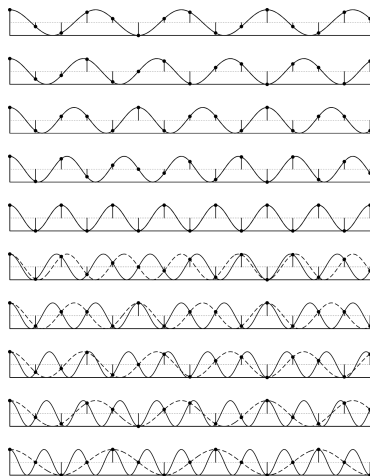
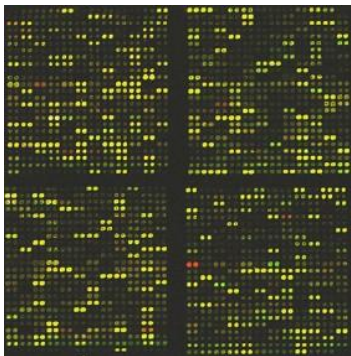


# PCA - Contextualization [ <https://bit.ly/2UhS65O> ]



Sensor Signal Analysis

Microarray Experiments



## Databases

NEAD ADD db march 2009  
simplified part 2/2

ADMIN_TABLE		
LIST_ADMIN	VARCHAR2(50 char)	pk>
USER_ADMIN	VARCHAR2(100 char)	pk>
ROBOT_ADMIN	VARCHAR2(200 char)	pk>
ROLE_ADMIN	VARCHAR2(20 char)	pk>
DATE_ADMIN	DATE	
UPDATE_ADMIN	DATE	
RECEPTION_ADMIN	VARCHAR2(20 char)	
COMMENT_ADMIN	VARCHAR2(150 char)	
SUBSCRIBED_ADMIN	NUMBER	
INCLUDED_ADMIN	NUMBER	
INCLUDE_SOURCES_ADMIN	VARCHAR2(50 char)	
INFO_ADMIN	VARCHAR2(150 char)	
PROFILE_ADMIN	VARCHAR2(20 char)	

ADDPREFS		
STAFF_KEY	NUMBER(3)	pk>
ITEM	VARCHAR2(15 char)	pk>
VALUE	VARCHAR2(255 char)	
COMMENTS	VARCHAR2(255 char)	

DIVISIONS		
DIV_KEY	VARCHAR2(10 char)	pk>
REAL_DIV	NUMBER(1)	
DIV_STAFF	VARCHAR2(4 char)	
DIVISION	VARCHAR2(20 char)	
DIV_BIP	VARCHAR2(20 char)	
DIV_FR	VARCHAR2(4 char)	

GEN_COR		
SECTION	VARCHAR2(10 char)	
KEY	VARCHAR2(15 char)	
VALUE	VARCHAR2(255 char)	
COMMENTS	VARCHAR2(255 char)	

GSEC		
STAFF_KEY	NUMBER(3)	pk>
GROUP_KEY	NUMBER(4)	pk>

IAEA_REQUEST		
EMAIL	VARCHAR2(50 char)	
SEQ	NUMBER	
CDATE	DATE	
QUERY	CLOB	

NETIDMAP_TABLE		
NETID_NETIDMAP	VARCHAR2(100 char)	pk>
IDP_NETIDMAP	VARCHAR2(100 char)	pk>
ROBOT_NETIDMAP	VARCHAR2(200 char)	pk>
EMAIL_NETIDMAP	VARCHAR2(100 char)	pk>

MISSIONS		
DECID	VARCHAR2(20 char)	pk>
STAFF_KEY	NUMBER(3)	
MISSION_DATE	DATE	
WHEREX	VARCHAR2(70 char)	
SENT	VARCHAR2(2 char)	
SUBJECT	VARCHAR2(50 char)	
STAFF_C	VARCHAR2(200 char)	
PURPOSE	VARCHAR2(2048 char)	
CONTACTS	VARCHAR2(2048 char)	
RESULTS	VARCHAR2(2048 char)	
FOLLOWUP	POINTS	
ACTIONS	VARCHAR2(1024 char)	
ATTACHFILE	VARCHAR2(50 char)	
ATTACHMENT	BLOB	

MISSIONS_DOCS		
DECID	VARCHAR2(20 char)	
ATTACHFILE	VARCHAR2(50 char)	
ATTACHMENT	BLOB	

MISSIONS_LIST		
REFERENCE_NUMBER	VARCHAR2(50 char)	
STAFF_NAME	VARCHAR2(100 char)	
DOCUMENT_START_DATE	DATE	
DOCUMENT_END_DATE	VARCHAR2(20 char)	
DESCRIPTION	VARCHAR2(250 char)	
TITLE	VARCHAR2(500 char)	

