Dimensionality Reduction and Data Visualization

Wojtek Kowalczyk

Leiden Institute of Advanced Computer Science

Universiteit Leiden

Exploratory Data Analysis (EDA)

- EDA the process of getting an insight into the data with help of:
  - visual inspection of the data files
  - descriptive statistics/summaries
  - *plots, bar charts, histograms, ...*
  - feedback from domain experts
  - ***thinking!***

Goals of EDA:

- discover (and fix) obvious errors in data
- discover inconsistencies
- get an idea about data quality
- get a feeling of what is important and what is not
- make first discoveries
- get some feedback from domain experts
- formulate first hypotheses

Case 1: German Credit Data

https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/

- 1000 records, 300 “bad”, 700 “good” clients
- 20 attributes: 7 numerical, 13 categorical
- cost matrix:

misclassifying a bad client as a good one costs 5x more than misclassifying a good client as a bad one

A\P	BAD	GOOD
BAD	0	5
GOOD	1	0

Some variables

Age

Sex

Loan duration

Amount

Credit history

Housing type: own, rented, social

AGE

mean=35.54

median=33

mode=27

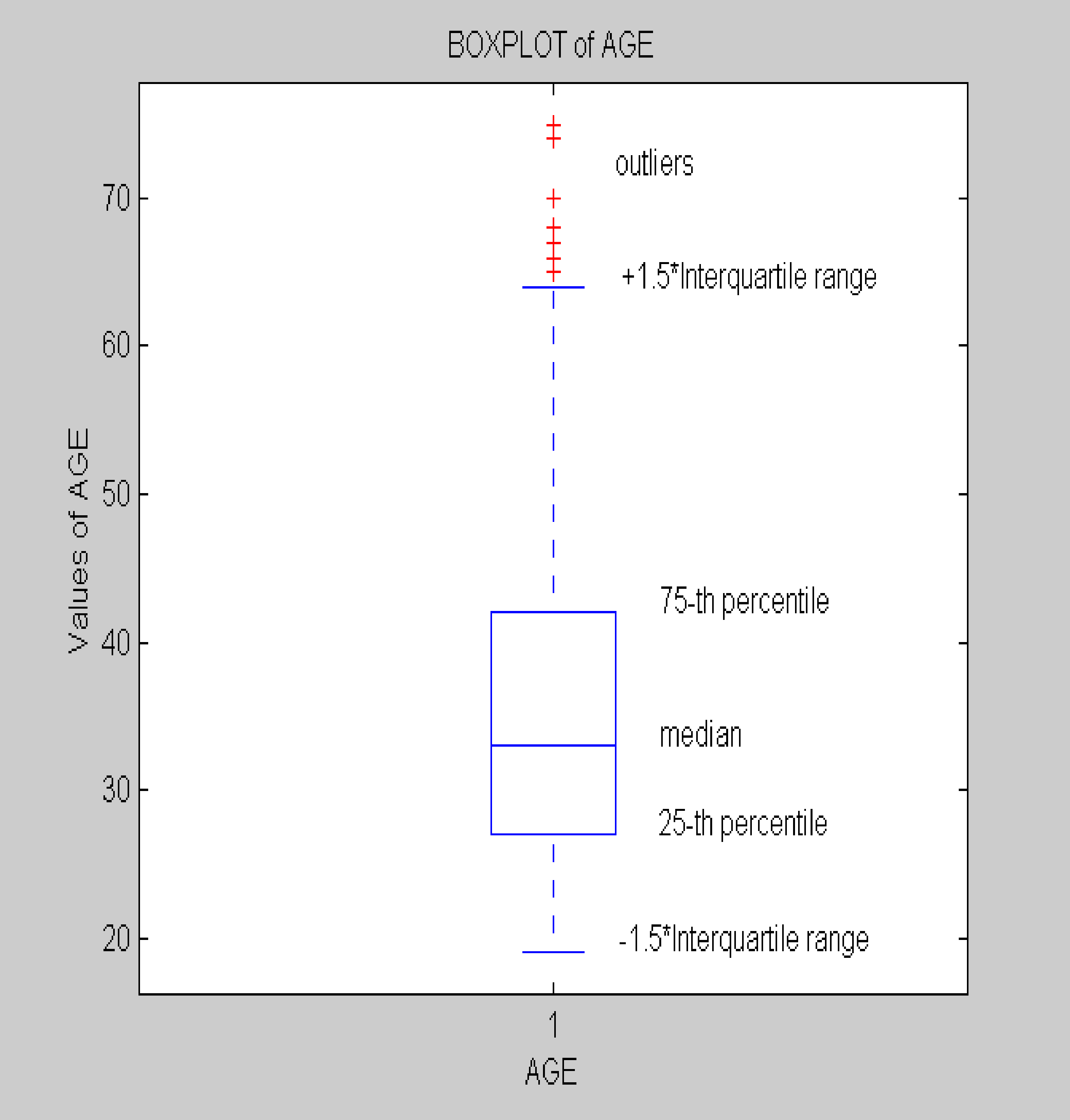
range=56

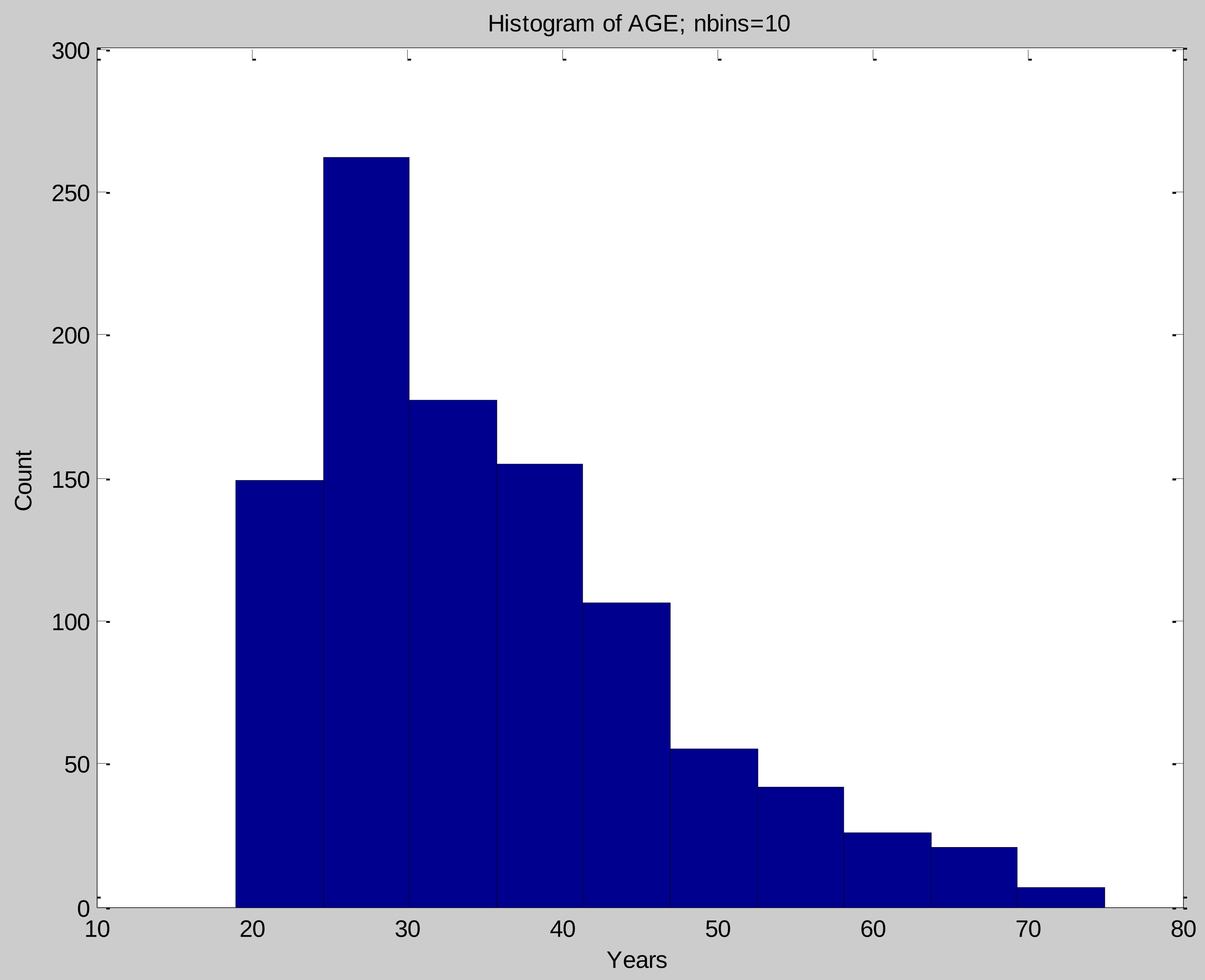
iqr=15

std=11.37

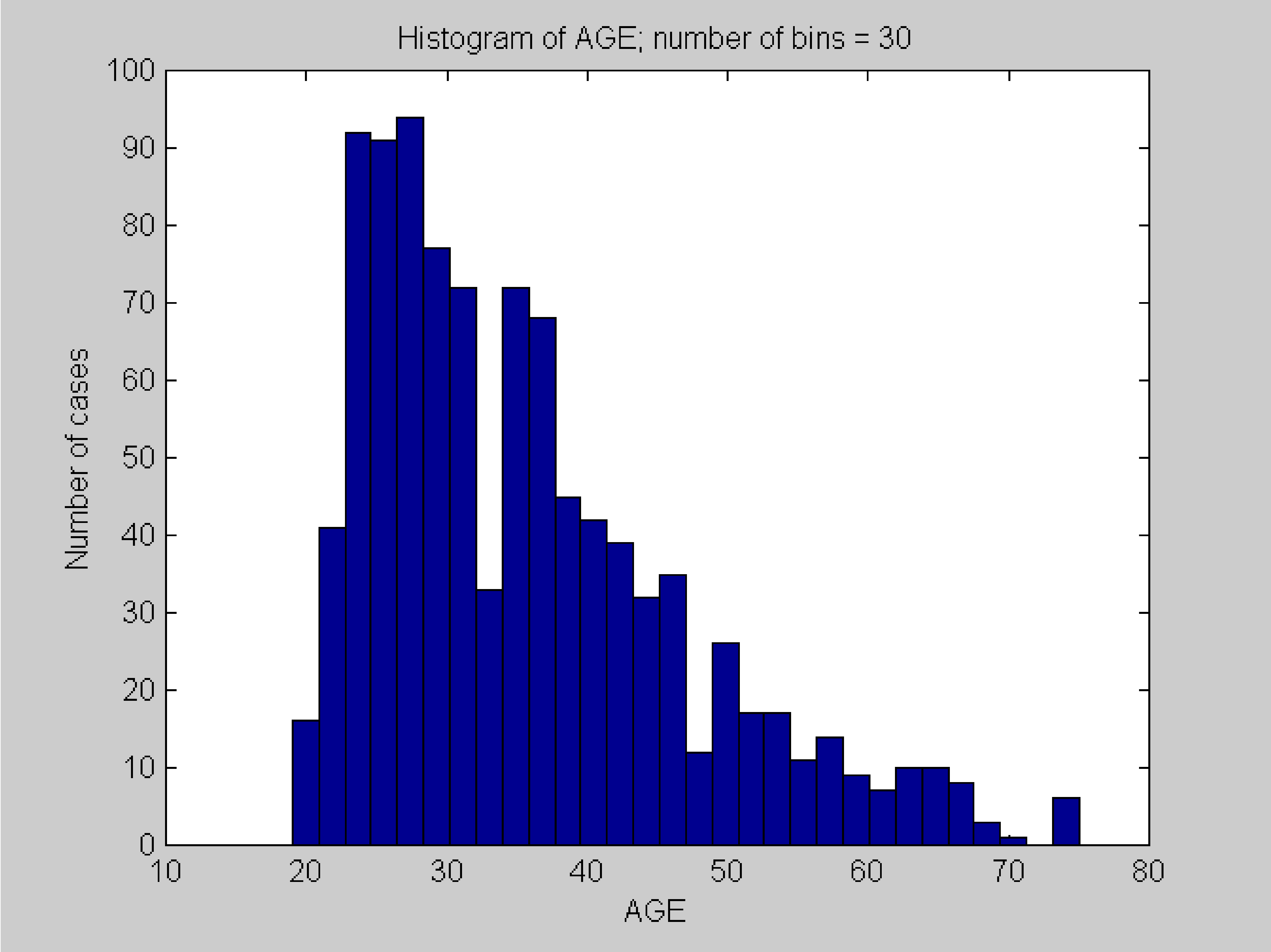
skewness=1.01

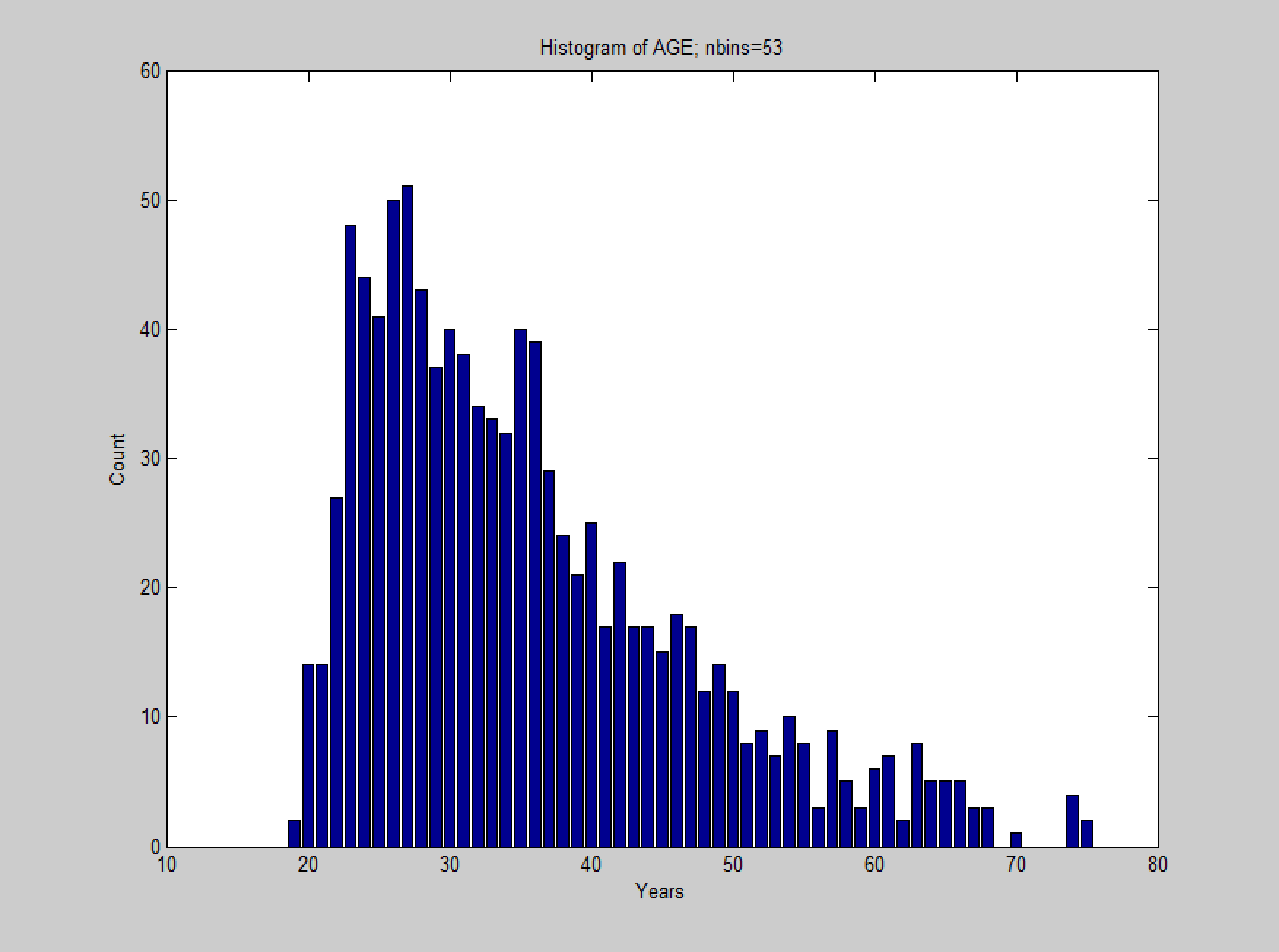
kurtosis=3.58





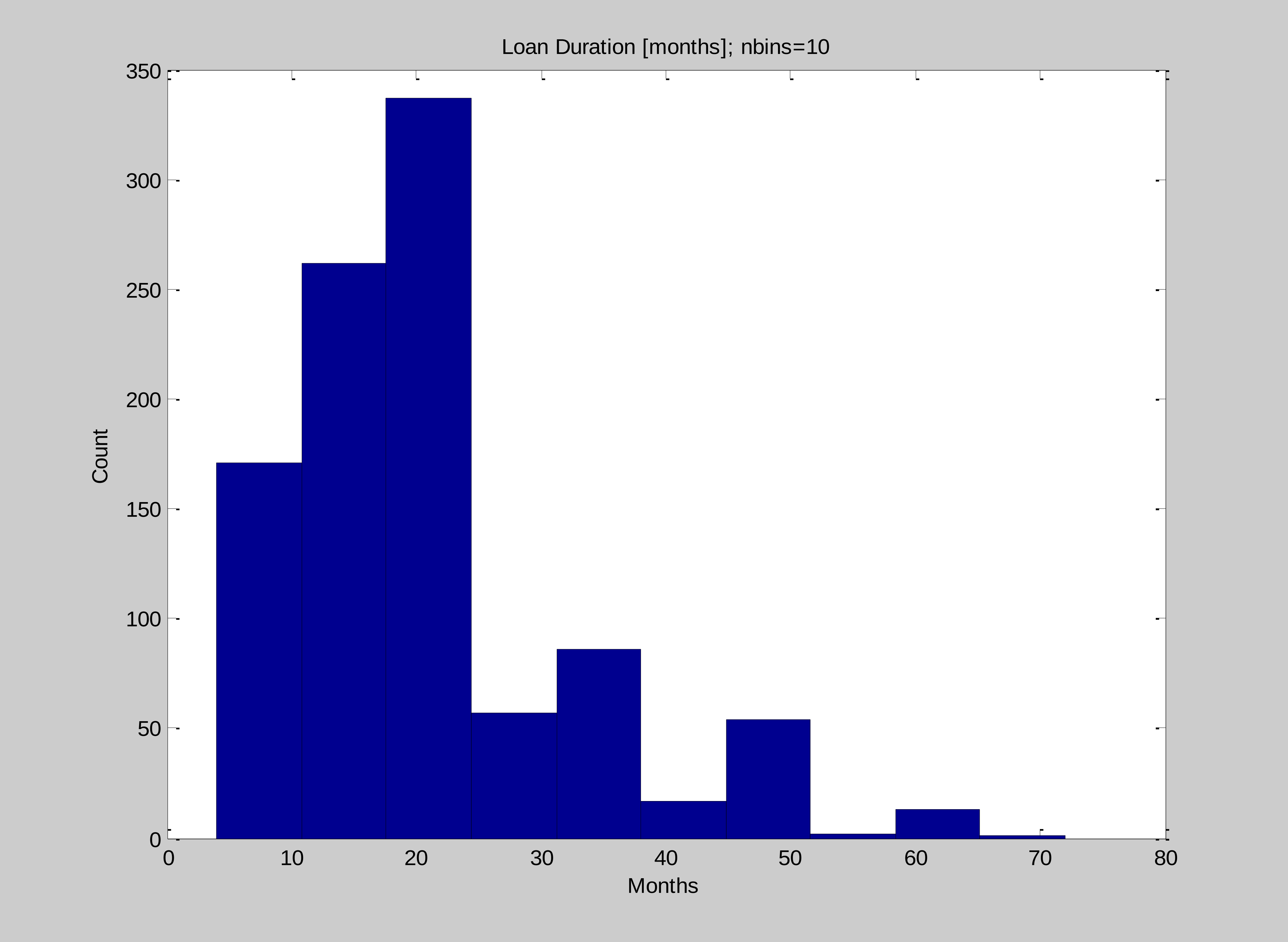
Histogram of AGE; nbins=10
Count
Years
300
250
200
