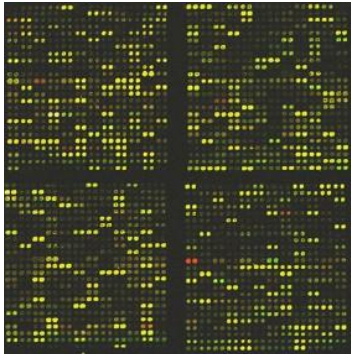
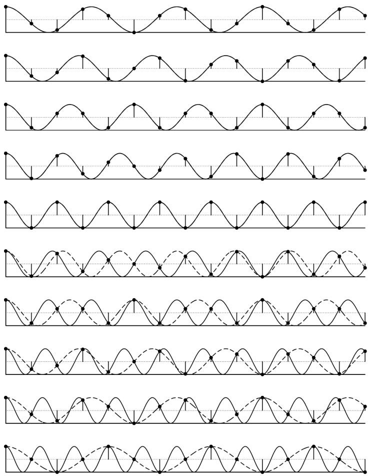
# PCA - Contextualization [https://bit.ly/2UhS65O]

Microarray Experiments

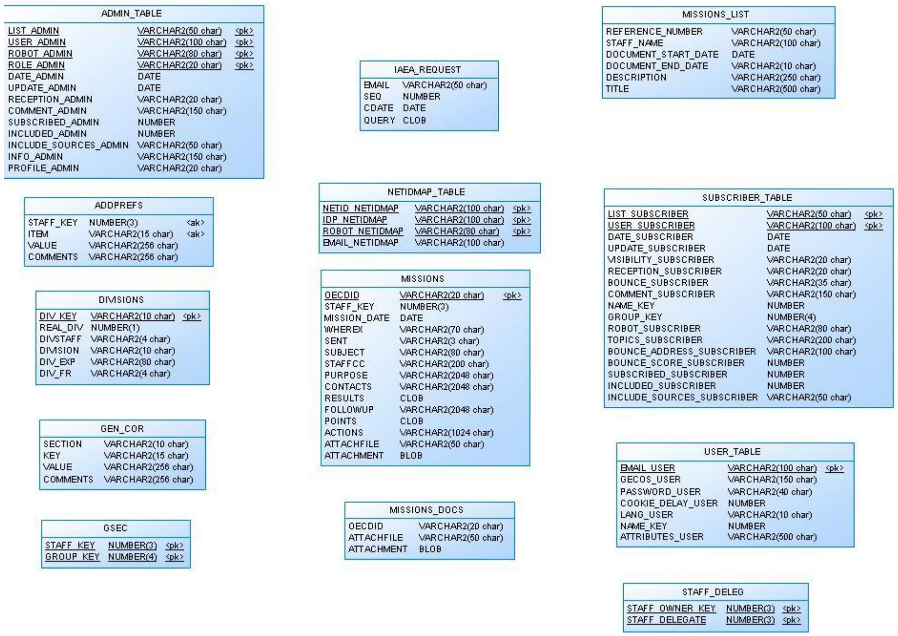




Sensor Signal Analysis

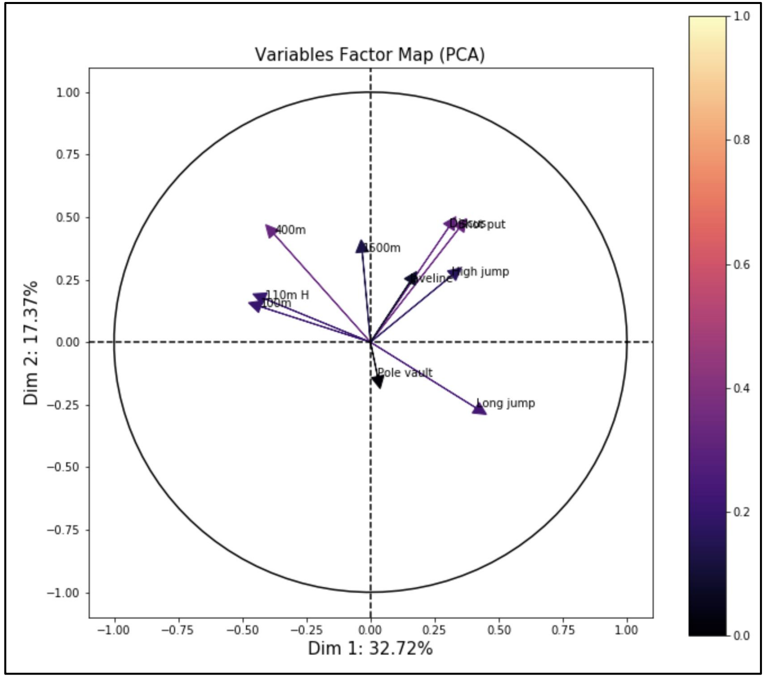
Databases

