Ollscoil na Gaillimhe
University of Galway

CT4101
Machine Learning

Dr. Frank Glavin
Room 404, IT Building
Frank.Glavin@University*of*Galway.ie

School of Computer Science

University *of* Galway.ie

Data Processing and Exploration

University *of* Galway.ie

Summary of topic

- Data normalisation
- Binning
- Sampling
- The Curse of Dimensionality & feature selection
- Covariance and correlation matrices


Ollscoil na Gaillimhe
University of Galway
Data Normalisation
University
ofGalway.ie

Data normalisation

Problem — Scaling:

Attribute 1 has range 0-10,

Attribute 2 has range 0-1000

Attribute 2 will dominate calculations

Solution:

Rescale all dimensions independently

Mean=0, Std deviation=1 [Z-Normalisation]

(HousePrices-1NN.xlsx on Blackboard has a worked example with k-NN)

D ← (D - Mean) / StDev

Min=0, Max=1 [0-1 Normalisation] (also referred to as “range normalisation”)

D ← (D – Min) / (Max – Min)

Normalisation is important in many other areas of machine learning and optimisation

Normalisation example

	Height			Sponsorship Earnings		
	Values	Range	Standard	Values	Range	Standard
	192	0.500	-0.073	561	0.315	-0.649
	197	0.679	0.533	1,312	0.776	0.762
	192	0.500	-0.073	1,359	0.804	0.850
	182	0.143	-1.283	1,678	1.000	1.449
	206	1.000	1.622	314	0.164	-1.114
	192	0.500	-0.073	427	0.233	-0.901
	190	0.429	-0.315	1,179	0.694	0.512
	178	0.000	-1.767	1,078	0.632	0.322
	196	0.643	0.412	47	0.000	-1.615
	201	0.821	1.017	1111	0.652	0.384
Max	206			1,678		
Min	178			47		
Mean	193			907		
Std Dev	8.26			532.18		

Normalisation in scikit-learn

It is generally good practice to normalise continuous variables before developing an ML model. Some algorithms (e.g. *k*-NN) are much more susceptible to the effects of the relative scale of attributes than others (e.g. decision trees are more robust to the effects of scale)

z-normalisation is easily accomplished in scikit-learn using the `StandardScaler` utility class
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
0-1 normalisation can be accomplished using the `MinMaxScaler` utility class
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

See the following link for more information:
https://scikit-learn.org/stable/modules/preprocessing.html#standardization-or-mean-removal-and-variance-scaling

Binning

University *of* Galway.ie

Binning

Binning involves converting a continuous feature into a categorical feature

To perform binning, we define a series of ranges (called **bins**) for the continuous feature that correspond to the levels of the new categorical feature we are creating.

We will introduce two of the more popular ways of defining bins:

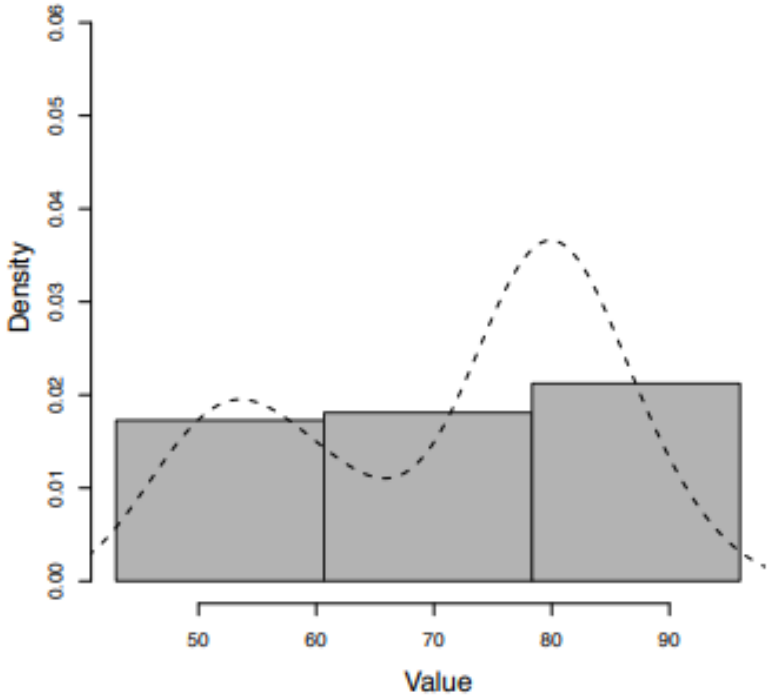
- equal-width binning
- equal-frequency binning

Setting the number of bins

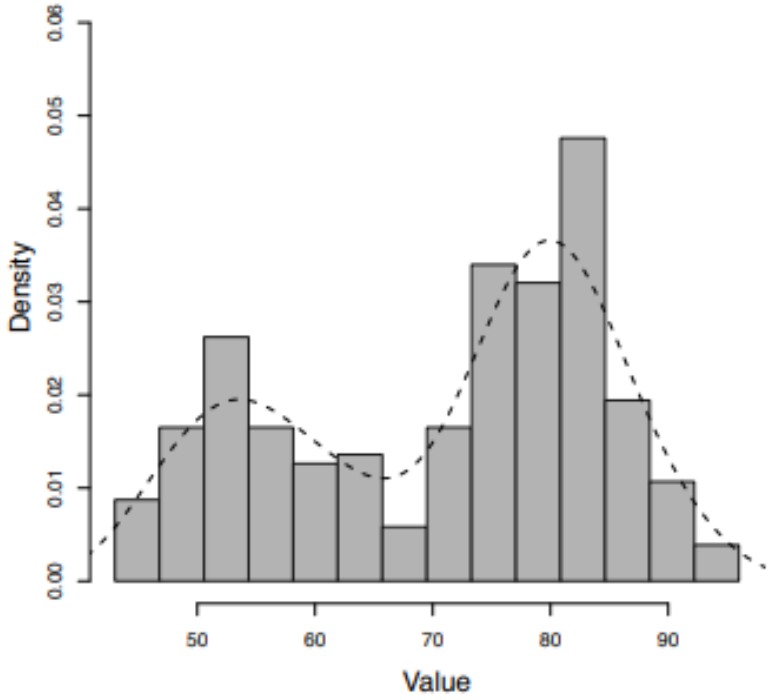
Deciding on the number of bins can be difficult. The general trade-off is this:

- If we set the number of bins to a very low number we may lose a lot of information
- If we set the number of bins to a very high number then we might have very few instances in each bin or even end up with empty bins.

