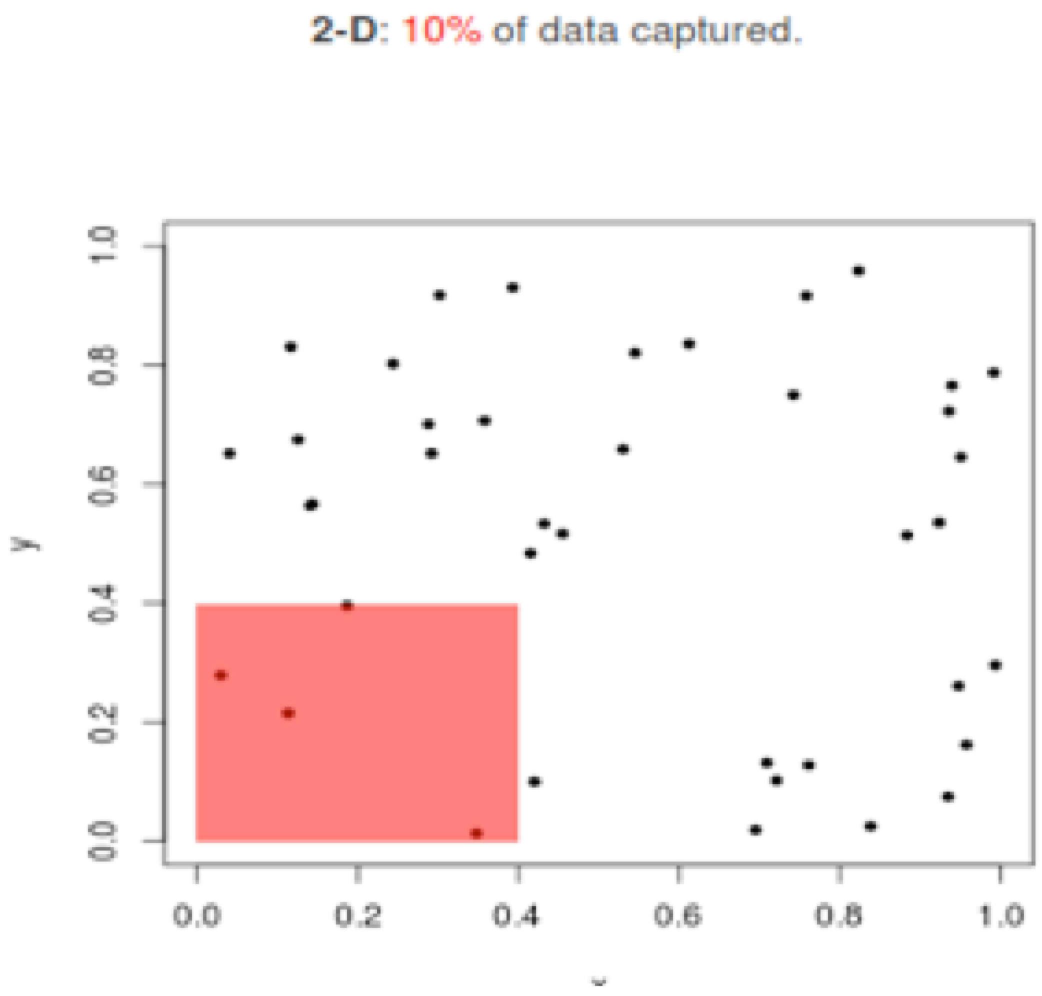
# Dimensionality reduction

- Distance metrics can not deal with high-dimensional datasets.
- This concept is known as curse of dimensionality.
- The problem of finding similar digits can be solved with dimensionality reduction techniques such as PCA and t-SNE.