150
100
50
0
10
20
30
40
50
60
70
80


Histogram of AGE; number of bins = 30
Number of cases
AGE
0
10
20
30
40
50
60
70
80
90
100
10
20
30
40
50
60
70
80

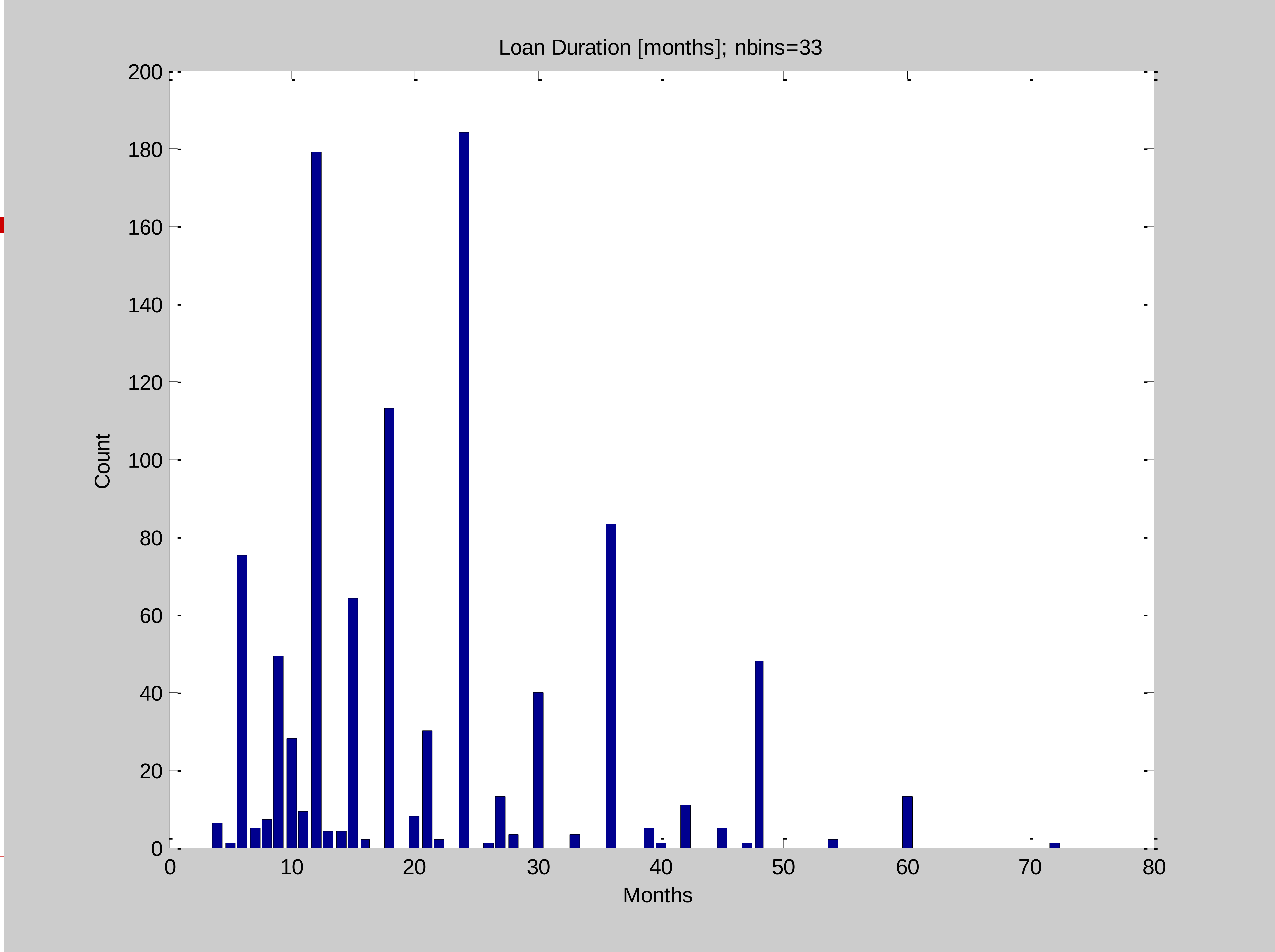

Histogram of AGE; nbins=53
Count
Years
0
10
20
30
40
50
60
10
20
30
40
50
60
70
80

Observations

- descriptive statistics (mean, median, …) are not very informative
- boxplot gives a rough idea of the distribution; main advantage: may be used to visualize several variables on a single figure
- histogram: very useful, but might be misleading (when the number of bins is badly chosen)!



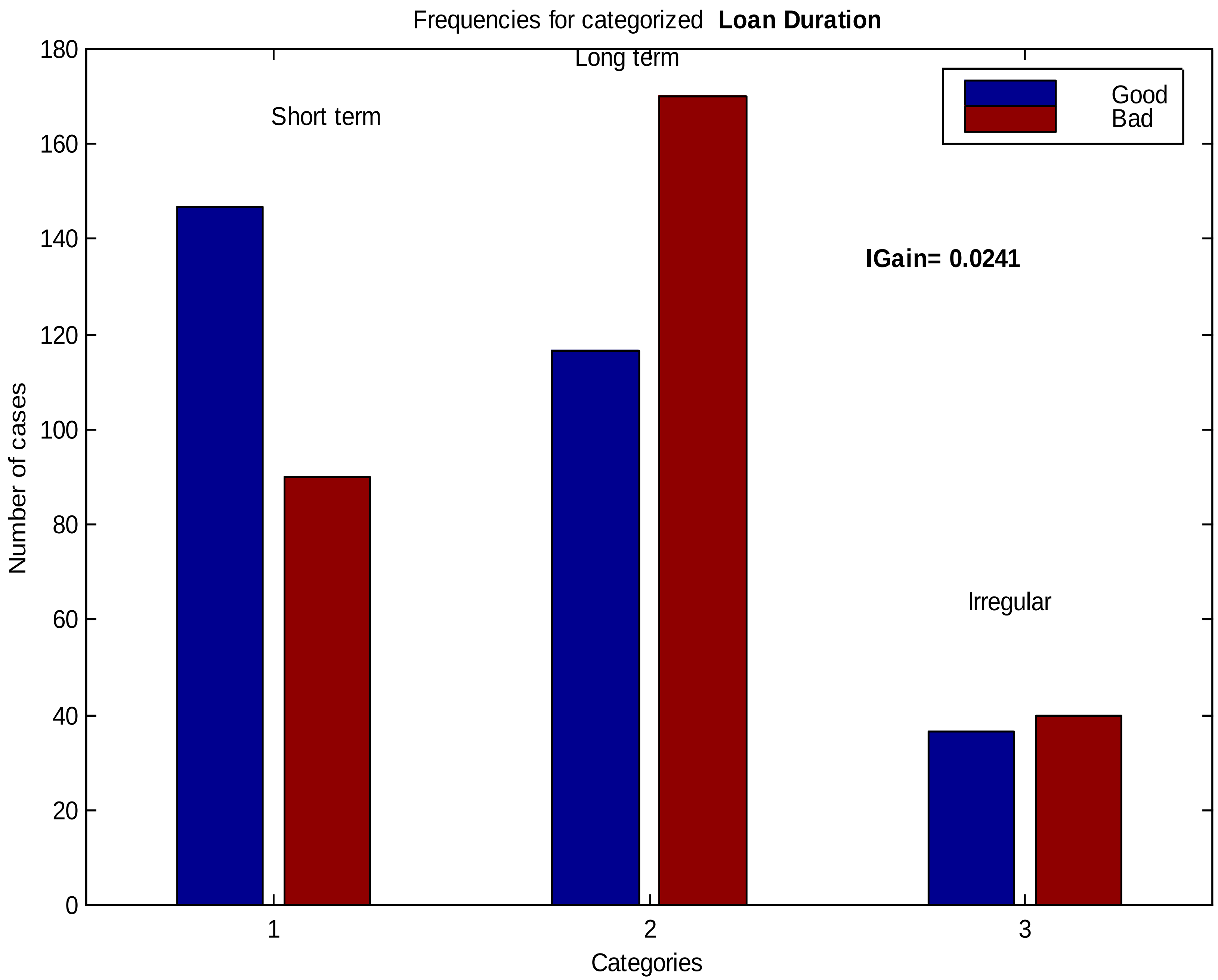
Loan Duration [months]; nbins=10
Count
Months
0
50
100
150
200
250
300
350
0
10
20
30
40
50
60
70
80