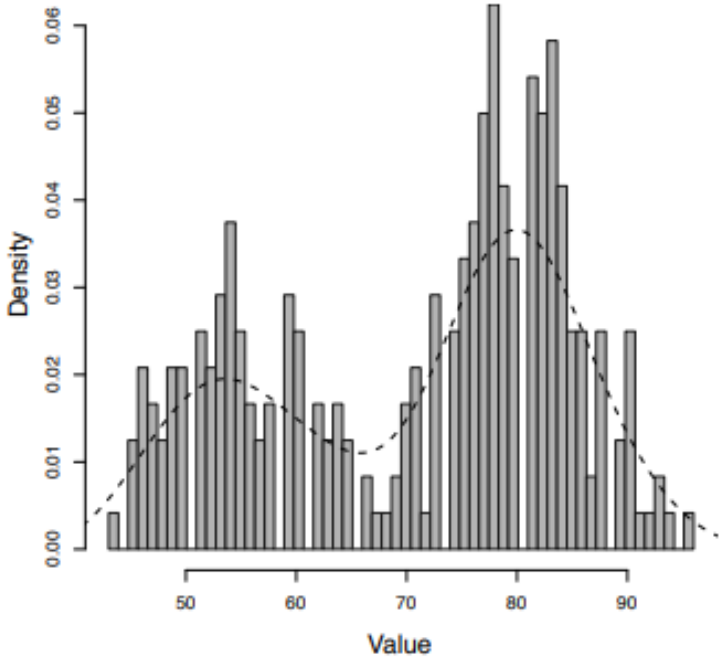
The effect of different numbers of bins



(e) 3 bins



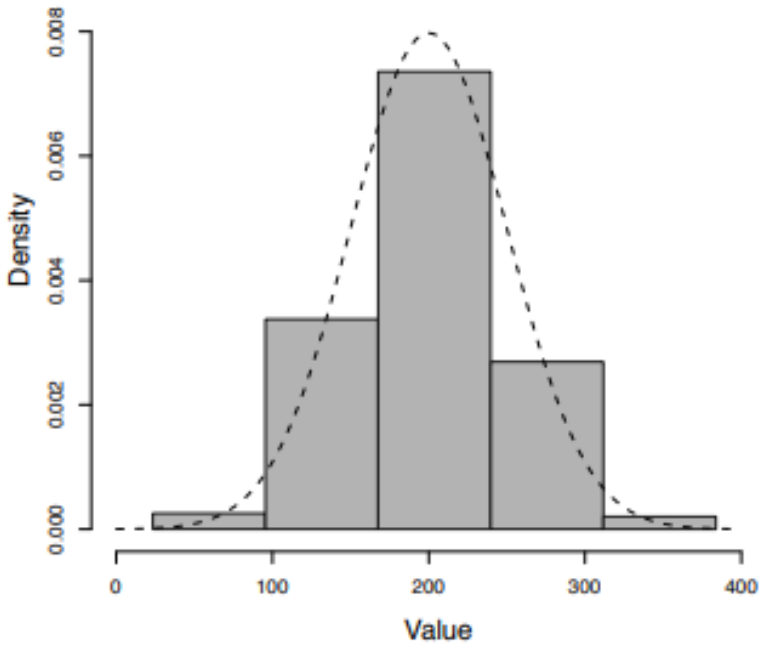
(f) 14 bins



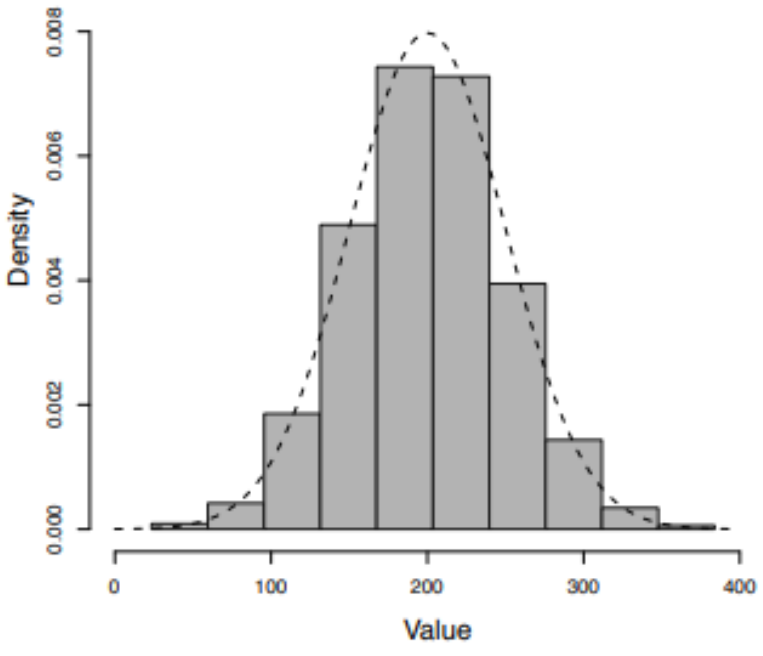
(g) 60 bins

Equal-width binning

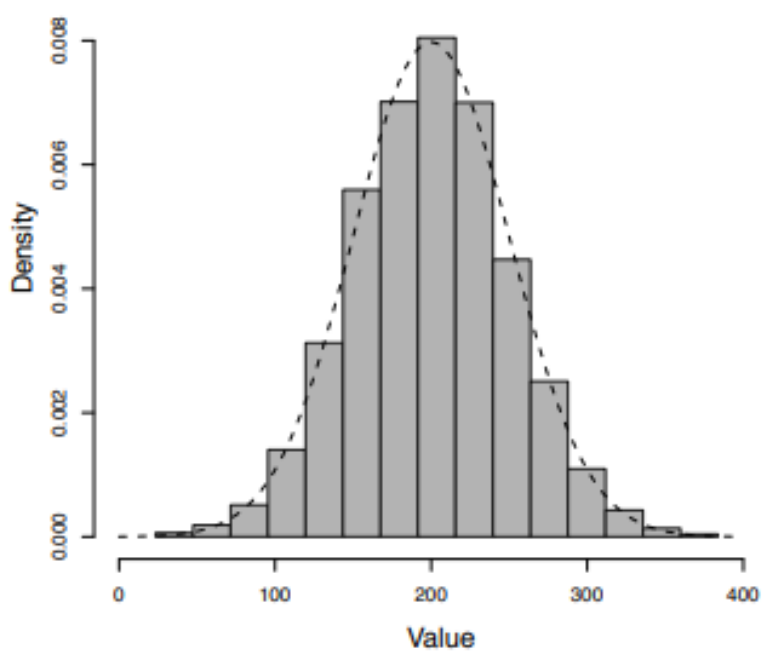
The equal-width binning approach splits the range of the feature values into b bins each of size range/b