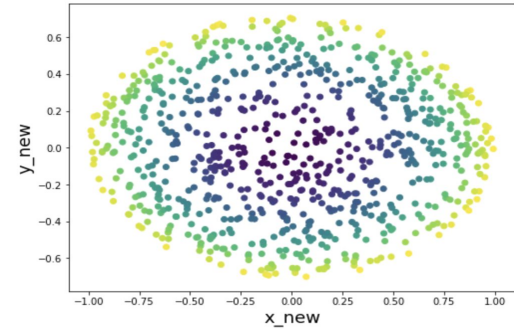
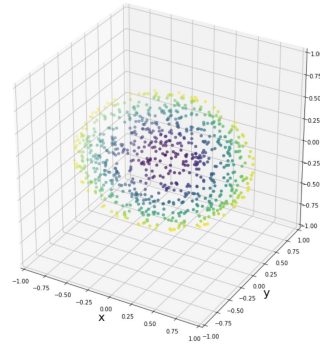
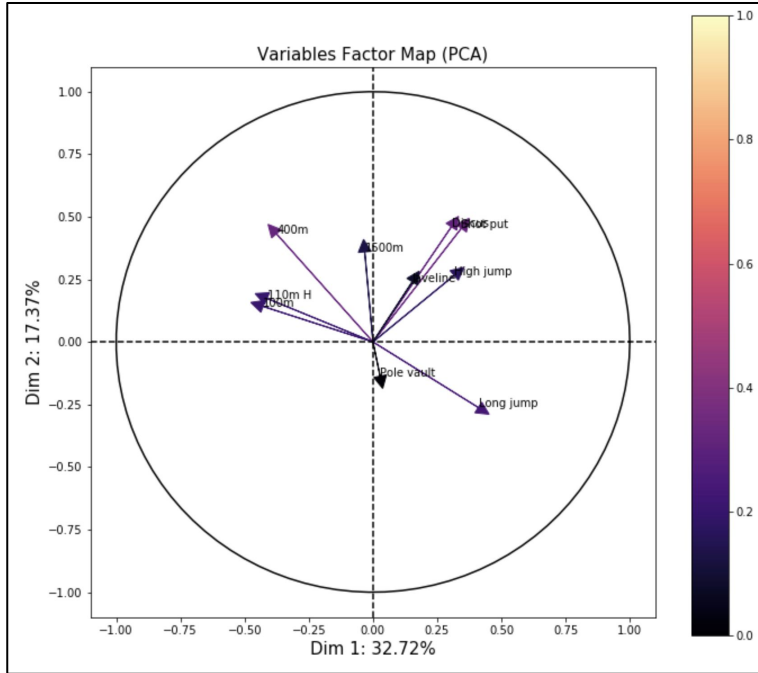
SUBSCRIBER_TABLE		
LIST_SUBSCRIBER	VARCHAR2(250 char)	pk>
USER_SUBSCRIBER	VARCHAR2(100 char)	pk>
DATE_SUBSCRIBER	DATE	
UPDATE_SUBSCRIBER	DATE	
VISIBILITY_SUBSCRIBER	VARCHAR2(20 char)	
RECEPTION_SUBSCRIBER	VARCHAR2(20 char)	
BOUNCE_SUBSCRIBER	VARCHAR2(25 char)	
COMMENT_SUBSCRIBER	VARCHAR2(150 char)	
NAME_KEY	NUMBER	
GROUP_KEY	NUMBER(4)	
ROBOT_SUBSCRIBER	VARCHAR2(50 char)	
TOPICS_SUBSCRIBER	VARCHAR2(200 char)	
BOUNCE_ADDRESS_SUBSCRIBER	VARCHAR2(100 char)	
BOUNCE_SCORE_SUBSCRIBER	NUMBER	
SUBSCRIBED_SUBSCRIBER	NUMBER	
INCLUDED_SUBSCRIBER	NUMBER	
INCLUDE_SOURCES_SUBSCRIBER	VARCHAR2(50 char)	

USER_TABLE		
EMAIL_USER	VARCHAR2(100 char)	pk>
GECOS_USER	VARCHAR2(150 char)	
PASSWORD_USER	VARCHAR2(40 char)	
COOKIE_DELAY_USER	NUMBER	
LANG_USER	VARCHAR2(10 char)	
NAME_KEY	NUMBER	
ATTRIBUTES_USER	VARCHAR2(500 char)	