NEAD ADD db march 2009
simplified part 2/2

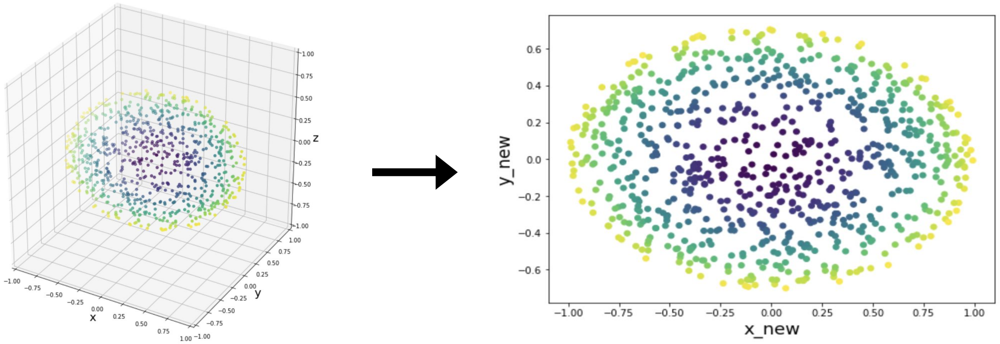



Images from https://commons.wikimedia.org/

# Objectives

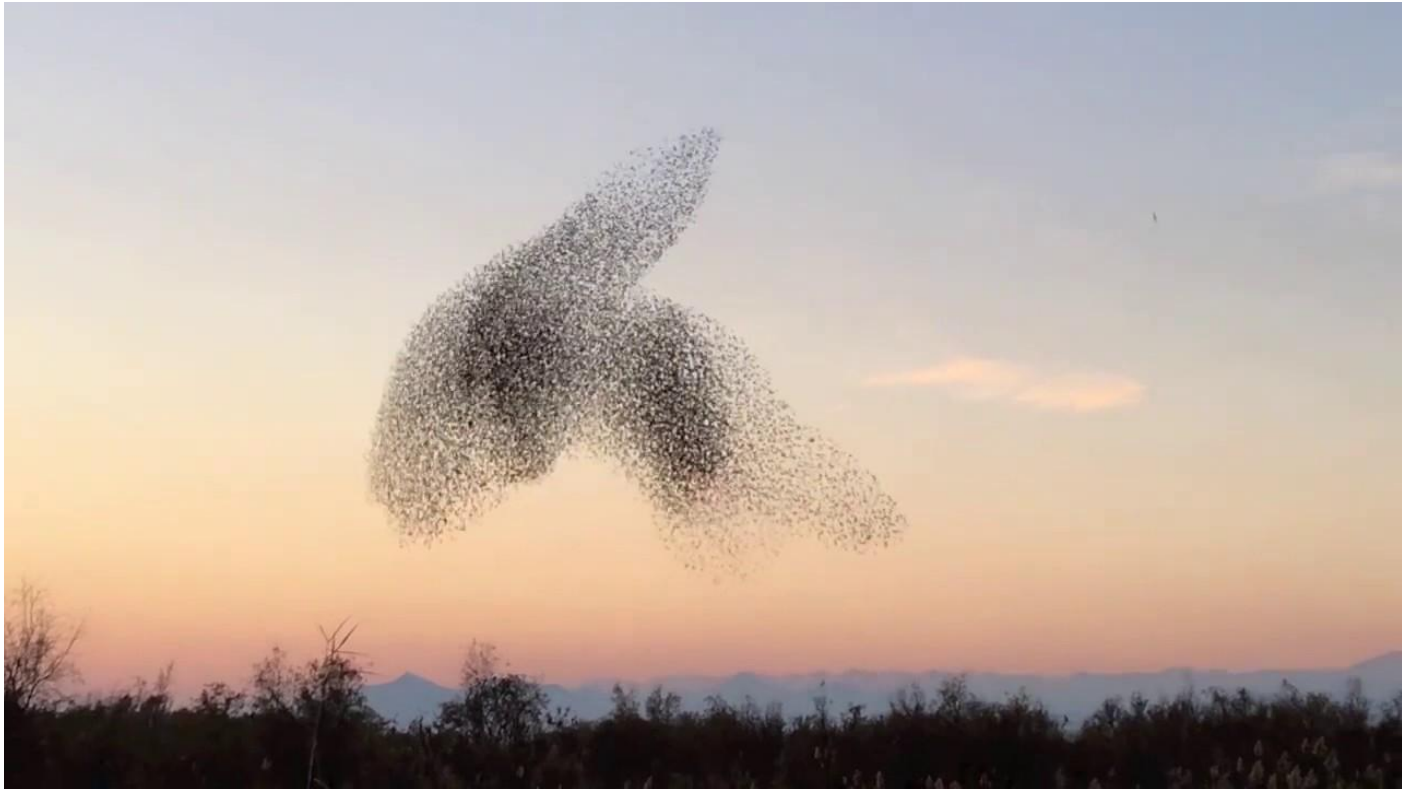


Variables Factor Map (PCA)
400m
1500m
110m H
Pole vault
Long jump
Dim 1: 32.72%
Dim 2: 17.37%

