Discriminant analysis

Discriminant analysis

- ANOVA and MANOVA: predict a (counted/measured) response from group membership.
- Discriminant analysis: predict group membership based on counted/measured variables.
- Covers same ground as logistic regression (and its variations), but emphasis on classifying observed data into correct groups.
- Does so by searching for linear combination of original variables that best separates data into groups (canonical variables).
- Assumption here that groups are known (for data we have). If trying to "best separate" data into unknown groups, see *cluster analysis*.
- Examples: revisit seed yield and weight data, peanut data, professions/activities data; remote-sensing data.

Packages

```
library(MASS)
library(tidyverse)
library(ggrepel)
library(ggbiplot)
library(MVTests) # for Box M test
library(conflicted)
conflict_prefer("arrange", "dplyr")
conflict_prefer("summarize", "dplyr")
conflict_prefer("select", "dplyr")
conflict_prefer("filter", "dplyr")
conflict_prefer("mutate", "dplyr")
```

- `ggrepel` allows labelling points on a plot so they don't overwrite each other.
- `ggbiplot` uses `plyr` rather than `dplyr`, which has functions by similar names.

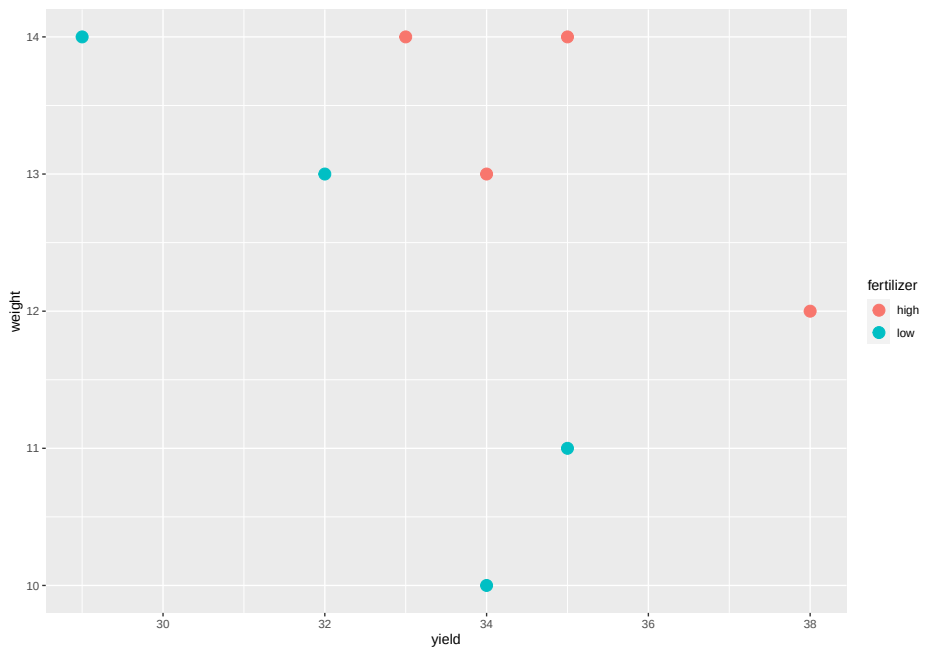
About `select`

- Both `dplyr` (in `tidyverse`) and `MASS` have a function called `select`, and *they do different things*.
- How do you know which `select` is going to get called?
- With `library`, the one loaded *last* is visible, and others are not.
- Thus we can access the `select` in `dplyr` but not the one in `MASS`. If we wanted that one, we'd have to say `MASS::select`.
- You can load `MASS` before `tidyverse`. If you do it the other way around, the `tidyverse` `select`, which you want to use, would be the invisible one.
- Alternative: load `conflicted` package. Any time you load two packages containing functions with same name, you get error and have to choose between them.

Example 1: seed yields and weights

```
my_url <- "http://ritsokiguess.site/datafiles/manova1.txt"
hilo <- read_delim(my_url, " ")
g <- ggplot(hilo, aes(x = yield, y = weight,
  colour = fertilizer)) + geom_point(size = 4)
```

Recall data from MANOVA: needed a multivariate analysis to find difference in seed yield and weight based on whether they were high or low fertilizer.



Basic discriminant analysis

```
hilo.1 <- lda(fertilizer ~ yield + weight, data = hilo)
```

- Uses `lda` from package MASS.
- "Predicting" group membership from measured variables.

Output

```
hilo.1
```

```
## Call:
## lda(fertilizer ~ yield + weight, data = hilo)
##
## Prior probabilities of groups:
## high  low
##  0.5  0.5
##
## Group means:
##      yield weight
## high  35.0  13.25
## low   32.5  12.00
##
## Coefficients of linear discriminants:
##                LD1
## yield  -0.7666761
## weight -1.2513563
```

Things to take from output

- Group means: high-fertilizer plants have (slightly) higher mean yield and weight than low-fertilizer plants.
- "Coefficients of linear discriminants": `LD1`, `LD2`, ...are scores constructed from observed variables that best separate the groups.
- For any plant, get LD1 score by taking -0.76 times yield plus -1.25 times weight, add up, standardize.
- the LD1 coefficients are like slopes:
  - if yield higher, LD1 score for a plant lower
  - if weight higher, LD1 score for a plant lower
- High-fertilizer plants have higher yield and weight, thus low (negative) LD1 score. Low-fertilizer plants have low yield and weight, thus high (positive) LD1 score.
- One LD1 score for each observation. Plot with actual groups.

How many linear discriminants?

- Smaller of these:
  - Number of variables
  - Number of groups *minus 1*
- Seed yield and weight: 2 variables, 2 groups, $\min(2, 2-1) = 1$.

Getting LD scores

Feed output from LDA into `predict`:

```
hilo.pred <- predict(hilo.1)
```

Component `x` contains LD score(s), here in descending order:

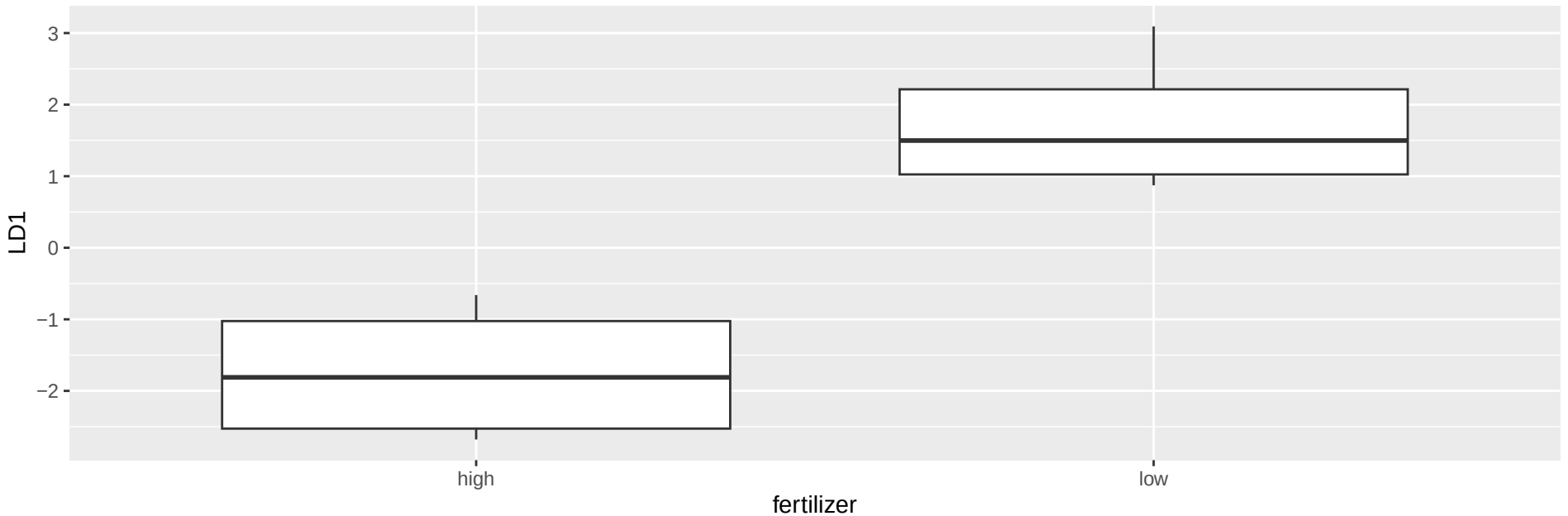
```
d <- cbind(hilo, hilo.pred$x) %>% arrange(desc(LD1))
d
```

	fertilizer	yield	weight	LD1
1	low	34	10	3.0931414
2	low	29	14	1.9210963
3	low	35	11	1.0751090
4	low	32	13	0.8724245
7	high	34	13	-0.6609276
5	high	33	14	-1.1456079
6	high	38	12	-2.4762756
8	high	35	14	-2.6789600

Plotting LD1 scores

With one LD score, plot against (true) groups, eg. boxplot:

```
ggplot(d, aes(x = fertilizer, y = LD1)) + geom_boxplot()
```



Potentially misleading

```
hilo.1$scaling
```

```
##                LD1
## yield  -0.7666761
## weight -1.2513563
```

- These are like regression slopes: change in LD1 score for 1-unit change in variables.