STAFF_DELEG		
STAFF_OWNER_KEY	NUMBER(3)	pk>
STAFF_DELEGATE	NUMBER(3)	pk>

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

# Objectives



# PCA - Going for the Best Point of View



PCA - Best Point of View  $\sim$  Maximize our Line of Sight

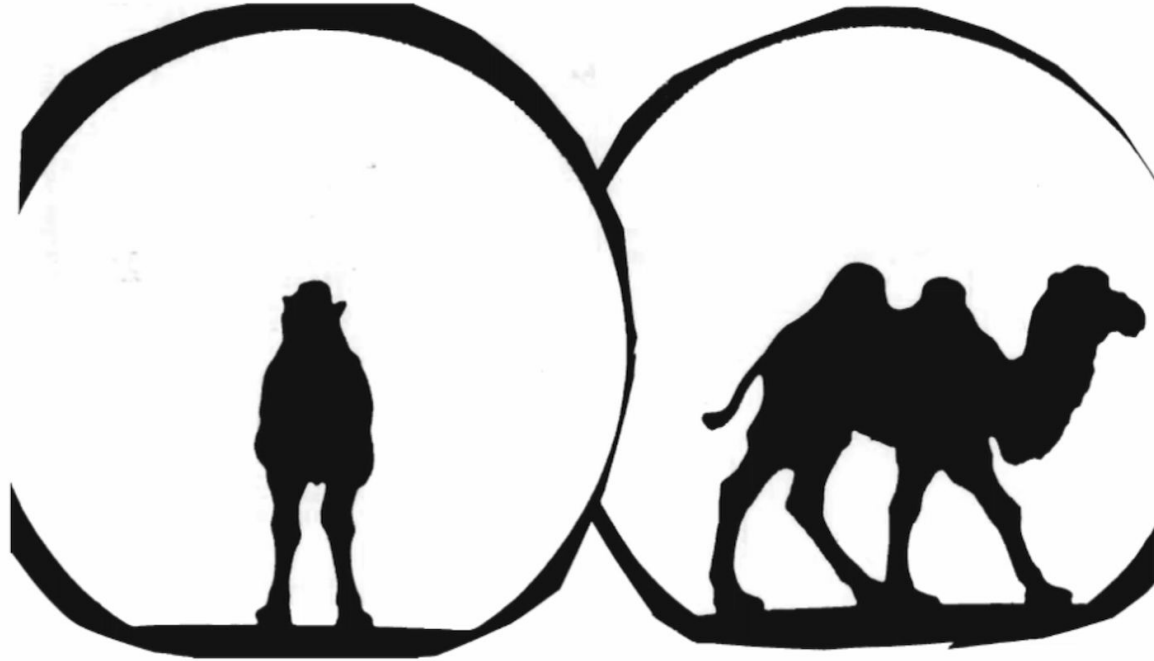
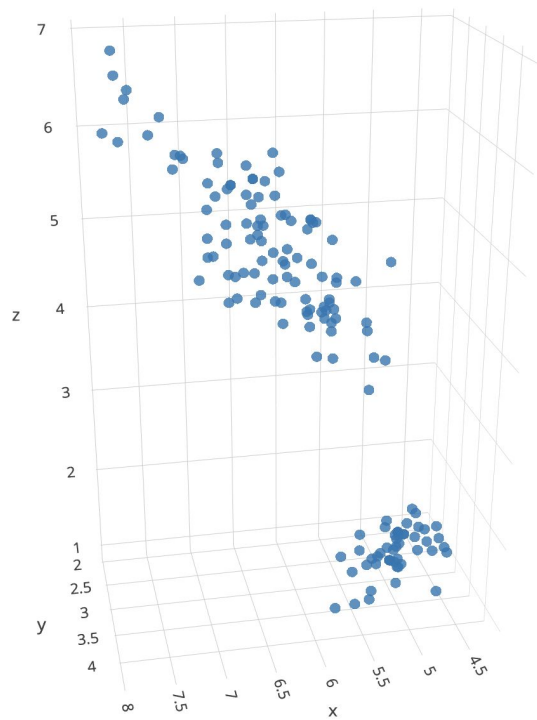


Figure: Camel or dromedary? (*illustration by J.P. F  nelon*)

