A Look at Multivariate Techniques

Dr. Lee Fawcett

Newcastle University

Introduction

Introduction

Multivariate Data Analysis, often abbreviated to MVA, provides the foundations of some of the buzz phrases being used in modern data applications:

- Data Science
- Big Data Analytics
- Machine Learning
- Artificial Intelligence

Over the course of the next 60 minutes we will have a whistle-stop tour of MVA, focusing primarily on the 'workhorse' of modern MVA techniques: **Principal Components Analysis**.

Background

With today's data collection systems and databases, a wealth of information is available to analysts at most organisations:

Background

With today's data collection systems and databases, a wealth of information is available to analysts at most organisations:

- *Amazon*: World leader in collecting, storing, processing and analysing personal information from you and every other customer

Background

With today's data collection systems and databases, a wealth of information is available to analysts at most organisations:

- *Amazon*: World leader in collecting, storing, processing and analysing personal information from you and every other customer
- 2020 US election campaign

Background

With today's data collection systems and databases, a wealth of information is available to analysts at most organisations:

- *Amazon*: World leader in collecting, storing, processing and analysing personal information from you and every other customer
- 2020 US election campaign
- 2019 UK General Election

Background

With today's data collection systems and databases, a wealth of information is available to analysts at most organisations:

- *Amazon*: World leader in collecting, storing, processing and analysing personal information from you and every other customer
- 2020 US election campaign
- 2019 UK General Election
- My own experience: impact work with road safety engineers

Background

With today's data collection systems and databases, a wealth of information is available to analysts at most organisations:

- *Amazon*: World leader in collecting, storing, processing and analysing personal information from you and every other customer
- 2020 US election campaign
- 2019 UK General Election
- My own experience: impact work with road safety engineers

Background

Multivariate Analysis – often **two steps**:

- Exploratory Data Analysis (building upon material in lecture 1)
- Modelling (which we'll cover in later lectures)

Why ‘multivariate’?

If you could solve all problems by taking a single measurement on a system, the world would be a much simpler place!

However, complex systems often require multiple measurements, on many variables, to understand them.

Example: **Predicting the weather**. Meteorologists require information on pressure, wind conditions, humidity, season, . . .

Why ‘multivariate’?

Multivariate Data Analysis, often abbreviated to MVA, is the investigation of many variables, simultaneously, in order to understand the relationships that might exist between them.

An MVA might be as simple as looking at the relationship between two variables through a scatterplot, right up to analysing **millions of variables together**.

Why ‘multivariate’?

An MVA can be **highly graphical**, although the usual output graphs might not be those you are used to seeing.

Why ‘multivariate’?

An MVA can be **highly graphical**, although the usual output graphs might not be those you are used to seeing.

For example, you might be interested in the association between height and weight for a cohort of students; this could easily be explored through a 2-D scatterplot.

Why ‘multivariate’?

An MVA can be **highly graphical**, although the usual output graphs might not be those you are used to seeing.

For example, you might be interested in the association between height and weight for a cohort of students; this could easily be explored through a 2-D scatterplot.

Throw a third variable into the mix – perhaps shoe size – and we’re into the realms of 3-D scatterplots.

Why ‘multivariate’?

An MVA can be **highly graphical**, although the usual output graphs might not be those you are used to seeing.

For example, you might be interested in the association between height and weight for a cohort of students; this could easily be explored through a 2-D scatterplot.

Throw a third variable into the mix – perhaps shoe size – and we’re into the realms of 3-D scatterplots.

Graphically displaying the results for more than three variables **requires some ingenuity**, since we live in a 3-D world!

Example: A look at the heptathlon dataset

Name	100M hurdles	High jump	Shot put	200M	Long jump	Javelin	800M
Jessica Ennis	12.54	1.86	14.28	22.83	6.48	47.49	128.65
Lilli Schwarzkopf	13.26	1.83	14.77	24.77	6.30	51.73	130.50
Austra Skujyte	14.00	1.92	17.31	25.43	6.25	51.13	140.59
Antoinette Nana Djimou Ida	12.96	1.80	14.26	24.72	6.13	55.87	133.62
Jessica Zelinka	12.65	1.68	14.81	23.32	5.91	45.75	129.15
Kristina Savitskaya	13.37	1.83	14.77	24.46	6.21	43.70	132.27
Laura Ikauniece	13.71	1.83	12.64	24.16	6.13	51.27	132.13
Hanna Melnychenko	13.32	1.80	12.96	24.09	6.40	43.86	132.90
Brianne Theisen	13.30	1.83	12.89	24.35	6.01	46.47	129.27
Dafne Schippers	13.48	1.80	13.67	22.83	6.28	36.63	135.52
Nadine Broersen	13.64	1.86	13.57	25.13	5.94	51.98	136.98
Jessica Samuelsson	13.58	1.77	14.18	24.25	6.18	42.02	131.31
Katarina Johnson-Thompson	13.48	1.89	11.32	23.73	6.19	38.37	130.76
Sharon Day	13.57	1.77	14.28	24.36	5.85	43.90	131.31
Yana Maksimava	13.97	1.89	14.09	25.43	5.99	42.33	133.37
Eliska Klucinova	14.01	1.80	12.93	25.00	6.13	45.65	136.08
Ellen Sprunger	13.35	1.71	12.62	23.59	5.88	45.63	137.54
Olga Kurban	13.46	1.80	13.71	23.88	5.83	40.36	139.82
Marisa De Aniceto	13.74	1.71	13.09	25.26	5.76	51.98	136.20
Gyorgyi Farkas	14.33	1.80	13.55	25.72	6.07	46.52	137.83
Grit Sadeiko	13.50	1.74	12.43	24.25	6.11	44.12	143.01
Sofia Ifadidou	13.82	1.68	12.96	25.91	5.81	56.96	142.03
Ivona Dadic	14.58	1.80	12.19	24.29	6.00	41.82	135.90
Sarah Cowley	13.95	1.80	12.37	25.60	6.00	41.90	139.01
Louise Hazel	13.48	1.59	12.81	24.48	5.77	47.38	138.78
Ida Marcussen	14.08	1.68	14.26	25.15	5.82	42.26	133.62
Chantae McMillan	13.49	1.68	14.92	25.25	5.37	49.78	160.55
Julia Machtig	14.54	1.68	14.99	25.38	4.06	44.40	140.34
Irina Karpova	14.21	1.68	11.68	25.42	5.70	35.75	148.93

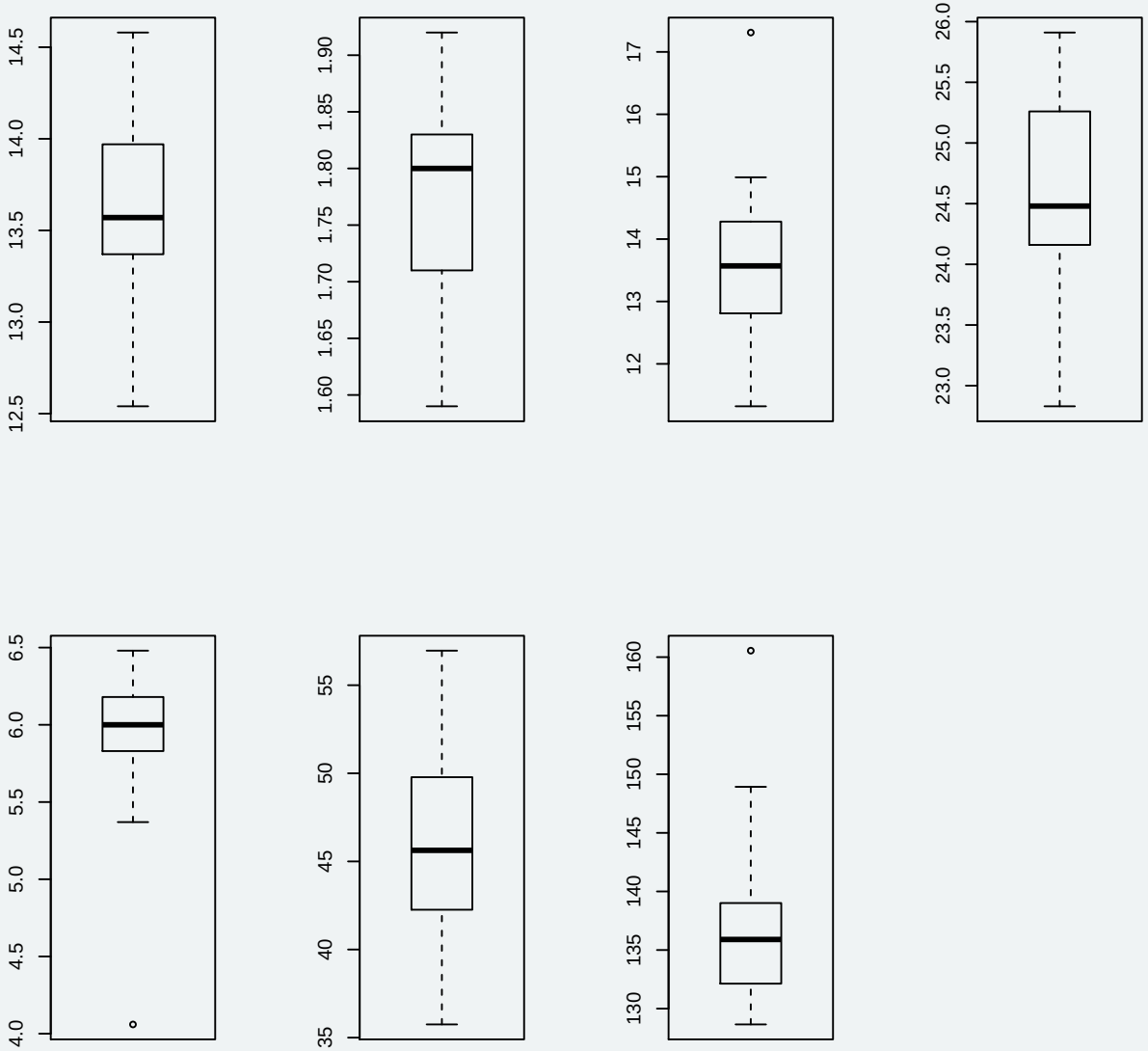
Example: A look at the heptathlon dataset

```
> head(heptathlon)
                                H100M   HJ    SP R200M   LJ    JT  R800M
Jessica Ennis (GBR)             12.54 1.86 14.28 22.83 6.48 47.49 128.65
Lilli Schwarzkopf (GER)         13.26 1.83 14.77 24.77 6.30 51.73 130.50
Austra Skujyte (LTU)            14.00 1.92 17.31 25.43 6.25 51.13 140.59
Antoinette Nana Djimou Ida (FRA) 12.96 1.80 14.26 24.72 6.13 55.87 133.62
Jessica Zelinka (CAN)           12.65 1.68 14.81 23.32 5.91 45.75 129.15
Kristina Savitskaya (RUS)       13.37 1.83 14.77 24.46 6.21 43.70 132.27

> heptathlon[,1]
 [1] 12.54 13.26 14.00 12.96 12.65 13.37 13.71 13.32 13.30 13.48 13.64 13.58
[13] 13.48 13.57 13.97 14.01 13.35 13.46 13.74 14.33 13.50 13.82 14.58 13.95
[25] 13.48 14.08 13.49 14.54 14.21

> heptathlon$H100M
 [1] 12.54 13.26 14.00 12.96 12.65 13.37 13.71 13.32 13.30 13.48 13.64 13.58
[13] 13.48 13.57 13.97 14.01 13.35 13.46 13.74 14.33 13.50 13.82 14.58 13.95
[25] 13.48 14.08 13.49 14.54 14.21

> heptathlon[1,]
                    H100M   HJ    SP R200M   LJ    JT  R800M
Jessica Ennis (GBR) 12.54 1.86 14.28 22.83 6.48 47.49 128.65

> apply(heptathlon, 2, mean)
     H100M         HJ         SP      R200M         LJ         JT      R800M
 13.633448   1.776207  13.596897  24.587586   5.950345  45.758966 136.481724
```