x
y
z
x_new
y_new

# PCA - Going for the Best Point of View



# PCA - Best Point of View ~= Maximize our Line of Sight

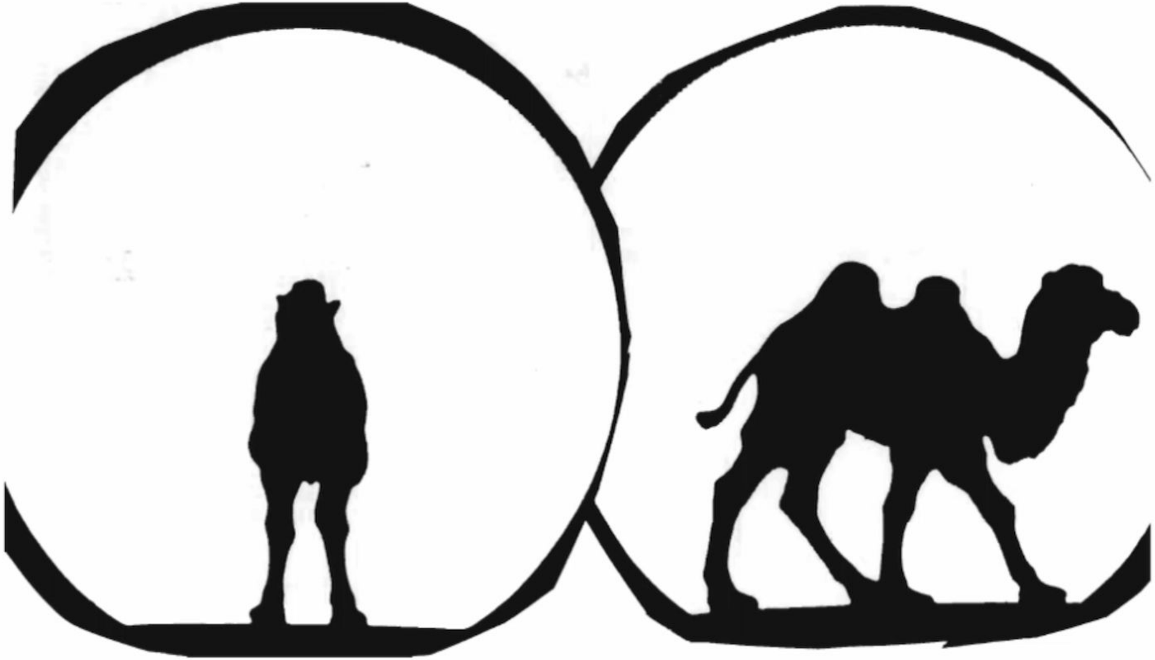
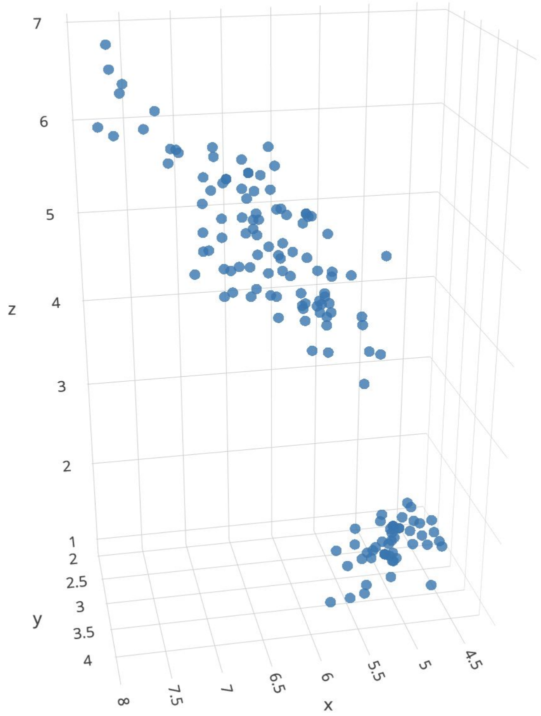


Figure: Camel or dromedary? (*illustration by J.P. Fénelon*)

