Repeated Measures ANOVA

Repeated measures by profile analysis

- More than one response *measurement* for each subject. Might be
- measurements of the same thing at different times
- measurements of different but related things
- Generalization of matched pairs ("matched triples", etc.).
- Variation: each subject does several different treatments at different times (called *crossover design*).
- Expect measurements on same subject to be correlated, so assumptions of independence will fail.
- Called *repeated measures*. Different approaches, but *profile analysis* uses `Manova` (set up right way).
- Another approach uses *mixed models* (random effects).

Packages

```
library(car)
library(tidyverse)
```

Example: histamine in dogs

- 8 dogs take part in experiment.
- Dogs randomized to one of 2 different drugs.
- Response: log of blood concentration of histamine 0, 1, 3 and 5 minutes after taking drug. (Repeated measures.)
- Data in `dogs.txt`, column-aligned.

Read in data

```
# my_url <- "http://ritsokiguess.site/datafiles/dogs.txt"
my_url <- "~/teaching/d29/notes/dogs.txt"
dogs <- read_table(my_url)
```

```
##
## -- Column specification --------------------------------
## cols(
##   dog = col_character(),
##   drug = col_character(),
##   x = col_character(),
##   lh0 = col_double(),
##   lh1 = col_double(),
##   lh3 = col_double(),
##   lh5 = col_double()
## )
```

Setting things up

```
dogs
```

dog	drug	x	lh0	lh1	lh3	lh5
A	Morphine	N	-3.22	-1.61	-2.30	-2.53
B	Morphine	N	-3.91	-2.81	-3.91	-3.91
C	Morphine	N	-2.66	0.34	-0.73	-1.43
D	Morphine	N	-1.77	-0.56	-1.05	-1.43
E	Trimethaphan	N	-3.51	-0.48	-1.17	-1.51
F	Trimethaphan	N	-3.51	0.05	-0.31	-0.51
G	Trimethaphan	N	-2.66	-0.19	0.07	-0.22
H	Trimethaphan	N	-2.41	1.14	0.72	0.21

```
response <- with(dogs, cbind(lh0, lh1, lh3, lh5))
dogs.1 <- lm(response ~ drug, data = dogs)
```

The repeated measures MANOVA

Get list of response variable names; we call them `times`. Save in data frame.

```
times <- colnames(response)
times.df <- data.frame(times=factor(times))
dogs.2 <- Manova(dogs.1,
  idata = times.df,
  idesign = ~times
)
```

The output (there is a lot)

```
summary(dogs.2)
```

```
##
## Type II Repeated Measures MANOVA Tests:
##
## ----------------------------------------
##
## Term: (Intercept)
##
##  Response transformation matrix:
##     (Intercept)
## lh0           1
## lh1           1
## lh3           1
## lh5           1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)     285.366
##
## Multivariate Tests: (Intercept)
##                  Df test stat approx F num Df den Df    Pr(>F)
## Pillai            1  0.763467 19.36642      1      6 0.0045648 **
## Wilks             1  0.236533 19.36642      1      6 0.0045648 **
## Hotelling-Lawley  1  3.227738 19.36642      1      6 0.0045648 **
## Roy               1  3.227738 19.36642      1      6 0.0045648 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ----------------------------------------
##
## Term: drug
```

What there is here xxx

- three sets of tests, for
  - times
  - drug
  - their interaction
- two *types* of test for each of these:
  - multivariate
  - univariate
- multivariate is the same as MANOVA
- univariate is more powerful *if* it applies

Wide and long format

- Interaction significant. Pattern of response over time different for the two drugs.
- Want to investigate interaction.

The wrong shapeF

- But data frame has several observations per line ("wide format"):

```
dogs %>% slice(1:6)
```

dog	drug	x	lh0	lh1	lh3	lh5
A	Morphine	N	-3.22	-1.61	-2.30	-2.53
B	Morphine	N	-3.91	-2.81	-3.91	-3.91
C	Morphine	N	-2.66	0.34	-0.73	-1.43
D	Morphine	N	-1.77	-0.56	-1.05	-1.43
E	Trimethaphan	N	-3.51	-0.48	-1.17	-1.51
F	Trimethaphan	N	-3.51	0.05	-0.31	-0.51

- Plotting works with data in "long format": one response per line.
- The responses are log-histamine at different times, labelled `lh`-something. Call them all `lh` and put them in one column, with the time they belong to labelled.