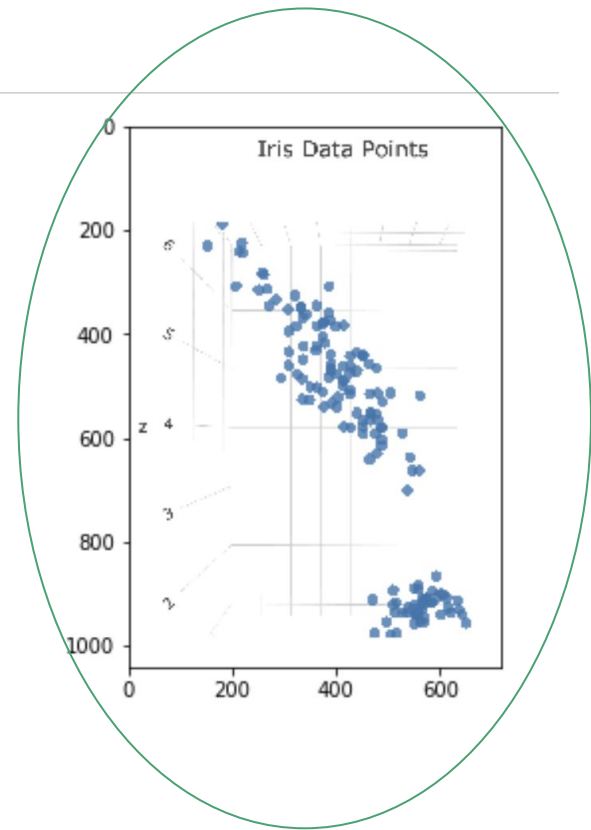
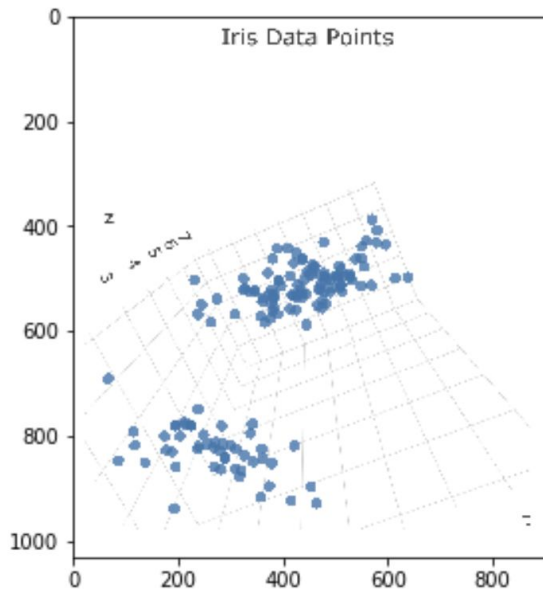
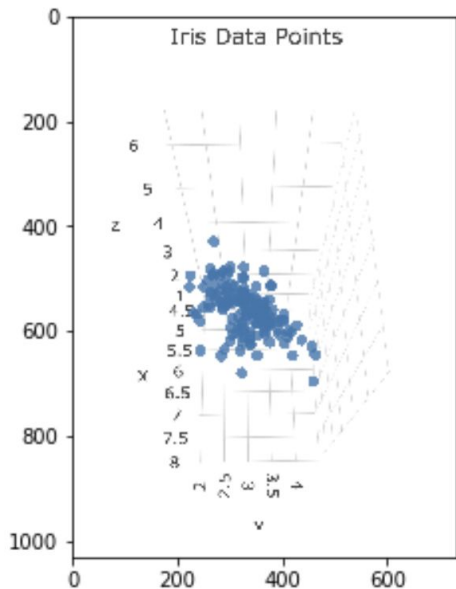
# From Pictures to Points



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

# From Pictures to Points

Automatic Way??