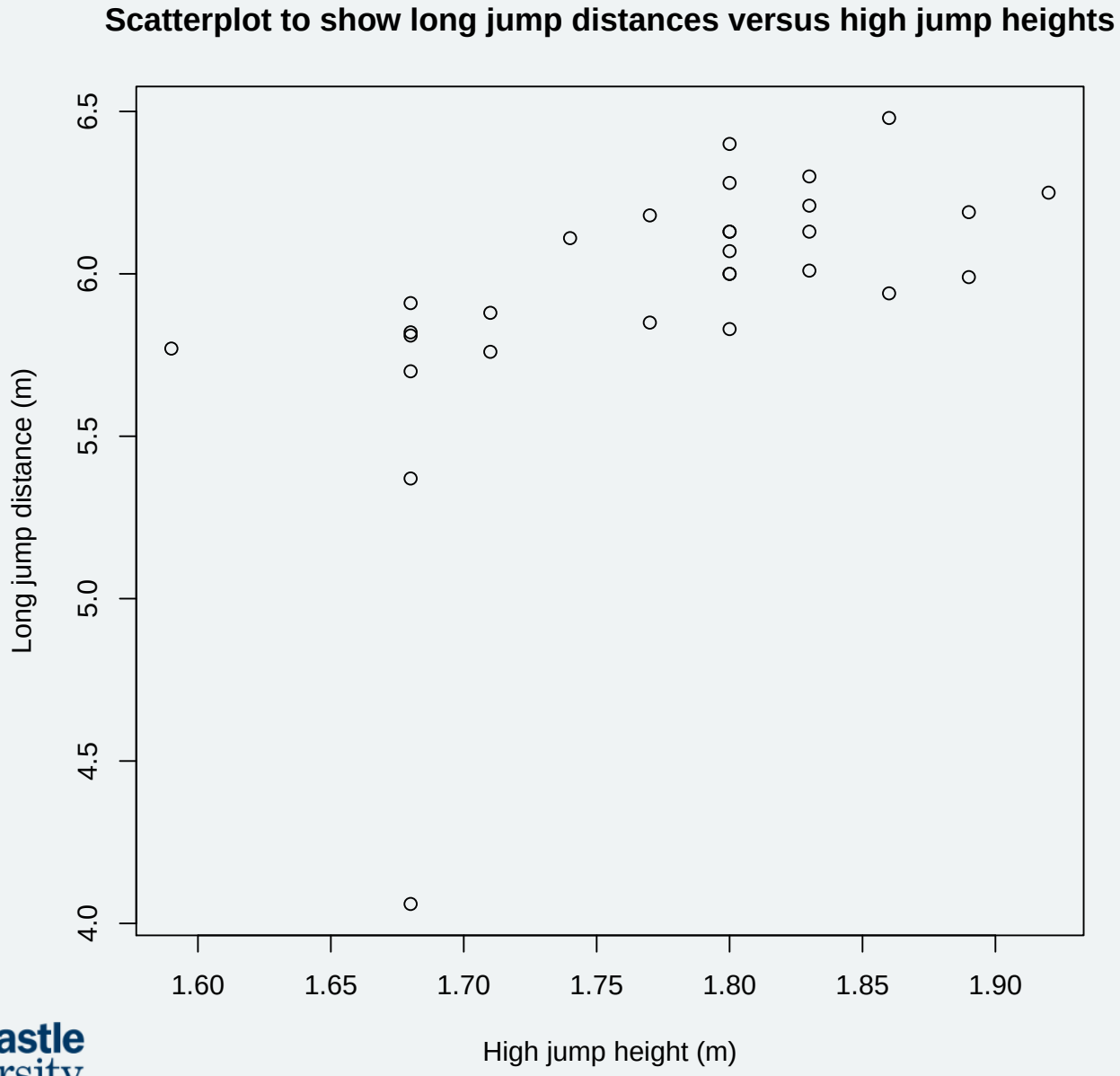
Example: A look at the heptathlon dataset

```
> par(mfrow=c(2,4))
> apply(heptathlon, 2, boxplot)

> par(mfrow=c(1,1))
> plot(heptathlon$HJ, heptathlon$LJ, xlab="High jump height (m)", ylab="Long jump
    distance (m)", main="Scatterplot to show long jump distances versus high jump
    heights")
```

Example: A look at the heptathlon dataset



Example: A look at the heptathlon dataset

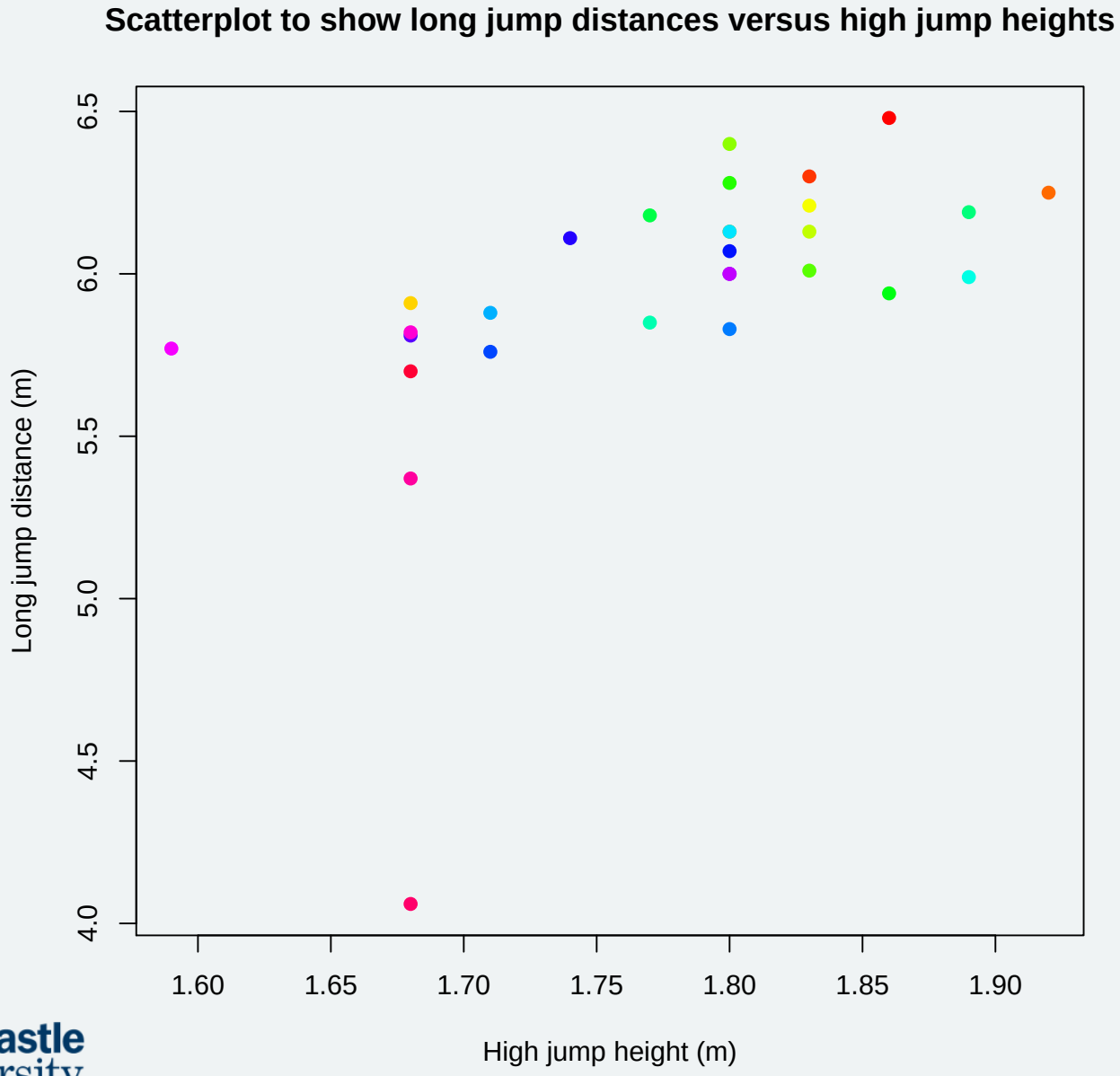


Newcastle University

Example: A look at the heptathlon dataset

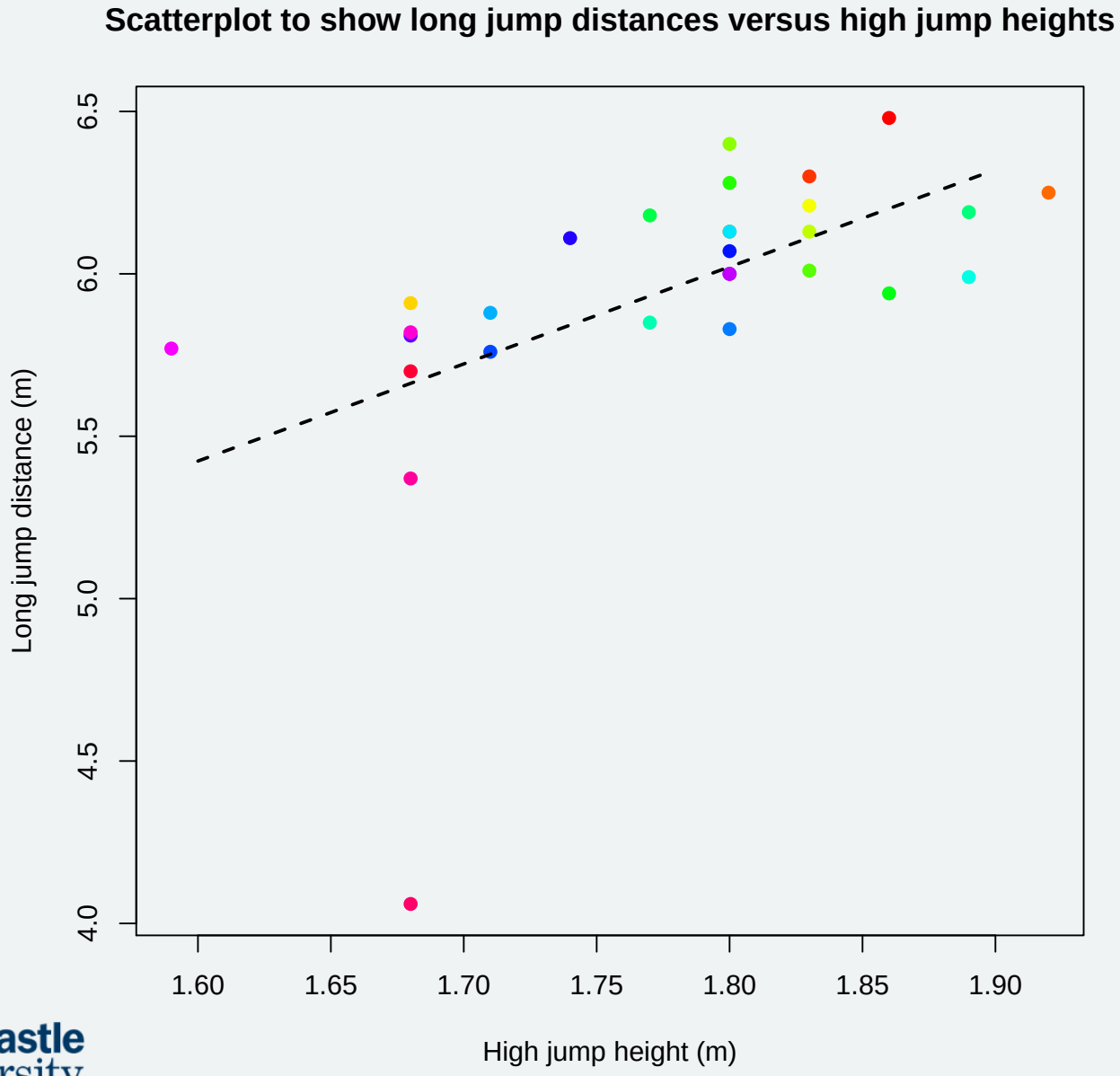
```
plot(heptathlon$HJ, heptathlon$LJ, xlab="High jump height (m)", ylab="Long jump
     distance (m)", main="Scatterplot to show long jump distances versus high jump
     heights", pch=19, col=rainbow(29))
```