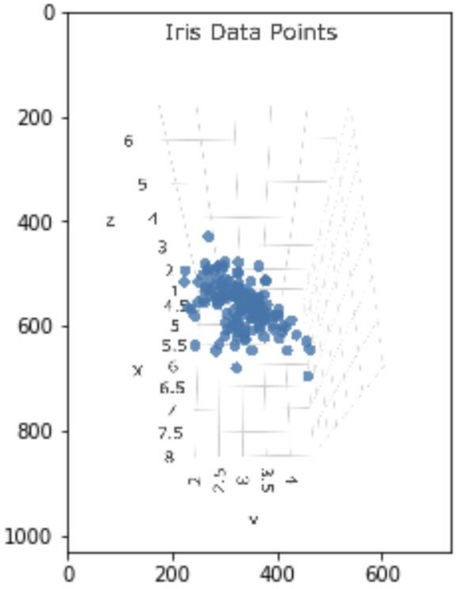
# From Pictures to Points

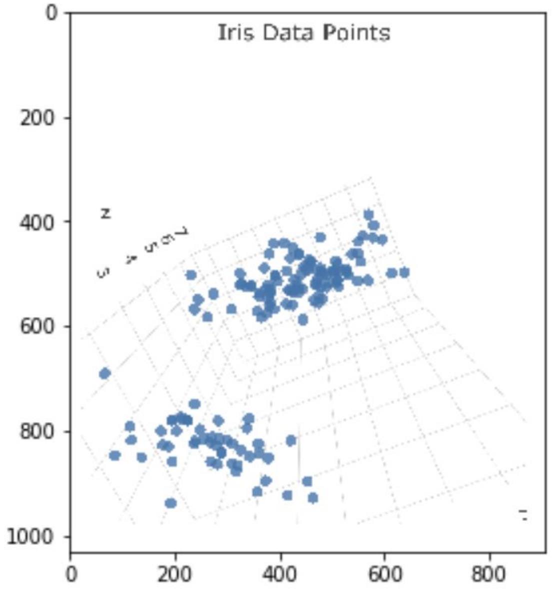


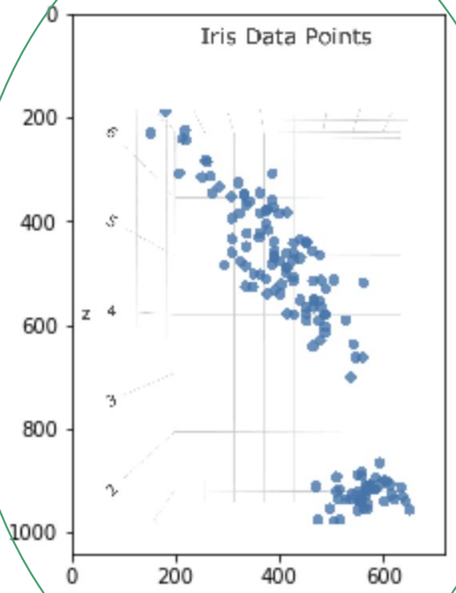

https://plot.ly/~c.miyashiro/2/#/

# From Pictures to Points

Automatic Way??



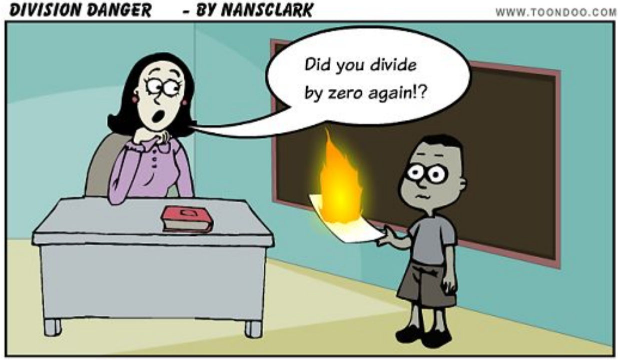




https://plot.ly/~c.miyashiro/2/#/

$$dist(O, A') = Xw$$

$$w^t w = 1$$

$$Cw = \lambda w$$




$$\frac{\partial \mathcal{L}}{\partial w} = 2wC - 2\lambda w$$

$$cov(X) = \frac{1}{m-1} X^t X$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = w^t w - 1$$

$$\mathcal{L} = w^t C w - \lambda(w^t w - 1)$$

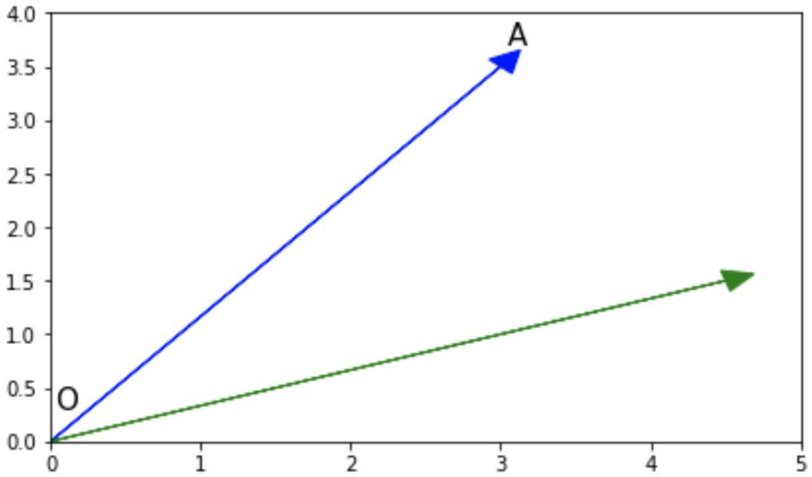
$$\underset{w}{arg\ max}\ cov(X^t w) = w^t C w$$