Loan Duration [months]; nbins=33
Count
Months
0
20
40
60
80
100
120
140
160
180
200
0
10
20
30
40
50
60
70
80

Observations

- A bar chart with frequencies of possible values is very useful: it allows us to identify 3 groups of loan duration:

   A) typical 'short term loans': shorter than 18 months
   B) typical 'long term loans': 18, 24, 30, 36, 48, 60 months
   C) 'special arrangements': all other durations


Frequencies for categorized **Loan Duration**
Long term
Short term
Irregular
Good
Bad
IGain= 0.0241
Number of cases
Categories
0
20
40
60
80
100
120
140
160
180
1
2
3

Iris Data Set: visualisation

- 150=3x50 records, 3 categories of Iris
- 4 variables

	Sepal length	Sepal width	Petal length	Petal width	Type
1	5.1	3.5	1.4	0.2	Iris setosa
2	4.9	3.0	1.4	0.2	Iris setosa
…					
51	7.0	3.2	4.7	1.4	Iris versicolor
52	6.4	3.2	4.5	1.5	Iris versicolor
…					
101	6.3	3.3	6.0	2.5	Iris virginica
102	5.8	2.7	5.1	1.9	Iris virginica
…					

Visualisation multi-dimensional data

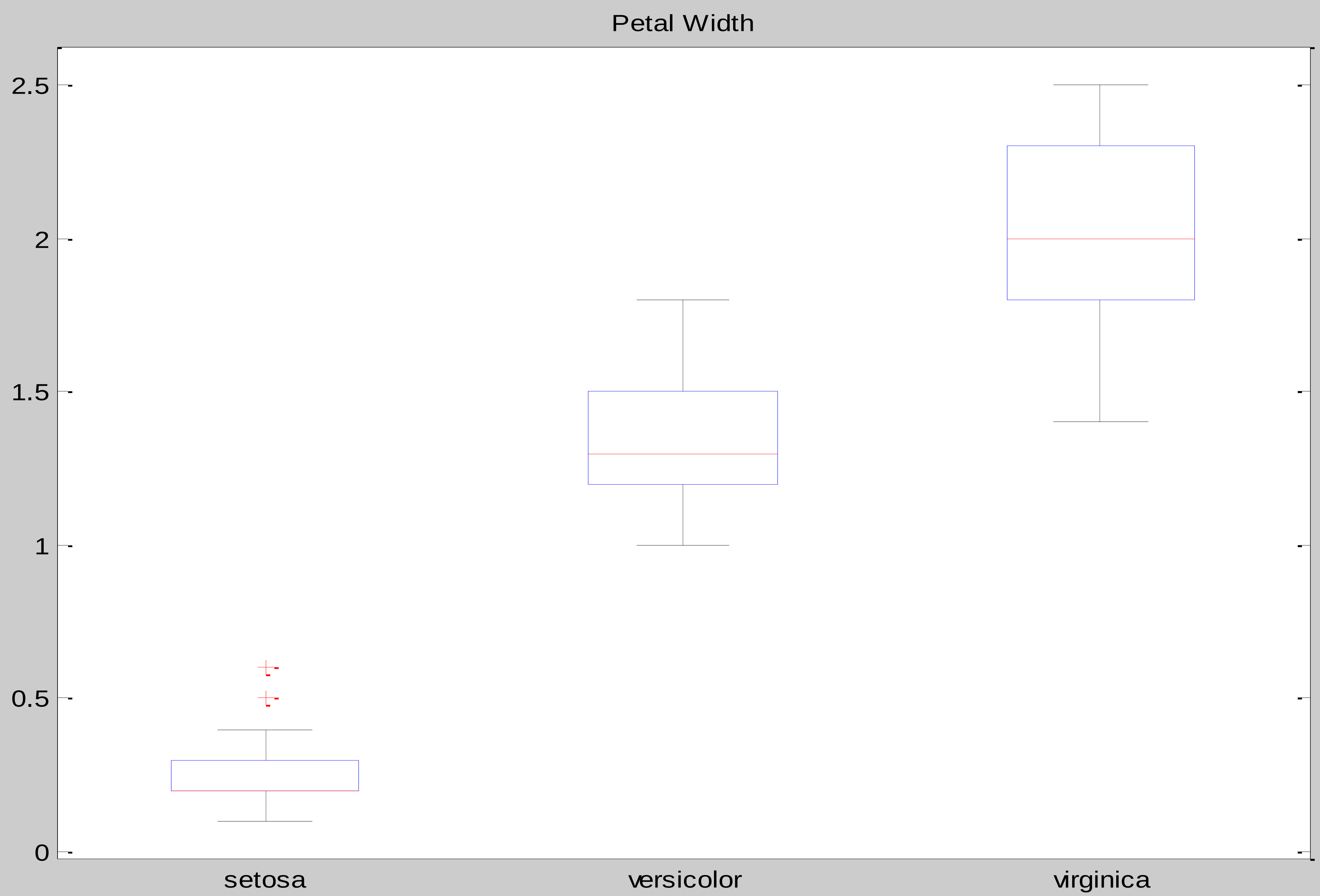
- Multiple boxplots
- 3D interactive scatter plots

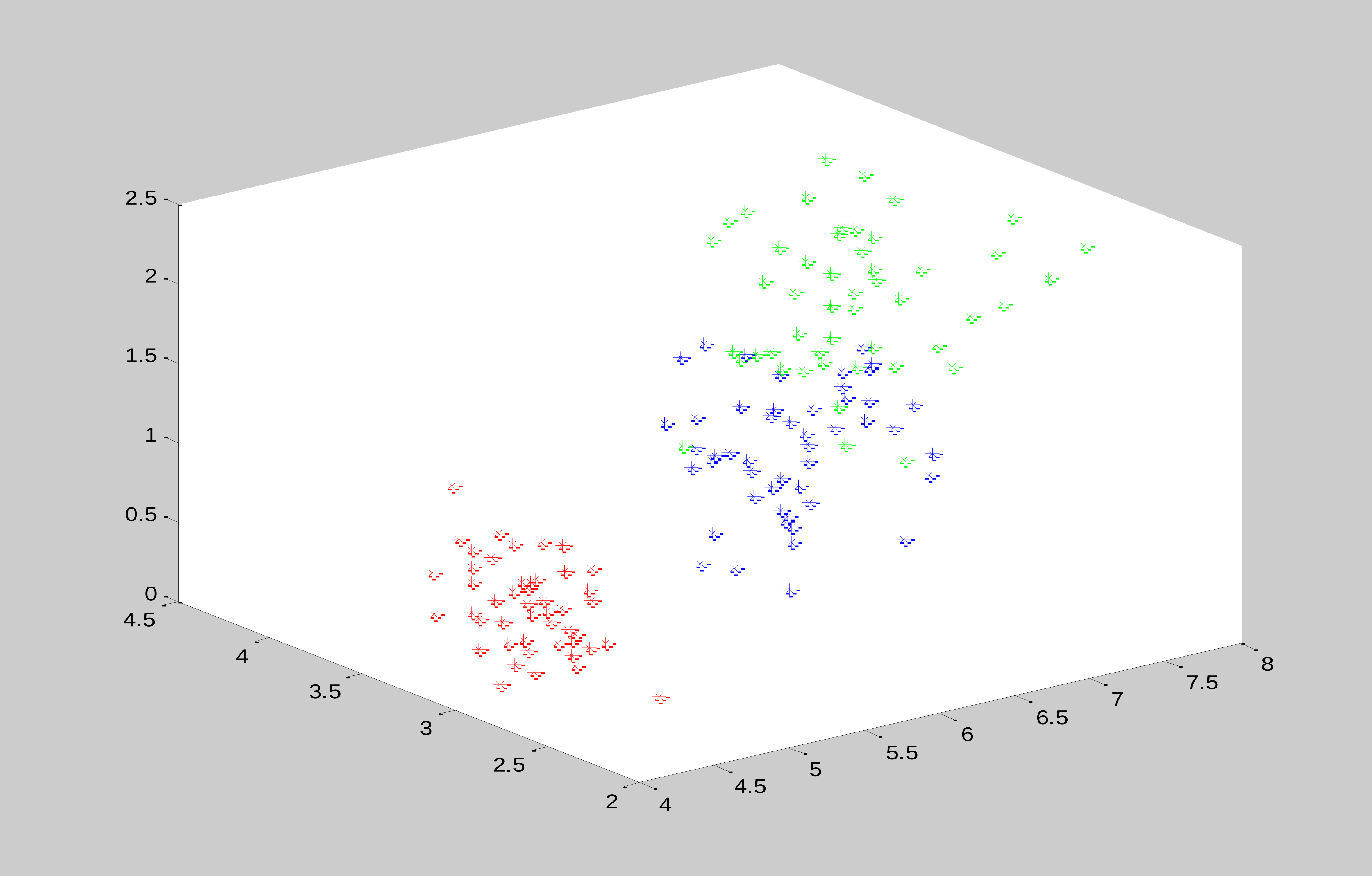
- **Parallel Coordinates Plots:**
  each record is plotted as a line