# Curse of dimensionality

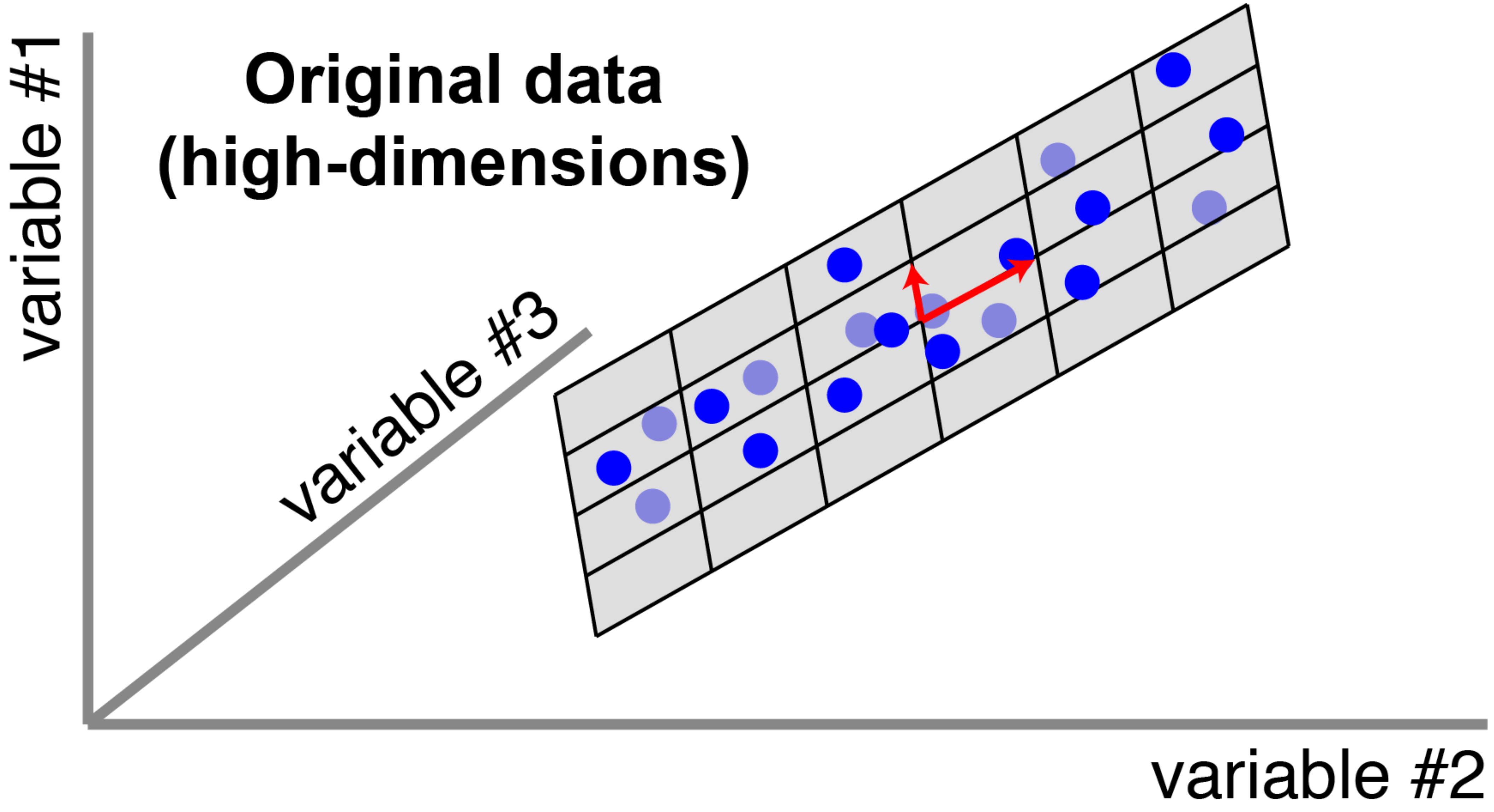
- Coined by Richard Bellman
- Describes the problems that arise when the number of dimensions grows



