Principal Components

- Have measurements on (possibly large) number of variables on some individuals.
- Question: can we describe data using fewer variables (because original variables correlated in some way)?
- Look for direction (linear combination of original variables) in which values *most spread out*. This is *first principal component*.
- Second principal component then direction uncorrelated with this in which values then most spread out. And so on.
- See whether small number of principal components captures most of variation in data.
- Might try to interpret principal components.
- If 2 components good, can make plot of data.
- (Akin to ideas in discriminant/canonical variables analysis, but no groups here.)

Small example: 2 test scores for 8 people

Principal component analysis

Strongly correlated, so data nearly 1-dimensional. Make a score summarizing this one dimension.
Code like this:

```
options linesize=70;

data test;
  infile "test12.dat";
  input score1 score2;

proc princomp out=fred;

proc print;
```

The output

Correlation Matrix

	score1	score2
score1	1.0000	0.9891
score2	0.9891	1.0000

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	1.98907796	1.97815591	0.9945	0.9945
2	0.01092204		0.0055	1.0000

- The two variables are very highly correlated.
- Look at *eigenvalues*:
  - First one is much bigger than rest, so data “almost” 1-dimensional.
  - Last column: first principal component accounts for almost all (99.45%) of variability in data, so we do fine by summarizing data by 1st principal component.
  - Generally: consider retaining components with eigenvalues bigger than 1.

Eigenvectors

Eigenvectors

	Prin1	Prin2
score1	0.707107	0.707107
score2	0.707107	-.707107

- Eigenvectors show how principal components depend on original variables (standardized).
- 1st principal component is basically sum of `score1` and `score2`, standardized.
- If correlation between 2 test scores had been negative, 1st eigenvector would have said “look at difference”.

Output data set

Obs	score1	score2	Prin1	Prin2
1	2	9	-1.93801	-0.13749
2	16	40	1.60878	-0.05216
3	8	17	-0.71306	0.19418
4	18	43	2.03571	0.03979
5	10	25	-0.00698	0.00698
6	4	10	-1.62274	0.06612
7	10	27	0.10468	-0.10468
8	12	30	0.53161	-0.01273

- Values on `Prin1` identify each person as a “high scorer” (positive) or “low scorer” (negative).
- `Prin1` and `Prin2` called *principal component scores*.

Track running data

(1984) track running records for distances 100m to marathon, arranged by country. Countries labelled by (mostly) Internet domain names — actual names not in file.

```
 10.39  20.81  46.84  1.81  3.70  14.04  29.36  137.72  ar (Argentina)
 10.31  20.06  44.84  1.74  3.57  13.28  27.66  128.30  au (Australia)
 10.44  20.81  46.82  1.79  3.60  13.26  27.72  135.90  at (Austria)
 10.34  20.68  45.04  1.73  3.60  13.22  27.45  129.95  be (Belgium)
 10.28  20.58  45.91  1.80  3.75  14.68  30.55  146.62  bm (Bermuda)
 10.22  20.43  45.21  1.73  3.66  13.62  28.62  133.13  br (Brazil)
 10.64  21.52  48.30  1.80  3.85  14.45  30.28  139.95  bu (Burma)
 10.17  20.22  45.68  1.76  3.63  13.55  28.09  130.15  ca (Canada)
 10.34  20.80  46.20  1.79  3.71  13.61  29.30  134.03  cl (Chile)
 ....
 10.71  21.43  47.60  1.79  3.67  13.56  28.58  131.50  tr (Turkey)
  9.93  19.75  43.86  1.73  3.53  13.20  27.43  128.22  us (United States)
 10.07  20.00  44.60  1.75  3.59  13.20  27.53  130.55  ru (USSR)
 10.82  21.86  49.00  2.02  4.24  16.28  34.71  161.83  ws (Western Samoa)
```

Data and aims

- Times in seconds 100m-400m, in minutes for rest (800m, 1500m, 5000m, 10000m, marathon).
- This taken care of by (automatic) standardization.
- 8 variables; can we summarize by fewer and gain some insight?
- In particular, if 2 components tell most of story, what do we see in a plot?

Code

```
options linesize=70;

data track;
  infile "men_track_field.dat";
  input m100 m200 m400 m800 m1500 m5000 m10000 marat
    country $;

proc princomp out=PC;

proc print data=PC;
  var country Prin1 Prin2;

proc plot data=PC;
  plot Prin2*Prin1 = '*' $ country;
```

Correlation matrix

Correlation Matrix

	m100	m200	m400	m800	m1500	m5000	m10000	marath
m100	1.0000	0.9226	0.8411	0.7560	0.7002	0.6195	0.6325	0.51
m200	0.9226	1.0000	0.8507	0.8066	0.7750	0.6954	0.6965	0.59
m400	0.8411	0.8507	1.0000	0.8702	0.8353	0.7786	0.7872	0.70
m800	0.7560	0.8066	0.8702	1.0000	0.9180	0.8636	0.8690	0.80
m1500	0.7002	0.7750	0.8353	0.9180	1.0000	0.9281	0.9347	0.86
m5000	0.6195	0.6954	0.7786	0.8636	0.9281	1.0000	0.9746	0.93
m10000	0.6325	0.6965	0.7872	0.8690	0.9347	0.9746	1.0000	0.94
marathon	0.5199	0.5962	0.7050	0.8065	0.8655	0.9322	0.9432	1.00

All variables positively correlated, but less so as gap between running distances increases.