What else is in `hilo.pred`?

```
names(hilo.pred)
```

```
## [1] "class"     "posterior" "x"
```

- `class`: predicted fertilizer level (based on values of `yield` and `weight`).
- `posterior`: predicted probability of being low or high fertilizer given `yield` and `weight`.
- `x`: scores for each linear discriminant (here is only LD1) on each observation.

Predictions and predicted groups

...based on `yield` and `weight`:

```
cbind(hilo, predicted = hilo.pred$class)
```

fertilizer	yield	weight	predicted
low	34	10	low
low	29	14	low
low	35	11	low
low	32	13	low
high	33	14	high
high	38	12	high
high	34	13	high
high	35	14	high

Count up correct and incorrect classification

```
table(obs = hilo$fertilizer, pred = hilo.pred$class)
```

```
##       pred
## obs    high low
##   high    4   0
##   low     0   4
```

- Each predicted fertilizer level is exactly same as observed one (perfect prediction).
- Table shows no errors: all values on top-left to bottom-right diagonal.

Posterior probabilities

show how clear-cut the classification decisions were:

```
pp <- round(hilo.pred$posterior, 4)
d <- cbind(hilo, hilo.pred$x, pp)
d
```

fertilizer	yield	weight	LD1	high	low
low	34	10	3.0931414	0.0000	1.0000
low	29	14	1.9210963	0.0012	0.9988
low	35	11	1.0751090	0.0232	0.9768
low	32	13	0.8724245	0.0458	0.9542
high	33	14	-1.1456079	0.9818	0.0182
high	38	12	-2.4762756	0.9998	0.0002
high	34	13	-0.6609276	0.9089	0.0911
high	35	14	-2.6789600	0.9999	0.0001

Only obs. 7 has any doubt: `yield` low for a high-fertilizer, but high `weight` makes up for it.

Example 2: the peanuts

```
my_url <- "http://ritsokiguess.site/datafiles/peanuts.txt"
peanuts <- read_delim(my_url, " ")
peanuts
```

obs	location	variety	y	smk	w
1	1	5	195.3	153.1	51.4
2	1	5	194.3	167.7	53.7
3	2	5	189.7	139.5	55.5
4	2	5	180.4	121.1	44.4
5	1	6	203.0	156.8	49.8
6	1	6	195.9	166.0	45.8
7	2	6	202.7	166.1	60.4
8	2	6	197.6	161.8	54.1
9	1	8	193.5	164.5	57.8
10	1	8	187.0	165.1	58.6
11	2	8	201.5	166.8	65.0
12	2	8	200.0	173.8	67.2

- Recall: `location` and `variety` both significant in MANOVA. Make combo of them (over):

Location-variety combos

```
peanuts %>%
   unite(combo, c(variety, location)) -> peanuts.combo
peanuts.combo
```

obs	combo	y	smk	w
1	5_1	195.3	153.1	51.4
2	5_1	194.3	167.7	53.7
3	5_2	189.7	139.5	55.5
4	5_2	180.4	121.1	44.4
5	6_1	203.0	156.8	49.8
6	6_1	195.9	166.0	45.8
7	6_2	202.7	166.1	60.4
8	6_2	197.6	161.8	54.1
9	8_1	193.5	164.5	57.8
10	8_1	187.0	165.1	58.6
11	8_2	201.5	166.8	65.0
12	8_2	200.0	173.8	67.2

Discriminant analysis

```
peanuts.1 <- lda(combo ~ y + smk + w, data = peanuts.combo)
peanuts.1
```

```
## Call:
## lda(combo ~ y + smk + w, data = peanuts.combo)
##
## Prior probabilities of groups:
##       5_1       5_2       6_1       6_2       8_1       8_2
## 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
##
## Group means:
##          y    smk     w
## 5_1 194.80 160.40 52.55
## 5_2 185.05 130.30 49.95
## 6_1 199.45 161.40 47.80
## 6_2 200.15 163.95 57.25
## 8_1 190.25 164.80 58.20
## 8_2 200.75 170.30 66.10
##
## Coefficients of linear discriminants:
##            LD1         LD2         LD3
## y    0.4027356  0.02967881  0.18839237
## smk  0.1727459 -0.06794271 -0.09386294
## w   -0.5792456 -0.16300221  0.07341123
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.8424 0.1317 0.0258
```

Comments

- Now 3 LDs (3 variables, 6 groups, $\min(3, 6-1) = 3$).
- Relationship of LDs to original variables. Look for coeffs far from zero:

```
peanuts.1$scaling
```

```
##              LD1          LD2         LD3
## y     0.4027356  0.02967881  0.18839237
## smk   0.1727459 -0.06794271 -0.09386294
## w    -0.5792456 -0.16300221  0.07341123
```

- high `LD1` mainly high `y` or low `w`.
- high `LD2` mainly low `w`.
- Proportion of trace values show relative importance of LDs: `LD1` much more important than `LD2`; `LD3` worthless.

The predictions and misclassification

```
peanuts.pred <- predict(peanuts.1)
table(
  obs = peanuts.combo$combo,
  pred = peanuts.pred$class
)
```

```
##       pred
## obs    5_1 5_2 6_1 6_2 8_1 8_2
##   5_1    2   0   0   0   0   0
##   5_2    0   2   0   0   0   0
##   6_1    0   0   2   0   0   0
##   6_2    1   0   0   1   0   0
##   8_1    0   0   0   0   2   0
##   8_2    0   0   0   0   0   2
```

Actually classified very well. Only one 6_2 classified as a 5_1, rest all correct.

Posterior probabilities

```
pp <- round(peanuts.pred$posterior, 2)
peanuts.combo %>%
  select(-c(y, smk, w)) %>%
  cbind(., pred = peanuts.pred$class, pp)
```

obs	combo	pred	5_1	5_2	6_1	6_2	8_1	8_2
1	5_1	5_1	0.69	0	0	0.31	0.00	0.00
2	5_1	5_1	0.73	0	0	0.27	0.00	0.00
3	5_2	5_2	0.00	1	0	0.00	0.00	0.00
4	5_2	5_2	0.00	1	0	0.00	0.00	0.00
5	6_1	6_1	0.00	0	1	0.00	0.00	0.00
6	6_1	6_1	0.00	0	1	0.00	0.00	0.00
7	6_2	6_2	0.13	0	0	0.87	0.00	0.00
8	6_2	5_1	0.53	0	0	0.47	0.00	0.00
9	8_1	8_1	0.02	0	0	0.02	0.75	0.21
10	8_1	8_1	0.00	0	0	0.00	0.99	0.01
11	8_2	8_2	0.00	0	0	0.00	0.03	0.97
12	8_2	8_2	0.00	0	0	0.00	0.06	0.94

Some doubt about which combo each plant belongs in, but not too much. The one misclassified plant was a close call.

Discriminant scores, again

- How are discriminant scores related to original variables?
- Construct data frame with original data and discriminant scores side by side:

```
peanuts.1$scaling
```

```
##              LD1         LD2         LD3
## y    0.4027356  0.02967881  0.18839237
## smk  0.1727459 -0.06794271 -0.09386294
## w   -0.5792456 -0.16300221  0.07341123
```

```
lds <- round(peanuts.pred$x, 2)
mm <- with(peanuts.combo,
           data.frame(combo, y, smk, w, lds))
```

- LD1 positive if y large and/or w small.
- LD2 positive if w small.

