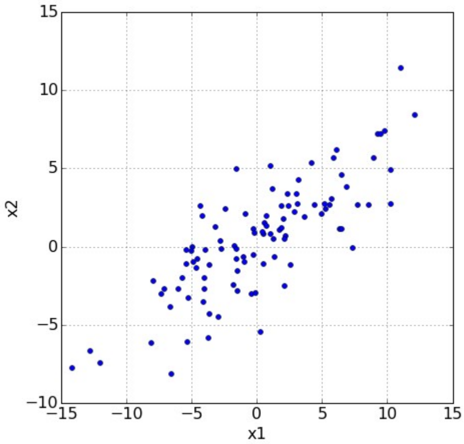
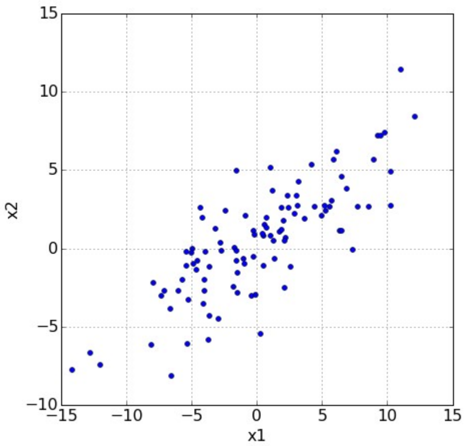
principle components analysis



- unsupervised
- dimensionality reduction
- feature extraction: principle components
  - orthogonal
  - direction of largest variance
- centered data

principle components analysis

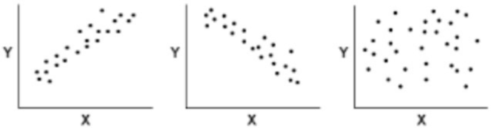


objective of PCA is to **rotate** the axes of the feature space to new positions (**principal axes**) that have the following properties:

- ordered such that principal axis 1 has the **highest variance**, axis 2 has the next highest variance, ... , and axis *m* has the lowest variance
- covariance among each pair of the principal axes is zero (the principal axes are **uncorrelated**).

principle components analysis

$$cov(x_i, x_j) = \sum_{k=1}^{n} \frac{(x_i^k - \bar{x}_i)(x_j^k - \bar{x}_j)}{n}$$



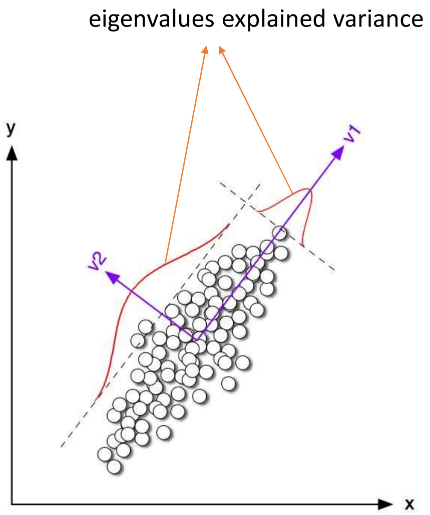
$$\begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

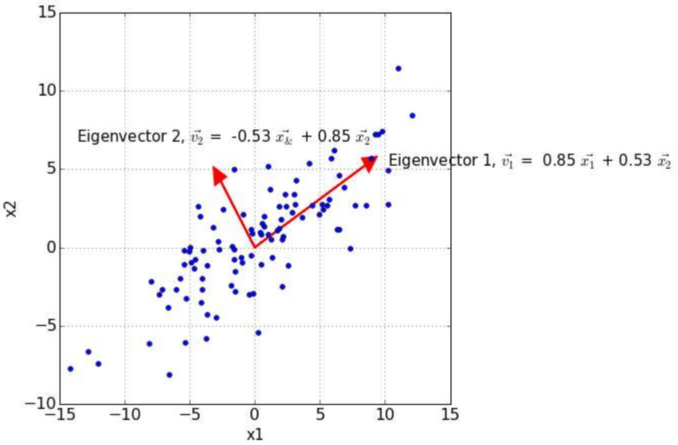
principle components analysis

$$A = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

$$Av = \begin{pmatrix} 1 & 2 \\ 8 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 5 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \lambda v$$

principle components analysis





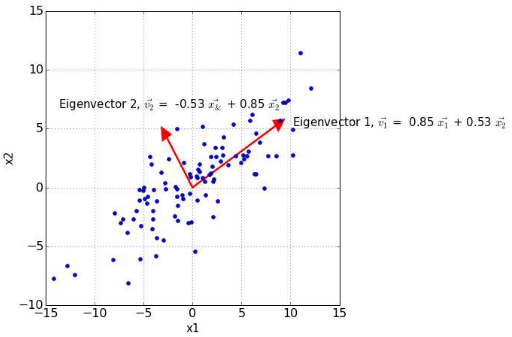
principle components analysis

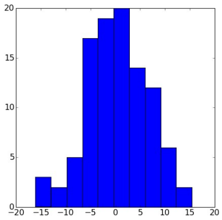
- The chart below shows the two PCs for a data set with two variables. The variance or eigenvalue of PC1 is 38.81 and of PC2 is 3.48. This means that PC1 **explains** 91.78% of the variance in the data set and PC2 explains 8.2%.
- The two PCs are $v = \begin{bmatrix} 0.85 & -0.53 \\ 0.53 & 0.85 \end{bmatrix}$
- To project a data point *x* = [x_1,x_2] onto the two PCs we use:

$$v^T x = \begin{bmatrix} 0.85 & 0.53 \\ -0.53 & 0.85 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- So, to reduce the dimensionality we project *x* only on PC1:

$$\begin{bmatrix} 0.85 & 0.53 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$





$\sigma^2 = 38.81$

```python
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
dataset_big_projected = pd.DataFrame(pca.fit(dataset_big).transform(dataset_big),
                                     columns=['PC1','PC2'])
dataset_big_projected['label'] = targets

sns.lmplot(x="PC1", y="PC2", data=dataset_big_projected, hue='label', markers=['+','o'],
           fit_reg=False, size=5.5, scatter_kws={"s": 80})
plt.show()
```

```python
print "Variance explained by PC1: %f" % pca.explained_variance_ratio_[0]
print "Variance explained by PC2: %f" % pca.explained_variance_ratio_[1]
```

multidimensional scaling

$$\begin{pmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,n} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,n} \\ \vdots & \vdots & & \vdots \\ \delta_{n,1} & \delta_{n,2} & \cdots & \delta_{n,n} \end{pmatrix}$$

- unsupervised
- dimensionality reduction
- data transformation
- starts from distance matrix Δ

The goal of MDS is to find vectors

$$x^{(1)}, \ldots, x^{(n)}$$

such that

$$\|x^{(i)} - x^{(j)}\| \approx \delta_{i,j} \qquad i,j \in 1, \ldots, n$$

```python
from sklearn import manifold

mds = manifold.MDS(n_components=2, max_iter=3000, eps=1e-9, random_state=2,
                   dissimilarity="precomputed", n_jobs=1)
mds_coordinates = mds.fit(sim_matrix_cities).embedding_

compomics_import.plot_scatter_annotated(pd.DataFrame(mds_coordinates),
                                        sim_matrix_cities.columns.values)
plt.show()
```