- **Andrews Plots:**
  a tuple of numbers defines a “smooth” curve
  (e.g., (a, b, c) -> ax^2+bx+c), so every record can be “plotted” as a curve. [Actually, instead of polynomials, trigonometric functions are used]

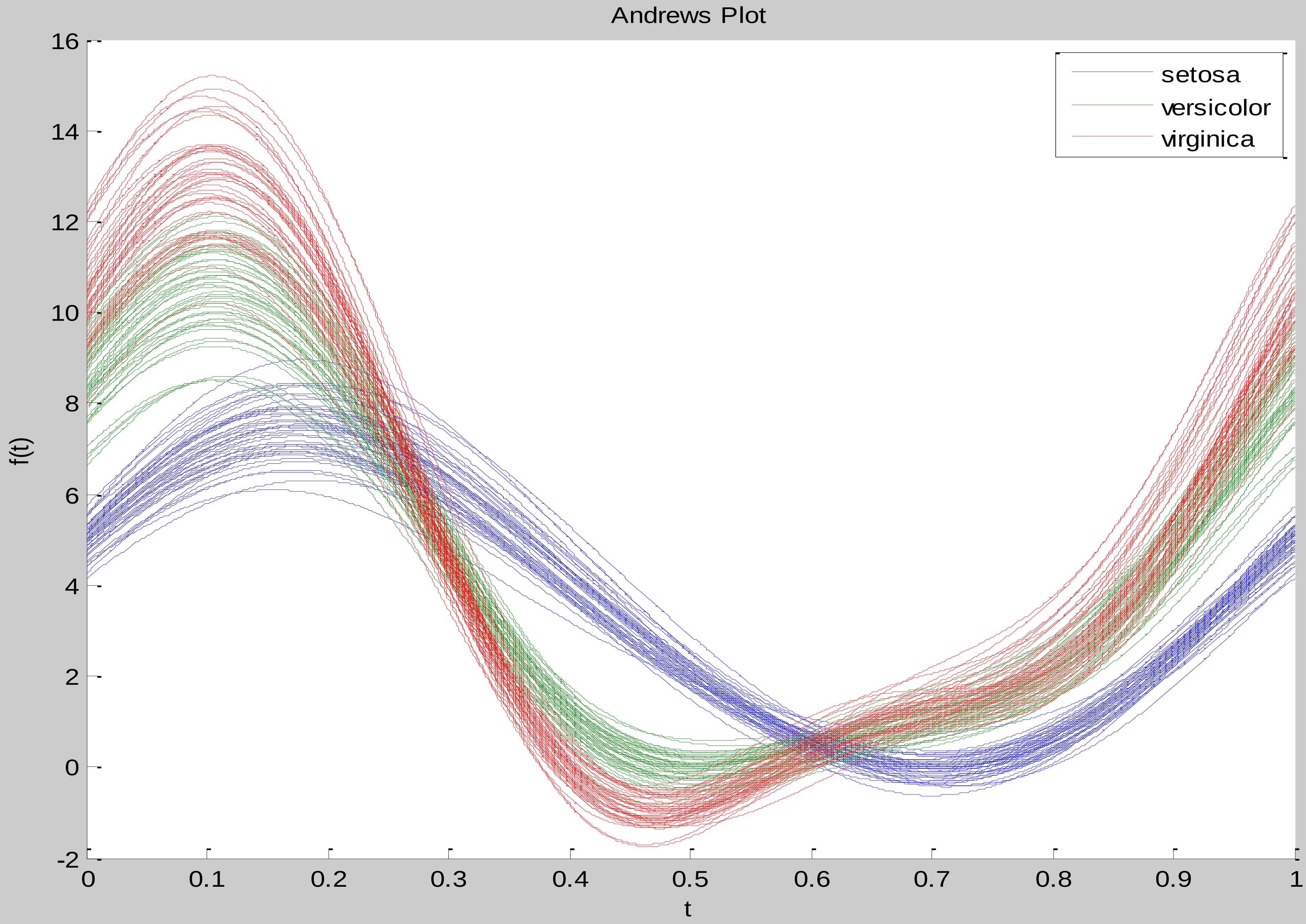
- **Glyphplots**

- **Principal Components (dimensionality reduction)**

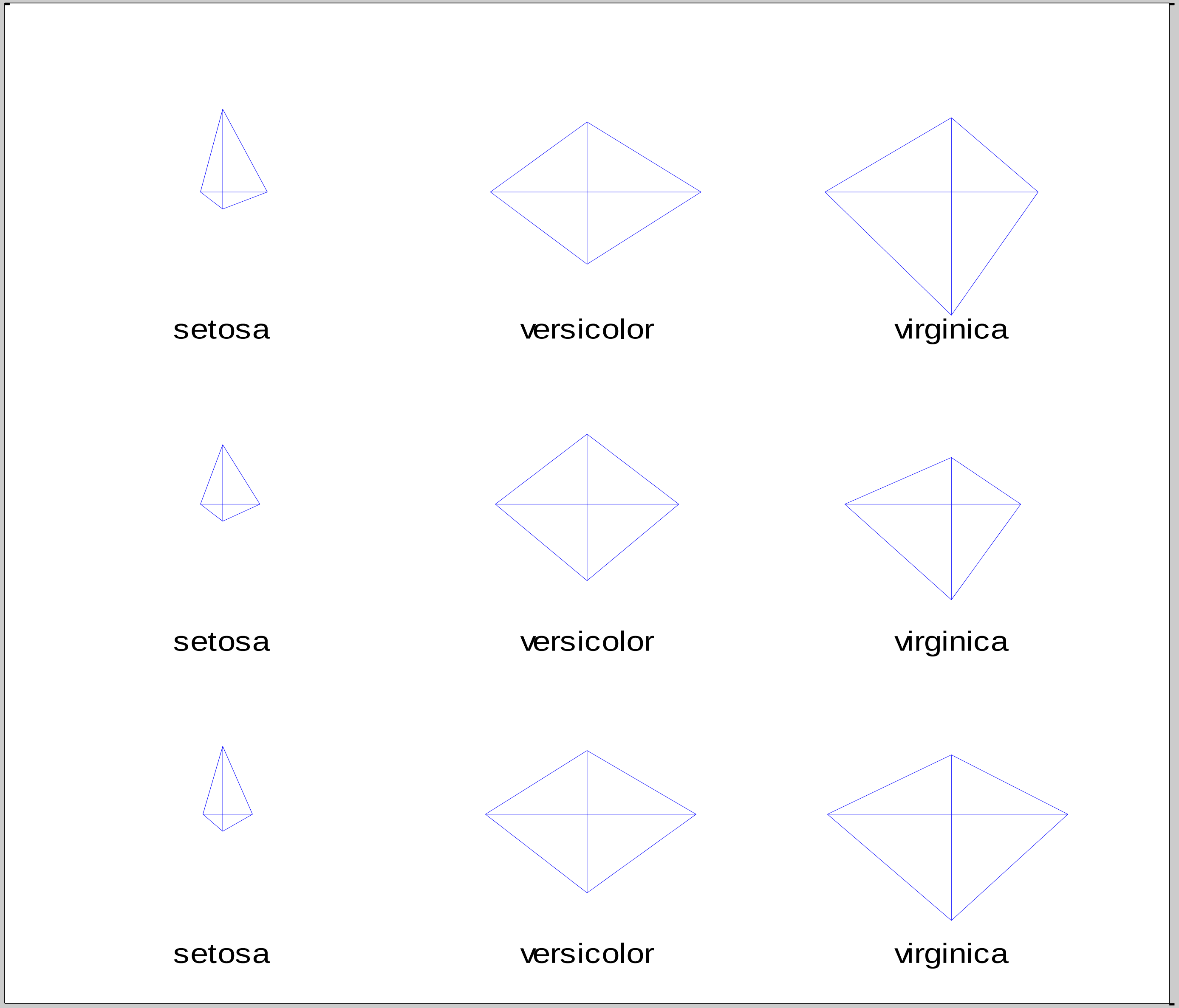

Petal Width
2.5
2
1.5
1
0.5
0
setosa
versicolor
virginica

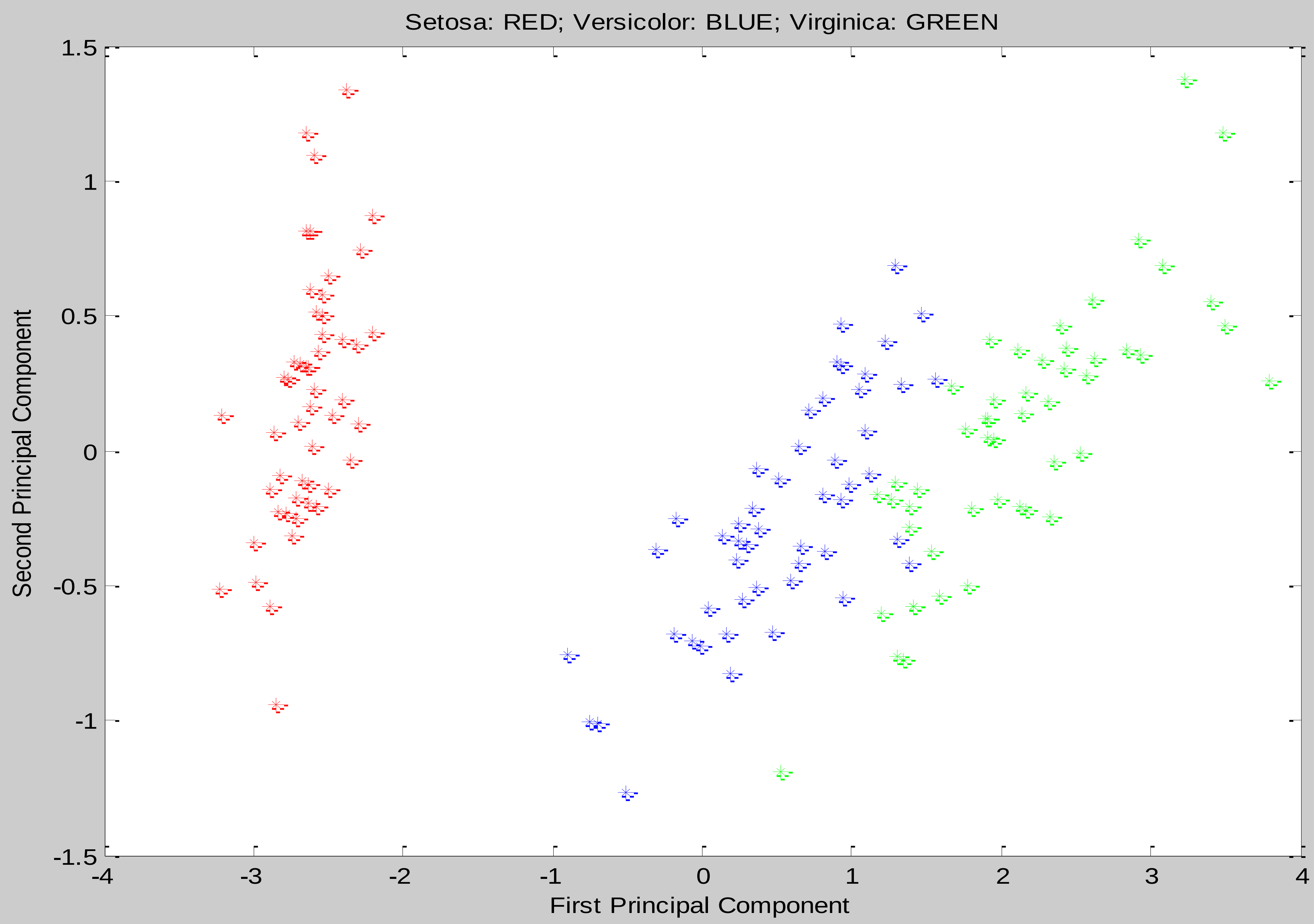

2.5
2
1.5
1
0.5
0
4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8