Discriminant scores for data

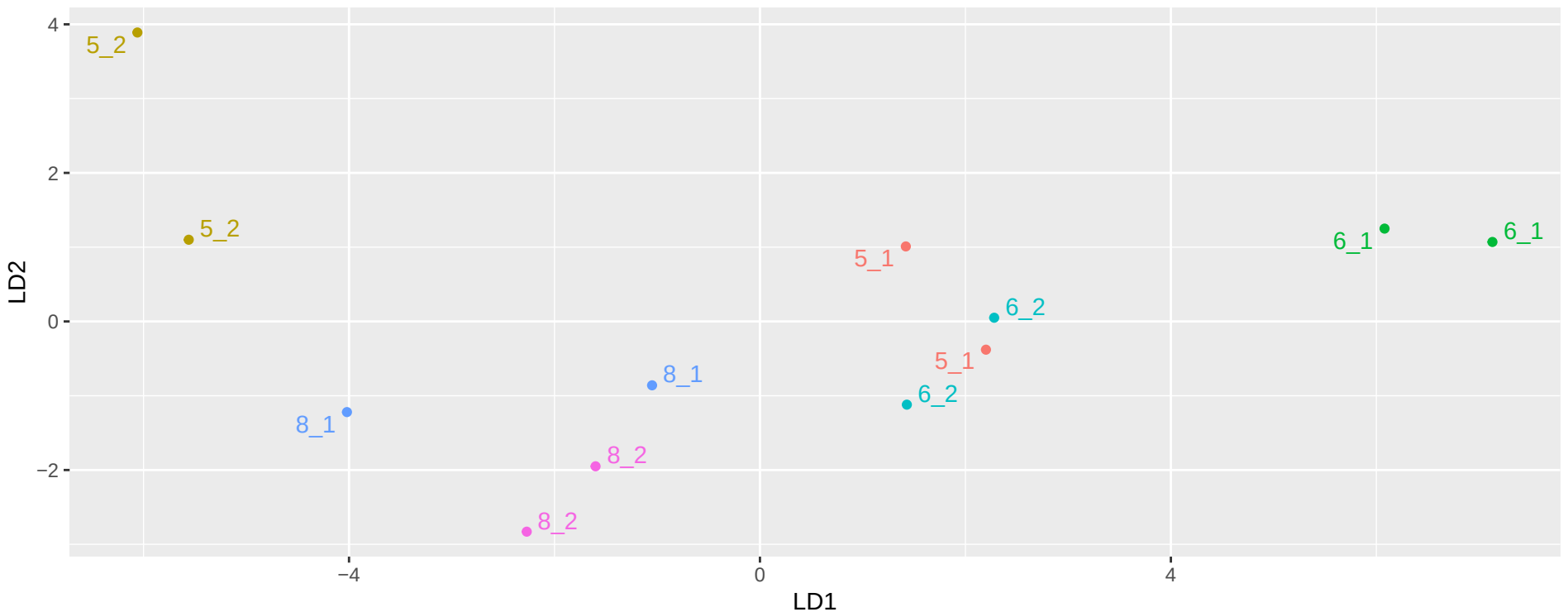
```
mm
```

combo	y	smk	w	LD1	LD2	LD3
5_1	195.3	153.1	51.4	1.42	1.01	0.26
5_1	194.3	167.7	53.7	2.20	-0.38	-1.13
5_2	189.7	139.5	55.5	-5.56	1.10	0.79
5_2	180.4	121.1	44.4	-6.06	3.89	-0.05
6_1	203.0	156.8	49.8	6.08	1.25	1.25
6_1	195.9	166.0	45.8	7.13	1.07	-1.24
6_2	202.7	166.1	60.4	1.43	-1.12	1.10
6_2	197.6	161.8	54.1	2.28	0.05	0.08
8_1	193.5	164.5	57.8	-1.05	-0.86	-0.67
8_1	187.0	165.1	58.6	-4.02	-1.22	-1.90
8_2	201.5	166.8	65.0	-1.60	-1.95	1.15
8_2	200.0	173.8	67.2	-2.27	-2.83	0.37

- Obs. 5 and 6 have most positive `LD1`: large `y`, small `w`.
- Obs. 4 has most positive `LD2`: small `w`.

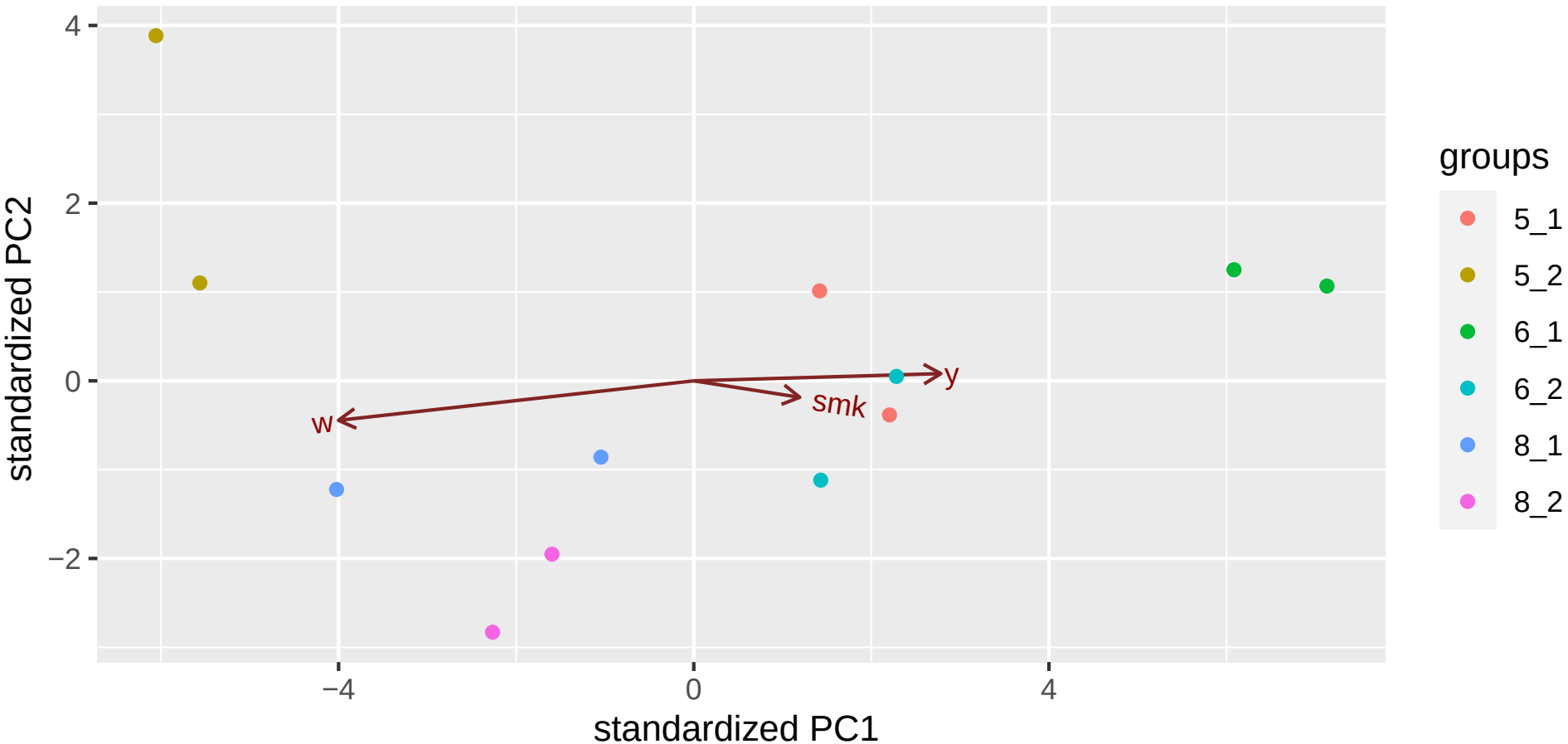
Plot LD1 vs. LD2, labelling by combo

```
g <- ggplot(mm, aes(x = LD1, y = LD2, colour = combo,
                    label = combo)) + geom_point() +
  geom_text_repel() + guides(colour = "none")
g
```



“Bi-plot” from ggbiplot

```
ggbiplot(peanuts.1, groups = factor(peanuts.combo$combo))
```



Installing ggbiplot

- ggbiplot not on CRAN, so usual install.packages will not work.
- Install package devtools first (once):

```
install.packages("devtools")
```

- Then install ggbiplot (once):

```
library(devtools)
install_github("vqv/ggbiplot")
```

Cross-validation

- So far, have predicted group membership from same data used to form the groups — dishonest!
- Better: *cross-validation*: form groups from all observations *except one*, then predict group membership for that left-out observation.
- No longer cheating!
- Illustrate with peanuts data again.