The eigenvalues

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	6.62214613	5.74452784	0.8278	0.8278
2	0.87761829	0.71829715	0.1097	0.9375
3	0.15932114	0.03527176	0.0199	0.9574
4	0.12404939	0.04416911	0.0155	0.9729
5	0.07988027	0.01191512	0.0100	0.9829
6	0.06796515	0.02154562	0.0085	0.9914
7	0.04641953	0.02381943	0.0058	0.9972
8	0.02260010		0.0028	1.0000

Only 1st is bigger than 1, but 2nd is much bigger than others, so include that as well.

The eigenvectors

Eigenvectors

	Prin1	Prin2	Prin3	Prin4
m100	0.317556	0.566878	0.332262	0.127628
m200	0.336979	0.461626	0.360657	-.259116
m400	0.355645	0.248273	-.560467	0.652341
m800	0.368684	0.012430	-.532482	-.479999
m1500	0.372810	-.139797	-.153443	-.404510
m5000	0.364374	-.312030	0.189764	0.029588
m10000	0.366773	-.306860	0.181752	0.080069
marathon	0.341926	-.438963	0.263209	0.299512

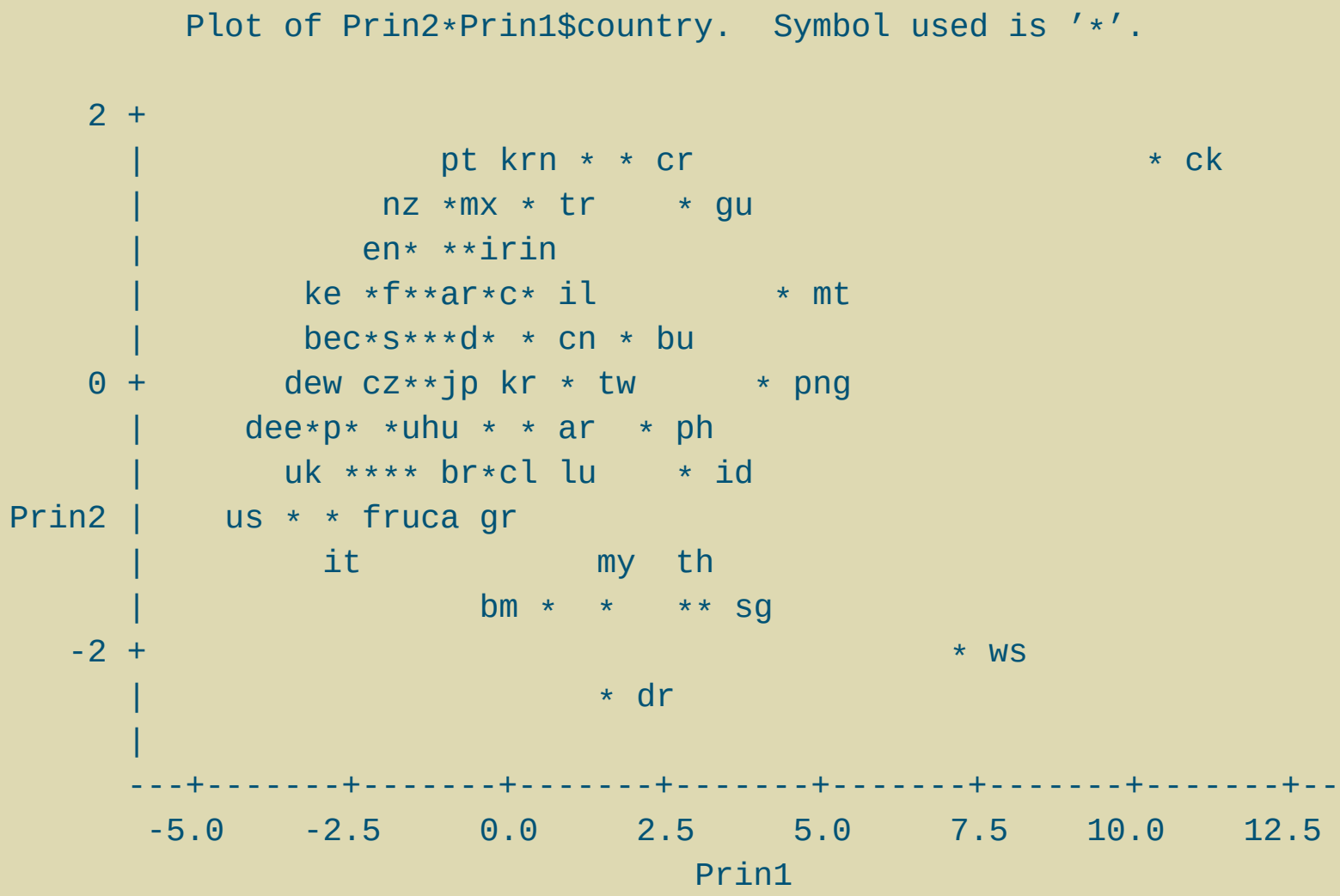
- `Prin1` basically average of record times: “how good a country is at running”.
- `Prin2` contrasts sprinting with distance running.
- `Prin3` contrasts longer sprints with everything else.
- More, but not going to keep `Prin3` and beyond.