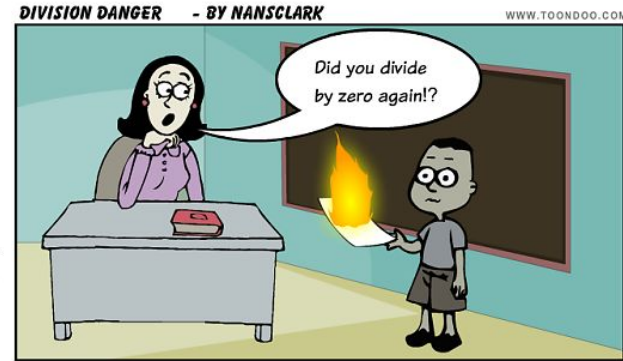
<https://plot.ly/~c.miyashiro/2/#/>

$$\text{dist}(O, A') = Xw \quad w^t w = 1$$

$$Cw = \lambda w$$

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

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



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

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

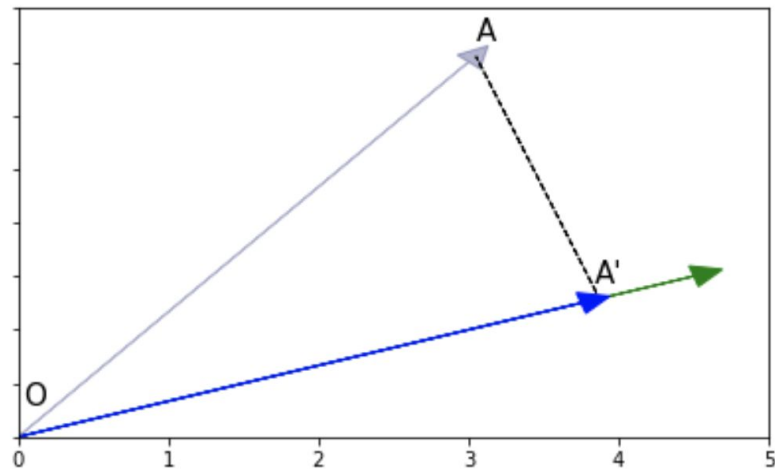
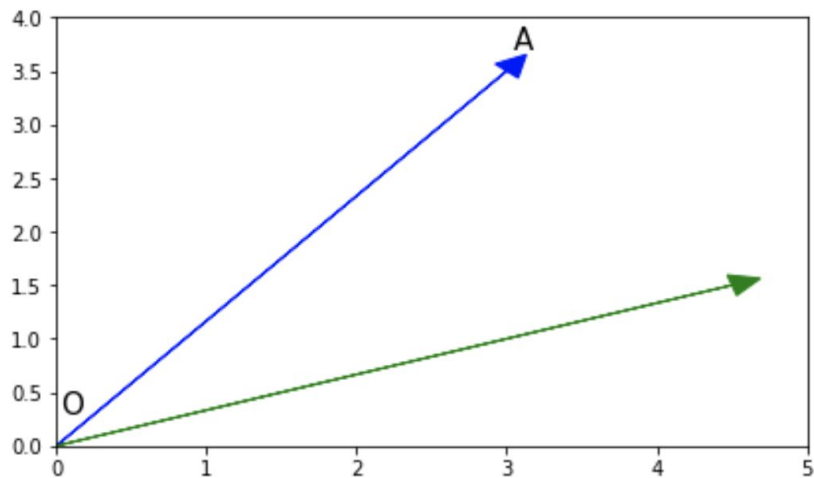
$$\arg \max_w \text{cov}(X^t w) = w^t Cw$$

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

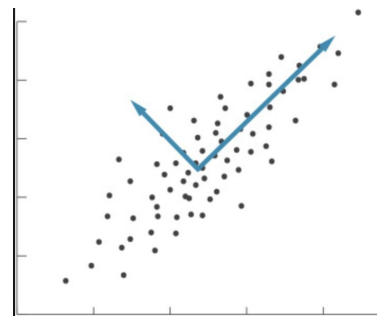
$$w^t \left( \frac{1}{m-1} X^t X \right) w$$

$$\text{cov}(X^t w) = w^t Cw = w^t \lambda w = \lambda w^t w = \lambda$$

# Changes in Perspective = Data Projection



Maximize our Line of Sight  $\rightarrow$  Maximize Projection Variance





# Decathlon Dataset - Standardising and fitting



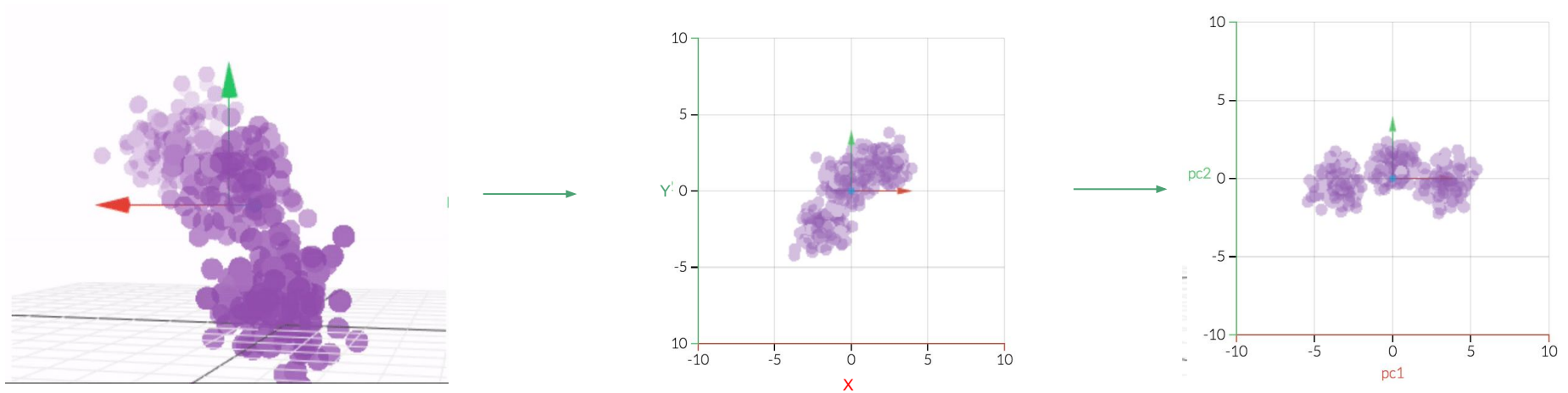
```
In [4]: df = pd.read_csv('data_PCA_Decathlon.csv', sep=';', index_col=0)
print(f'Dataset shape: {df.shape}')
df.head()
```

Dataset shape: (41, 13)

Out[4]:

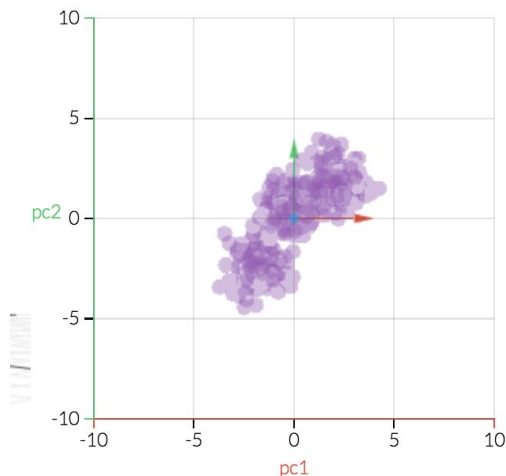
	◆ 100m ◆	◆ Long jump ◆	◆ Shot put ◆	◆ High jump ◆	◆ 400m ◆	◆ 110m H ◆	◆ Discus ◆	◆ Pole vault ◆	◆ Javeline ◆	◆ 1500m ◆	◆ Rank ◆	◆ Points ◆	◆ Competition ◆
<b>Sebrle</b>	10.85	7.84	16.36	2.12	48.36	14.05	48.72	5.0	70.52	280.01	1	8893	OlympicG
<b>Clay</b>	10.44	7.96	15.23	2.06	49.19	14.13	50.11	4.9	69.71	282.00	2	8820	OlympicG
<b>Karpov</b>	10.50	7.81	15.93	2.09	46.81	13.97	51.65	4.6	55.54	278.11	3	8725	OlympicG
<b>Macey</b>	10.89	7.47	15.73	2.15	48.97	14.56	48.34	4.4	58.46	265.42	4	8414	OlympicG
<b>Warners</b>	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



# How many PC's? Variance Explained

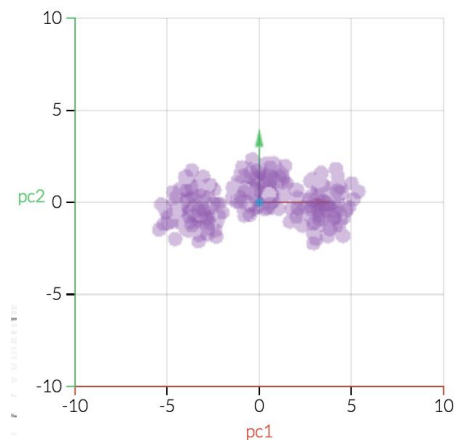
Original Data



Total Variance Data =  
Sum of variances of each  
variable

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

PCA Data

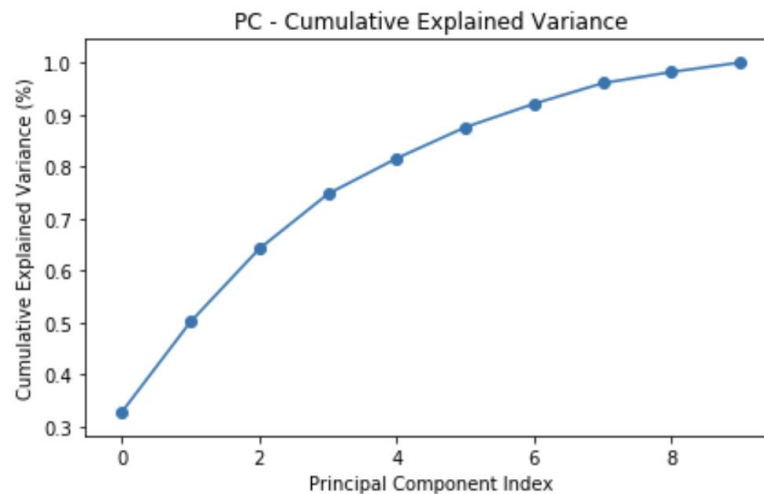
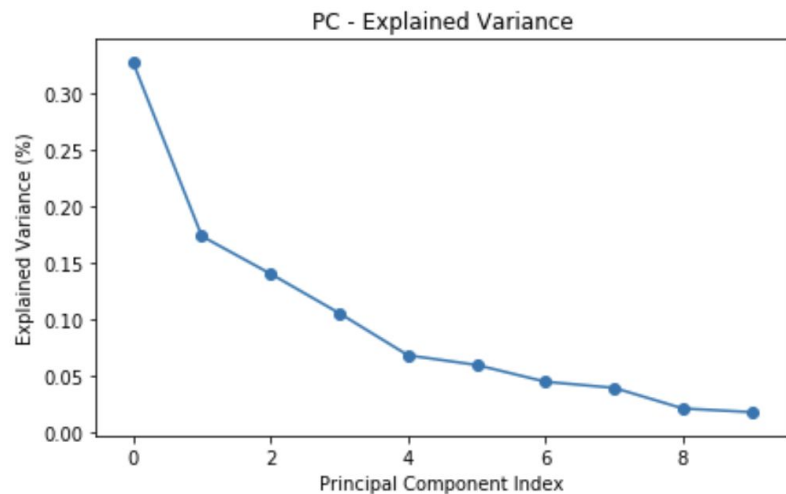


Total Variance PCA = Total  
Variance Data = Sum of  
'**explained\_variance\_**' in scikit.