Misclassifications

- Fitting and prediction all in one go:

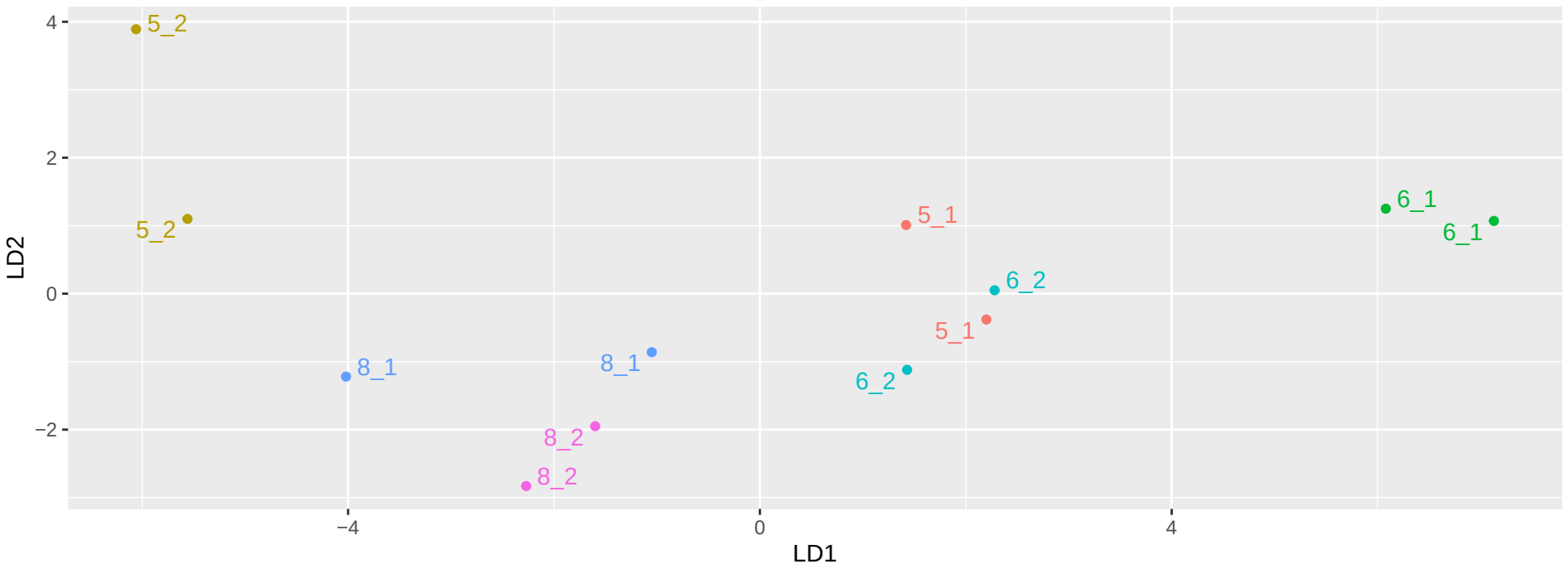
```
peanuts.cv <- lda(combo ~ y + smk + w,
  data = peanuts.combo, CV = T)
table(obs = peanuts.combo$combo,
      pred = peanuts.cv$class)
```

```
##       pred
## obs    5_1 5_2 6_1 6_2 8_1 8_2
##   5_1    0   0   0   2   0   0
##   5_2    0   1   0   0   1   0
##   6_1    0   0   2   0   0   0
##   6_2    1   0   0   1   0   0
##   8_1    0   1   0   0   0   1
##   8_2    0   0   0   0   0   2
```

- Some more misclassification this time.

Repeat of LD plot

```
gg
```



Posterior probabilities

```
pp <- round(peanuts.cv$posterior, 3)
data.frame(
  obs = peanuts.combo$combo,
  pred = peanuts.cv$class, pp
)
```

obs	pred	X5_1	X5_2	X6_1	X6_2	X8_1	X8_2
5_1	6_2	0.162	0.00	0.000	0.838	0.000	0.000
5_1	6_2	0.200	0.00	0.000	0.799	0.000	0.000
5_2	8_1	0.000	0.18	0.000	0.000	0.820	0.000
5_2	5_2	0.000	1.00	0.000	0.000	0.000	0.000
6_1	6_1	0.194	0.00	0.669	0.137	0.000	0.000
6_1	6_1	0.000	0.00	1.000	0.000	0.000	0.000
6_2	6_2	0.325	0.00	0.000	0.667	0.001	0.008
6_2	5_1	0.821	0.00	0.000	0.179	0.000	0.000
8_1	8_2	0.000	0.00	0.000	0.000	0.000	1.000
8_1	5_2	0.000	1.00	0.000	0.000	0.000	0.000
8_2	8_2	0.001	0.00	0.000	0.004	0.083	0.913
8_2	8_2	0.000	0.00	0.000	0.000	0.167	0.833

Why more misclassification?

- When predicting group membership for one observation, only uses the *other one* in that group.
- So if two in a pair are far apart, or if two groups overlap, great potential for misclassification.
- Groups 5_1 and 6_2 overlap.
- 5_2 closest to 8_1s looks more like an 8_1 than a 5_2 (other one far away).
- 8_1s relatively far apart and close to other things, so one appears to be a 5_2 and the other an 8_2.

Example 3: professions and leisure activities

- 15 individuals from three different professions (politicians, administrators and belly dancers) each participate in four different leisure activities: reading, dancing, TV watching and skiing. After each activity they rate it on a 0–10 scale.
- How can we best use the scores on the activities to predict a person's profession?
- Or, what combination(s) of scores best separate data into profession groups?

The data

```
my_url <- "http://ritsokiguess.site/datafiles/profile.txt"
active <- read_delim(my_url, " ")
active
```

job	reading	dance	tv	ski
bellydancer	7	10	6	5
bellydancer	8	9	5	7
bellydancer	5	10	5	8
bellydancer	6	10	6	8
bellydancer	7	8	7	9
politician	4	4	4	4
politician	6	4	5	3
politician	5	5	5	6
politician	6	6	6	7
politician	4	5	6	5
admin	3	1	1	2
admin	5	3	1	5
admin	4	2	2	5
admin	7	1	2	4
admin	6	3	3	3

Discriminant analysis

```
active.1 <- lda(job ~ reading + dance + tv + ski, data = active)
active.1
```

```
## Call:
## lda(job ~ reading + dance + tv + ski, data = active)
##
## Prior probabilities of groups:
##       admin bellydancer  politician
##   0.3333333   0.3333333   0.3333333
##
## Group means:
##             reading dance  tv ski
## admin           5.0   2.0 1.8 3.8
## bellydancer     6.6   9.4 5.8 7.4
## politician      5.0   4.8 5.2 5.0
##
## Coefficients of linear discriminants:
##                 LD1        LD2
## reading -0.01297465 -0.4748081
## dance   -0.95212396 -0.4614976
## tv      -0.47417264  1.2446327
## ski      0.04153684 -0.2033122
##
## Proportion of trace:
##    LD1    LD2
## 0.8917 0.1083
```

Comments

- Two discriminants, first fair bit more important than second.
- `LD1` depends (negatively) most on `dance`, a bit on `tv`.
- `LD2` depends mostly (negatively) on `tv`.

Misclassification

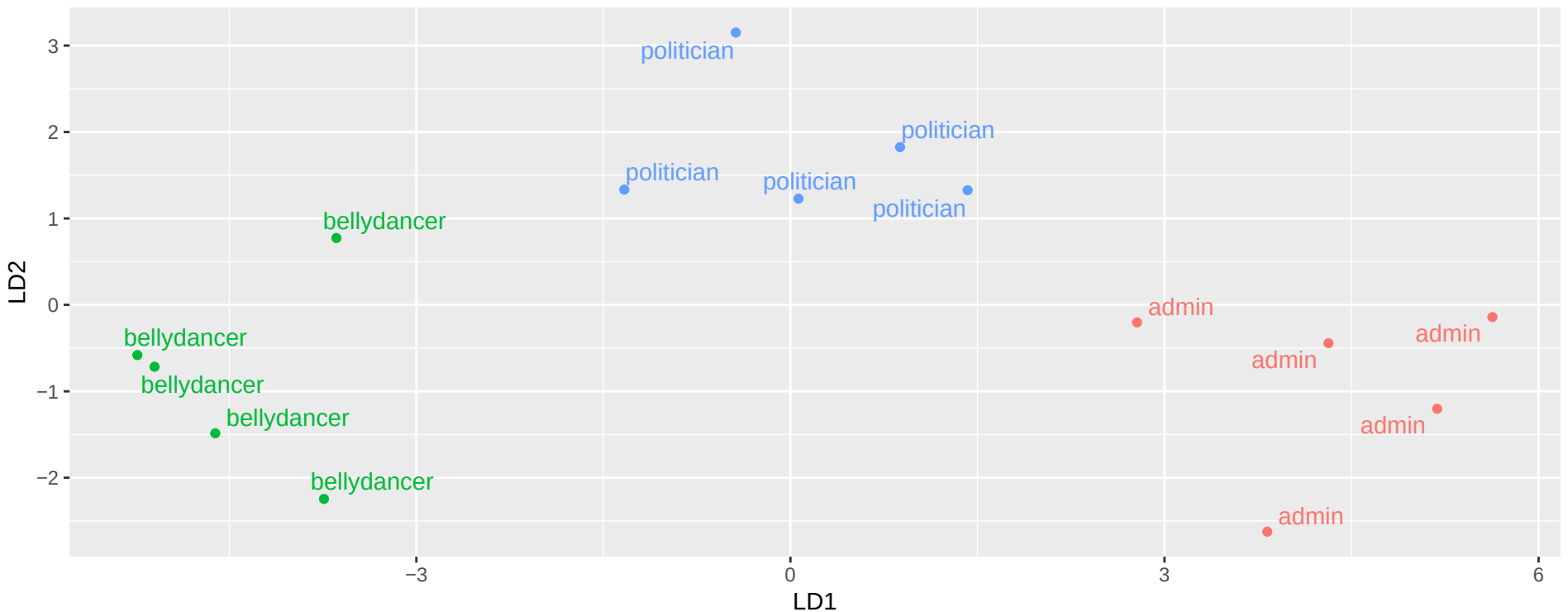
```
active.pred <- predict(active.1)
table(obs = active$job, pred = active.pred$class)
```

```
##                pred
## obs             admin bellydancer politician
##   admin             5           0          0
##   bellydancer       0           5          0
##   politician        0           0          5
```

Everyone correctly classified.