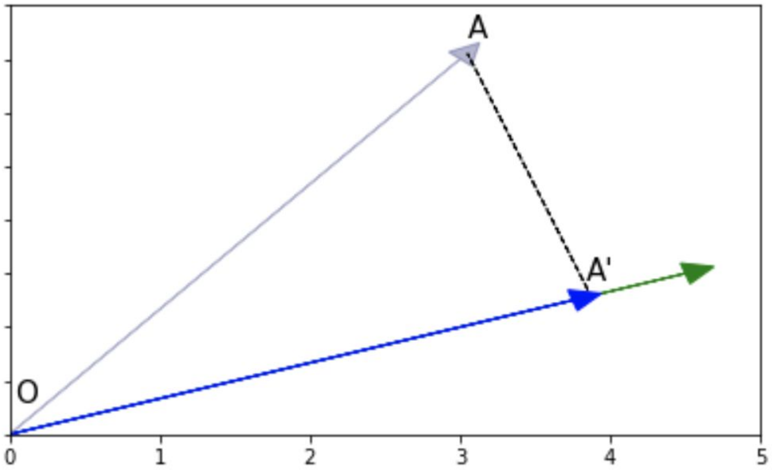
$$cov(Xw) = \frac{1}{m-1} (Xw)^t Xw$$

$$w^t (\frac{1}{m-1} X^t X) w$$

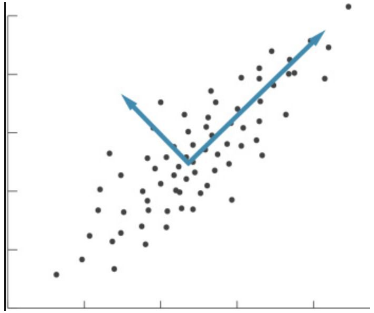
$$cov(X^t w) = w^t C w = w^t \lambda w = \lambda w^t w = \lambda$$

# Changes in Perspective = Data Projection




Maximize our Line of Sight ➔ Maximize Projection Variance



# Decathlon Dataset - Standardising and fitting

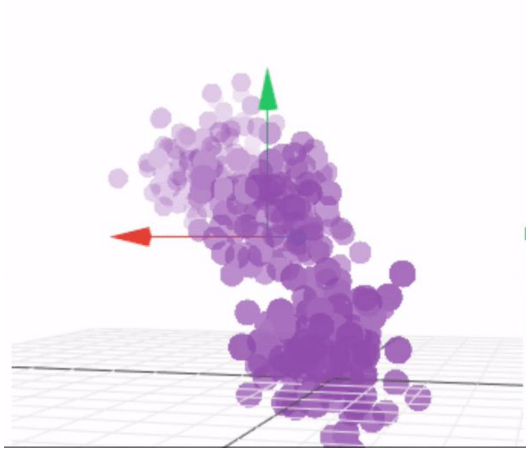
In [4]:
```
df = pd.read_csv('data_PCA_Decathlon.csv', sep=';', index_col=0)
print(f'Dataset shape: {df.shape}')
df.head()
```

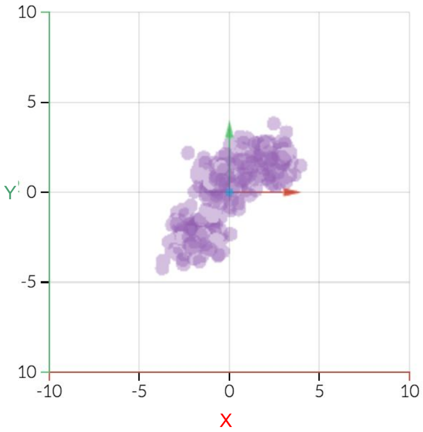
```
Dataset shape: (41, 13)
```

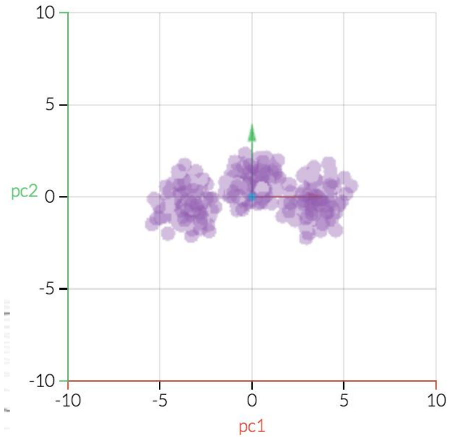
Out[4]:

| | 100m | Long jump | Shot put | High jump | 400m | 110m H | Discus | Pole vault | Javeline | 1500m | Rank | Points | Competition |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Sebrle** | 10.85 | 7.84 | 16.36 | 2.12 | 48.36 | 14.05 | 48.72 | 5.0 | 70.52 | 280.01 | 1 | 8893 | OlympicG |
| **Clay** | 10.44 | 7.96 | 15.23 | 2.06 | 49.19 | 14.13 | 50.11 | 4.9 | 69.71 | 282.00 | 2 | 8820 | OlympicG |
| **Karpov** | 10.50 | 7.81 | 15.93 | 2.09 | 46.81 | 13.97 | 51.65 | 4.6 | 55.54 | 278.11 | 3 | 8725 | OlympicG |
| **Macey** | 10.89 | 7.47 | 15.73 | 2.15 | 48.97 | 14.56 | 48.34 | 4.4 | 58.46 | 265.42 | 4 | 8414 | OlympicG |
| **Warners** | 10.62 | 7.74 | 14.48 | 1.97 | 47.97 | 14.01 | 43.73 | 4.9 | 55.39 | 278.05 | 5 | 8343 | OlympicG |