Parallel Coordinates
Coordinate Value
0
1
2
3
4
5
6
7
8
Sepal Length
Sepal Width
Petal Length
Petal Width
setosa
versicolor
virginica



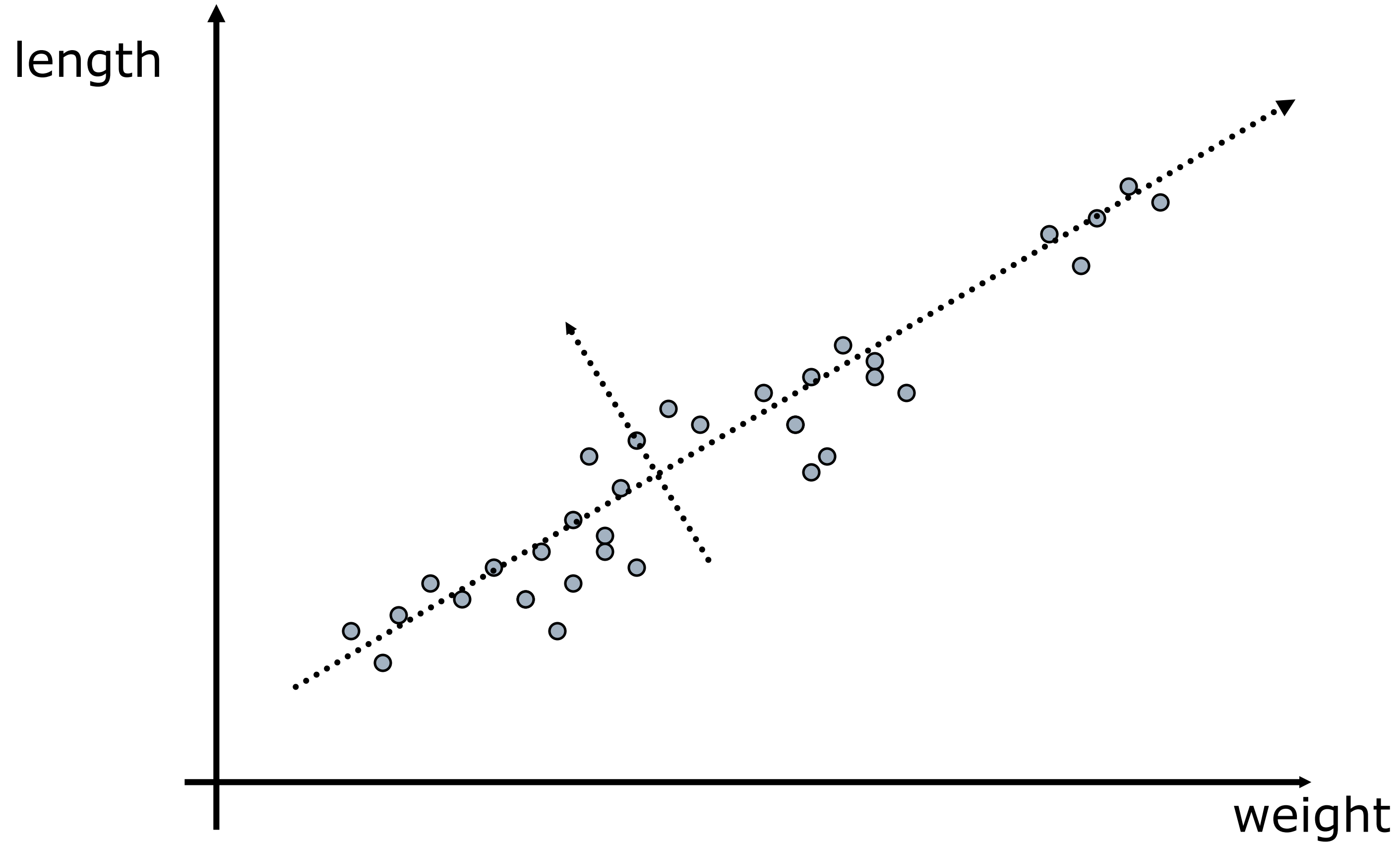
Andrews Plot
f(t)
t
setosa
versicolor
virginica
16
14
12
10
8
6
4
2
0
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


setosa
versicolor
virginica
setosa
versicolor
virginica
setosa
versicolor
virginica



Setosa: RED; Versicolor: BLUE; Virginica: GREEN
Second Principal Component
First Principal Component
1.5
1
0.5
0
-0.5
-1
-1.5
-4
-3
-2
-1
0
1
2
3
4