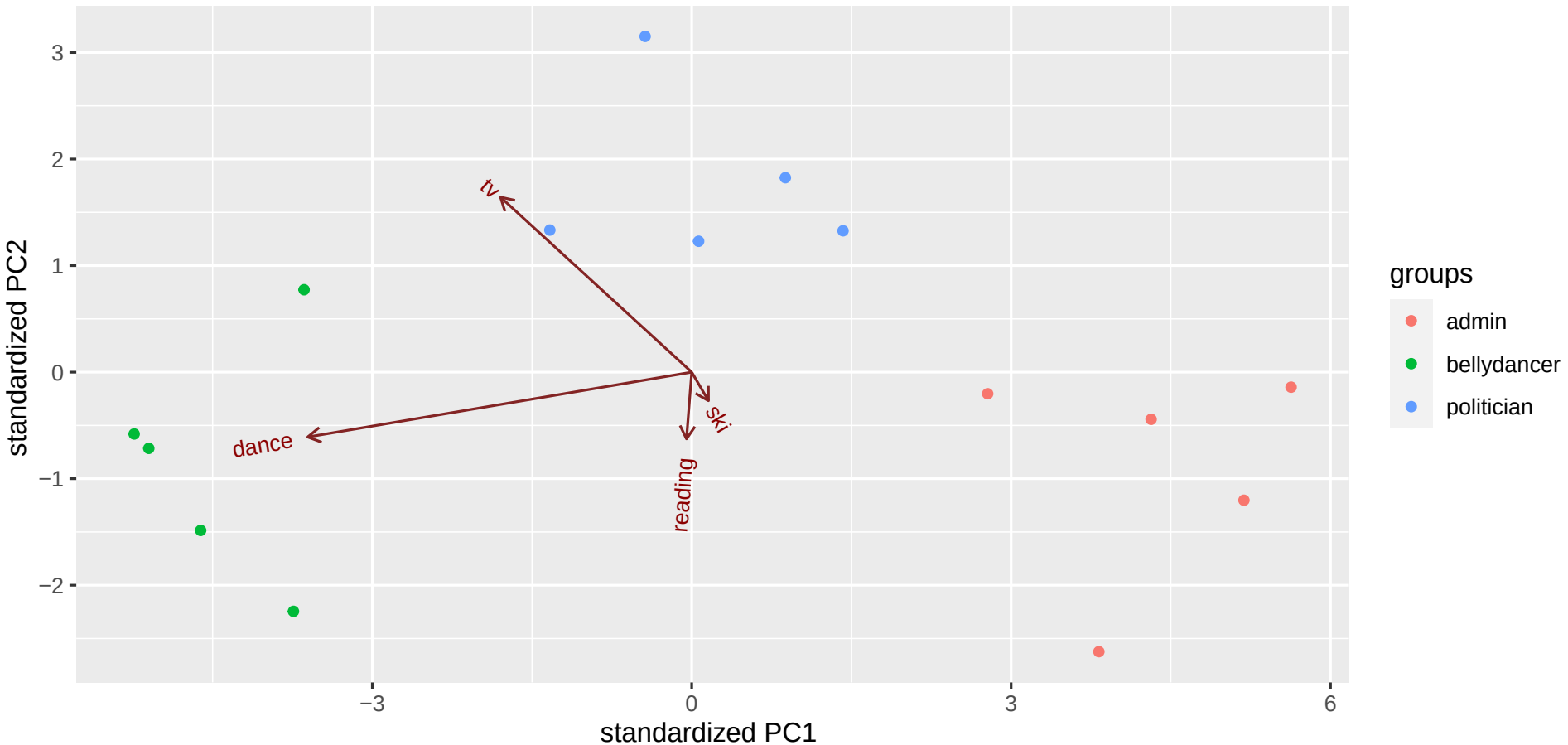
Plotting LDs

```
mm <- data.frame(job = active$job, active.pred$x, person = 1:15)
g <- ggplot(mm, aes(x = LD1, y = LD2, colour = job, label = job)) +
  geom_point() + geom_text_repel() + guides(colour = "none")
g
```



Biplot

```
ggbiplot(active.1, groups = active$job)
```



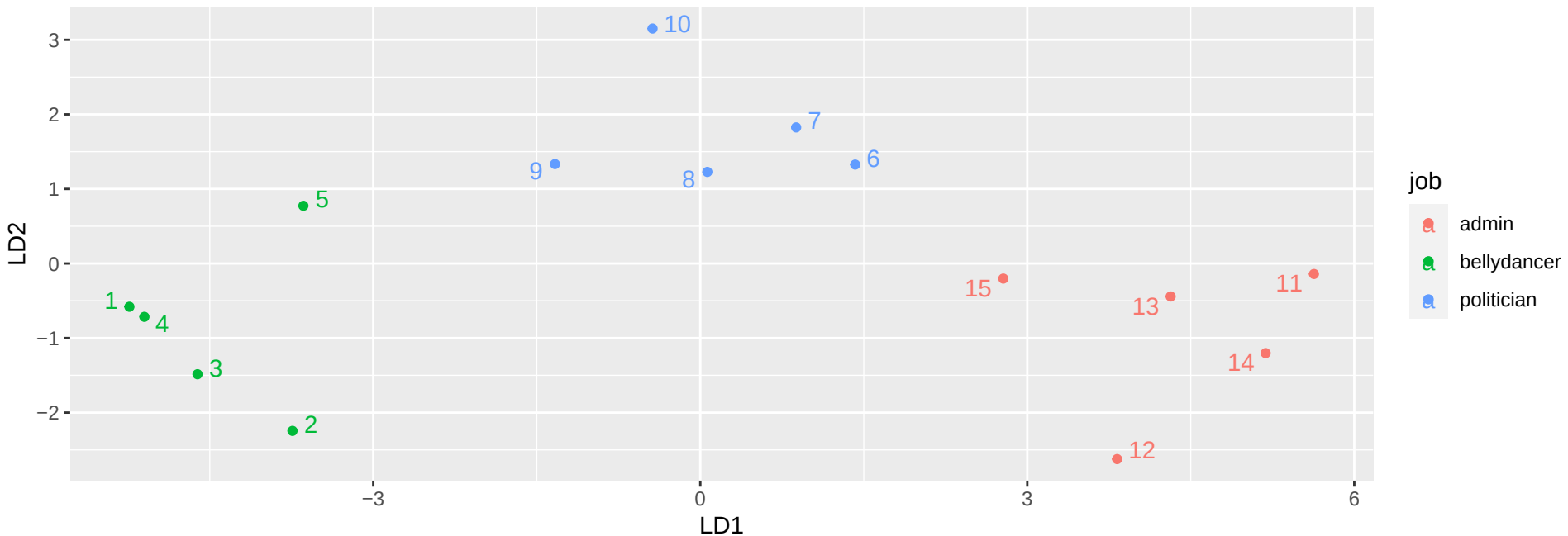
Comments on plot

- Groups well separated: bellydancers top left, administrators top right, politicians lower middle.
- Bellydancers most negative on `LD1`: like dancing most.
- Administrators most positive on `LD1`: like dancing least.
- Politicians most negative on `LD2`: like TV-watching most.

Plotting individual persons

Make `label` be identifier of person. Now need legend:

```
ggplot(mm, aes(x = LD1, y = LD2,  colour = job,
               label = person)) +
  geom_point() + geom_text_repel()
```



Posterior probabilities

```
pp <- round(active.pred$posterior, 3)
data.frame(obs = active$job, pred = active.pred$class, pp)
```

obs	pred	admin	bellydancer	politician
bellydancer	bellydancer	0.000	1.000	0.000
bellydancer	bellydancer	0.000	1.000	0.000
bellydancer	bellydancer	0.000	1.000	0.000
bellydancer	bellydancer	0.000	1.000	0.000
bellydancer	bellydancer	0.000	0.997	0.003
politician	politician	0.003	0.000	0.997
politician	politician	0.000	0.000	1.000
politician	politician	0.000	0.000	1.000
politician	politician	0.000	0.002	0.998
politician	politician	0.000	0.000	1.000
admin	admin	1.000	0.000	0.000
admin	admin	1.000	0.000	0.000
admin	admin	1.000	0.000	0.000
admin	admin	1.000	0.000	0.000
admin	admin	0.982	0.000	0.018

Not much doubt.