Running pivot_longer, try 1

```
dogs %>% pivot_longer(starts_with("lh"),
                      names_to = "time", values_to = "lh")
```

dog	drug	x	time	lh
A	Morphine	N	lh0	-3.22
A	Morphine	N	lh1	-1.61
A	Morphine	N	lh3	-2.30
A	Morphine	N	lh5	-2.53
B	Morphine	N	lh0	-3.91
B	Morphine	N	lh1	-2.81
B	Morphine	N	lh3	-3.91
B	Morphine	N	lh5	-3.91
C	Morphine	N	lh0	-2.66
C	Morphine	N	lh1	0.34
C	Morphine	N	lh3	-0.73
C	Morphine	N	lh5	-1.43
D	Morphine	N	lh0	-1.77
D	Morphine	N	lh1	-0.56
D	Morphine	N	lh3	-1.05
D	Morphine	N	lh5	-1.43

Getting the times

Not quite right: for the times, we want just the numbers, not the letters `lh` every time. Want new variable containing just number in `time`: `parse_number`.

```
dogs %>%
  pivot_longer(starts_with("lh"),
               names_to = "timex", values_to = "lh") %>%
  mutate(time = parse_number(timex))
```

dog	drug	x	timex	lh	time
A	Morphine	N	lh0	-3.22	0
A	Morphine	N	lh1	-1.61	1
A	Morphine	N	lh3	-2.30	3
A	Morphine	N	lh5	-2.53	5
B	Morphine	N	lh0	-3.91	0
B	Morphine	N	lh1	-2.81	1
B	Morphine	N	lh3	-3.91	3
B	Morphine	N	lh5	-3.91	5
C	Morphine	N	lh0	-2.66	0

What I did differently

- I realized that `pivot_longer` was going to produce something like `lh1`, which I needed to do something further with, so this time I gave it a temporary name `timex`.
- This enabled me to use the name `time` for the actual numeric time.
- This works now, so next save into a new data frame `dogs.long`.

Saving the pipelined results

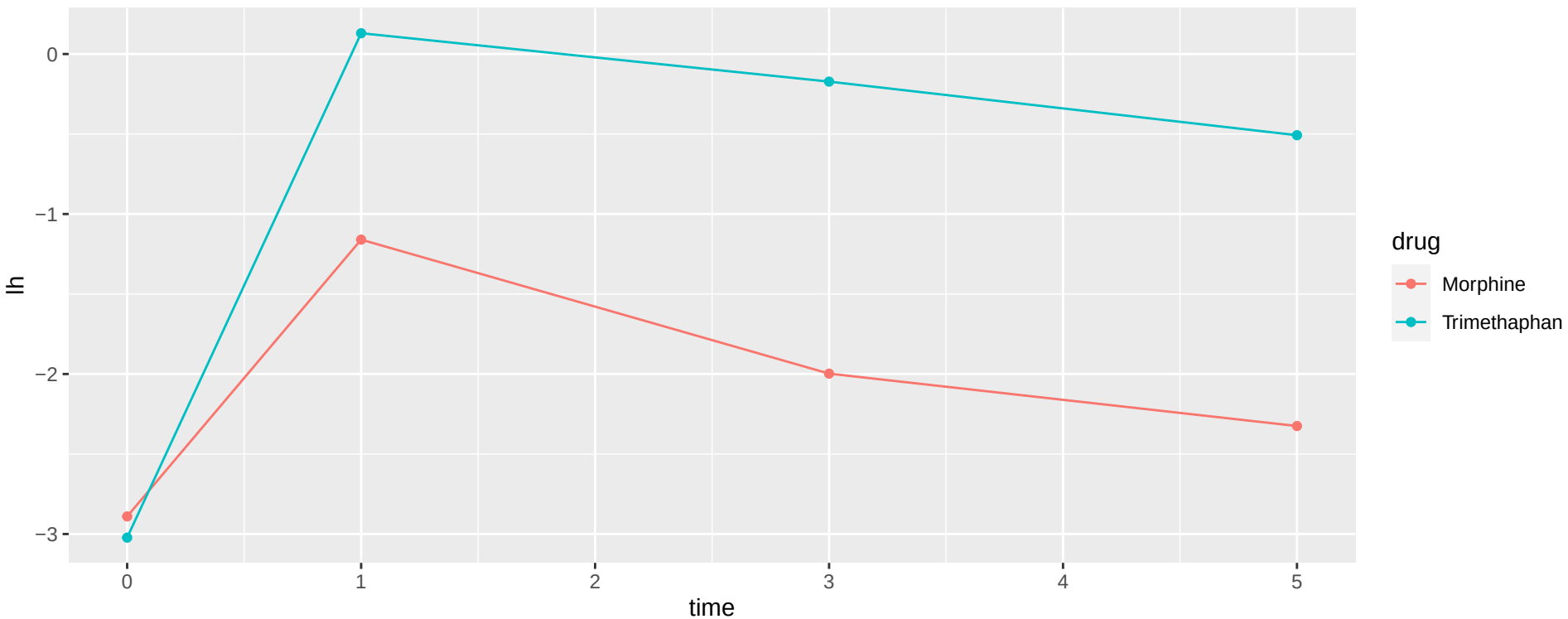
```
dogs %>%
  pivot_longer(starts_with("lh"),
               names_to = "timex", values_to = "lh") %>%
  mutate(time = parse_number(timex)) -> dogs.long
```