# PCA = Rotating and Transforming



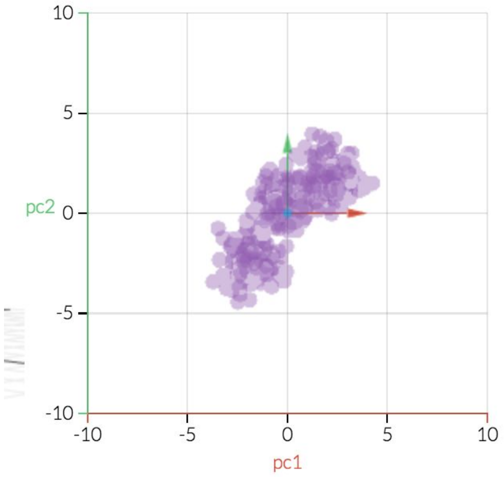


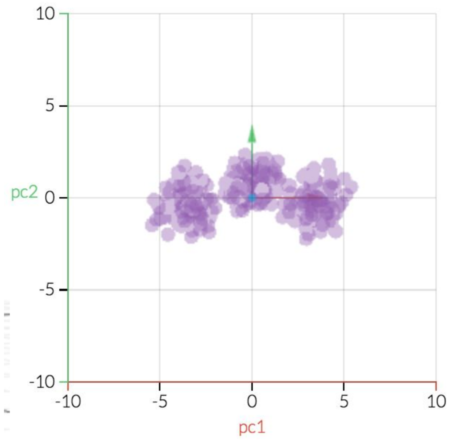

http://setosa.io/ev/principal-component-analysis/

# How many PC's? Variance Explained

Original Data




PCA Data




<u>Total Variance Data</u> = Sum of variances of each variable

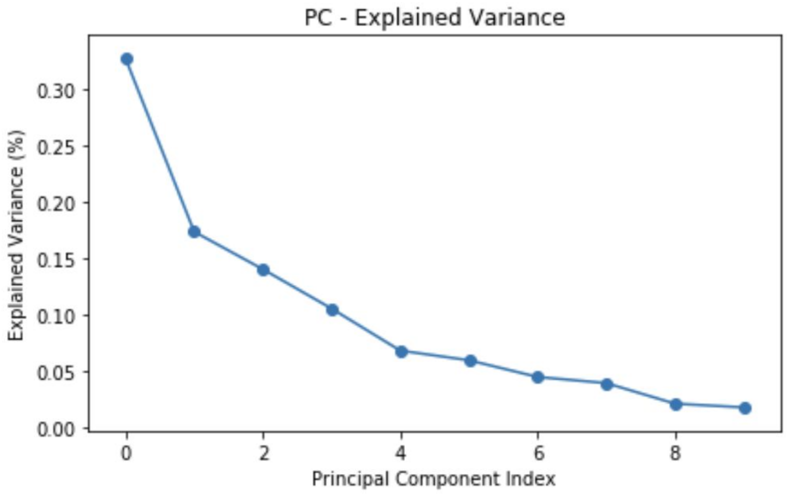
$$\sigma^2_{total} = \sum_{i=1}^{m} \sigma_i^2$$

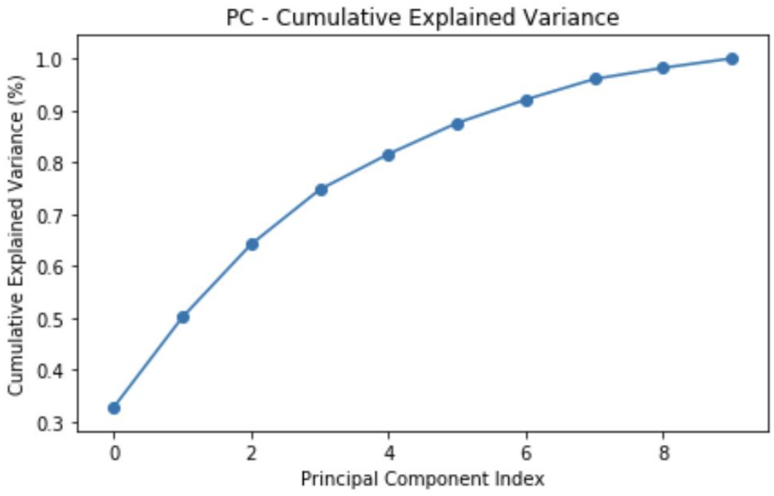
<u>Total Variance PCA</u> = Total Variance Data = Sum of '**explained_variance_**' in scikit.