Cross-validating the jobs-activities data

Recall: no need for `predict`. Just pull out `class` and make a table:

```
active.cv <- lda(job ~ reading + dance + tv + ski,
  data = active, CV = T
)
table(obs = active$job, pred = active.cv$class)
```

```
##                pred
## obs             admin bellydancer politician
##   admin             5           0          0
##   bellydancer       0           4          1
##   politician        0           0          5
```

This time one of the bellydancers was classified as a politician.

and look at the posterior probabilities

picking out the ones where things are not certain:

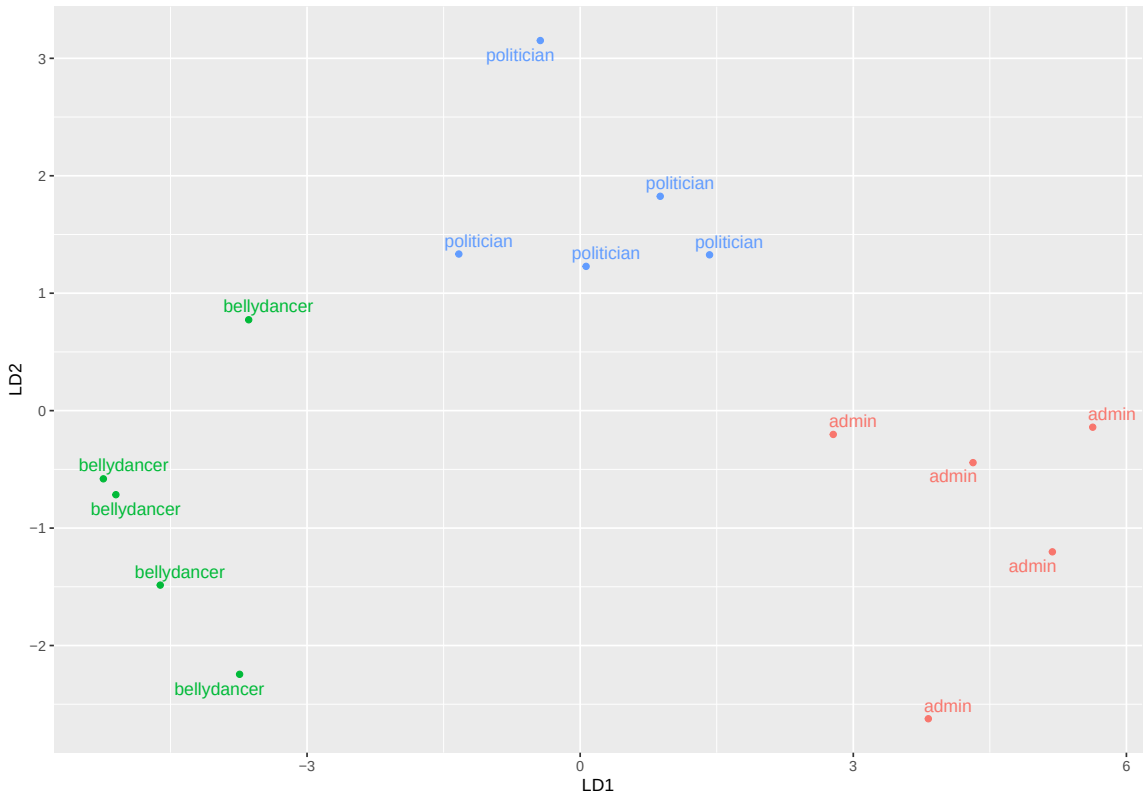
```
pp <- round(active.cv$posterior, 3)
data.frame(obs = active$job, pred = active.cv$class, pp) %>%
  mutate(max = pmax(admin, bellydancer, politician)) %>%
  filter(max < 0.9995)
```

	obs	pred	admin	bellydancer	politician	max
5	bellydancer	politician	0.000	0.001	0.999	0.999
6	politician	politician	0.006	0.000	0.994	0.994
7	politician	politician	0.001	0.000	0.999	0.999
9	politician	politician	0.000	0.009	0.991	0.991
15	admin	admin	0.819	0.000	0.181	0.819

- Bellydancer was "definitely" a politician!
- One of the administrators might have been a politician too.

Why did things get misclassified?

- Go back to plot of discriminant scores:
- one bellydancer much closer to the politicians,
- one administrator a bit closer to the politicians.



Example 4: remote-sensing data

- View 38 crops from air, measure 4 variables `x1`–`x4`.
- Go back and record what each crop was.
- Can we use the 4 variables to distinguish crops?

The data (some)

```
my_url <- "http://ritsokiguess.site/datafiles/remote-sensing.txt"
crops <- read_table(my_url)
crops
```

crop	x1	x2	x3	x4	cr
Corn	16	27	31	33	r
Corn	15	23	30	30	r
Corn	16	27	27	26	r
Corn	18	20	25	23	r
Corn	15	15	31	32	r
Corn	15	32	32	15	r
Corn	12	15	16	73	r
Soybeans	20	23	23	25	y
Soybeans	24	24	25	32	y
Soybeans	21	25	23	24	y
Soybeans	27	45	24	12	y
Soybeans	12	13	15	42	y
Soybeans	22	32	31	43	y
Cotton	31	32	33	34	t
Cotton	29	24	26	28	t

Discriminant analysis

```
crops.lda <- lda(crop ~ x1 + x2 + x3 + x4, data = crops)
crops.lda
```

```
## Call:
## lda(crop ~ x1 + x2 + x3 + x4, data = crops)
##
## Prior probabilities of groups:
##     Clover       Corn     Cotton   Soybeans Sugarbeets
##  0.3055556  0.1944444  0.1666667  0.1666667  0.1666667
##
## Group means:
##                  x1       x2       x3       x4
## Clover     46.36364 32.63636 34.18182 36.63636
## Corn       15.28571 22.71429 27.42857 33.14286
## Cotton     34.50000 32.66667 35.00000 39.16667
## Soybeans   21.00000 27.00000 23.50000 29.66667
## Sugarbeets 31.00000 32.16667 20.00000 40.50000
##
## Coefficients of linear discriminants:
##               LD1          LD2         LD3          LD4
## x1 -6.147360e-02  0.009215431 -0.02987075 -0.014680566
## x2 -2.548964e-02  0.042838972  0.04631489  0.054842132
## x3  1.642126e-02 -0.079471595  0.01971222  0.008938745
## x4  5.143616e-05 -0.013917423  0.05381787 -0.025717667
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4
## 0.7364 0.1985 0.0576 0.0075
```

Assessing

- 4 LDs (four variables, six groups).
- 1st one important, maybe 2nd as well.

```
round(crops.lda$scaling, 3)
```

```
##        LD1    LD2    LD3    LD4
## x1 -0.061  0.009 -0.030 -0.015
## x2 -0.025  0.043  0.046  0.055
## x3  0.016 -0.079  0.020  0.009
## x4  0.000 -0.014  0.054 -0.026
```

- Links original variables to LDs.
- LD1 mostly x1 (minus)
- LD2 x3 (minus), x2 (plus)

Predictions

- Thus:

```
crops.pred <- predict(crops.lda)
table(obs = crops$crop, pred = crops.pred$class)
```

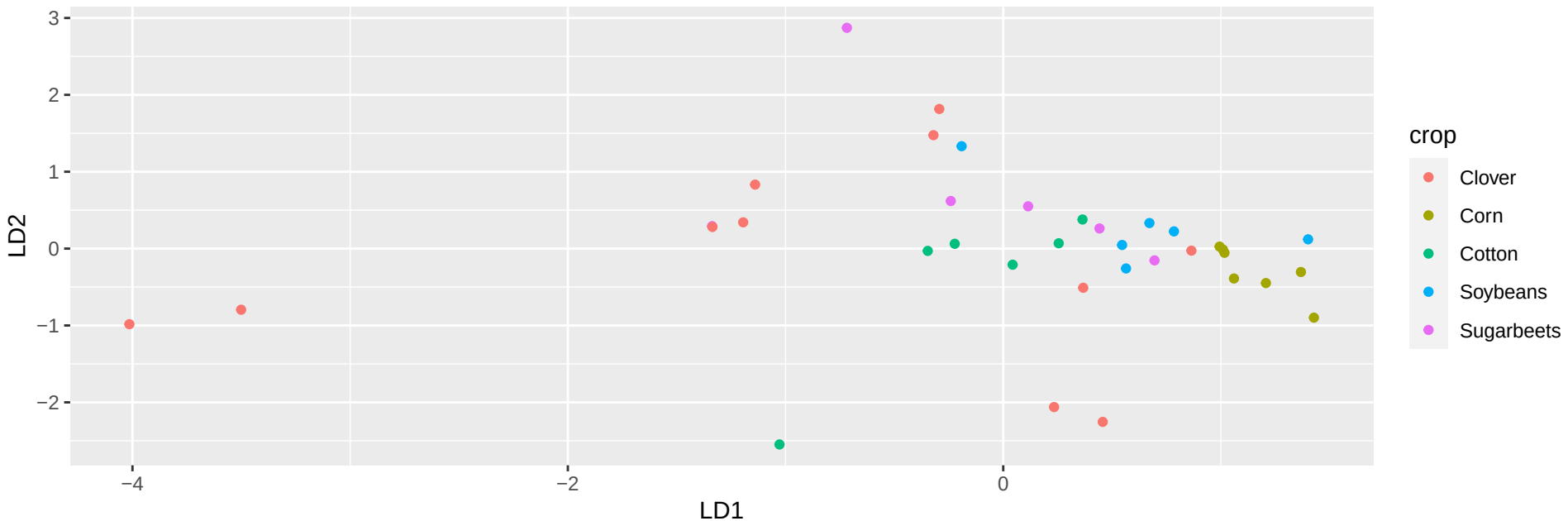
```
##             pred
## obs          Clover Corn Cotton Soybeans Sugarbeets
##   Clover          6    0      3        0           2
##   Corn            0    6      0        1           0
##   Cotton          3    0      1        2           0
##   Soybeans        0    1      1        3           1
##   Sugarbeets      1    1      0        2           2
```