This says:

- Take data frame dogs, and then:
- Combine the columns `lh0` through `lh5` into one column called `lh`, with the column that each `lh` value originally came from labelled by `timex`, and then:
- Pull out numeric values in `timex`, saving in `time` and then:
- save the result in a data frame `dogs.long`.

Interaction plot

```
ggplot(dogs.long, aes(x = time, y = lh,
                      colour = drug, group = drug)) +
  stat_summary(fun = mean, geom = "point") +
  stat_summary(fun = mean, geom = "line")
```



Comments

- Plot mean `lh` value at each time, joining points on same drug by lines.
- drugs same at time 0
- after that, Trimethaphan higher than Morphine.
- Effect of drug not consistent over time: significant interaction.

Take out time zero

- Lines on interaction plot would then be parallel, and so interaction should no longer be significant.
- Go back to original "wide" dogs data frame.

```
response <- with(dogs, cbind(lh1, lh3, lh5)) # excl time 0
dogs.1 <- lm(response ~ drug, data = dogs)
times <- colnames(response)
times.df <- data.frame(times=factor(times))
dogs.2 <- Manova(dogs.1,
  idata = times.df,
  idesign = ~times
)
```

Results and comments

```
dogs.2
```

```
##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##             Df test stat approx F num Df den Df   Pr(>F)
## (Intercept)  1   0.54582   7.2106      1      6 0.036281 *
## drug         1   0.44551   4.8207      1      6 0.070527 .
## times        1   0.85429  14.6569      2      5 0.008105 **
## drug:times   1   0.43553   1.9289      2      5 0.239390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Correct: interaction no longer significant.
- Significant effect of time.
- Drug effect not quite significant (some variety among dogs within drug).

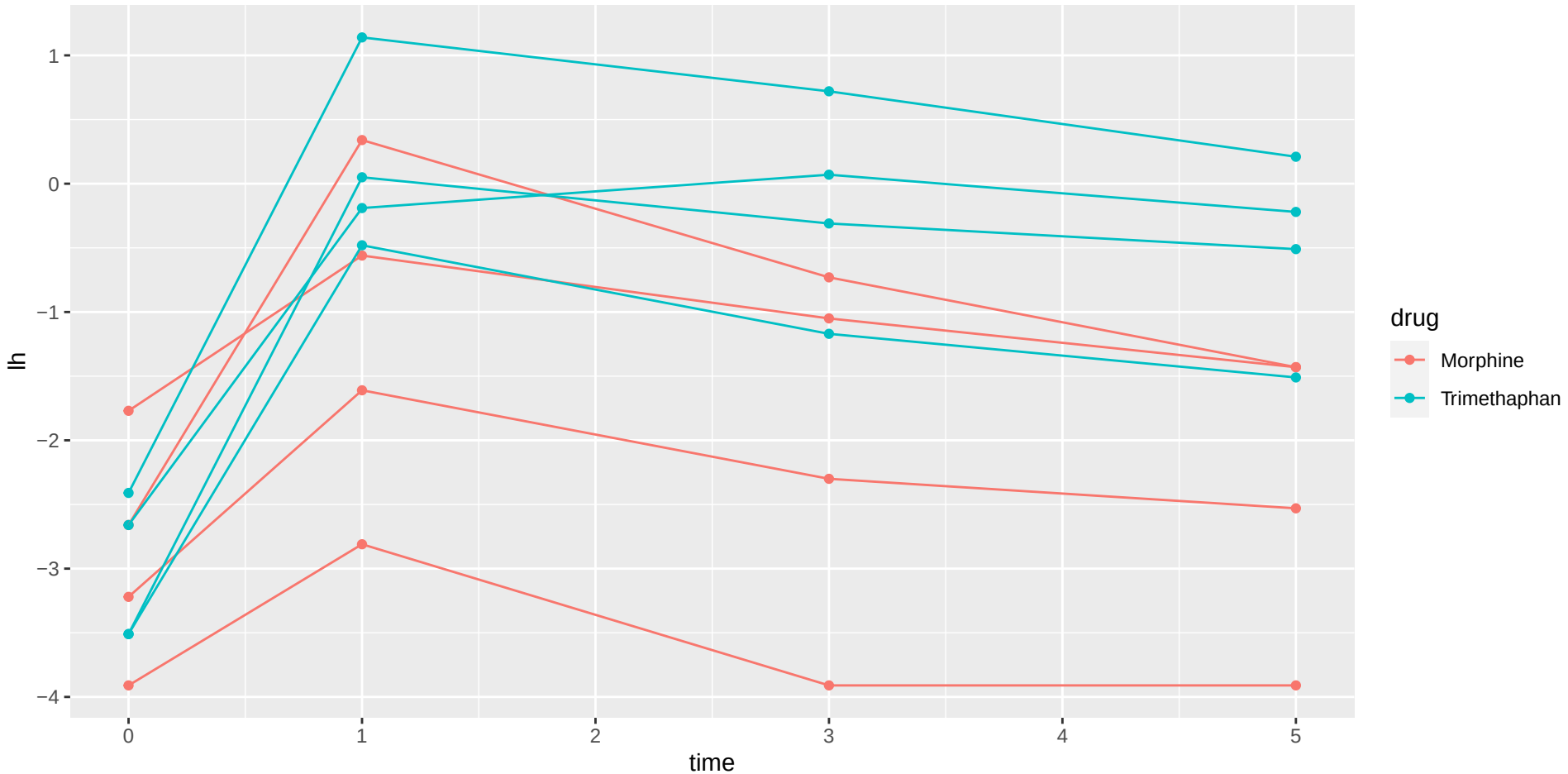
Is the non-significant drug effect reasonable?

- Plot *actual data*: `lh` against `days`, labelling observations by drug: "spaghetti plot".
- Uses long data frame (confusing, yes I know):
- Plot (time,lh) points coloured by drug
- and connecting measurements for each *dog* by lines.
- This time, we want `group=dog` (want the measurements for each *dog* joined by lines), but `colour=drug`:

```
g <- ggplot(dogs.long, aes(
  x = time, y = lh,
  colour = drug, group = dog
)) +
  geom_point() + geom_line()
```

The spaghetti plot