# Principal component analysis (PCA)

- Linear feature extraction technique: creates new independent features



# PCA in R

## PCA with default parameters

```
pca_result <- prcomp(mnist[, -1])
```

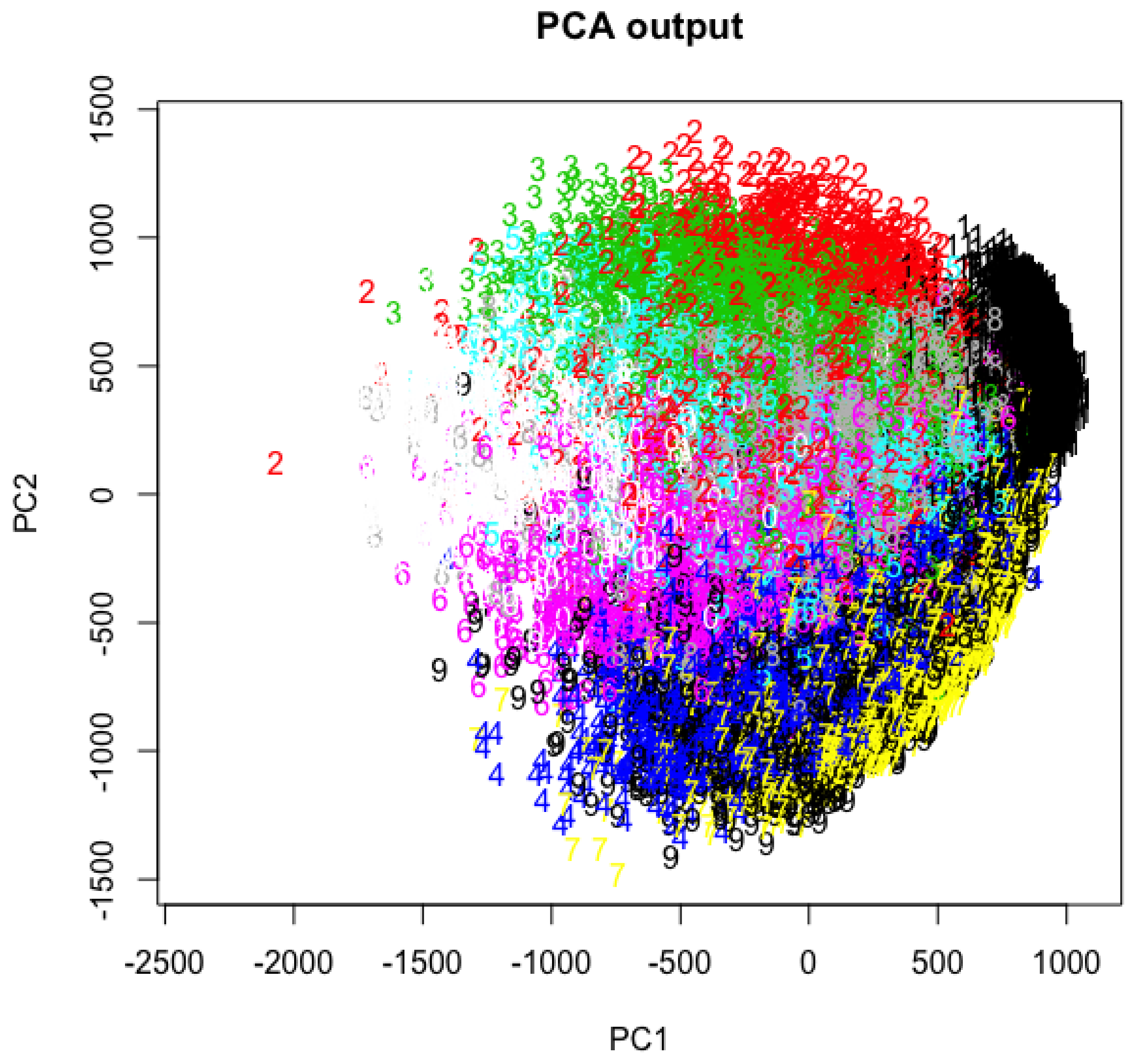
## PCA with two principal components

```
pca_result <- prcomp(mnist[, -1], rank = 2)
```