Example: A look at the heptathlon dataset



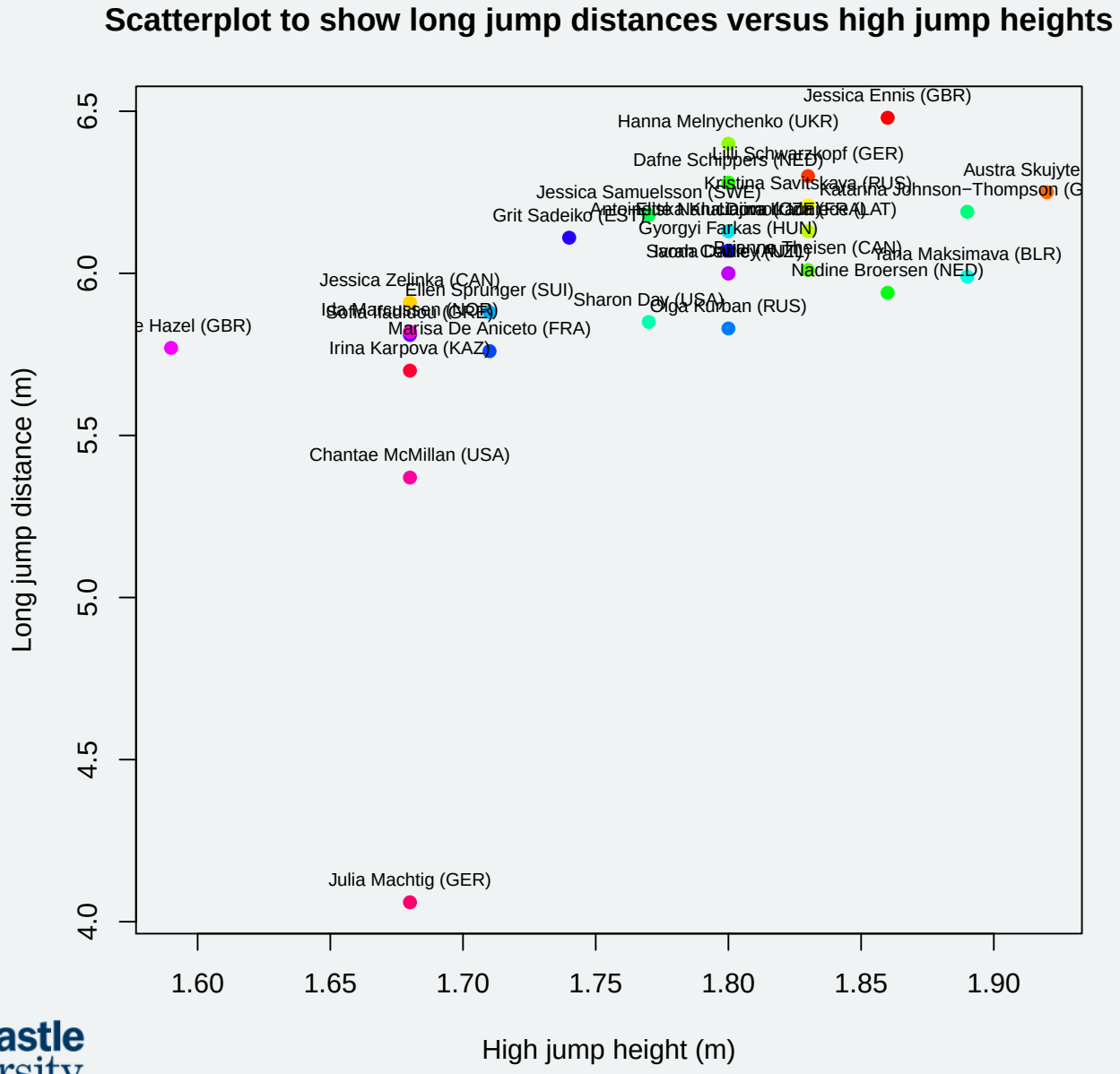
Newcastle University

Example: A look at the heptathlon dataset



Newcastle University

Example: A look at the heptathlon dataset



Newcastle University

Introduction

We will now break for a problem session covering this section.

Please pause the video here and go to **Problem Session 1** of **Lesson 2**.

See you in a bit!

Simple exploration

Simple methods for summarising multivariate datasets

Before embarking on any rigorous analysis of a multivariate dataset (or *any* dataset, for that matter!), **plot it**!

Simple methods for summarising multivariate datasets

Before embarking on any rigorous analysis of a multivariate dataset (or *any* dataset, for that matter!), **plot it**!

- Univariate datasets: histograms, bar charts/pie charts, ...

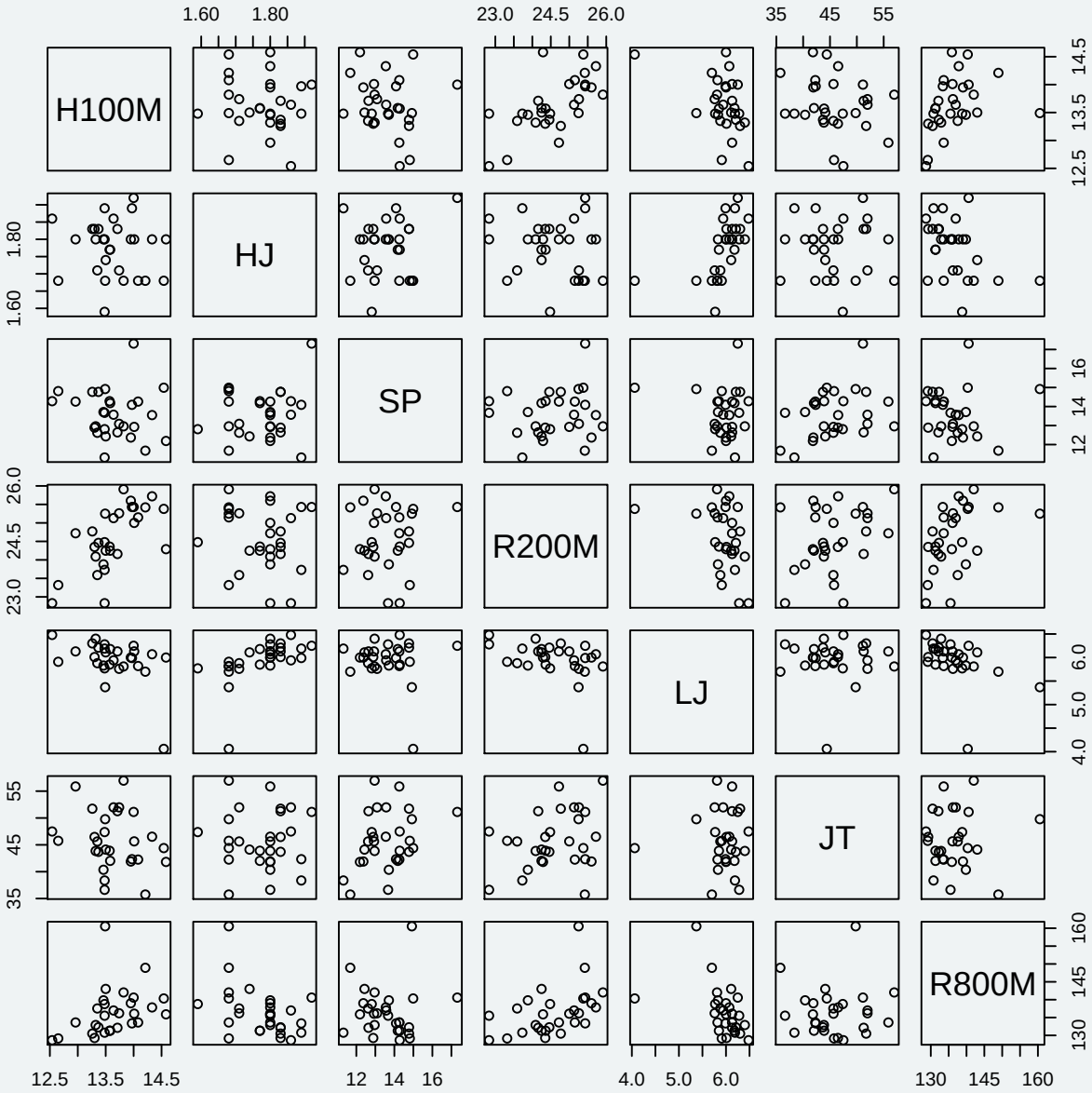
Simple methods for summarising multivariate datasets

Before embarking on any rigorous analysis of a multivariate dataset (or *any* dataset, for that matter!), **plot it**!