Principal Component Analysis



Principal Component Analysis

- **Given:** n variables $x_1, x_2, \ldots, x_n$
- PCs are linear combinations of $x_1, \ldots, x_n$ such that:

  1) they are orthogonal to each other

  2) they maximize the "variance of projections"

  3) the first PC explains most of the variance, second PC less,

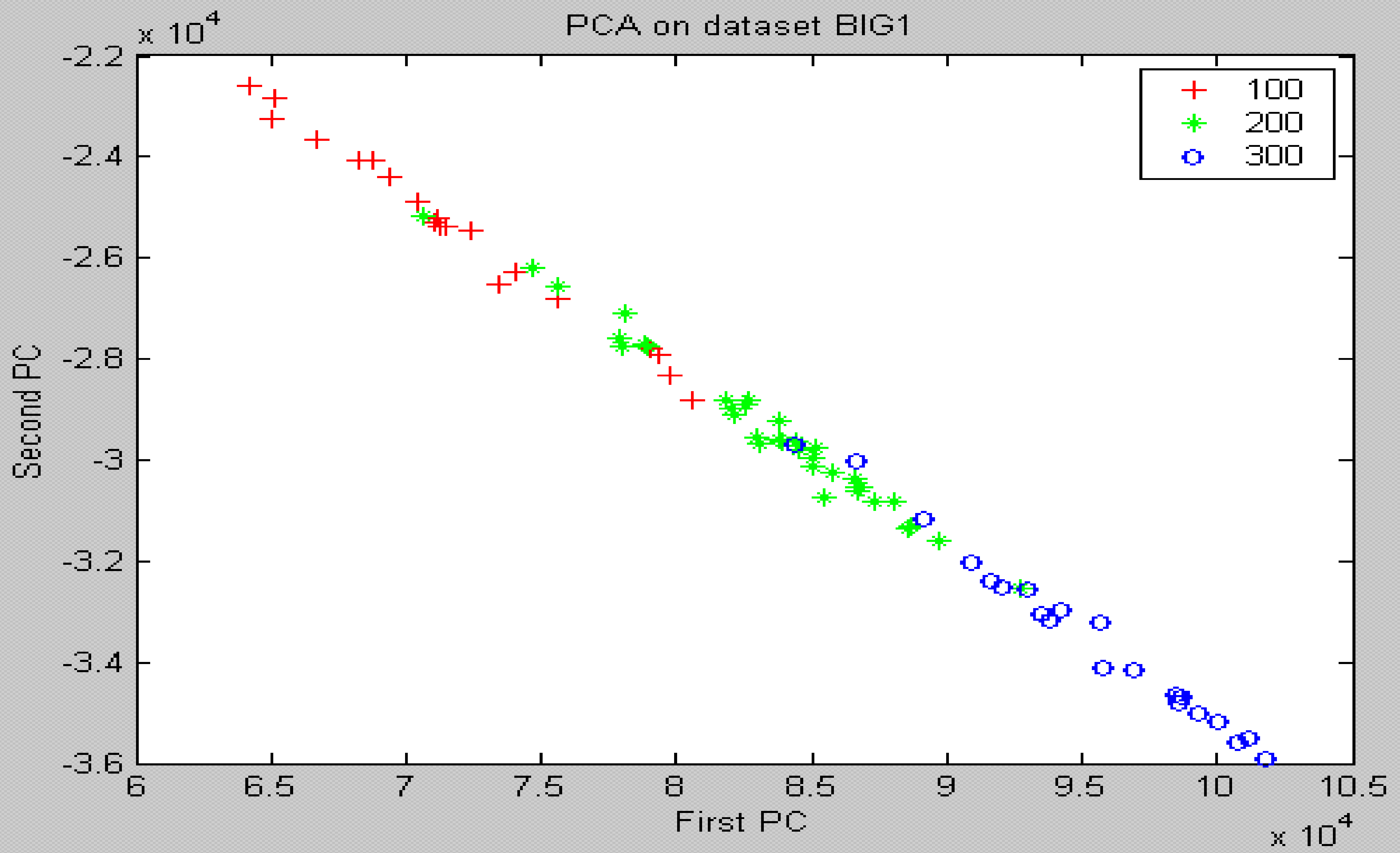
- **In practice the first few PCs (2-3) explain most of the variance**

Example application of PCAs

- a digital camera takes several photos of a flower
- some features are extracted from images (sizes, proportions, diameters, etc): 30 numbers
- some flowers are manually labeled as “nice”; others as “ugly”

Problem:

develop a system that would automatically estimate flower quality from 30 features


PCA on dataset BIG1
x 10^4
-2.2
-2.4
-2.6
-2.8
-3
-3.2
-3.4
-3.6
Second PC
6
6.5
7
7.5
8
8.5
9
9.5
10
10.5
First PC
x 10^4
100
200
300

More advanced dimensionality reduction methods

- Variants of PCA
- Multi Dimensional Scaling
- Locally Linear Embeddings
- …
- “Latent Spaces” (Variational Autoencoders, Generative Adversarial Networks)

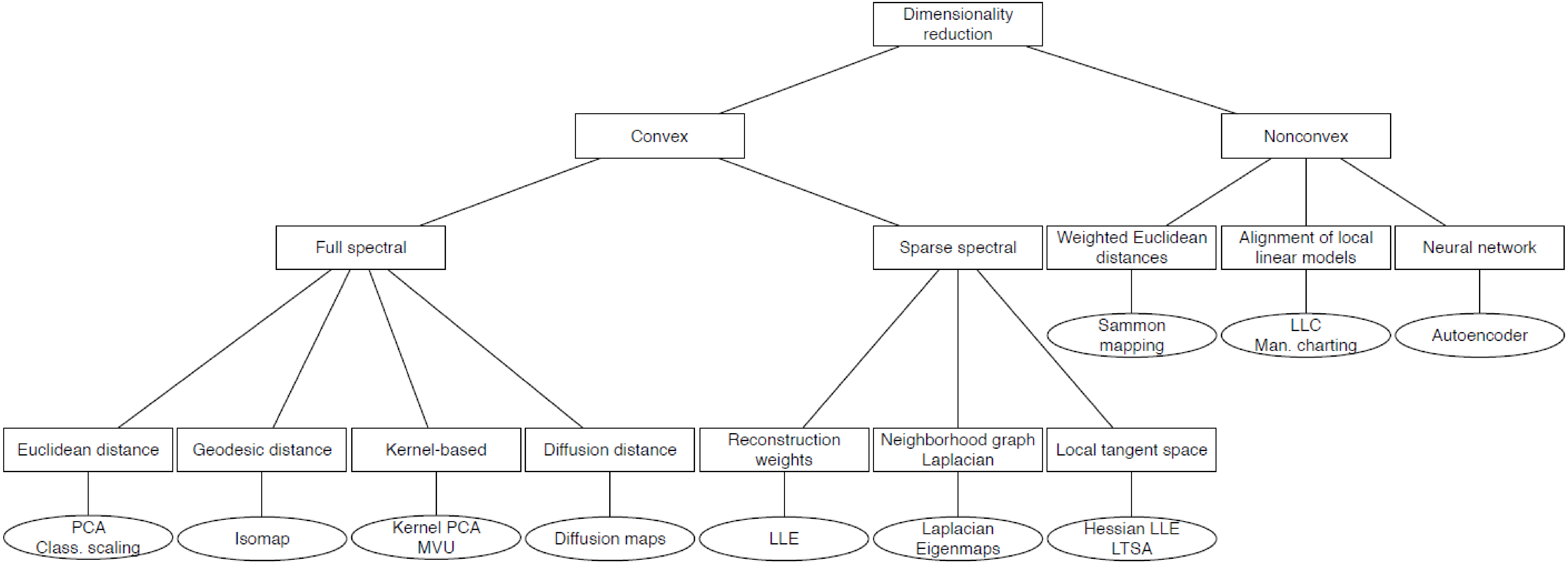


Figure 1: Taxonomy of dimensionality reduction techniques.

L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik.
Dimensionality Reduction: A Comparative Review

Metric Multi-Dimensional Scaling: the key idea

Given n points $p_1, \ldots, p_n$ (in a highly dimensional space)

find a mapping $p_i \to q_i$ (q_i in a low dimensional space)
that preserves the original distances as much as possible:

$$\mathrm{dist}(p_i, p_j) \approx \mathrm{dist}(q_i, q_j) \text{ for all } i, j;$$

$$\text{Total error: } \mathrm{sum}((\mathrm{dist}(p_i,p_j) - \mathrm{dist}(q_i,q_j)^2)$$

An optimization problem! PCA does (most of) the job!