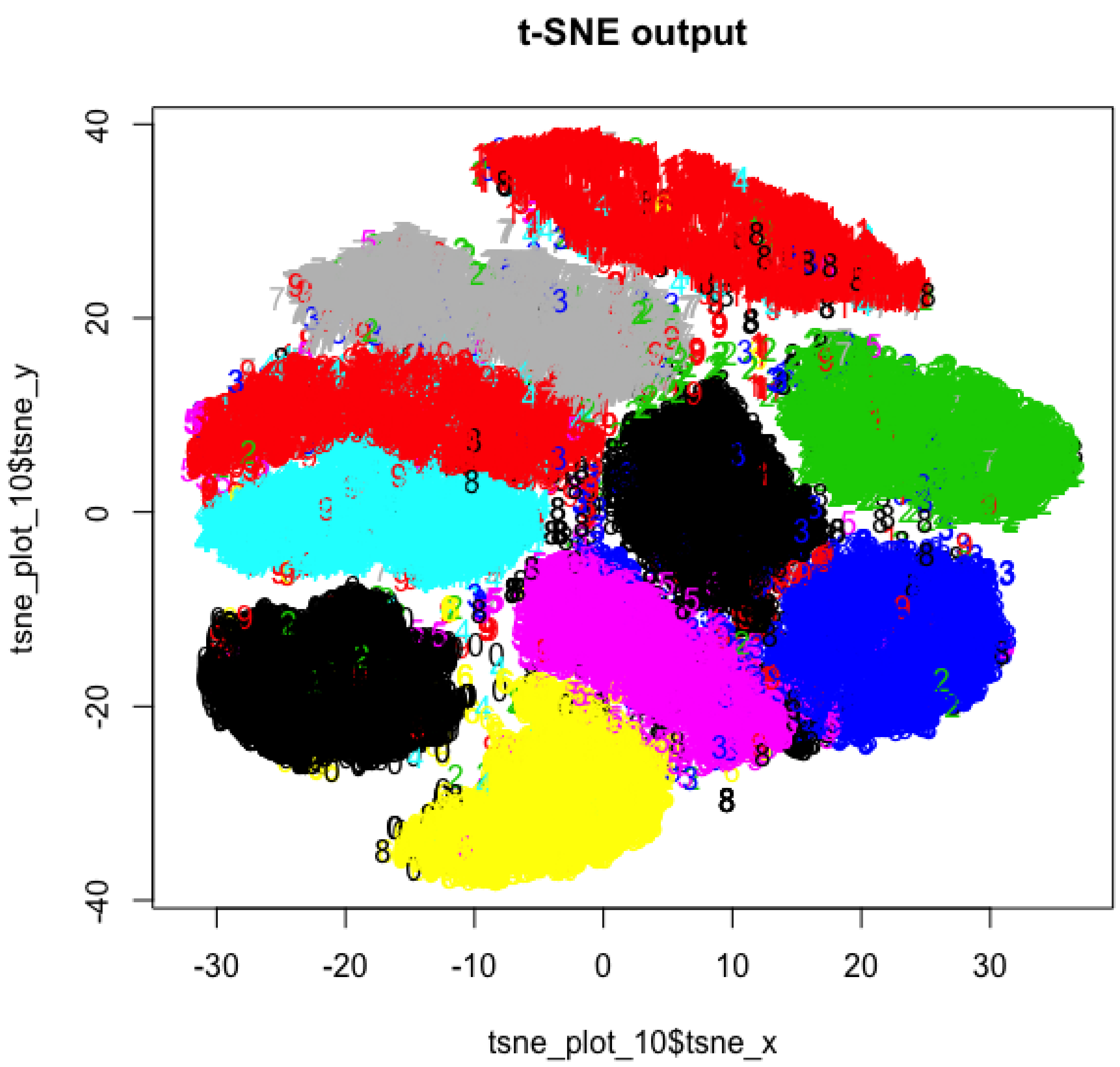
```
summary(pca_result)
```

```
Importance of first k=2 (out of 784) components:
                             PC1      PC2
Standard deviation     578.60227 495.8680
Proportion of Variance   0.09749   0.0716
Cumulative Proportion    0.09749   0.1691
```

```r
plot(pca_result$x[,1:2], pch = as.character(mnist$label),
     col = mnist$label, main = "PCA output")
```



```r
plot(tsne$tsne_x, tsne$tsne_y, pch = as.character(mnist$label),
     col = mnist$label+1, main = "t-SNE output")
```



# Let's practice!

ADVANCED DIMENSIONALITY REDUCTION IN R