- Univariate datasets: histograms, bar charts/pie charts, ...
- Bivariate datasets: histograms for the 'margins', **scatterplots** for the dependence
- Multivariate datasets...?

```
> pairs(heptathlon)
```

Scatterplot matrices



Covariance and Correlation matrices

```
> cov(heptathlon)
              H100M            HJ           SP         R200M           LJ
H100M   0.233480542 -0.003236453 -0.08428534  0.264269335 -0.09268695
HJ     -0.003236453  0.006324384  0.01539495 -0.008963054  0.01890850
SP     -0.084285345  0.015394951  1.51165074  0.110874384 -0.06231675
R200M   0.264269335 -0.008963054  0.11087438  0.693497537 -0.13611342
LJ     -0.092686946  0.018908498 -0.06231675 -0.136113424  0.18704631
JT     -0.564292734 -0.007568350  2.11291453  1.423186700  0.02329680
R800M   1.150536700 -0.225578941 -0.01903017  2.544207882 -1.28646490
                JT        R800M
H100M  -0.56429273  1.15053670
HJ     -0.00756835 -0.22557894
SP      2.11291453 -0.01903017
R200M   1.42318670  2.54420788
LJ      0.02329680 -1.28646490
JT     27.65580246  1.06724828
R800M   1.06724828 43.98550764
```

Note that the **total variation** is:

```
> sum(diag(cov(heptathlon)))
[1] 74.27331
```

Newcastle University

Covariance and Correlation matrices

```
> cor(heptathlon)
              H100M           HJ            SP       R200M           LJ           JT
H100M   1.00000000 -0.08422381 -0.141873405   0.6567483 -0.44352557 -0.22206800
HJ     -0.08422381  1.00000000  0.157450284  -0.1353395  0.54976024 -0.01809668
SP     -0.14187340  0.15745028  1.000000000   0.1082887 -0.11719376  0.32678572
R200M   0.65674831 -0.13533946  0.108288672   1.0000000 -0.37792353  0.32497225
LJ     -0.44352557  0.54976024 -0.117193760  -0.3779235  1.00000000  0.01024303
JT     -0.22206800 -0.01809668  0.326785718   0.3249723  0.01024303  1.00000000
R800M   0.35902139 -0.42769545 -0.002333792   0.4606545 -0.44850632  0.03059973
               R800M
H100M   0.359021391
HJ     -0.427695445
SP     -0.002333792
R200M   0.460654511
LJ     -0.448506323
JT      0.030599725
R800M   1.000000000
```

Covariance and Correlation matrices

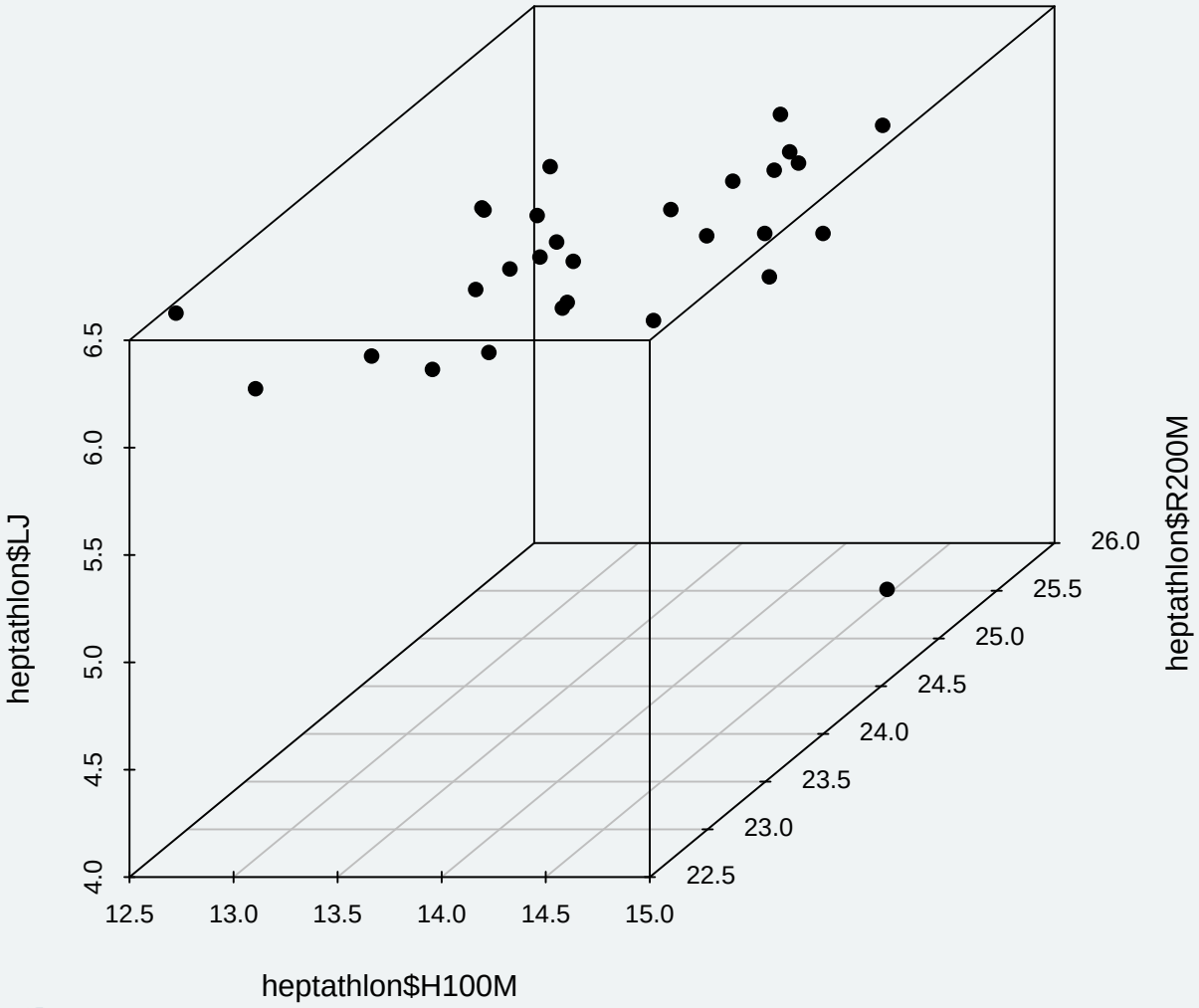
And what if you want to check the **significance** of these correlations?

```
> install.packages("Hmisc") #Install
> library(Hmisc)            #Load
> rcorr(as.matrix(heptathlon))
       H100M    HJ    SP R200M    LJ    JT R800M
H100M   1.00 -0.08 -0.14   0.66 -0.44 -0.22   0.36
HJ     -0.08  1.00  0.16  -0.14  0.55 -0.02  -0.43
SP     -0.14  0.16  1.00   0.11 -0.12  0.33   0.00
R200M   0.66 -0.14  0.11   1.00 -0.38  0.32   0.46
LJ     -0.44  0.55 -0.12  -0.38  1.00  0.01  -0.45
JT     -0.22 -0.02  0.33   0.32  0.01  1.00   0.03
R800M   0.36 -0.43  0.00   0.46 -0.45  0.03   1.00

n= 29

P
       H100M  HJ     SP     R200M  LJ     JT     R800M
H100M         0.6640 0.4629 0.0001 0.0160 0.2469 0.0558
HJ     0.6640         0.4147 0.4839 0.0020 0.9258 0.0206
SP     0.4629 0.4147        0.5761 0.5449 0.0836 0.9904
R200M  0.0001 0.4839 0.5761        0.0432 0.0854 0.0119
LJ     0.0160 0.0020 0.5449 0.0432        0.9579 0.0147
JT     0.2469 0.9258 0.0836 0.0854 0.9579        0.8748
R800M  0.0558 0.0206 0.9904 0.0119 0.0147 0.8748
```

3-D Scatterplots

```
install.packages("scatterplot3d") #Install
library(scatterplot3d)            #Load
scatterplot3d(heptathlon$H100M, heptathlon$R200M, heptathlon$LJ)
```

3-D Scatterplots



Other methods

There are a variety of other methods available for summarising multivariate datasets both graphically and numerically:

- **Chernoff's Faces**: Each individual is represented by a face, the size/colour of its features being proportional to the measurements made on different variables

Other methods

There are a variety of other methods available for summarising multivariate datasets both graphically and numerically:

- **Chernoff's Faces**: Each individual is represented by a face, the size/colour of its features being proportional to the measurements made on different variables
- **Star symbol plots**: Each individual is represented by a star, these stars having the same number of rays as we have variables. The length of the rays are proportional to the measurements made on each variable

Other methods

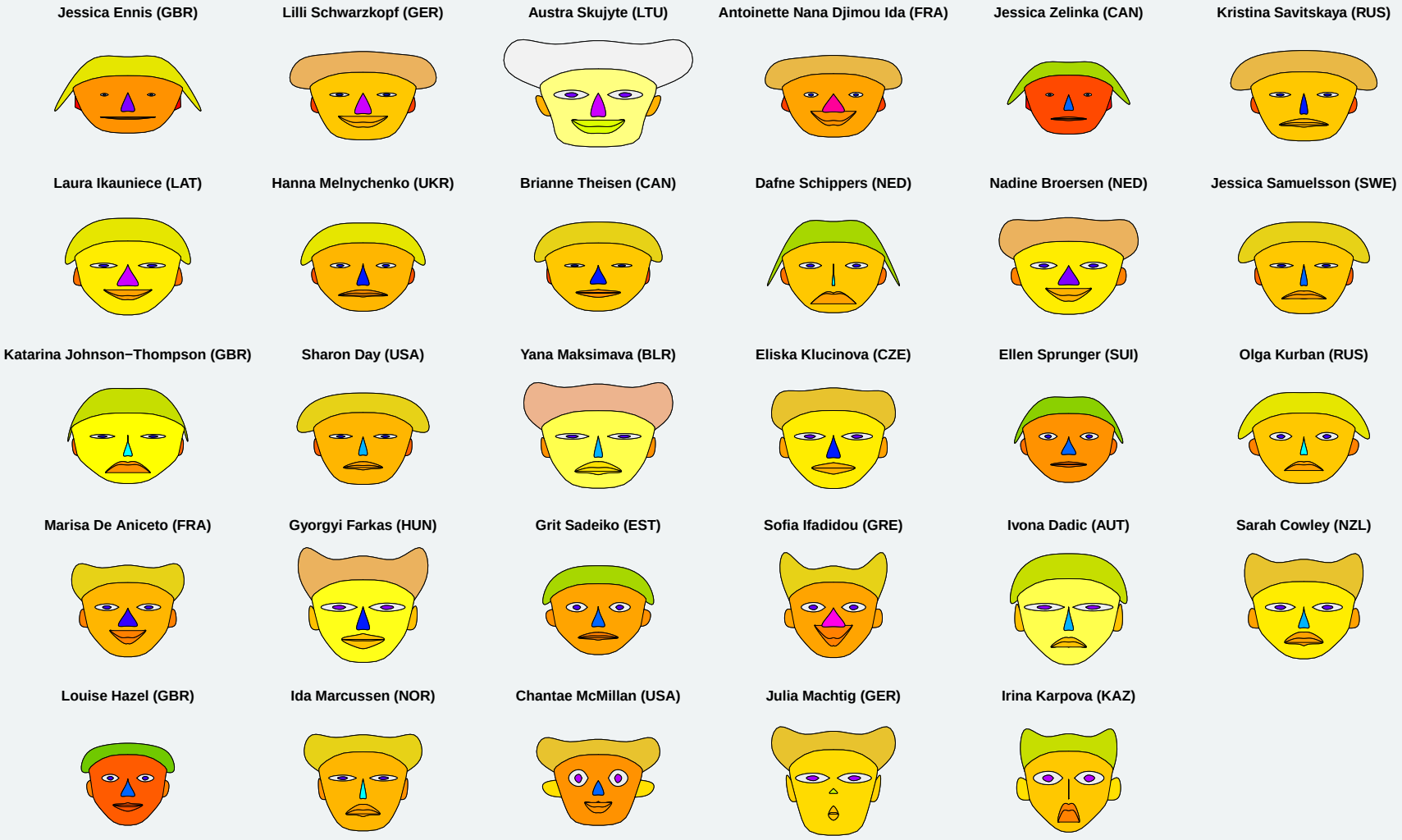
There are a variety of other methods available for summarising multivariate datasets both graphically and numerically:

- **Chernoff's Faces**: Each individual is represented by a face, the size/colour of its features being proportional to the measurements made on different variables
- **Star symbol plots**: Each individual is represented by a star, these stars having the same number of rays as we have variables. The length of the rays are proportional to the measurements made on each variable
- **Mahalanobis distance**: Put simply, this is the distance between two points in multivariate space. It is a numerical summary often used to detect outliers in multivariate space