(h) 5 Equal-width bins



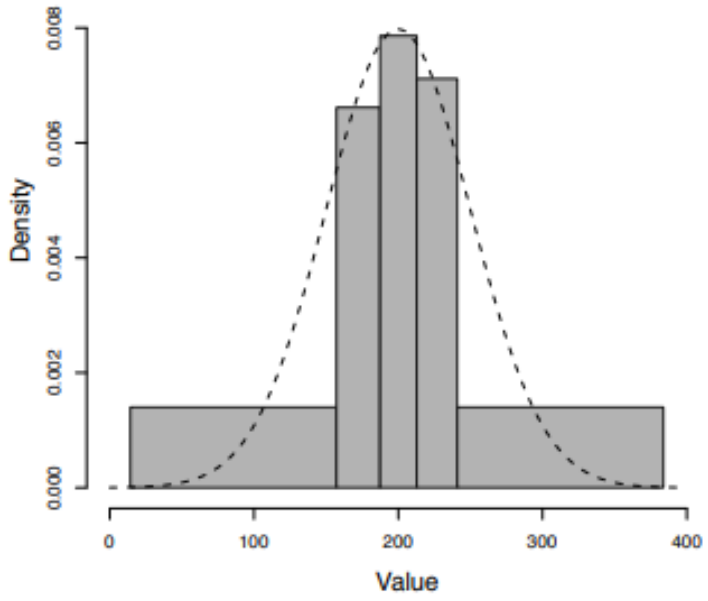
(i) 10 Equal-width bins

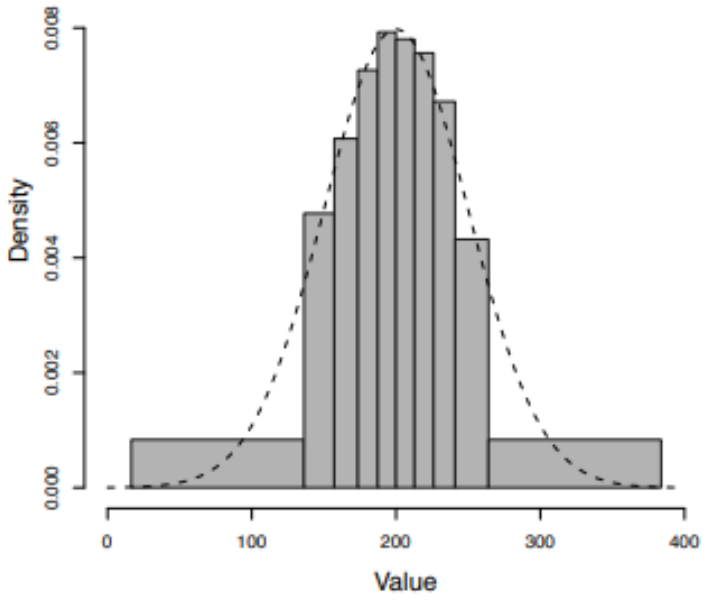


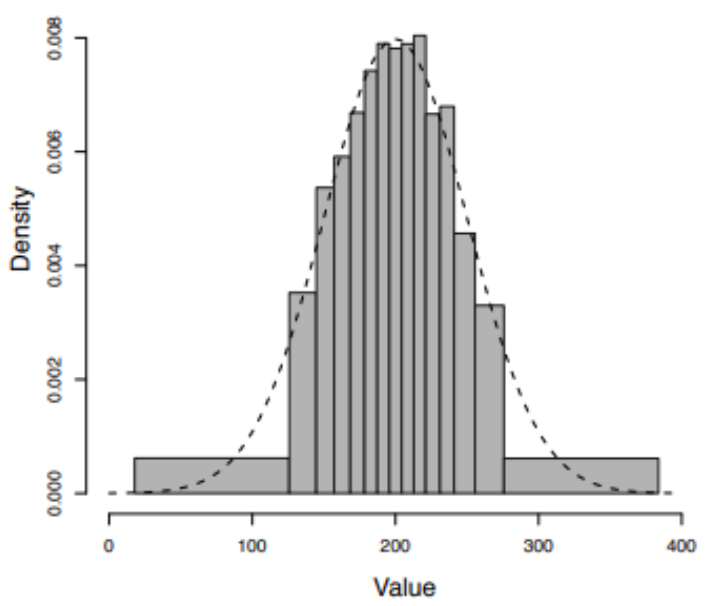
(j) 15 Equal-width bins

Equal-frequency binning

Equal-frequency binning first **sorts the continuous feature values into ascending order** and then places an equal number of instances into each bin, starting with bin 1.
The number of instances placed in each bin is simply the total number of instances divided by the number of bins, b.







(k) 5 Equal-frequency bins (l) 10 Equal-frequency bins (m) 15 Equal-frequency bins

Sampling

University *of* Galway.ie

Sampling

Sometimes the dataset we have is so large that we do not use all the data available to us and instead sample a smaller percentage from the larger dataset.

E.g. we may wish to use only part of the data because training will take a long time with very many examples for some algorithms. Or in the case of *k*-NN, a very large training set may lead to long prediction times

We need to be careful when sampling, however, to ensure that the resulting datasets are still representative of the original data and **that no unintended bias is introduced** during this process. Common forms of sampling include: top sampling, random sampling, stratified sampling, under-sampling, over-sampling

Top sampling

Top sampling simply selects the top *s*% of instances from a dataset to create a sample.
Top sampling runs a serious risk of introducing bias, however, as the sample will be **affected by any ordering** of the original dataset.

Therefore top sampling should be avoided.

Random sampling

A good default sampling strategy is random sampling
Random sampling randomly selects a proportion of *s*% of the instances from a large dataset to create a smaller set.

Random sampling is a good choice in most cases as the random nature of the selection of instances should avoid introducing bias.

Stratified sampling

Stratified sampling is a sampling method that ensures that the relative frequencies of the levels of a specific stratification feature are maintained in the sampled dataset.

To perform stratified sampling:

the instances in a dataset are divided into groups (or strata), where each group contains only instances that have a particular level for the stratification feature

s% of the instances in each stratum are randomly selected these selections are combined to give an overall sample of *s*% of the original dataset.

Dealing with imbalanced datasets

In contrast to stratified sampling, sometimes we would like a sample to contain different relative frequencies of the levels of a particular discrete feature to the distribution in the original dataset.

To do this, we can use **under-sampling** or **over-sampling**.

Under-sampling

Under-sampling begins by dividing a dataset into groups, where each group contains only instances that have a particular level for the feature to be under-sampled.
The number of instances in the smallest group is the under-sampling target size.
Each group containing more instances than the smallest one is then randomly sampled by the appropriate percentage to create a subset that is the under-sampling target size.

These under-sampled groups are then combined to create the overall under-sampled dataset.

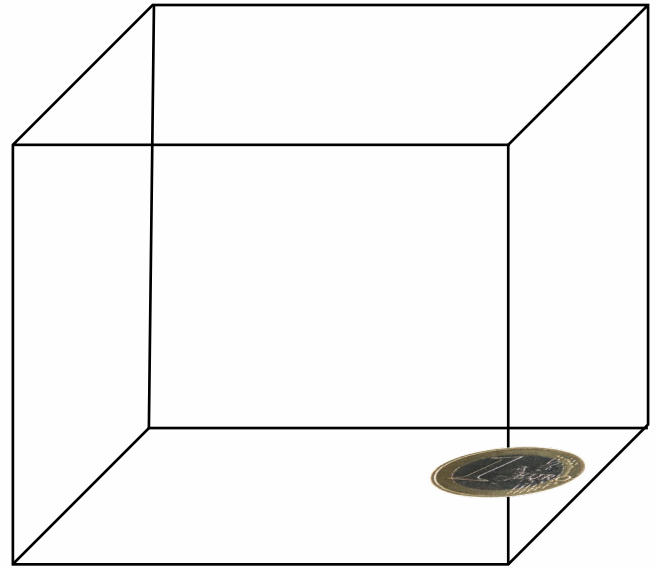
Over-sampling

Over-sampling addresses the same issue as under-sampling but in the opposite way around.
After dividing the dataset into groups, the number of instances in the largest group becomes the over-sampling target size.