- Not very good, eg. only 6 of 11 `Clover` classified correctly.
- Set up for plot:

```
mm <- data.frame(crop = crops$crop, crops.pred$x)
```

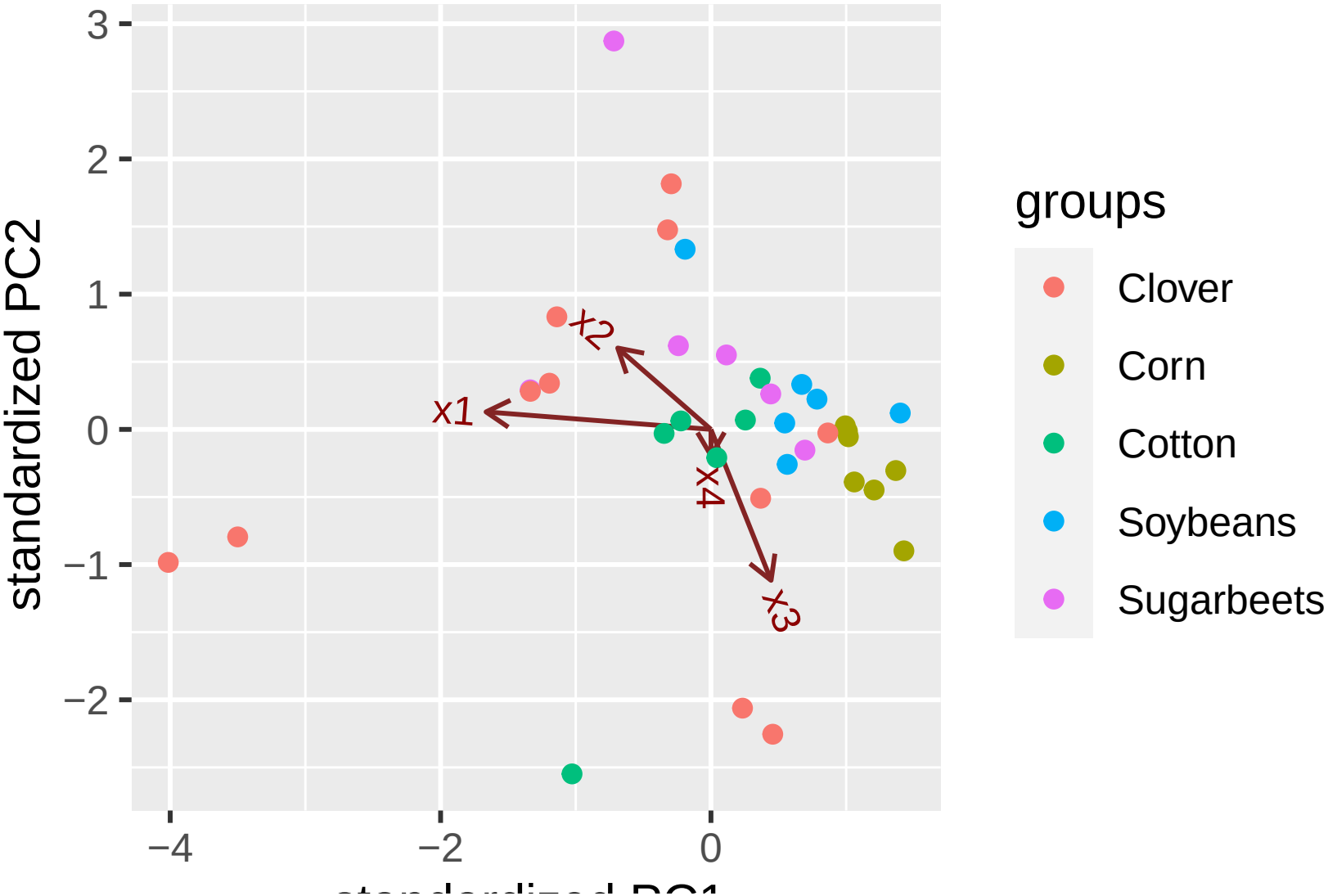
Plotting the LDs

```
ggplot(mm, aes(x = LD1, y = LD2, colour = crop)) +
  geom_point()
```



Biplot

```
ggbiplot(crops.lda, groups = crops$crop)
```



Comments

- Corn high on LD1 (right).
- Clover all over the place, but mostly low on LD1 (left).
- Sugarbeets tend to be high on LD2.
- Cotton tends to be low on LD2.
- Very mixed up.

Try removing Clover

- the dplyr way:

```
crops %>% filter(crop != "Clover") -> crops2
crops2.lda <- lda(crop ~ x1 + x2 + x3 + x4, data = crops2)
```

- LDs for crops2 will be different from before.
- Concentrate on plot and posterior probs.

```
crops2.pred <- predict(crops2.lda)
mm <- data.frame(crop = crops2$crop, crops2.pred$x)
```

lda output

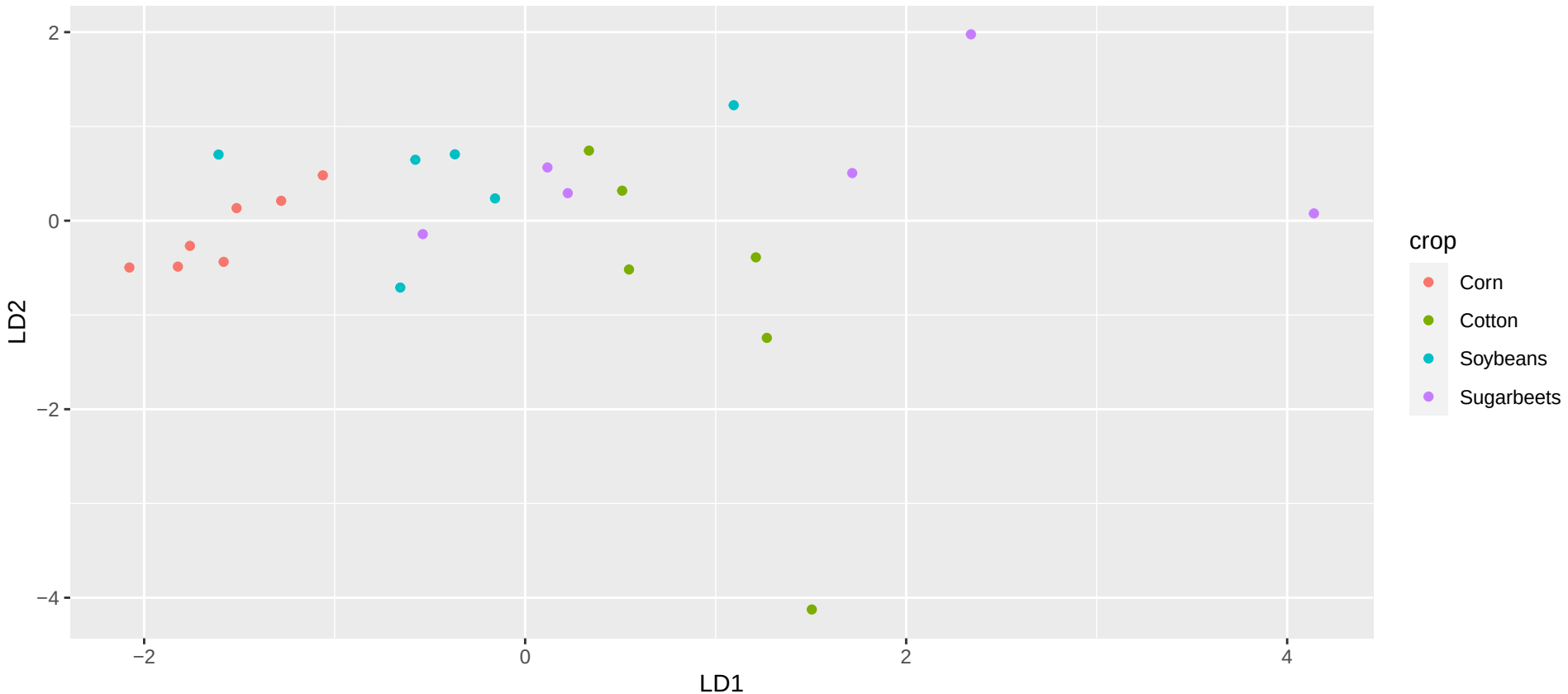
```
crops2.lda
```

```
## Call:
## lda(crop ~ x1 + x2 + x3 + x4, data = crops2)
##
## Prior probabilities of groups:
##       Corn     Cotton   Soybeans Sugarbeets
##       0.28       0.24       0.24       0.24
##
## Group means:
##                  x1       x2       x3       x4
## Corn       15.28571 22.71429 27.42857 33.14286
## Cotton     34.50000 32.66667 35.00000 39.16667
## Soybeans   21.00000 27.00000 23.50000 29.66667
## Sugarbeets 31.00000 32.16667 20.00000 40.50000
##
## Coefficients of linear discriminants:
##            LD1          LD2           LD3
## x1  0.14077479  0.007780184 -0.0312610362
## x2  0.03006972  0.007318386  0.0085401510
## x3 -0.06363974 -0.099520895 -0.0005309869
## x4 -0.00677414 -0.035612707  0.0577718649
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.8044 0.1832 0.0124
```