Other methods

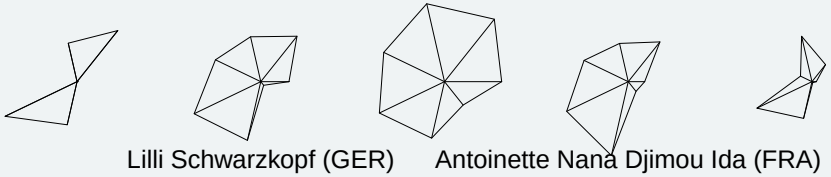
```
> install.packages("aplpack") #Install
> library("aplpack")          #Load
> faces(heptathlon)
effect of variables:
 modified item        Var
 "height of face    " "H100M"
 "width of face     " "HJ"
 "structure of face" "SP"
 "height of mouth   " "R200M"
 "width of mouth    " "LJ"
 "smiling           " "JT"
 "height of eyes    " "R800M"
 "width of eyes     " "H100M"
 "height of hair    " "HJ"
 "width of hair     " "SP"
 "style of hair     " "R200M"
 "height of nose    " "LJ"
 "width of nose     " "JT"
 "width of ear      " "R800M"
 "height of ear     " "H100M"

> install.packages(graphics)  #Install
> library(graphics)           #Load
> stars(heptathlon)
```

Other methods

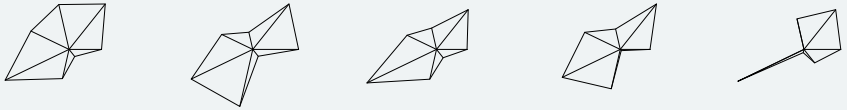


Newcastle University

Other methods



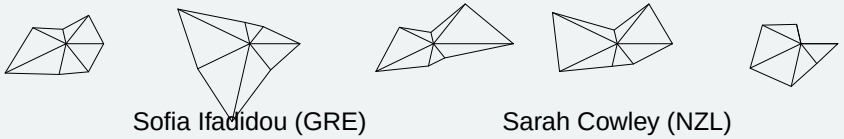
Lilli Schwarzkopf (GER) Antoinette Nana Djimou Ida (FRA)
Jessica Ennis (GBR) Austra Skujyte (LTU) Jessica Zelinka (CAN)



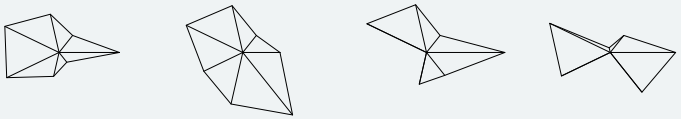
Kristina Savitskaya (RUS) Hanna Melnychenko (UKR) Dafne Schippers (NED)
Laura Ikauniece (LAT) Brianne Theisen (CAN)

Jessica Samuelsson (SWE) Sharon Day (USA)
Nadine Broersen (NED) Katarina Johnson−Thompson (GBR) Yana Maksimava (BLR)

Eliska Klucinova (CZE) Olga Kurban (RUS) Gyorgyi Farkas (HUN)
Ellen Sprunger (SUI) Marisa De Aniceto (FRA)



Sofia Ifadidou (GRE) Sarah Cowley (NZL)
Grit Sadeiko (EST) Ivona Dadic (AUT) Louise Hazel (GBR)



Ida Marcussen (NOR) Julia Machtig (GER)
Chantae McMillan (USA) Irina Karpova (KAZ)

Newcastle University

Other methods

```
> mahalanobis(heptathlon, apply(heptathlon, 2, mean), cov(heptathlon))
            Jessica Ennis (GBR)          Lilli Schwarzkopf (GER)
                       6.790333                         3.213907
           Austra Skujyte (LTU) Antoinette Nana Djimou Ida (FRA)
                      15.581773                         5.234548
          Jessica Zelinka (CAN)        Kristina Savitskaya (RUS)
                       9.033625                         2.874082
          Laura Ikauniece (LAT)          Hanna Melnychenko (UKR)
                       6.676844                         1.528953
          Brianne Theisen (CAN)            Dafne Schippers (NED)
                       3.652364                         8.590237
          Nadine Broersen (NED)         Jessica Samuelsson (SWE)
                       4.105886                         2.820154
 Katarina Johnson-Thompson (GBR)                 Sharon Day (USA)
                       9.014777                         1.692851
           Yana Maksimava (BLR)           Eliska Klucinova (CZE)
                       6.026769                         1.533981
           Ellen Sprunger (SUI)                Olga Kurban (RUS)
                       3.685668                         3.269338
         Marisa De Aniceto (FRA)            Gyorgyi Farkas (HUN)
                       3.180311                         3.792612
             Grit Sadeiko (EST)            Sofia Ifadidou (GRE)
                       3.046474                         8.056892
              Ivona Dadic (AUT)              Sarah Cowley (NZL)
                      11.559556                         5.705398
             Louise Hazel (GBR)             Ida Marcussen (NOR)
                       6.892786                         7.733176
          Chantae McMillan (USA)             Julia Machtig (GER)
                      17.140631                        23.099988
            Irina Karpova (KAZ)
                      10.466088
```

Simple exploration

We will now break for a problem session covering this section.

Please pause the video here and go to **Problem Session 2** of **Lesson 2**.

See you in a bit!

Principal Components Analysis

Principal Components Analysis

- **Principal Components Analysis** (PCA) is the analysis of variability in a multivariate dataset
- PCA is often referred to as the "**workhorse of multivariate methods**"
- PCA provides some powerful graphical tools for exploratory analysis, and enables datasets with many variables to be displayed graphically in 2-D space

Principal Components Analysis

- **Principal Components Analysis** (PCA) is the analysis of variability in a multivariate dataset
- PCA is often referred to as the "**workhorse of multivariate methods**"
- PCA provides some powerful graphical tools for exploratory analysis, and enables datasets with many variables to be displayed graphically in 2-D space
- Thus, PCA is also referred to as a **dimension reduction** technique

Principal Components Analysis

- The main aim is to find new variables which are linear combinations of the original variables

Principal Components Analysis

- The main aim is to find new variables which are linear combinations of the original variables
- The new variables are such that a few of them will explain most of the variability in the data, hence effectively **reducing the number of variables**

Principal Components Analysis

- The main aim is to find new variables which are linear combinations of the original variables
- The new variables are such that a few of them will explain most of the variability in the data, hence effectively **reducing the number of variables**
- Unlike **stepwise regression**, PCA includes information on *all* variables through these new variables

PCA of the heptathlon dataset

Recall we have seven variables, which we will now label:

$$
\begin{aligned}
X_1 &= \text{Time in 100 metre hurdles} \\
X_2 &= \text{Height in high jump} \\
X_3 &= \text{Distance in shot put} \\
X_4 &= \text{Time in 200 metres} \\
X_5 &= \text{Distance in long jump} \\
X_6 &= \text{Distance thrown in javelin} \\
X_7 &= \text{Time in 800 metres}
\end{aligned}
$$

PCA of the heptathlon dataset

The aim of a PCA is to **construct new variables**:

$$
\begin{aligned}
Y_1 &= a_{1,1}X_1 + a_{2,1}X_2 + a_{3,1}X_3 + a_{4,1}X_4 + a_{5,1}X_5 + a_{6,1}X_6 + a_{7,1}X_7 \\
Y_2 &= a_{1,2}X_1 + a_{2,2}X_2 + a_{3,2}X_3 + a_{4,2}X_4 + a_{5,2}X_5 + a_{6,2}X_6 + a_{7,2}X_7 \\
Y_3 &= a_{1,3}X_1 + a_{2,3}X_2 + a_{3,3}X_3 + a_{4,3}X_4 + a_{5,3}X_5 + a_{6,3}X_6 + a_{7,3}X_7 \\
Y_4 &= a_{1,4}X_1 + a_{2,4}X_2 + a_{3,4}X_3 + a_{4,4}X_4 + a_{5,4}X_5 + a_{6,4}X_6 + a_{7,4}X_7 \\
\vdots \quad & \quad \vdots
\end{aligned}
$$

PCA of the heptathlon dataset

Why would we want to do all of this?

- Y_1 is known as the **first principal component** in our analysis
- Notice it contains information on all of our original variables X_1 through to X_7

PCA of the heptathlon dataset

Why would we want to do all of this?

- Y_1 is known as the **first principal component** in our analysis
- Notice it contains information on all of our original variables X_1 through to X_7
- Y_2 is the **second principal component**, and it too contains information on all of our original variables

PCA of the heptathlon dataset

Why would we want to do all of this?

- Y_1 is known as the **first principal component** in our analysis
- Notice it contains information on all of our original variables X_1 through to X_7
- Y_2 is the **second principal component**, and it too contains information on all of our original variables
- Y_3 is the **third principal component**, and so on and so forth

PCA of the heptathlon dataset

Why would we want to do all of this?

- Y_1 is known as the **first principal component** in our analysis
- Notice it contains information on all of our original variables X_1 through to X_7
- Y_2 is the **second principal component**, and it too contains information on all of our original variables
- Y_3 is the **third principal component**, and so on and so forth

Each of our principal components is a function of all of our original X variables, and so each one will **capture the overall variability** of our multivariate dataset.

PCA of the heptathlon dataset

The mathematics underpinning the estimation of the 'a's is such that

$$\text{Var}(Y_1) > \text{Var}(Y_2) > \text{Var}(Y_3) > \cdots ,$$

and $\sum \text{Var}(Y_i) = \text{total variation}$ (See last section).

PCA of the heptathlon dataset

The mathematics underpinning the estimation of the 'a's is such that

$$\text{Var}(Y_1) > \text{Var}(Y_2) > \text{Var}(Y_3) > \cdots ,$$

and $\sum \text{Var}(Y_i) = \text{total variation}$ (See last section).

It's often the case that the first two principal components, considered together, **capture most of the variability** present in our original multivariate dataset.

Thus, our aim is to work with – perhaps – just Y_1 and Y_2, rather than $X_1, X_2, \ldots, X_7$.

PCA of the heptathlon dataset

The mathematics underpinning the estimation of the 'a's is such that

$$\text{Var}(Y_1) > \text{Var}(Y_2) > \text{Var}(Y_3) > \cdots ,$$

and $\sum \text{Var}(Y_i) = \text{total variation}$ (See last section).

It's often the case that the first two principal components, considered together, **capture most of the variability** present in our original multivariate dataset.

Thus, our aim is to work with – perhaps – just Y_1 and Y_2, rather than $X_1, X_2, \ldots, X_7$.

Both Y_1 and Y_2 will likely capture most of the variability in our multivariate dataset *and* then we've '**reduced the dimension**' of our analysis from 7-D to just 2-D.