From each smaller group, we then create a sample containing that number of instances using random sampling with replacement.
These larger samples are combined to form the overall over-sampled dataset.

Feature Selection

The Curse of Dimensionality [1]



The Curse of Dimensionality [2]

Problem — Curse of Dimensionality:

- Some attributes are much more significant than others
- This is particularly problematic for *k*-NN if all attributes are considered equally in distance metric, possibly leading to bad predictions.
- *k*-NN uses all attributes when making a prediction, whereas other algorithms e.g., decision trees use only the most useful features so are not as badly affected by the Curse of Dimensionality
- With many attributes, everything becomes 'distant' [see next slide]

Solution a:

- Assign weighting to each dimension
- (**not** same as distance-weighted k-NN!)
- Optimise weighting to minimise error

Solution b:

- Give some dimensions 0 weight:
- **Feature Subset Selection**

Any algorithm that considers all attributes in a high-dimensional space equally has this problem, not just *k*-NN + Euclidean Distance!

Ollscoil na Gaillimhe
University of Galway

The Curse of Dimensionality [3]

Russell & Norvig:

Consider N cases with d dimensions, in hypercube of **unit** volume

Assume neighbourhoods are hypercubes, length b: volume is b^d

To contain k points, average neighbourhood must occupy k/N of entire volume

$\Rightarrow b^d = k/N$

$\Rightarrow b = (k/N)^{1/d}$

High dimensions:

k =10; N =1,000,000; d =100 => b = 0.89

i.e. neighbourhood spans nearly 90% of each dimension of space!

Low dimensions:

k and N unchanged; d =2 => b = 0.003 [OK]

High-D spaces are generally very sparse: all neighbours far away

Feature Selection

Fortunately, some algorithms partially mitigate the effects of the curse of dimensionality (e.g., decision tree learning). This is not true for all algorithms however, and heuristics for search can sometimes be misleading!

K-NN and many other algorithms use all attributes when making a prediction
Acquiring more data is not (always) a realistic option
The best way to avoid the curse is to use only the most useful features during learning, this process is known as feature selection

Types of features

We may wish to distinguish between different types of descriptive features:
Predictive: provides information that is useful when estimating the correct target value
Interacting: provides useful information only when considered in conjunction with other features
Redundant: features that have a strong correlation with another feature
Irrelevant: doesn't provide any useful information for estimating the target value
Ideally, a good feature selection approach should identify the smallest subset of features that maintain prediction performance

Feature Selection Approaches

Rank and prune:

Rank features according to their predictive power and keep only the top X%
A **filter** is a measure of predictive power used during ranking, e.g. information gain
Drawback: features evaluated in isolation, so we will miss useful interacting features

Search for useful feature subsets:

We can pick out useful interacting features by evaluating feature subsets
Could generate, evaluate and rank all possible feature subsets then pick best (essentially a brute force approach, computationally expensive/infeasible?)
Better approach: **greedy local search**, build feature subset iteratively by starting out with an empty selection, then trying to add additional features incrementally. Requires evaluation experiments along the way. Stop trying to add more features to the selection once termination conditions are met.

Covariance and Correlation

Example dataset: professional basketball players

ID	Position	Height	Weight	Career Stage	Age	Sponsorship Earnings	Shoe Sponsor
1	forward	192	218	veteran	29	561	yes
2	center	218	251	mid-career	35	60	no
3	forward	197	221	rookie	22	1,312	no
4	forward	192	219	rookie	22	1,359	no
5	forward	198	223	veteran	29	362	yes
6	guard	166	188	rookie	21	1,536	yes
7	forward	195	221	veteran	25	694	no
8	guard	182	199	rookie	21	1,678	yes
9	guard	189	199	mid-career	27	385	yes
10	forward	205	232	rookie	24	1,416	no
11	center	206	246	mid-career	29	314	no
12	guard	185	207	rookie	23	1,497	yes
13	guard	172	183	rookie	24	1,383	yes
14	guard	169	183	rookie	24	1,034	yes
15	guard	185	197	mid-career	29	178	yes
16	forward	215	232	mid-career	30	434	no
17	guard	158	184	veteran	29	162	yes
18	guard	190	207	mid-career	27	648	yes
19	center	195	235	mid-career	28	481	no
20	guard	192	200	mid-career	32	427	yes
21	forward	202	220	mid-career	31	542	no
22	forward	184	213	mid-career	32	12	no
23	forward	190	215	rookie	22	1,179	no
24	guard	178	193	rookie	21	1,078	no
25	guard	185	200	mid-career	31	213	yes
26	forward	191	218	rookie	19	1,855	no
27	center	196	235	veteran	32	47	no
28	forward	198	221	rookie	22	1,409	no
29	center	207	247	veteran	27	1,065	no
30	center	201	244	mid-career	25	1,111	yes

Ollscoil na Gaillimhe
University of Galway

Scatter plot example

Note strong positive correlation between weight and height – almost linear relationship

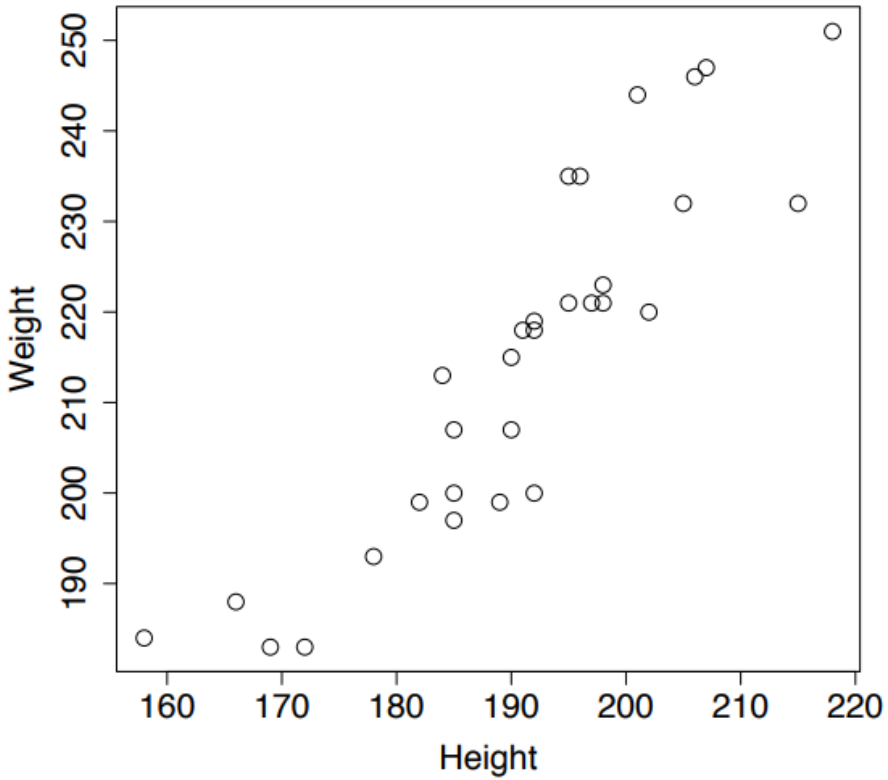


Figure: An example scatter plot showing the relationship between the HEIGHT and WEIGHT features from the professional basketball squad