General Multi-Dimensional Scaling: the key idea

Given n *objects* $p_1, \ldots, p_n$ and a similarity measure between them

find a mapping $p_i \to q_i$ (q_i in a low dimensional space)
that preserves the original distances as much as possible:

$$\mathit{sim}(p_i, p_j) \approx \mathrm{dist}(q_i, q_j) \text{ for all } i, j;$$

Total error: $\mathrm{sum}((\mathit{sim}(p_i,p_j) - \mathrm{dist}(q_i,q_j)^2)$

An optimization problem! Initialize at random; iterate SGD

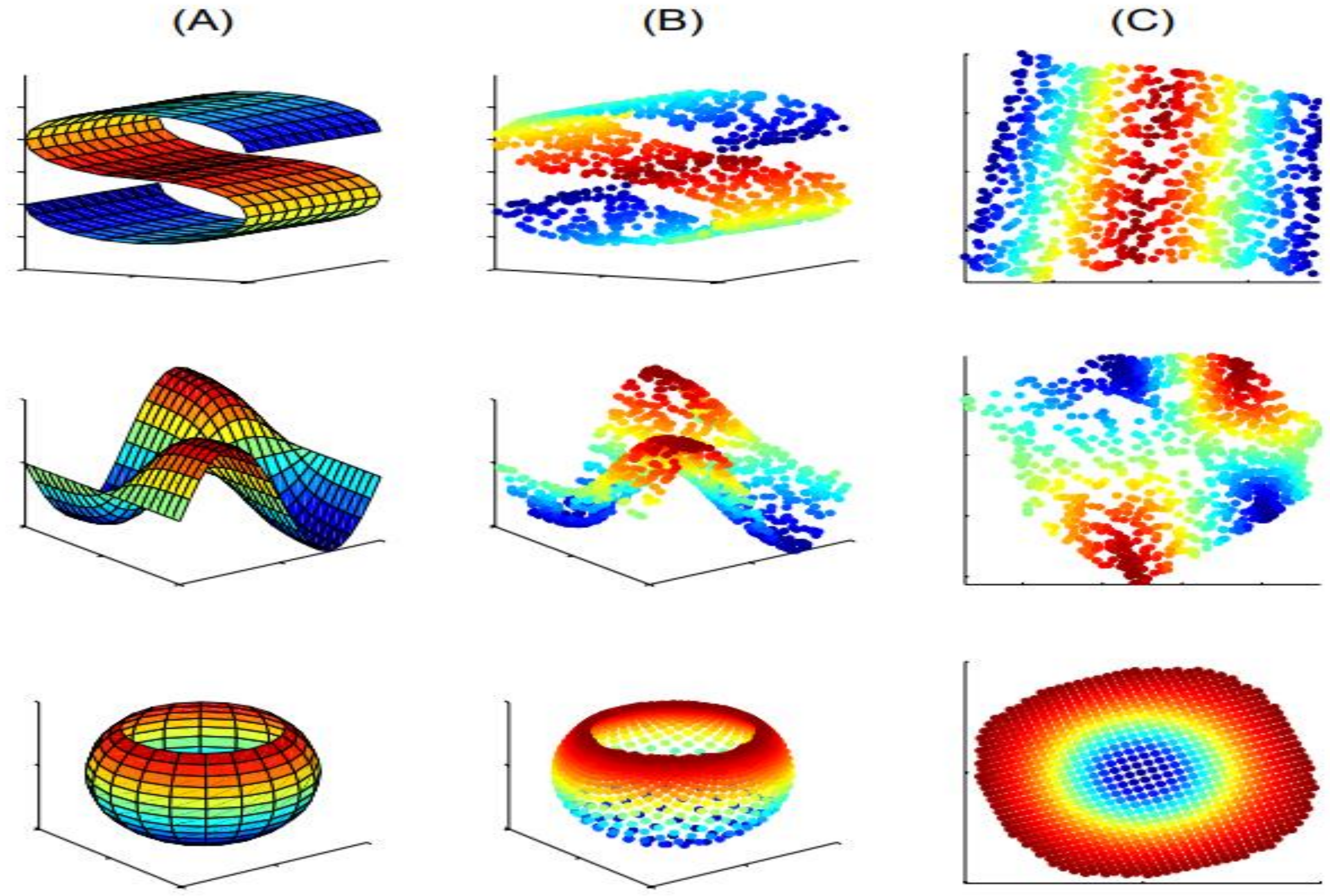
Locally Linear Embeddings

Often, interesting data “lives” in “highly dimensional spaces” (images, word embeddings, spectra, …)

We want to see it in low dimensions!

The key idea:

- for each object X_i find a few neighboring objects;
- measure distances between X_i and these neighbours
- find Y_i in low dimensional space that preserve all mutual distances => a very simple optimization problem!


(A)
(B)
(C)

1. **Compute the neighbors of each data point, $\vec{X}_i$.**

2. **Compute the weights W_{ij} that best reconstruct each data point $\vec{X}_i$ from its neighbors, minimizing the cost in Equation (1) by constrained linear fits.**

3. **Compute the vectors $\vec{Y}_i$ best reconstructed by the weights W_{ij}, minimizing the quadratic form in Equation (2) by its bottom nonzero eigenvectors.**

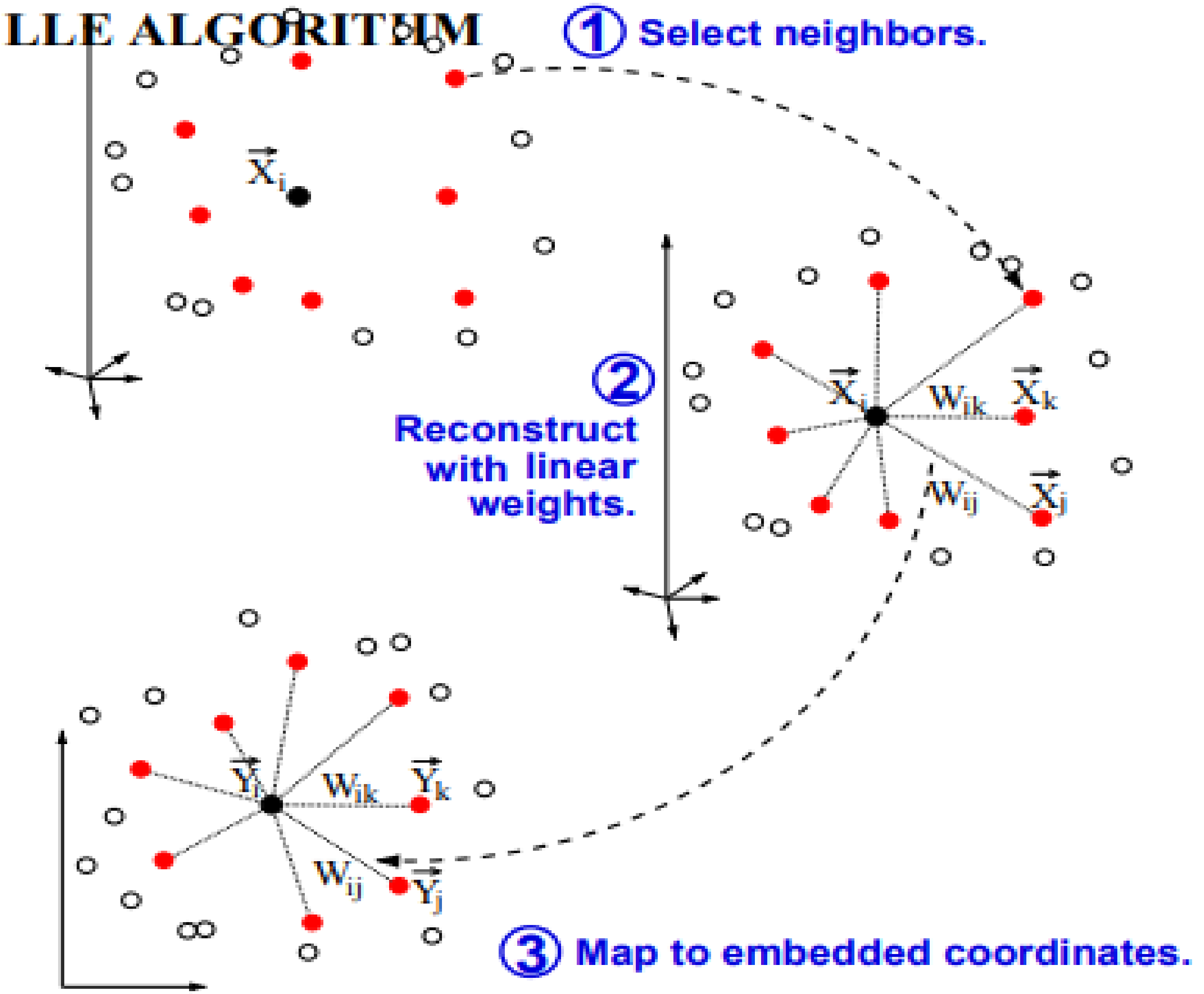


Figure 2: Summary of the LLE algorithm, mapping high dimensional inputs $\vec{X}_i$ to low dimensional outputs $\vec{Y}_i$ via local linear reconstruction weights W_{ij}.

References:

- An overview of dimensionality reduction methods:
  https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf

- LLE
  https://www.jmlr.org/papers/volume4/saul03a/saul03a.pdf

- t-SNE
  https://www.youtube.com/watch?v=EMD106bB2vY
  https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf

- PCA/Dimensionality reduction
  Chapter 11, MMDS book (http://www.mmds.org/)