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

Math = Sum of eigenvalues

# How many PC's? Variance Explained



# Evaluation - Correlation Circle

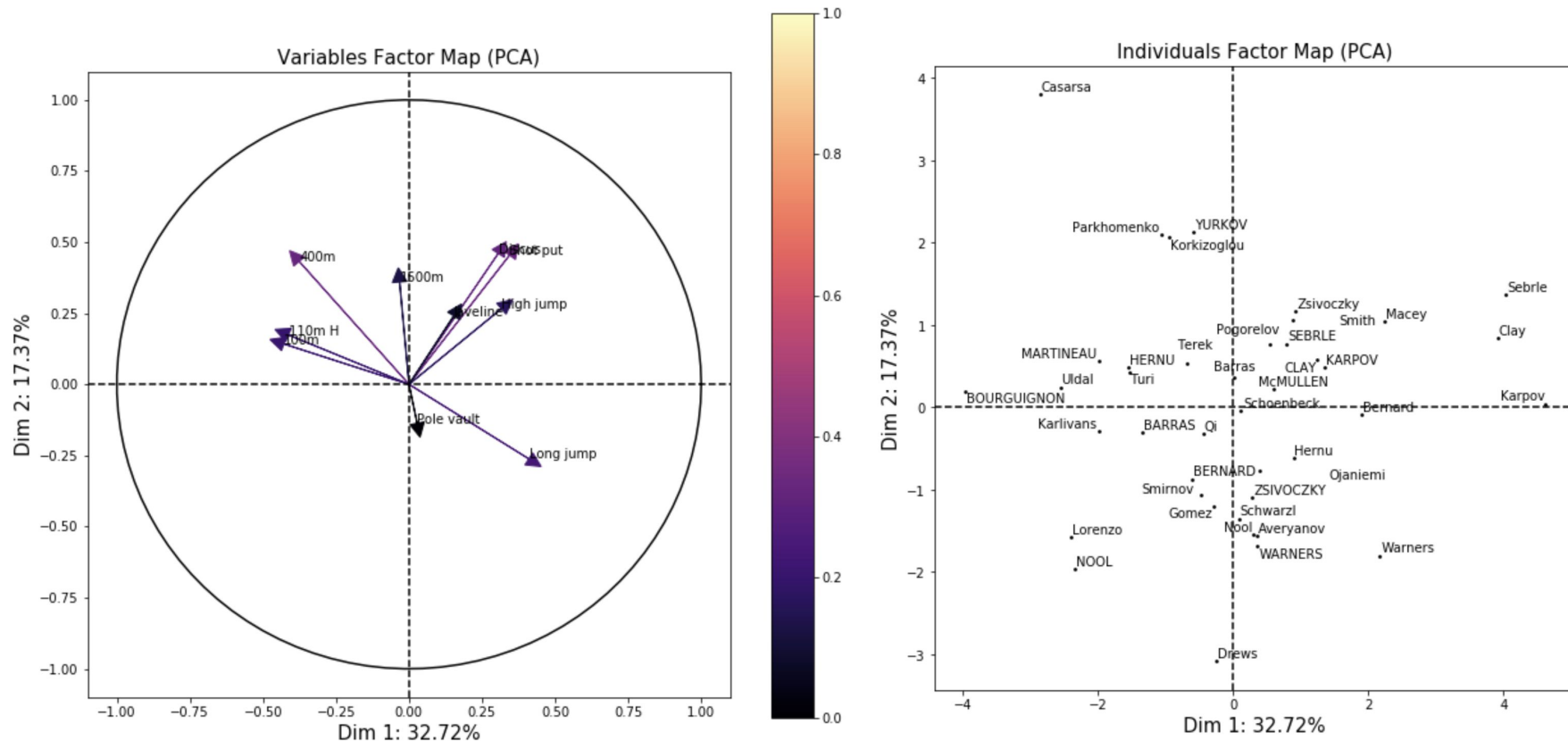
## Proxy to correlation matrix

```
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



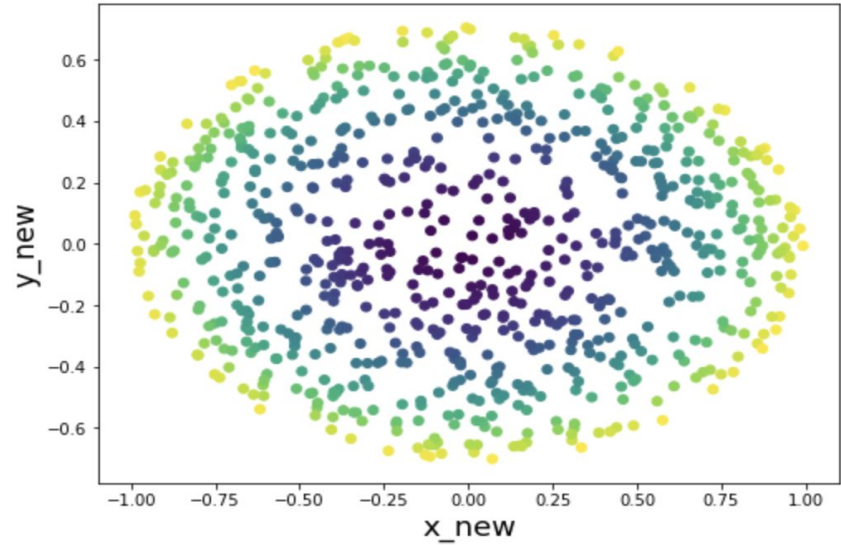
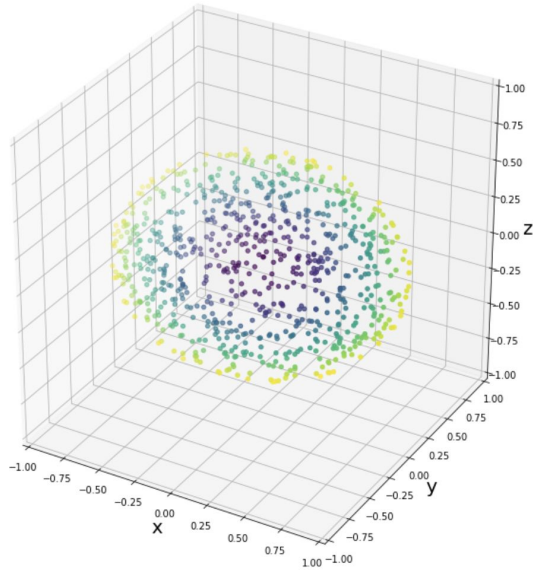
# Evaluation - Correlation Circle