Plot

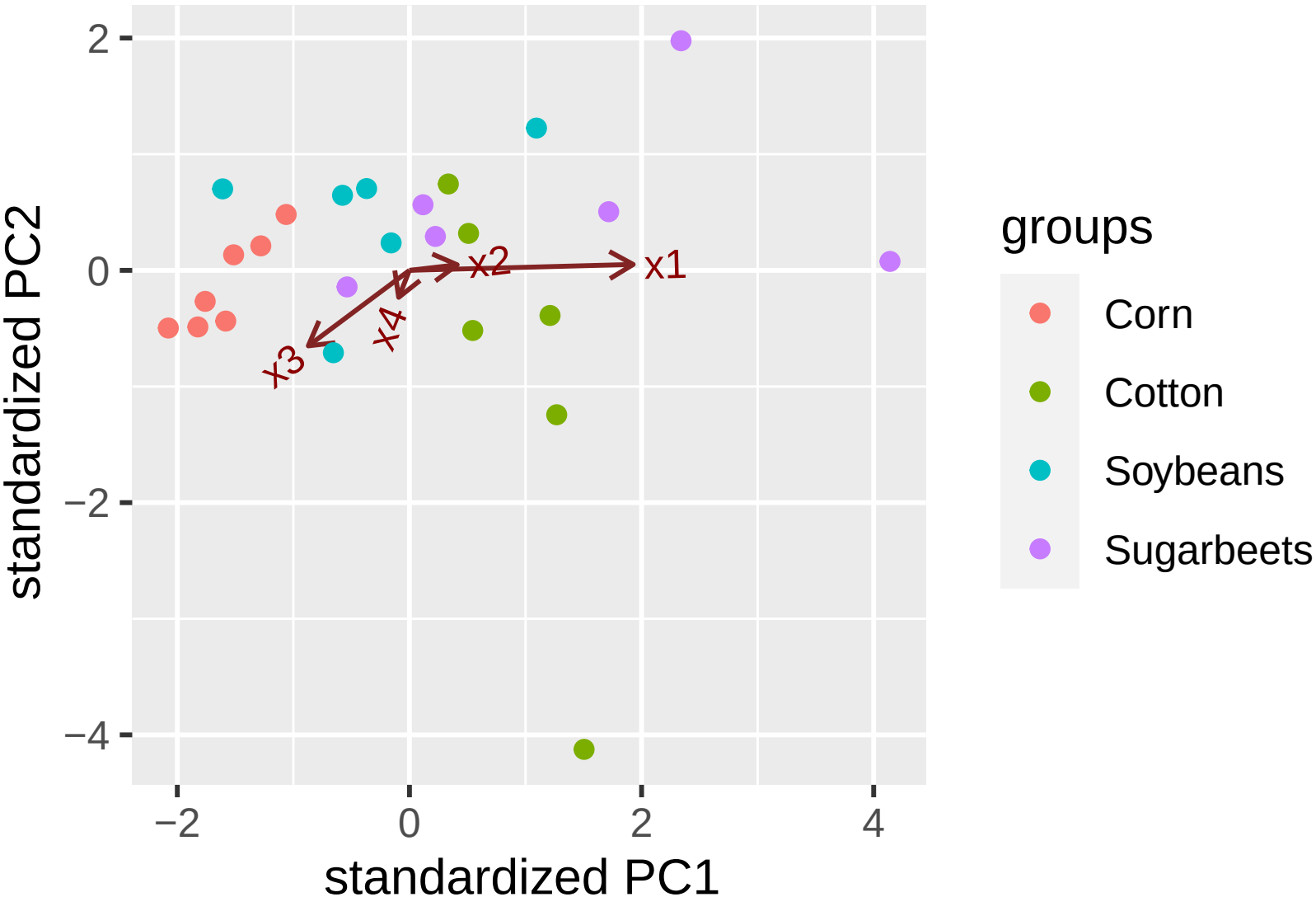
A bit more clustered:

```
ggplot(mm, aes(x = LD1, y = LD2, colour = crop)) +
  geom_point()
```



Biplot

```
ggbiplot(crops2.lda, groups = crops2$crop)
```



Quality of classification

```
table(obs = crops2$crop, pred = crops2.pred$class)
```

```
##                pred
## obs             Corn Cotton Soybeans Sugarbeets
##   Corn             6      0        1           0
##   Cotton           0      4        2           0
##   Soybeans         2      0        3           1
##   Sugarbeets       0      0        3           3
```

Better.

Posterior probs (some)

```
post <- round(crops2.pred$posterior, 3)
data.frame(obs = crops2$crop, pred = crops2.pred$class, post) %>%
  filter(obs != pred)
```

	obs	pred	Corn	Cotton	Soybeans	Sugarbeets
4	Corn	Soybeans	0.443	0.034	0.494	0.029
11	Soybeans	Sugarbeets	0.010	0.107	0.299	0.584
12	Soybeans	Corn	0.684	0.009	0.296	0.011
13	Soybeans	Corn	0.467	0.199	0.287	0.047
15	Cotton	Soybeans	0.056	0.241	0.379	0.324
17	Cotton	Soybeans	0.066	0.138	0.489	0.306
20	Sugarbeets	Soybeans	0.381	0.146	0.395	0.078
21	Sugarbeets	Soybeans	0.106	0.144	0.518	0.232
24	Sugarbeets	Soybeans	0.088	0.207	0.489	0.216

Comments

- These were the misclassified ones, but the posterior probability of being correct was not usually too low.
- The correctly-classified ones are not very clear-cut either.

MANOVA

Began discriminant analysis as a followup to MANOVA. Do our variables significantly separate the crops (excluding Clover)?

```
response <- with(crops2, cbind(x1, x2, x3, x4))
crops2.manova <- manova(response ~ crop, data = crops2)
summary(crops2.manova)
```

```
##           Df Pillai approx F num Df den Df  Pr(>F)
## crop       3 0.9113   2.1815     12     60 0.02416 *
## Residuals 21
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Box's M test

We should also run Box's M test to check for equal variance of each variable across crops:

```
summary(BoxM(response, crops2$crop))
```

```
##         Box's M Test
##
## Chi-Squared Value = 69.42634 , df = 30  and p-value: 5.79e-05
```

- The P-value for the M test is smaller even than our guideline of 0.001. So we should not take the MANOVA seriously.
- *Apparently* at least one of the crops differs (in means) from the others. So it is worth doing this analysis.
- We did this the wrong way around, though!

The right way around

- *First*, do a MANOVA to see whether any of the groups differ significantly on any of the variables.
- Check that the MANOVA is believable by using Box's M test.
- *If the MANOVA is significant*, do a discriminant analysis in the hopes of understanding how the groups are different.
- For remote-sensing data (without Clover):
  - LD1 a fair bit more important than LD2 (definitely ignore LD3).
  - LD1 depends mostly on `x1`, on which Cotton was high and Corn was low.
- Discriminant analysis in MANOVA plays the same kind of role that Tukey does in ANOVA.