Math = Sum of eigenvalues

$$\sigma^2_{total} = \sum_{i=1}^{m} \lambda_i$$

# How many PC's? Variance Explained



# Evaluation - Correlation Circle

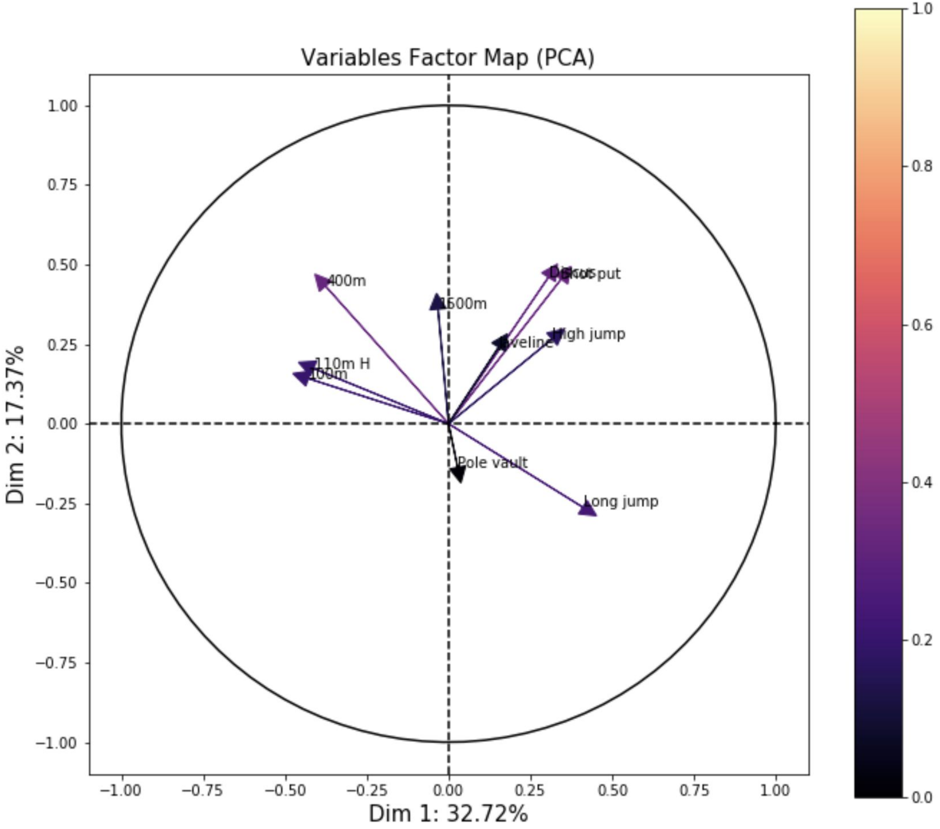
## Proxy to correlation matrix

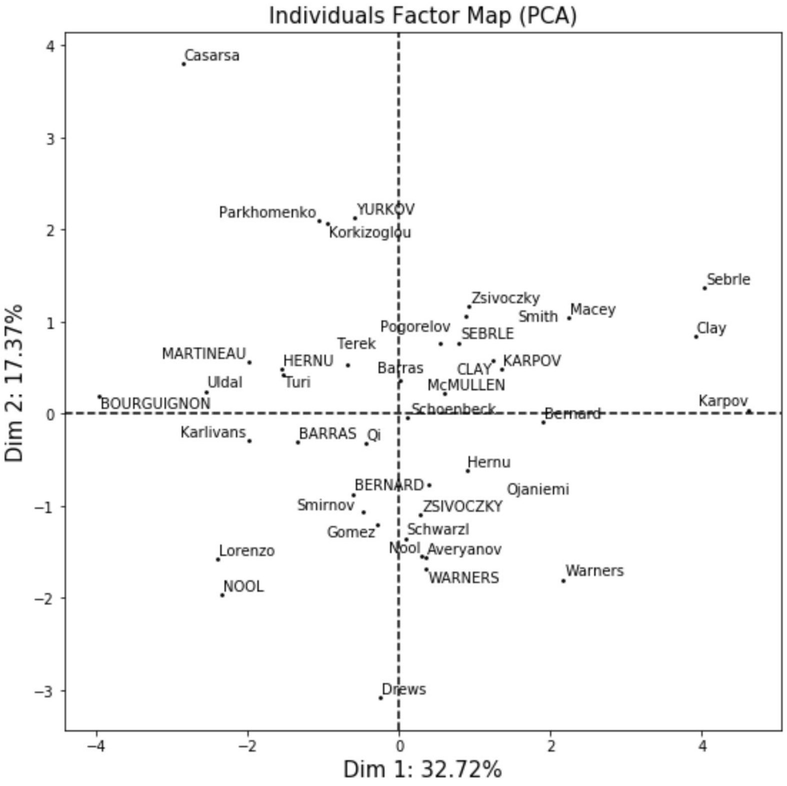
In [23]:
```
np.round(df.corr(), 2)
```

Out[23]:

| | 100m | Long jump | Shot put | High jump | 400m | 110m H | Discus | Pole vault | Javeline | 1500m | Rank | Points |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **100m** | 1.00 | -0.60 | -0.36 | -0.25 | 0.52 | 0.58 | -0.22 | -0.08 | -0.16 | -0.06 | 0.30 | -0.68 |
| **Long jump** | -0.60 | 1.00 | 0.18 | 0.29 | -0.60 | -0.51 | 0.19 | 0.20 | 0.12 | -0.03 | -0.60 | 0.73 |
| **Shot put** | -0.36 | 0.18 | 1.00 | 0.49 | -0.14 | -0.25 | 0.62 | 0.06 | 0.37 | 0.12 | -0.37 | 0.63 |
| **High jump** | -0.25 | 0.29 | 0.49 | 1.00 | -0.19 | -0.28 | 0.37 | -0.16 | 0.17 | -0.04 | -0.49 | 0.58 |
| **400m** | 0.52 | -0.60 | -0.14 | -0.19 | 1.00 | 0.55 | -0.12 | -0.08 | 0.00 | 0.41 | 0.56 | -0.67 |
| **110m H** | 0.58 | -0.51 | -0.25 | -0.28 | 0.55 | 1.00 | -0.33 | -0.00 | 0.01 | 0.04 | 0.44 | -0.64 |
| **Discus** | -0.22 | 0.19 | 0.62 | 0.37 | -0.12 | -0.33 | 1.00 | -0.15 | 0.16 | 0.26 | -0.39 | 0.48 |
| **Pole vault** | -0.08 | 0.20 | 0.06 | -0.16 | -0.08 | -0.00 | -0.15 | 1.00 | -0.03 | 0.25 | -0.32 | 0.20 |
| **Javeline** | -0.16 | 0.12 | 0.37 | 0.17 | 0.00 | 0.01 | 0.16 | -0.03 | 1.00 | -0.18 | -0.21 | 0.42 |
| **1500m** | -0.06 | -0.03 | 0.12 | -0.04 | 0.41 | 0.04 | 0.26 | 0.25 | -0.18 | 1.00 | 0.09 | -0.19 |
| **Rank** | 0.30 | -0.60 | -0.37 | -0.49 | 0.56 | 0.44 | -0.39 | -0.32 | -0.21 | 0.09 | 1.00 | -0.74 |
| **Points** | -0.68 | 0.73 | 0.63 | 0.58 | -0.67 | -0.64 | 0.48 | 0.20 | 0.42 | -0.19 | -0.74 | 1.00 |