```
g
```



Comments

- For each dog over time, there is a strong increase and gradual decrease in log-histamine. This explains the significant time effect.
- The pattern is more or less the same for each dog, regardless of drug. This explains the non-significant interaction.
- Most of the trimethaphan dogs (blue) have higher log-histamine throughout (time 1 and after), and some of the morphine dogs have lower.
- *But* two of the morphine dogs have log-histamine profiles like the trimethaphan dogs. This ambiguity is probably why the `drug` effect is not quite significant.

Assumptions

sphericity (or earlier)

Mixed models

- Another way to fit repeated measures
- Subjects (on whom repeated measures taken) are *random sample of all possible subjects* (random effects)
- Times and treatments are *the only ones we care about* (fixed effects)
- Use package `lme4` function `lmer` (like `lm` in some ways)
- Uses long-format "tidy" data

Fitting the model

```
library(lme4)
# dogs.long
dogs.2=lmer(lh~drug*time+(1|dog),data=dogs.long)
```

- note specification of random effect: each dog has "random intercept" that moves log-histamine up or down for that dog over all times

What can we drop?

- using `drop1`:

```
drop1(dogs.2,test="Chisq")
```

	npar	AIC	LRT	Pr(Chi)
	NA	113.2597	NA	NA
drug:time	1	114.2130	2.953368	0.0856988

- Interaction again not significant, but P-value smaller than before

Re-fit without interaction

```
dogs.3=update(dogs.2,.~.-drug:time)
drop1(dogs.3,test="Chisq")
```

	npar	AIC	LRT	Pr(Chi)
	NA	114.2130	NA	NA
drug	1	115.5691	3.356021	0.0669597
time	1	114.9631	2.750099	0.0972484

- This time neither drug nor (surprisingly) time is significant.
- MANOVA and `lmer` methods won't agree, but both valid ways to approach problem.

The exercise data

- 30 people took part in an exercise study.
- Each subject was randomly assigned to one of two diets ("low fat" or "non-low fat' ') and to one of three exercise programs ("at rest", "walking", "running").
- There are $2 \times 3 = 6$ experimental treatments, and thus each one is replicated $30/6 = 5$ times.
- Nothing unusual so far.
- However, each subject had their pulse rate measured at three different times (1, 15 and 30 minutes after starting their exercise), so have repeated measures.

Reading the data xxx

Separated by *tabs*:

```
# url <- "http://ritsokiguess.site/datafiles/exercise2.txt"
url <- "~/teaching/d29/notes/exercise.txt"
exercise.long <- read_tsv(url)
```

```
## Rows: 90 Columns: 5
## -- Column specification --------------------------------
## Delimiter: "\t"
## chr (3): diet, exertype, time
## dbl (2): id, pulse
##
## i Use `spec()` to retrieve the full column specification fo
## i Specify the column types or set `show_col_types = FALSE`
```

The data

```
exercise.long %>% slice(1:8)
```

id	diet	exertype	pulse	time
1	nonlowfat	atrest	85	min01
1	nonlowfat	atrest	85	min15
1	nonlowfat	atrest	88	min30
2	nonlowfat	atrest	90	min01
2	nonlowfat	atrest	92	min15
2	nonlowfat	atrest	93	min30
3	nonlowfat	atrest	97	min01
3	nonlowfat	atrest	97	min15

- This is "long format", which is usually what we want.
- But for repeated measures analysis, we want *wide* format!
- `pivot_wider`.

Making wide format

- `pivot_wider` needs: a column that is going to be split, and the column to make the values out of:

```
exercise.long %>% pivot_wider(names_from=time,
                              values_from=pulse) -> exercise.wide
exercise.wide %>% sample_n(5)
```

id	diet	exertype	min01	min15	min30
1	nonlowfat	atrest	85	85	88
3	nonlowfat	atrest	97	97	94
5	nonlowfat	atrest	91	92	91
14	nonlowfat	walking	95	96	100
15	nonlowfat	walking	89	96	95

- Normally `pivot_longer` `min01, min15, min30` into one column called `pulse` labelled by the number of minutes. But `Manova` needs it the other way.

Setting up the repeated-measures analysis

- Make a response variable consisting of `min01, min15, min30`:

```
response <- with(exercise.wide, cbind(min01, min15, min30))
```

- Predict that from `diet` and `exertype` and interaction using `lm`:

```
exercise.1 <- lm(response ~ diet * exertype,
  data = exercise.wide
)
```

- Run this through `Manova`:

```
times <- colnames(response)
times.df <- data.frame(times=factor(times))
exercise.2 <- Manova(exercise.1,
                     idata = times.df,
                     idesign = ~times)
```

Results

```
exercise.2
```

```
##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##                      Df test stat approx F num Df den Df    Pr(>F)
## (Intercept)           1   0.99767  10296.7      1     24 < 2.2e-16 ***
## diet                  1   0.37701     14.5      1     24 0.0008483 ***
## exertype              2   0.79972     47.9      2     24 4.166e-09 ***
## diet:exertype         2   0.28120      4.7      2     24 0.0190230 *
## times                 1   0.78182     41.2      2     23 2.491e-08 ***
## diet:times            1   0.25153      3.9      2     23 0.0357258 *
## exertype:times        2   0.83557      8.6      4     48 2.538e-05 ***
## diet:exertype:times   2   0.51750      4.2      4     48 0.0054586 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Three-way interaction significant, so cannot remove anything.
- Pulse rate depends on diet and exercise type *combination*, and *that* is different for each time.

Making some graphs

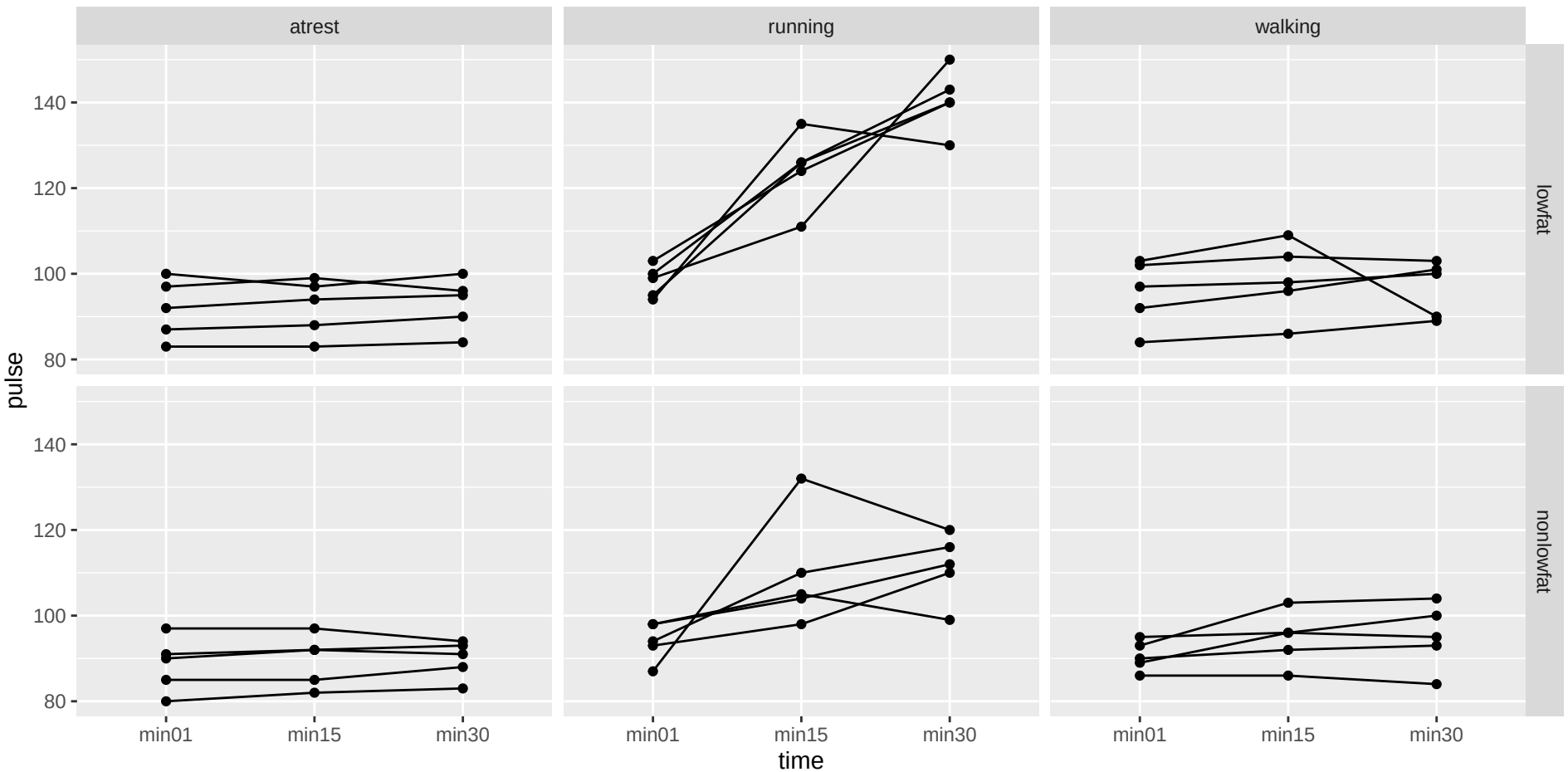
- Three-way interactions are difficult to understand. To make an attempt, look at some graphs.
- Plot time trace of pulse rates for each individual, joined by lines, and make *separate* plots for each `diet-exertype` combo.
- `ggplot` again. Using *long* data frame:

```
g <- ggplot(exercise.long, aes(
  x = time, y = pulse,
  group = id
)) + geom_point() + geom_line() +
  facet_grid(diet ~ exertype)
```