Some further scatter plot examples

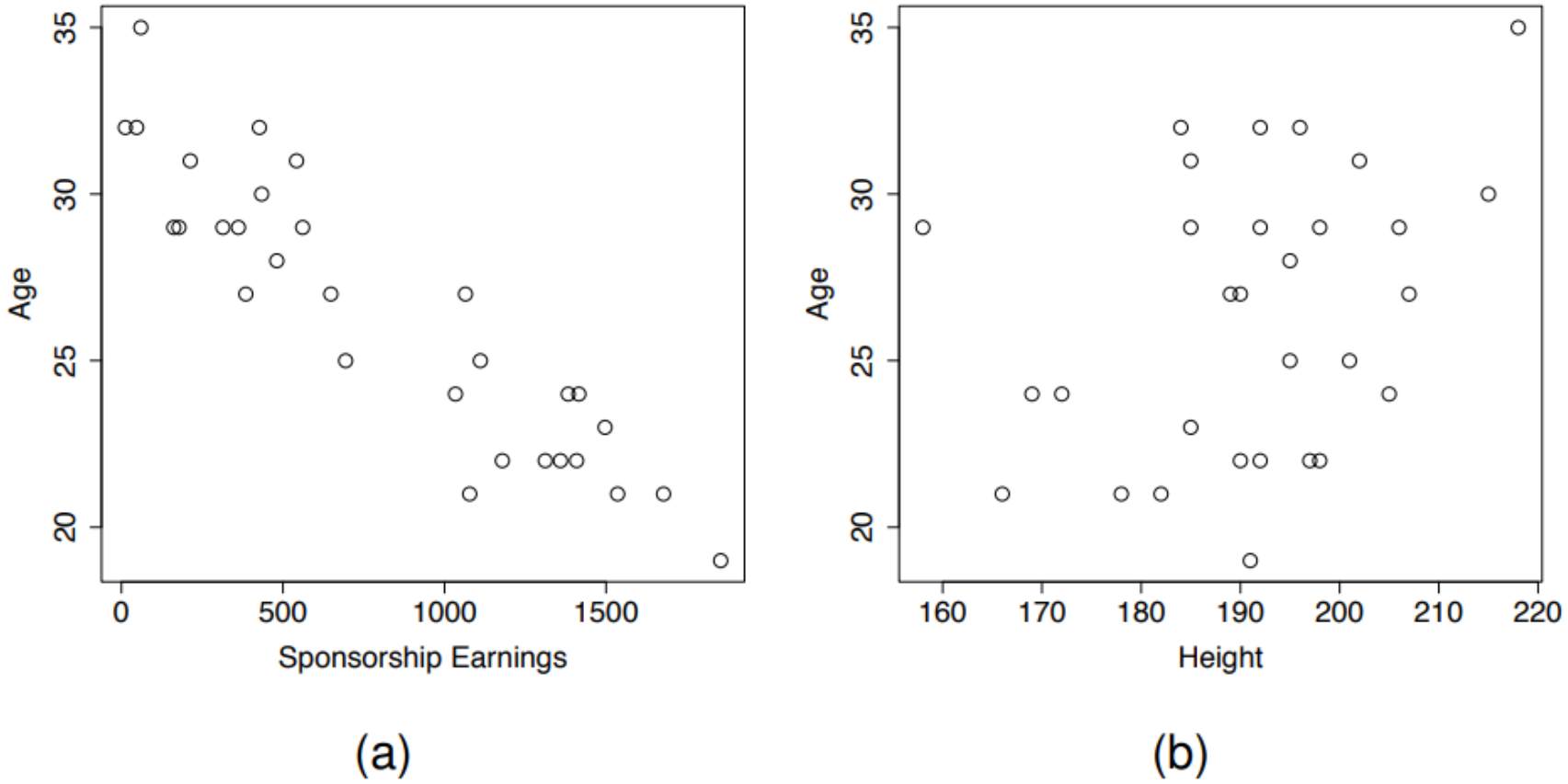
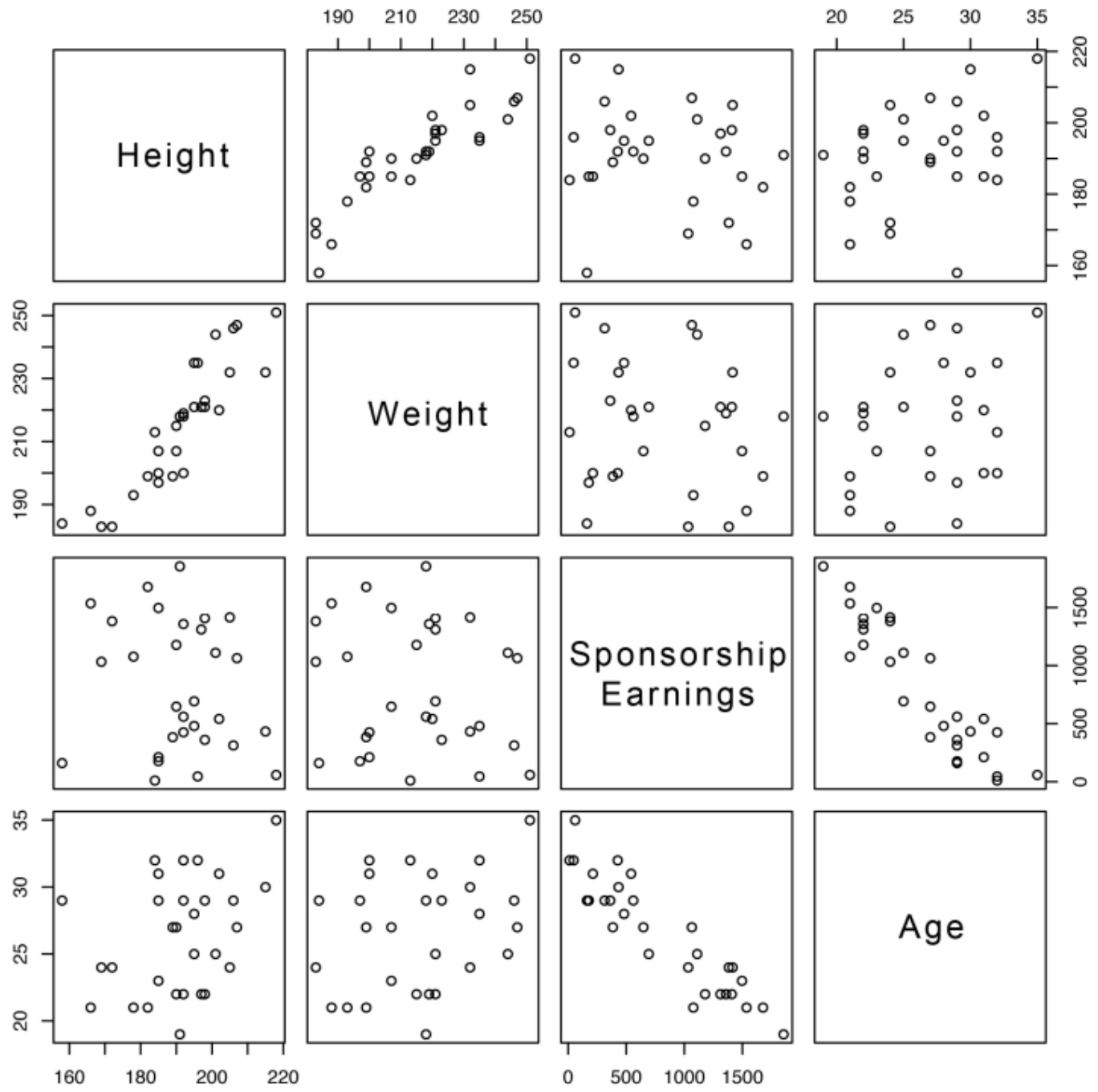