PCA of the heptathlon dataset

- In the jumping events large numbers are indicative of good performances

PCA of the heptathlon dataset

- In the jumping events large numbers are indicative of good performances
- In the running events, small numbers are indicative of good performances

PCA of the heptathlon dataset

- In the jumping events large numbers are indicative of good performances
- In the running events, small numbers are indicative of good performances
- It is good practice, when performing a PCA, to **re-scale variables** so that the ‘direction’ is interpreted in the same way for all variables

PCA of the heptathlon dataset

- In the jumping events large numbers are indicative of good performances
- In the running events, small numbers are indicative of good performances
- It is good practice, when performing a PCA, to **re-scale variables** so that the 'direction' is interpreted in the same way for all variables
- Thus, from this point forwards, we now work with a new version of the heptathlon dataset in which the running events have been adjusted so that big numbers indicate good performances

PCA of the heptathlon dataset

- In the jumping events large numbers are indicative of good performances
- In the running events, small numbers are indicative of good performances
- It is good practice, when performing a PCA, to **re-scale variables** so that the 'direction' is interpreted in the same way for all variables
- Thus, from this point forwards, we now work with a new version of the heptathlon dataset in which the running events have been adjusted so that big numbers indicate good performances
- Specifically, for each athlete j we now have

$$\begin{aligned} X^*_{1_j} &= \max(X_1) - X_{1_j}, \\ X^*_{4_j} &= \max(X_4) - X_{4_j} \quad \text{and} \\ X^*_{7_j} &= \max(X_7) - X_{7_j}. \end{aligned}$$

PCA of the heptathlon dataset

```
> heptathlon2 = heptathlon
> heptathlon2$H100M = max(heptathlon2$H100M) - heptathlon2$H100M
> heptathlon2$R200M = max(heptathlon2$R200M) - heptathlon2$R200M
> heptathlon2$R800M = max(heptathlon2$R800M) - heptathlon2$R800M
> head(heptathlon)
                                H100M   HJ    SP R200M   LJ    JT  R800M
Jessica Ennis (GBR)             12.54 1.86 14.28 22.83 6.48 47.49 128.65
Lilli Schwarzkopf (GER)         13.26 1.83 14.77 24.77 6.30 51.73 130.50
Austra Skujyte (LTU)            14.00 1.92 17.31 25.43 6.25 51.13 140.59
Antoinette Nana Djimou Ida (FRA) 12.96 1.80 14.26 24.72 6.13 55.87 133.62
Jessica Zelinka (CAN)           12.65 1.68 14.81 23.32 5.91 45.75 129.15
Kristina Savitskaya (RUS)       13.37 1.83 14.77 24.46 6.21 43.70 132.27

> head(heptathlon2)
                                H100M   HJ    SP R200M   LJ    JT R800M
Jessica Ennis (GBR)              2.04 1.86 14.28  3.08 6.48 47.49 31.90
Lilli Schwarzkopf (GER)          1.32 1.83 14.77  1.14 6.30 51.73 30.05
Austra Skujyte (LTU)             0.58 1.92 17.31  0.48 6.25 51.13 19.96
Antoinette Nana Djimou Ida (FRA) 1.62 1.80 14.26  1.19 6.13 55.87 26.93
Jessica Zelinka (CAN)            1.93 1.68 14.81  2.59 5.91 45.75 31.40
Kristina Savitskaya (RUS)        1.21 1.83 14.77  1.45 6.21 43.70 28.28
```

PCA of the heptathlon dataset

```
> (pca1 = prcomp(x=heptathlon2))
Standard deviations (1, .., p=7):
[1] 6.65449663 5.27527745 1.16042524 0.78461166 0.37146157 0.24271694 0.05774546
# Notice that ∑ Var(Y_i) = 6.654^2 + ... + 0.058^2 = 74.27331, the total variation (as found earlier)
Rotation (n x k) = (7 x 7):
                PC1           PC2           PC3           PC4          PC5
H100M  -0.025531133  0.0230747007  0.0283836192  0.479533759 -0.16226365
HJ     -0.005106786  0.0003381213  0.0109047147 -0.004792405 -0.09779270
SP      0.003090843  0.0800303918  0.9948011149 -0.001825383 -0.04390316
R200M  -0.060571386 -0.0455319386  0.0003525911  0.856545544  0.28288157
LJ     -0.029240732  0.0038010184 -0.0544354626  0.175398972 -0.93881291
JT      0.068543052  0.9929624023 -0.0804990437  0.032406717  0.02273520
R800M  -0.995032897  0.0707152225 -0.0016611050 -0.067348425  0.01646365
                PC6           PC7
H100M  -0.859374071 -0.0566054449
HJ      0.081637617 -0.9917680240
SP      0.040955621  0.0186585712
R200M   0.424917692  0.0032450898
LJ      0.266850000  0.1132426862
JT      0.035780059 -0.0003526592
R800M  -0.009484905  0.0030507417
```

Newcastle University

PCA of the heptathlon dataset

```
> (pca1 = prcomp(x=heptathlon2))
Standard deviations (1, .., p=7):
[1] 6.65449663 5.27527745 1.16042524 0.78461166 0.37146157 0.24271694 0.05774546
# Notice that ∑ Var(Y_i) = 6.654^2 + ... + 0.058^2 = 74.27331, the total variation (as found earlier)
Rotation (n x k) = (7 x 7):
                PC1           PC2           PC3           PC4           PC5
H100M  -0.025531133  0.0230747007  0.0283836192  0.479533759  -0.16226365
HJ     -0.005106786  0.0003381213  0.0109047147 -0.004792405  -0.09779270
SP      0.003090843  0.0800303918  0.9948011149 -0.001825383  -0.04390316
R200M  -0.060571386 -0.0455319386  0.0003525911  0.856545544   0.28288157
LJ     -0.029240732  0.0038010184 -0.0544354626  0.175398972  -0.93881291
JT      0.068543052  0.9929624023 -0.0804990437  0.032406717   0.02273520
R800M  -0.995032897  0.0707152225 -0.0016611050 -0.067348425   0.01646365
                PC6           PC7
H100M  -0.859374071 -0.0566054449
HJ      0.081637617 -0.9917680240
SP      0.040955621  0.0186585712
R200M   0.424917692  0.0032450898
LJ      0.266850000  0.1132426862
JT      0.035780059 -0.0003526592
R800M  -0.009484905  0.0030507417
```

$$
\begin{aligned}
Y_1 &= -0.026X_1^* - 0.005X_2 + 0.003X_3 - 0.061X_4^* - 0.029X_5 + 0.069X_6 - 0.995X_7^* \\
Y_2 &= 0.023X_1^* + 0.0003X_2 + 0.080X_3 - 0.0461X_4^* + 0.004X_5 + 0.993X_6 + 0.071X_7^* \\
Y_3 &= 0.028X_1^* + 0.011X_2 + 0.995X_3 + 0.00031X_4^* - 0.054X_5 - 0.080X_6 - 0.002X_7^* \\
\vdots \quad & \quad \vdots \quad \vdots
\end{aligned}
$$

Newcastle University

PCA of the heptathlon dataset

```
> (pca1 = prcomp(x=heptathlon2))
Standard deviations (1, .., p=7):
[1] 6.65449663 5.27527745 1.16042524 0.78461166 0.37146157 0.24271694 0.05774546
# Notice that ∑ Var(Y_i) = 6.654^2 + ... + 0.058^2 = 74.27331, the total variation (as found earlier)
Rotation (n x k) = (7 x 7):
                PC1           PC2           PC3           PC4           PC5
H100M  -0.025531133  0.0230747007  0.0283836192  0.479533759  -0.16226365
HJ     -0.005106786  0.0003381213  0.0109047147 -0.004792405  -0.09779270
SP      0.003090843  0.0800303918  0.9948011149 -0.001825383  -0.04390316
R200M  -0.060571386 -0.0455319386  0.0003525911  0.856545544   0.28288157
LJ     -0.029240732  0.0038010184 -0.0544354626  0.175398972  -0.93881291
JT      0.068543052  0.9929624023 -0.0804990437  0.032406717   0.02273520
R800M  -0.995032897  0.0707152225 -0.0016611050 -0.067348425   0.01646365
                PC6           PC7
H100M  -0.859374071 -0.0566054449
HJ      0.081637617 -0.9917680240
SP      0.040955621  0.0186585712
R200M   0.424917692  0.0032450898
LJ      0.266850000  0.1132426862
JT      0.035780059 -0.0003526592
R800M  -0.009484905  0.0030507417
```

$$
\begin{aligned}
Y_1 &= -0.026X_1^* - 0.005X_2 + 0.003X_3 - 0.061X_4^* - 0.029X_5 + 0.069X_6 - 0.995X_7^* \\
Y_2 &= 0.023X_1^* + 0.0003X_2 + 0.080X_3 - 0.0461X_4^* + 0.004X_5 + 0.993X_6 + 0.071X_7^* \\
Y_3 &= 0.028X_1^* + 0.011X_2 + 0.995X_3 + 0.00031X_4^* - 0.054X_5 - 0.080X_6 - 0.002X_7^* \\
\vdots \quad & \quad \vdots \quad \vdots
\end{aligned}
$$

Newcastle University

PCA of the heptathlon dataset

PCA is **not scale invariant**!

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Y_1 is completely dominated by the 800m

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Y_1 is completely dominated by the 800m
- Y_2 is dominated by the javelin throw

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Y_1 is completely dominated by the 800m
- Y_2 is dominated by the javelin throw
- Y_3 is dominated by the shot put

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Y_1 is completely dominated by the 800m
- Y_2 is dominated by the javelin throw
- Y_3 is dominated by the shot put

This is no surprise:

```
> apply(heptathlon2, 2, var)
       H100M          HJ          SP       R200M          LJ          JT
 0.233480542  0.006324384  1.511650739  0.693497537  0.187046305 27.655802463
       R800M
43.985507635
```

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Y_1 is completely dominated by the 800m
- Y_2 is dominated by the javelin throw
- Y_3 is dominated by the shot put

This is no surprise:

```
> apply(heptathlon2, 2, var)
       H100M          HJ          SP       R200M          LJ          JT
 0.233480542  0.006324384  1.511650739  0.693497537  0.187046305 27.655802463
       R800M
43.985507635
```

And what if the 800m had been recorded in minutes (not seconds)? Or the javelin throw in feet (not metres)? **Changing the scales would change the whole analysis!**

PCA of the heptathlon dataset

PCA is **not scale invariant**!

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Not usually an issue if the original variables all have the same scale of measurement
- If several variables have considerably larger means and/or variances then they can **dominate a PCA** based on the covariance matrix (as is the default method in R)

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Not usually an issue if the original variables all have the same scale of measurement
- If several variables have considerably larger means and/or variances then they can **dominate a PCA** based on the covariance matrix (as is the default method in R)
- Solution: First **standardise** all variables (subtract means and divide by standard deviations)

PCA of the heptathlon dataset

PCA is **not scale invariant**!