# Evaluation - Correlation Circle



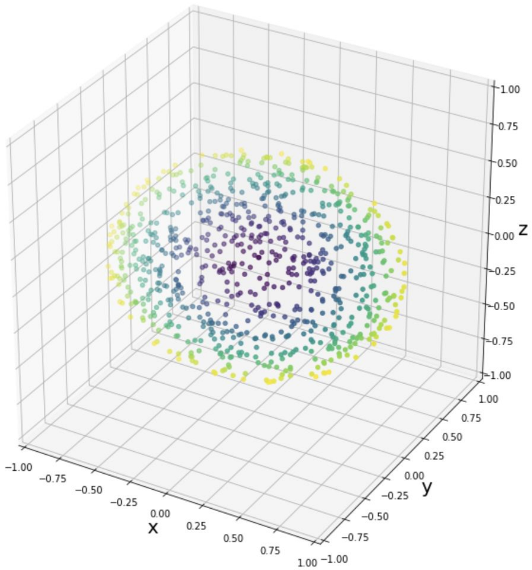
# Evaluation - Correlation Circle

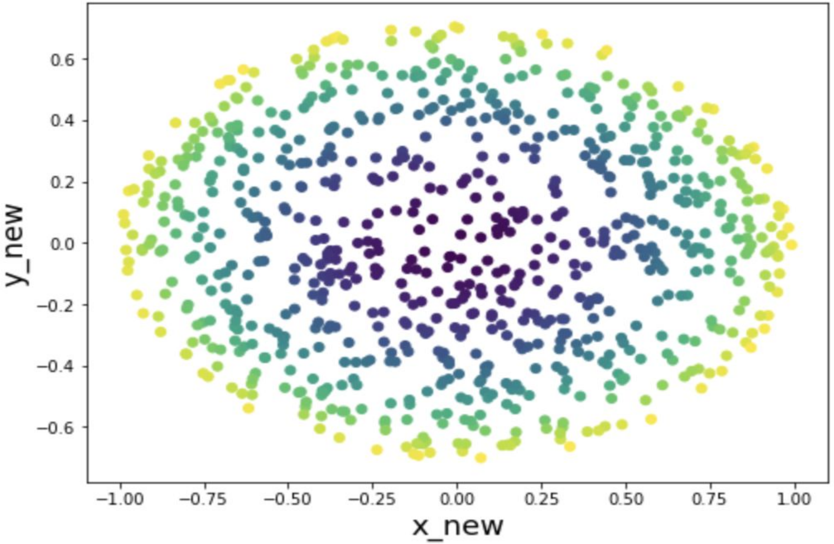
In [19]:
```
athlets = ['Parkhomenko', 'Warners']
df_scaled.loc[athlets]
```

Out[19]:

| | 100m | Long jump | Shot put | High jump | 400m | 110m H | Discus | Pole vault | Javeline | 1500m |
|---|---|---|---|---|---|---|---|---|---|---|
| **Parkhomenko** | 0.546396 | -2.079872 | 1.489512 | 0.605182 | 1.249593 | 0.588297 | -0.727015 | 0.136790 | 1.573837 | -0.094092 |
| **Warners** | -1.455178 | 1.535905 | 0.003594 | -0.077730 | -1.445050 | -1.278656 | -0.178519 | 0.500971 | -0.613850 | -0.084551 |

# PCA in Dimensionality Reduction



# PCA Applied to Images - Eigenfaces

Original images

First 3 components with colormap 'Greys_r' - positive values are white.

First 3 components with colormap 'Greys' - negative values are black.

Exercise!

Lets go to the repository:

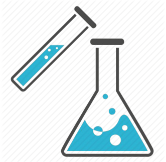
# PCA - Final Thoughts

- Unsupervised Learning - No labelling data!
- Dimensionality Reduction
- Simpler representation on variable correlation

- Assumes linear relationship among explanatory variables
- “Target Variable” is variance, we have to be careful about noises

- Only numerical features (Categorical proxy as supplementary variables)
- No missing value support

- Advances: Sparse PCA / Batch PCA



# Thank you!


https://dilbert.com/strip/2000-01-03