Principal component scores (selected)

`Prin1` measures “good overall” (negative best), `Prin2` measures “sprinting vs. distance” (negative: sprinting, positive: distance). (SAS has turned this around: check back with original data.)

Obs	country	Prin1	Prin2
1	ar	0.2619	-0.34488
2	au	-2.4464	-0.21617
4	be	-2.0413	0.26195
8	ca	-1.7464	-0.50035
12	ck	10.5556	1.50877
18	fr	-2.1719	-0.50289
19	dee	-2.5901	-0.31067
20	dew	-2.5527	-0.41137
29	it	-2.7269	-0.98986
30	jp	-1.2379	0.41357
31	ke	-2.1683	0.53371
37	mx	-0.6785	0.84175
43	pl	-2.0006	-0.46260
44	pt	-0.9164	1.30473
45	rm	-1.1965	0.53077
53	us	-3.4306	-1.11019
54	ru	-2.6269	-0.75696
55	ws	7.2312	-1.90208

Component scores plot



Data TYPE=CORR

- Just as data TYPE=DISTANCE (we used for clustering), also TYPE=CORR, used for matrices of correlations.
- Procedure PROC CORR will produce this as output.
- Example small data set (three variables):

```
 3  7 20
 4 10 16
 6 15 11
 9 18  8
```

  x_2 is just over twice x_1, while x_3 goes down as x_1 and x_2 goes up. So expect some high positive and negative correlations.

Using PROC CORR

- Code like this:

```
data xc;
  infile "xcorr.dat";
  input x1 x2 x3;

proc corr out=fred;

proc print;
```

- PROC CORR itself produces some output (ignored) and output data set looks like this:

Obs	_TYPE_	_NAME_	x1	x2	x3
1	MEAN		5.50000	12.5000	13.7500
2	STD		2.64575	4.9329	5.3151
3	N		4.00000	4.0000	4.0000
4	CORR	x1	1.00000	0.9705	-0.9600
5	CORR	x2	0.97054	1.0000	-0.9980
6	CORR	x3	-0.96001	-0.9980	1.0000

- Last 3 lines are matrix of correlations; values as expected.

TYPE=CORR data set

- Full data set includes mean and SD of each variable, and number of observations for each (same for each variable). Entry in new variable _TYPE_ says what each row of numbers is.
- Each correlation is of *two* variables, so entry in row of _NAME_ and variable column says which two variables involved.
- Can create TYPE=CORR data set yourself. Not everything has to be specified:
  - if _TYPE_ missing, CORR assumed.
  - if _NAME_ missing, variables may not get names (but OK if planning to use all in analysis).
  - Correlation eg. between x_1 and x_2 same as between x_2 and x_1, so can give redundant correlations as . (missing).
  - PROC PRINCOMP only uses correlations (not mean, SD or sample size — no testing). So can still do if you have only correlations.

Doing principal components with (above) correlations o

- Create data file like this:

```
1 0.9705 -0.9600
. 1      -0.9980
. .       1
```

- Code like below. Note: no actual data implies no component scores (and no output data set).

```
data yc(type=corr);
  infile "ycorr.dat";
  input x1 x2 x3;

proc princomp;
```

- Can also use PROC PRINT to check proper reading of data.
- PROC PRINCOMP handles this kind of data automatically.

Output

The PRINCOMP Procedure

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	2.95242147	2.90600335	0.9841	0.9841
2	0.04641812	0.04525771	0.0155	0.9996
3	0.00116041		0.0004	1.0000

Eigenvectors

	Prin1	Prin2	Prin3
x1	0.573007	0.811856	-.112035
x2	0.580528	-.305586	0.754721
x3	-.578489	0.497500	0.646408

- Data behind correlations effectively one-dimensional.
- Principal component made of first two variables minus third almost entirely summarizes data.