```
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



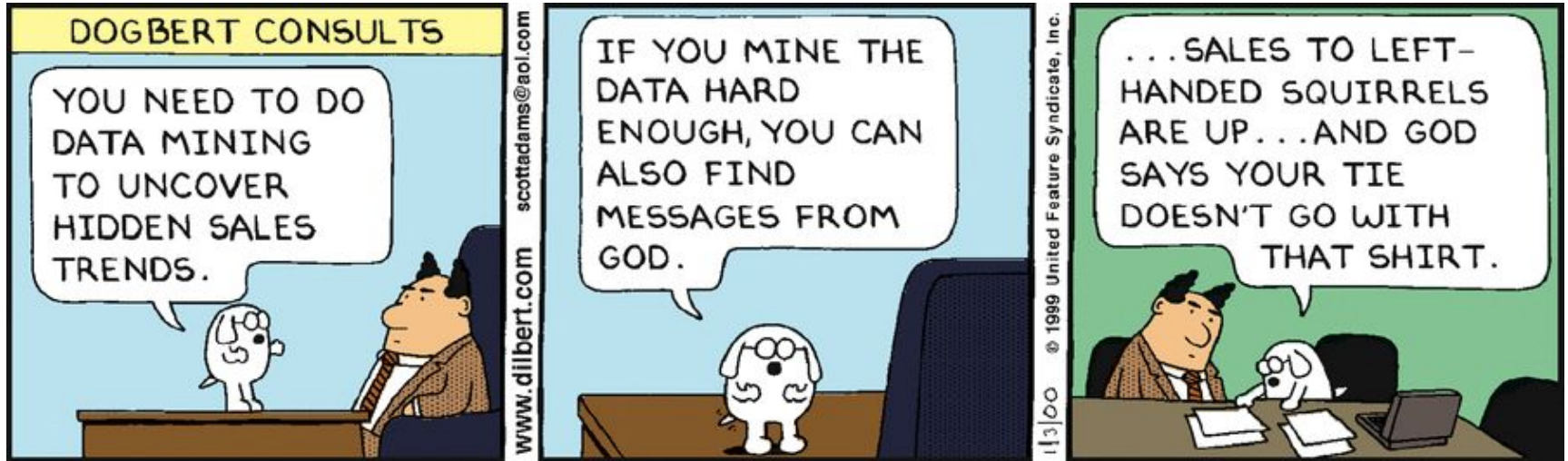
- Only numerical features (Categorical proxy as supplementary variables)
- No missing value support



- Advances: Sparse PCA / Batch PCA



Thank you!



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