Figure: Example scatter plots showing (a) the strong negative covariance between the SPONSORSHIP EARNINGS and AGE features and (b) the HEIGHT and AGE features from the dataset in Table 4 [4].

Example scatter plot matrix



Ollscoil na Gaillimhe
University of Galway

Measuring covariance and correlation

As well as visually inspecting scatter plots, we can calculate formal measures of the relationship between two continuous features using **covariance** and **correlation**.

For two features, a and b, in a dataset of n instances, the **sample covariance** between a and b is

$$cov(a,b) = \frac{1}{n-1}\sum_{i=1}^{n}\left((a_i - \bar{a}) \times \left(b_i - \bar{b}\right)\right) \quad (1)$$

where a_i and b_i are values of features a and b for the i^{th} instance in a dataset, and $\bar{a}$ and $\bar{b}$ are the sample means of features a and b.

Covariance

Covariance values fall into the range $[-\infty, \infty]$ where negative values indicate a negative relationship, positive values indicate a positive relationship, and values near zero indicate that there is little or no relationship between the features.

Example co-variance calculation (1/2)

ID	Height (h)	$h-\overline{h}$	Weight (w)	$w-\overline{w}$	$(h-\overline{h})\times(w-\overline{w})$	Age (a)	$a-\overline{a}$	$(h-\overline{h})\times(a-\overline{a})$
1	192	0.9	218	3.0	2.7	29	2.6	2.3
2	218	26.9	251	36.0	967.5	35	8.6	231.3
3	197	5.9	221	6.0	35.2	22	-4.4	-26.0
4	192	0.9	219	4.0	3.6	22	-4.4	-4.0
5	198	6.9	223	8.0	55.0	29	2.6	17.9
				...				
26	191	-0.1	218	3.0	-0.3	19	-7.4	0.7
27	196	4.9	235	20.0	97.8	32	5.6	27.4
28	198	6.9	221	6.0	41.2	22	-4.4	-30.4
29	207	15.9	247	32.0	508.3	27	0.6	9.5
30	201	9.9	244	29.0	286.8	25	-1.4	-13.9
Mean	191.1		215.0			26.4		
Std Dev	13.6		19.8			4.2		
Sum					7,009.9			570.8

Example co-variance calculation (2/2)

Covariance is measured in the same units as the features that it measures, so comparisons like the above don't really make sense (the pairs of features being measured should be in the same units

To solve this problem we can use the **correlation coefficient**. (Formally known as the Pearson product-moment correlation coefficient or Pearson's *r*)

$$cov(\text{HEIGHT}, \text{WEIGHT}) = \frac{7{,}009.9}{29} = 241.72$$

$$cov(\text{HEIGHT}, \text{AGE}) = \frac{570.8}{29} = 19.7$$

Correlation

Correlation is a normalized form of covariance that ranges between -1 and $+1$.

The correlation between two features, a and b, can be calculated as

$$corr(a, b) = \frac{cov(a, b)}{sd(a) \times sd(b)} \tag{2}$$

where $cov(a, b)$ is the covariance between features a and b and $sd(a)$ and $sd(b)$ are the standard deviations of a and b respectively.

Correlation

Correlation values fall into the range $[-1, 1]$, where values close to -1 indicate a very strong negative correlation (or covariance), values close to 1 indicate a very strong positive correlation, and values around 0 indicate no correlation.

Features that have no correlation are said to be **independent**.

Ollscoil na Gaillimhe
University of Galway

Calculating correlation

This example confirms what we observed earlier in the scatterplots:

- There is a strong positive relationship between height and weight
- There is little correlation between height and age (unsurprisingly!)

$$corr(Height, Weight) = \frac{241.72}{13.6 \times 19.8} = 0.898$$

$$corr(Height, Age) = \frac{19.7}{13.6 \times 4.2} = 0.345$$

Covariance matrices

The covariance matrix, usually denoted as $\sum$, between a set of continuous features, $\{a, b, \ldots, z\}$, is given as

$$\sum_{\{a,b,\ldots,z\}} = \begin{bmatrix} var(a) & cov(a,b) & \cdots & cov(a,z) \\ cov(b,a) & var(b) & \cdots & cov(b,z) \\ \vdots & \vdots & \ddots & \vdots \\ cov(z,a) & cov(z,b) & \cdots & var(z) \end{bmatrix} \quad (3)$$

Correlation matrices

Similarly, the **correlation matrix** is just a normalized version of the covariance matrix and shows the correlation between each pair of features:

$$\underset{\{a,b,\ldots,z\}}{\textit{correlation matrix}} = \begin{bmatrix} corr(a,a) & corr(a,b) & \cdots & corr(a,z) \\ corr(b,a) & corr(b,b) & \cdots & corr(b,z) \\ \vdots & \vdots & \ddots & \vdots \\ corr(z,a) & corr(z,b) & \cdots & corr(z,z) \end{bmatrix}$$

Covariance and correlation matrices for the basketball dataset

Covariance matrix

$$\sum_{<Height, Weight, Age>} = \begin{bmatrix} 185.128 & 241.72 & 19.7 \\ 241.72 & 392.102 & 24.469 \\ 19.7 & 24.469 & 17.697 \end{bmatrix}$$