- Not usually an issue if the original variables all have the same scale of measurement
- If several variables have considerably larger means and/or variances then they can **dominate a PCA** based on the covariance matrix (as is the default method in R)
- Solution: First **standardise** all variables (subtract means and divide by standard deviations)
- Note that this will **adjust the total variation** to give

$$\text{Total variation} = \sum_{i=1}^{p} 1 = p,$$

where p is the number of variables

PCA of the heptathlon dataset

```
> (pca2 = prcomp(x=heptathlon2, scale=TRUE))
Standard deviations (1, .., p=7):
[1] 1.6130032 1.2183228 1.0703309 0.8877202 0.7463326 0.5265001 0.3821523
# Check: ∑ Var(Y_i) = 1.613^2 + ... + 0.382^2 = 7,
Rotation (n x k) = (7 x 7):
                PC1         PC2         PC3        PC4         PC5        PC6
H100M  -0.444307758 -0.18648492  0.54748391 -0.1006979  0.18873506  0.1195608
HJ     -0.354825381 -0.16941702 -0.65972442  0.1312607  0.19560448  0.5826990
SP      0.001373487 -0.62701911  0.04473236  0.6872318  0.15204295 -0.3160055
R200M  -0.469979456  0.27204215  0.39768748  0.2519808  0.07615658  0.2929824
LJ     -0.486122948 -0.02282651 -0.25423876 -0.3911557  0.39416618 -0.5937738
JT      0.043843375 -0.68464607  0.14172860 -0.5227754 -0.20973167  0.2519431
R800M  -0.466443166 -0.01000331 -0.14193097  0.1027218 -0.83537221 -0.2107755
                PC7
H100M  -0.63876411
HJ     -0.12301165
SP      0.09787771
R200M   0.62595867
LJ      0.19387886
JT      0.35846022
R800M  -0.09675483
```

$$
\begin{aligned}
Y_1 &= -0.444X_1^* - 0.355X_2 + 0.001X_3 - 0.470X_4^* - 0.486X_5 + 0.044X_6 - 0.466X_7^* \\
Y_2 &= -0.186X_1^* - 0.169X_2 - 0.627X_3 + 0.2721X_4^* - 0.023X_5 - 0.685X_6 - 0.010X_7^* \\
Y_3 &= 0.547X_1^* - 0.660X_2 + 0.045X_3 + 0.3981X_4^* - 0.254X_5 + 0.142X_6 - 0.1422X_7^* \\
\vdots \quad & \quad \vdots \quad \vdots
\end{aligned}
$$

Newcastle University

PCA of the heptathlon dataset

```
> (pca2 = prcomp(x=heptathlon2, scale=TRUE))
Standard deviations (1, .., p=7):
[1] 1.6130032 1.2183228 1.0703309 0.8877202 0.7463326 0.5265001 0.3821523
# Check: ∑ Var(Y_i) = 1.613^2 + ... + 0.382^2 = 7,
Rotation (n x k) = (7 x 7):
                PC1         PC2         PC3        PC4         PC5        PC6
H100M  -0.444307758 -0.18648492  0.54748391 -0.1006979  0.18873506  0.1195608
HJ     -0.354825381 -0.16941702 -0.65972442  0.1312607  0.19560448  0.5826990
SP      0.001373487 -0.62701911  0.04473236  0.6872318  0.15204295 -0.3160055
R200M  -0.469979456  0.27204215  0.39768748  0.2519808  0.07615658  0.2929824
LJ     -0.486122948 -0.02282651 -0.25423876 -0.3911557  0.39416618 -0.5937738
JT      0.043843375 -0.68464607  0.14172860 -0.5227754 -0.20973167  0.2519431
R800M  -0.466443166 -0.01000331 -0.14193097  0.1027218 -0.83537221 -0.2107755
                PC7
H100M  -0.63876411
HJ     -0.12301165
SP      0.09787771
R200M   0.62595867
LJ      0.19387886
JT      0.35846022
R800M  -0.09675483
```

$$
\begin{aligned}
Y_1 &= -0.444X_1^* - 0.355X_2 + 0.001X_3 - 0.470X_4^* - 0.486X_5 + 0.044X_6 - 0.466X_7^* \\
Y_2 &= -0.186X_1^* - 0.169X_2 - 0.627X_3 + 0.2721X_4^* - 0.023X_5 - 0.685X_6 - 0.010X_7^* \\
Y_3 &= 0.547X_1^* - 0.660X_2 + 0.045X_3 + 0.3981X_4^* - 0.254X_5 + 0.142X_6 - 0.1422X_7^* \\
\vdots \quad & \vdots \quad \vdots
\end{aligned}
$$

Newcastle University

PCA of the heptathlon dataset: Summary so far

- 7 variables across 29 athletes. Do we need all 7 variables?

PCA of the heptathlon dataset: Summary so far

- 7 variables across 29 athletes. Do we need all 7 variables?
- After **re-scaling** and **standardisation**, we have 7 **Principal Components**:
  - **PC1:** Average measure of ability in running/jumping events
  - **PC2:** Average measure of ability in throwing events
  - **PC3:** Contrast between (i) jumping/middle-distance running events and (ii) throwing events and sprints

PCA of the heptathlon dataset: Summary so far

- 7 variables across 29 athletes. Do we need all 7 variables?
- After **re-scaling** and **standardisation**, we have 7 **Principal Components**:
  - **PC1:** Average measure of ability in running/jumping events
  - **PC2:** Average measure of ability in throwing events
  - **PC3:** Contrast between (i) jumping/middle-distance running events and (ii) throwing events and sprints
- Variability accounted for:

PC	1	2	3
%Var	$(1.613^2/7) \times 100 = 37.2\%$	$(1.218^2/7) \times 100 = 21.2\%$	$(1.070^2/7) \times 100 = 16.4\%$
Sum	37.2%	58.4%	74.8%

PCA of the heptathlon dataset: Summary so far

- 7 variables across 29 athletes. Do we need all 7 variables?
- After **re-scaling** and **standardisation**, we have 7 **Principal Components**:
  - **PC1:** Average measure of ability in running/jumping events
  - **PC2:** Average measure of ability in throwing events
  - **PC3:** Contrast between (i) jumping/middle-distance running events and (ii) throwing events and sprints
- Variability accounted for:

PC	1	2	3
%Var	$(1.613^2/7) \times 100 = 37.2\%$	$(1.218^2/7) \times 100 = 21.2\%$	$(1.070^2/7) \times 100 = 16.4\%$
Sum	37.2%	58.4%	74.8%

- Could probably just use first 3 PCs and still retain most of the information in the data!

Principal Components Analysis

We will now break for a problem session covering this section.

Please pause the video here and go to **Problem Session 3** of **Lesson 2**.

See you in a bit!

PC scores

So we can reduce the dimension of our dataset whilst still retaining much of the variation within it. So what?!

PC scores

So we can reduce the dimension of our dataset whilst still retaining much of the variation within it. So what?!

Plots of **PC scores** against each other, for the first two PCs, can often be used to reveal groupings or outliers.

PC scores

So we can reduce the dimension of our dataset whilst still retaining much of the variation within it. So what?!

Plots of **PC scores** against each other, for the first two PCs, can often be used to reveal groupings or outliers.

For example, in the heptathlon dataset, we had:

$$
\begin{aligned}
Y_1 &= -0.444X_1^* - 0.355X_2 + 0.001X_3 - 0.470X_4^* - 0.486X_5 + 0.044X_6 - 0.466X_7^* \\
Y_2 &= -0.186X_1^* - 0.169X_2 - 0.627X_3 + 0.2721X_4^* - 0.023X_5 - 0.685X_6 - 0.010X_7^*.
\end{aligned}
$$

PC scores

So we can reduce the dimension of our dataset whilst still retaining much of the variation within it. So what?!

Plots of **PC scores** against each other, for the first two PCs, can often be used to reveal groupings or outliers.

For example, in the heptathlon dataset, we had:

$$
\begin{aligned}
Y_1 &= -0.444X_1^* - 0.355X_2 + 0.001X_3 - 0.470X_4^* - 0.486X_5 + 0.044X_6 - 0.466X_7^* \\
Y_2 &= -0.186X_1^* - 0.169X_2 - 0.627X_3 + 0.2721X_4^* - 0.023X_5 - 0.685X_6 - 0.010X_7^*.
\end{aligned}
$$

The observed scores for each athlete, $Y_1^{(\text{obs})}$ and $Y_2^{(\text{obs})}$, can be found by substituting in their actual values for $X_1^*, X_2, \ldots$.

Tedious to do by hand – easy to do in R!

PC scores

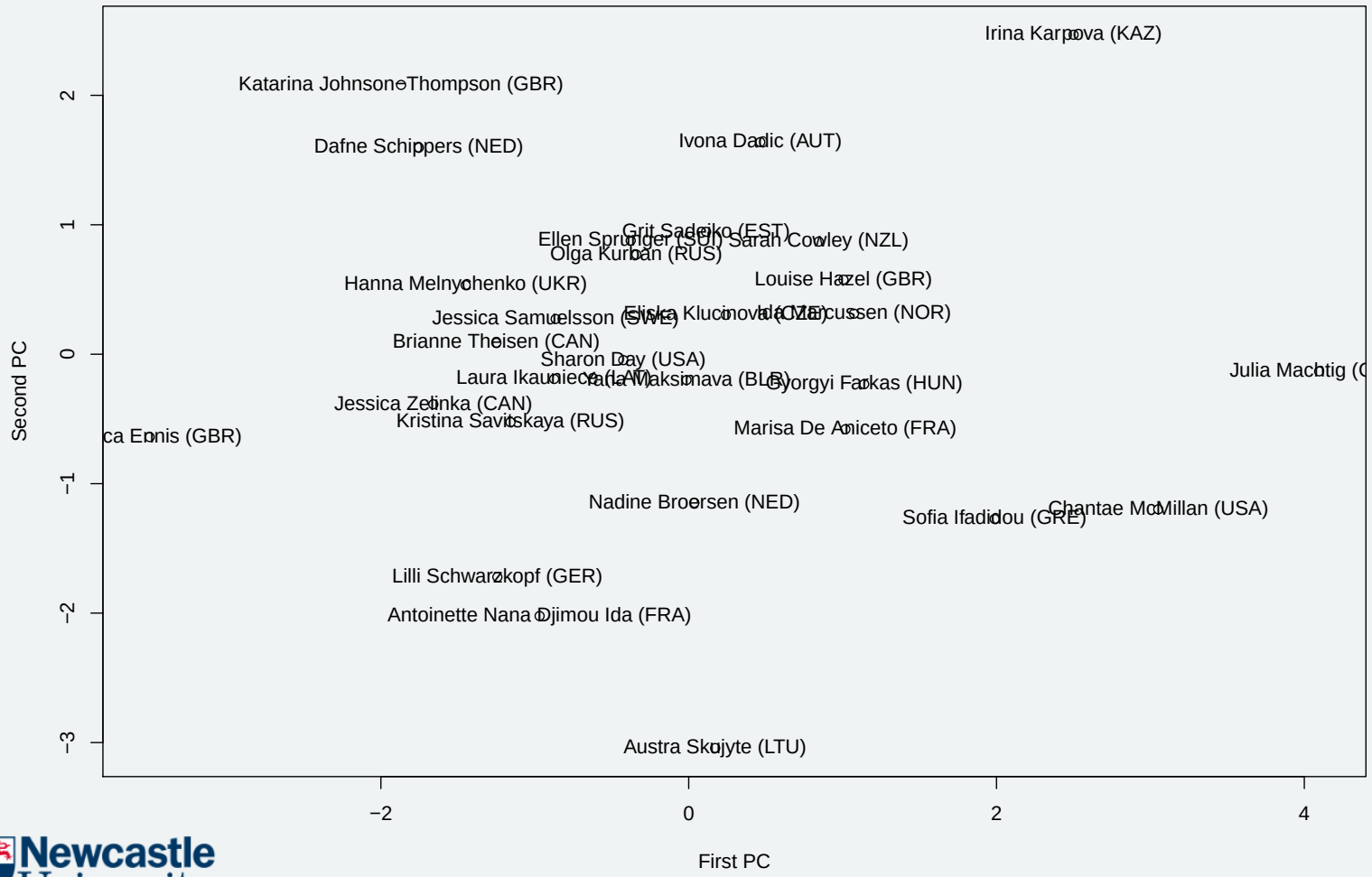
```
> pca2$x[,1]
            Jessica Ennis (GBR)          Lilli Schwarzkopf (GER)
                    -3.50217160                       -1.24307944
           Austra Skujyte (LTU) Antoinette Nana Djimou Ida (FRA)
                     0.17195226                       -0.96884007
          Jessica Zelinka (CAN)        Kristina Savitskaya (RUS)
                    -1.65943423                       -1.15818453
          Laura Ikauniece (LAT)          Hanna Melnychenko (UKR)
                    -0.87405110                       -1.44906334
          Brianne Theisen (CAN)            Dafne Schippers (NED)
                    -1.24982570                       -1.75337081
          Nadine Broersen (NED)         Jessica Samuelsson (SWE)
                     0.03678361                       -0.86435915
 Katarina Johnson-Thompson (GBR)                 Sharon Day (USA)
                    -1.86873533                       -0.42476488
           Yana Maksimava (BLR)           Eliska Klucinova (CZE)
                    -0.01428641                        0.24099396
           Ellen Sprunger (SUI)                Olga Kurban (RUS)
                    -0.37690278                       -0.33981366
        Marisa De Aniceto (FRA)            Gyorgyi Farkas (HUN)
                     1.01829329                        1.14004303
             Grit Sadeiko (EST)            Sofia Ifadidou (GRE)
                     0.11303330                        1.98774068
              Ivona Dadic (AUT)              Sarah Cowley (NZL)
                     0.46512623                        0.84474007
             Louise Hazel (GBR)             Ida Marcussen (NOR)
                     1.00597850                        1.07407876
         Chantae McMillan (USA)             Julia Machtig (GER)
                     3.05124041                        4.09640007
            Irina Karpova (KAZ)
                     2.50047888
```