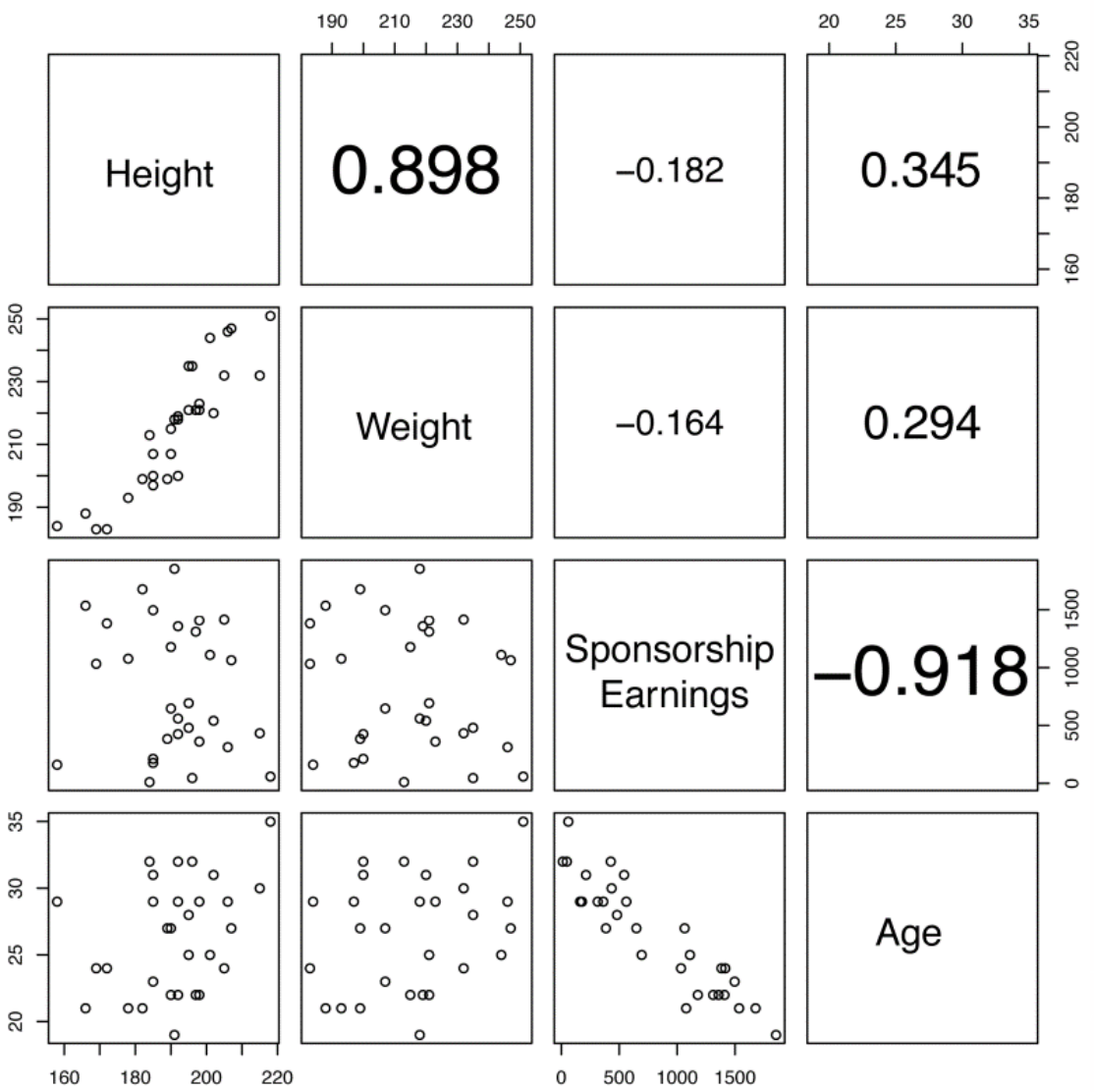
Correlation matrix

$$\text{correlation matrix}_{<Height, Weight, Age>} = \begin{bmatrix} 1.0 & 0.898 & 0.345 \\ 0.898 & 1.0 & 0.294 \\ 0.345 & 0.294 & 1.0 \end{bmatrix}$$

Ollscoil na Gaillimhe
University of Galway

Example scatter plot matrix

Relationship between scatter plots and corelation matrices



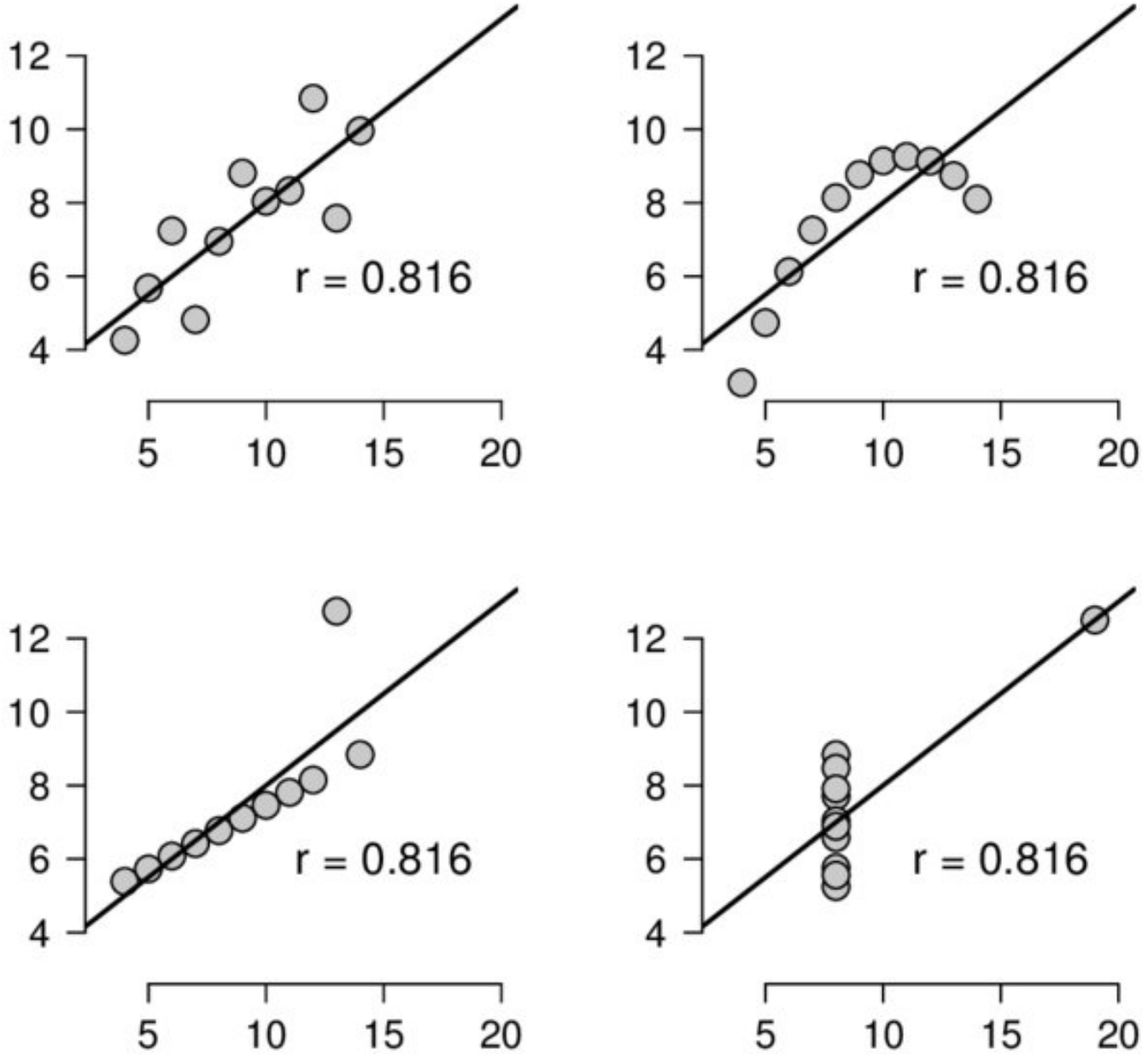
Shortcomings of covariance and correlation

Correlation is a good measure of the relationship between two continuous features, but it is not by any means perfect.
First, the correlation measure given earlier responds only to linear relationships between features.
In a linear relationship between two features, as one feature increases or decreases, the other feature increases or decreases by a corresponding amount.
Frequently, features will have very strong non-linear relationships that correlation does not respond to.
Some of the limitations of measuring correlation are illustrated very clearly in the famous example of Anscombe's quartet published by the famous statistician Francis Anscombe in 1973

Anscombe's quartet

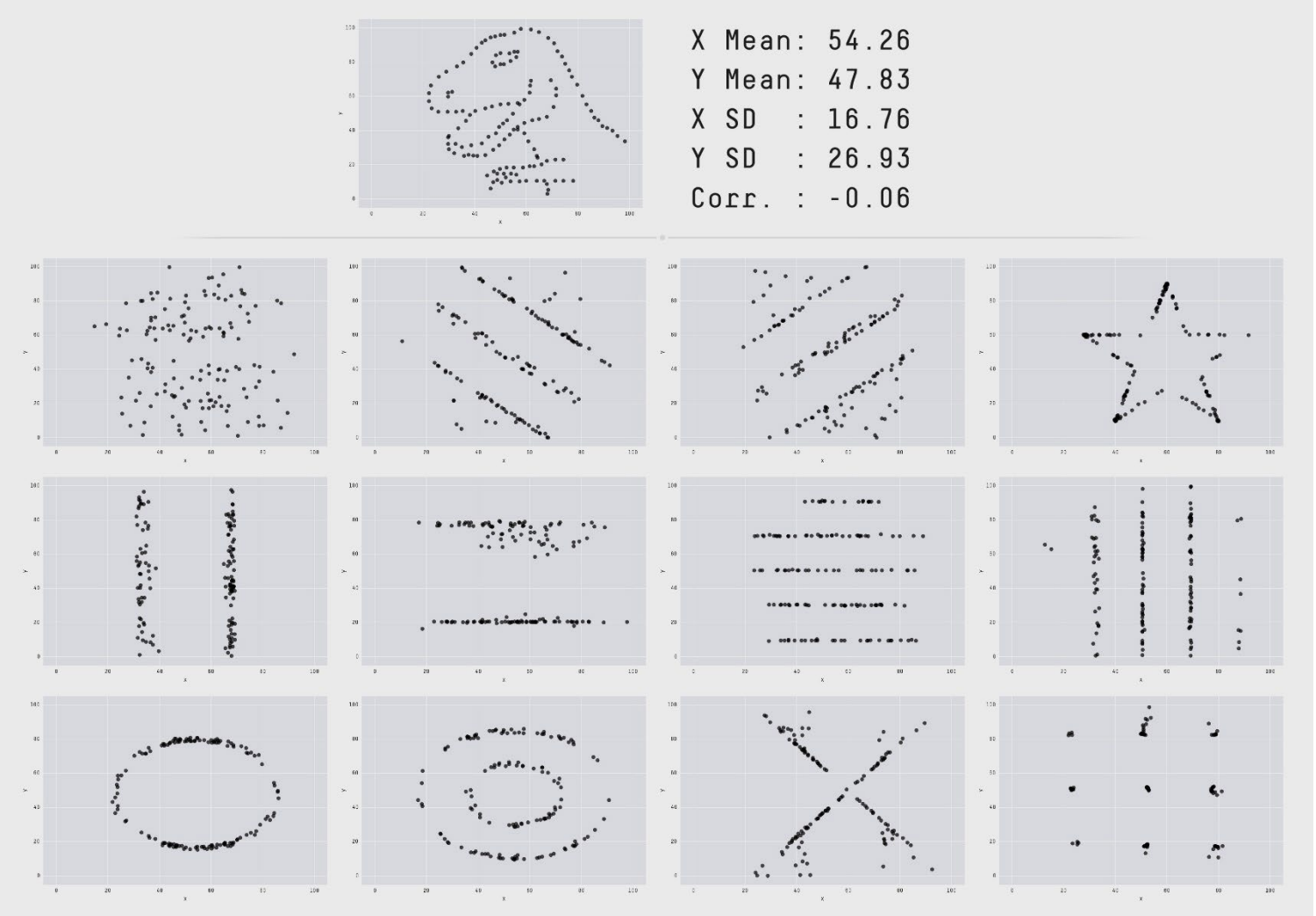
Here we see 4 pairs of features that all have the same correlation to one another: 0.816
Note the linearly increasing relationship as shown by the best fit linear regression line

Main lesson: it is important to **visualise your data** as well as looking at summary statistics!



A more extreme example – The Datasaurus Dozen (Autodesk)

https://www.autodesk.com/research/publications/same-stats-different-graphs



Correlation and causation

Perhaps the most important thing to remember in relation to correlation is that **correlation does not necessarily imply causation**.

Two main mistakes that are made:

1. Mistaking the order of a causal relationship
2. Inferring causation between two features, but neglecting a third hidden feature that has a causal relationship with the first two

Main lesson: before causation is concluded based on a strong correlation between two features, in-depth studies involving domain experts are required—correlation alone is just not enough

Mistaking the order of a causal relationship

Windmills are observed to spin faster when there is a stronger wind
Therefore, can we conclude that spinning windmills cause wind?
No – the relationship is the other way around, wind causes windmills to spin

Many basketball players are taller than average
Therefore, can we conclude that being a basketball player makes one taller?
No – basketball players are taller than average because the extra height gives them an advantage against shorter opponents

Hidden third factors

Every summer, ice cream sales increase. Every summer, drownings also increase
Should we conclude that ice cream causes drowning?
No – there is a hidden third factor (temperature) that causes ice cream sales to increase, and also increases the number of people swimming (hence the increase in drownings)

In 1999, a study was published in *Nature* (one of the world's top journals) claiming a causal relationship between night light use as a child and the development of near-sightedness later in life. However, other researchers could not replicate the results. Later it was discovered that near-sighted parents tend to use night lights in their children's bedrooms because of the parents' poor vision. Near-sighted parents are also more likely to have near sighted children – hence the hidden third factor in this case is the parents, explaining the correlation, while also ruling out a causal link.

Correlation matrices with Pandas

Correlation matrices are easily computed using the pandas library, specifically the `pandas.DataFrame.corr` function

Lots of tutorials available online, here's one example: https://likegeeks.com/python-correlation-matrix/

In this example the seaborn library is also used for visualising the correlation matrix