PC scores

```
> pca2$x[,2]
            Jessica Ennis (GBR)          Lilli Schwarzkopf (GER)
                     -0.6398539                        -1.7214147
           Austra Skujyte (LTU) Antoinette Nana Djimou Ida (FRA)
                     -3.0425385                        -2.0221641
          Jessica Zelinka (CAN)        Kristina Savitskaya (RUS)
                     -0.3869343                        -0.5248583
          Laura Ikauniece (LAT)          Hanna Melnychenko (UKR)
                     -0.1908915                         0.5337838
          Brianne Theisen (CAN)            Dafne Schippers (NED)
                      0.0882352                         1.5966040
          Nadine Broersen (NED)         Jessica Samuelsson (SWE)
                     -1.1480645                         0.2723521
 Katarina Johnson-Thompson (GBR)                 Sharon Day (USA)
                      2.0803685                        -0.0457769
           Yana Maksimava (BLR)           Eliska Klucinova (CZE)
                     -0.1995714                         0.3041175
           Ellen Sprunger (SUI)                Olga Kurban (RUS)
                      0.8778327                         0.7701118
        Marisa De Aniceto (FRA)            Gyorgyi Farkas (HUN)
                     -0.5792726                        -0.2312338
             Grit Sadeiko (EST)            Sofia Ifadidou (GRE)
                      0.9458016                        -1.2727123
              Ivona Dadic (AUT)              Sarah Cowley (NZL)
                      1.6386397                         0.8700351
             Louise Hazel (GBR)             Ida Marcussen (NOR)
                      0.5758550                         0.3134866
         Chantae McMillan (USA)             Julia Machtig (GER)
                     -1.1981219                        -0.1319795
            Irina Karpova (KAZ)
                      2.4681645
```

PC scores

```
> plot(pca2$x[,1],pca2$x[,2], xlab="First PC", ylab="Second PC")
> text(pca2$x[,1],pca2$x[,2],labels = rownames(heptathlon2))
```



PCA of the microarray dataset

To illustrate the power of PCA in the analysis of ‘big data’, we now consider a much more detailed dataset!

The **microrarray dataset** consists of measurements on the expression levels of 6,830 genes from 64 samples taken from malignant tumours in different patients.

This really is a **big dataset**!

- Scatterplot matrix: 23 million plots
- No simple way of visualising these data

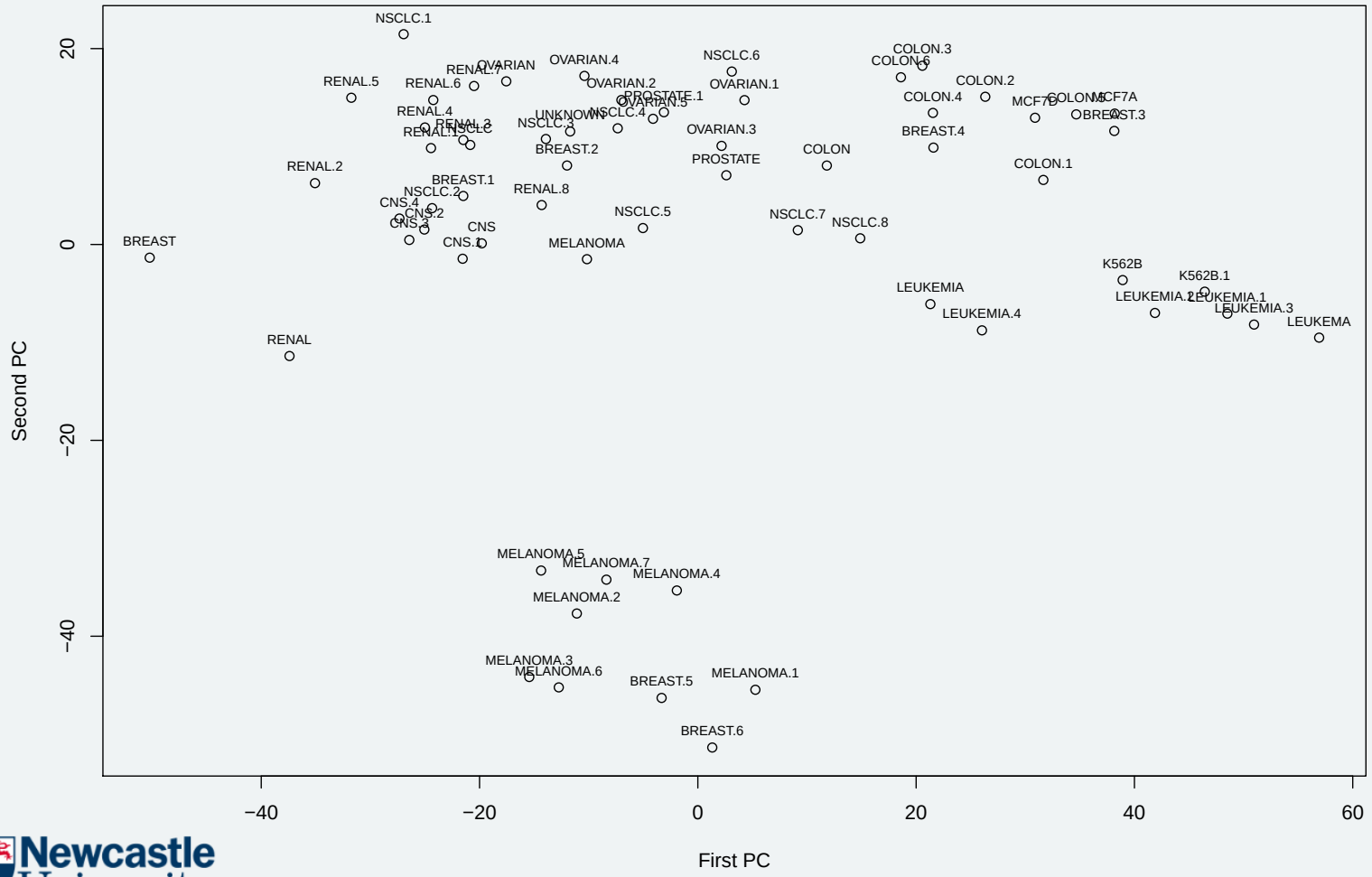
PCA of the microarray dataset

```
> dim(nci)
[1]    64  6830

> nci[1:10,1:10]
                 g1        g2        g3        g4        g5            g6
CNS        0.300000  1.180000  0.550000  1.140000 -0.265000 -7.00000e-02
CNS.1      0.679961  1.289961  0.169961  0.379961  0.464961  5.79961e-01
CNS.2      0.940000 -0.040000 -0.170000 -0.040000 -0.605000  0.00000e+00
RENAL      0.280000 -0.310000  0.680000 -0.810000  0.625000 -1.39000e-17
BREAST     0.485000 -0.465000  0.395000  0.905000  0.200000 -5.00000e-03
CNS.3      0.310000 -0.030000 -0.100000 -0.460000 -0.205000 -5.40000e-01
CNS.4     -0.830000  0.000000  0.130000 -1.630000  0.075000 -3.60000e-01
BREAST.1  -0.190000 -0.870000 -0.450000  0.080000  0.005000  3.50000e-01
NSCLC      0.460000  0.000000  1.150000 -1.400000 -0.005000 -7.00000e-01
NSCLC.1    0.760000  1.490000  0.280000  0.100000 -0.525000  3.60000e-01
                 g7        g8           g9          g10
CNS        0.350000 -0.315000 -0.45000000 -0.65498050
CNS.1      0.699961  0.724961 -0.04003899 -0.28501950
CNS.2      0.090000  0.645000  0.43000000  0.47501950
RENAL      0.170000  0.245000  0.02000000  0.09501949
BREAST     0.085000  0.110000  0.23500000  1.49001949
CNS.3     -0.640000 -0.585000 -0.77000000 -0.24498050
CNS.4      0.100000  0.155000 -0.29000000 -0.08498051
BREAST.1  -0.040000 -0.265000 -0.31000000 -0.24498050
NSCLC     -0.920000 -0.515000 -0.28000000 -0.11498050
NSCLC.1    0.600000  0.175000  0.58000000  1.14501900
```

Newcastle University

PCA of the microarray dataset

```
> pca3 = prcomp(x = nci)
> plot(pca3$x[,1], pca3$x[,2], xlab="First PC", ylab="Second PC")
> text(pca3$x[,1], pca3$x[,2],labels=rownames(nci), cex=0.7,pos=3)
```



Conclusions

1. **Simple graphical/numerical procedures** for exploring multivariate datasets
   - Scatterplot matrices and 3-D scatterplots
   - Correlation matrices, with p-values

Conclusions

1. **Simple graphical/numerical procedures** for exploring multivariate datasets
   - Scatterplot matrices and 3-D scatterplots
   - Correlation matrices, with p-values
2. **Other procedures** – e.g. Chernoff's faces, star symbol plots, Mahalanobis metric

Conclusions

1. **Simple graphical/numerical procedures** for exploring multivariate datasets
   - Scatterplot matrices and 3-D scatterplots
   - Correlation matrices, with p-values
2. **Other procedures** – e.g. Chernoff's faces, star symbol plots, Mahalanobis metric
3. **Principal Components Analysis**

Conclusions

1. **Simple graphical/numerical procedures** for exploring multivariate datasets
   - Scatterplot matrices and 3-D scatterplots
   - Correlation matrices, with p-values
2. **Other procedures** – e.g. Chernoff's faces, star symbol plots, Mahalanobis metric
3. **Principal Components Analysis**
   - Dimension reduction, whilst still retaining most of the information in the data
   - Spotting groupings/outliers using score plots
   - Be careful – might need to pre-process your data!

Conclusions

1. **Simple graphical/numerical procedures** for exploring multivariate datasets
   - Scatterplot matrices and 3-D scatterplots
   - Correlation matrices, with p-values
2. **Other procedures** – e.g. Chernoff's faces, star symbol plots, Mahalanobis metric
3. **Principal Components Analysis**
   - Dimension reduction, whilst still retaining most of the information in the data
   - Spotting groupings/outliers using score plots
   - Be careful – might need to pre-process your data!
4. **Further considerations** – use PCs, rather than original variables, in statistical models (see later lecture)