- `facet_grid(diet~exertype)`: do a separate plot for each combination of diet and exercise type, with diets going down the page and exercise types going across. (Graphs are usually landscape, so have the factor `exertype` with more levels going across.)

The graph(s)

```
g
```


atrest
running
walking
lowfat
nonlowfat
pulse
140
120
100
80
min01
min15
min30
time

Comments on graphs

- For subjects who were at rest, no change in pulse rate over time, for both diet groups.
- For walking subjects, not much change in pulse rates over time. Maybe a small increase on average between 1 and 15 minutes.
- For both running groups, an overall increase in pulse rate over time, but the increase is stronger for the `lowfat` group.
- No consistent effect of diet over all exercise groups.
- No consistent effect of exercise type over both diet groups.
- No consistent effect of time over all diet-exercise type combos.

"Simple effects" of diet for the subjects who ran

- Looks as if there is only any substantial time effect for the runners. For them, does diet have an effect?
- Pull out only the runners from the wide data:

```
exercise.wide %>%
  filter(exertype == "running") -> runners.wide
```

- Create response variable and do MANOVA. Some of this looks like before, but I have different data now:

```
response <- with(runners.wide, cbind(min01, min15, min30))
runners.1 <- lm(response ~ diet, data = runners.wide)
times <- colnames(response)
times.df <- data.frame(times=factor(times))
runners.2 <- Manova(runners.1,
  idata = times.df,
  idesign = ~times
)
```

Results

```
runners.2
```

```
##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##             Df test stat approx F num Df den Df    Pr(>F)
## (Intercept)  1   0.99912   9045.3      1      8 1.668e-13 ***
## diet         1   0.84986     45.3      1      8 0.0001482 ***
## times        1   0.92493     43.1      2      7 0.0001159 ***
## diet:times   1   0.68950      7.8      2      7 0.0166807 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

text under

- The `diet` by `time` interaction is still significant (at $\alpha = 0.05$): the effect of time on pulse rates is different for the two diets.
- At $\alpha = 0.01$, the interaction is not significant, and then we have only two (very) significant main effects of `diet` and `time`.

How is the effect of diet different over time?

- Table of means. Only I need long data for this, so make it (in a pipeline):

```
runners.wide %>%
  pivot_longer(starts_with("min"),
               names_to = "time", values_to = "pulse") %>%
  group_by(time, diet) %>%
  summarize(
    mean = mean(pulse),
    sd = sd(pulse)
  ) -> summ
```

```
## `summarise()` has grouped output by 'time'. You can overric
## the `.groups` argument.
```

- Result of `summarize` is data frame, so can save it (and do more with it if needed).

Understanding diet-time interaction

- The summary:

```
summ
```

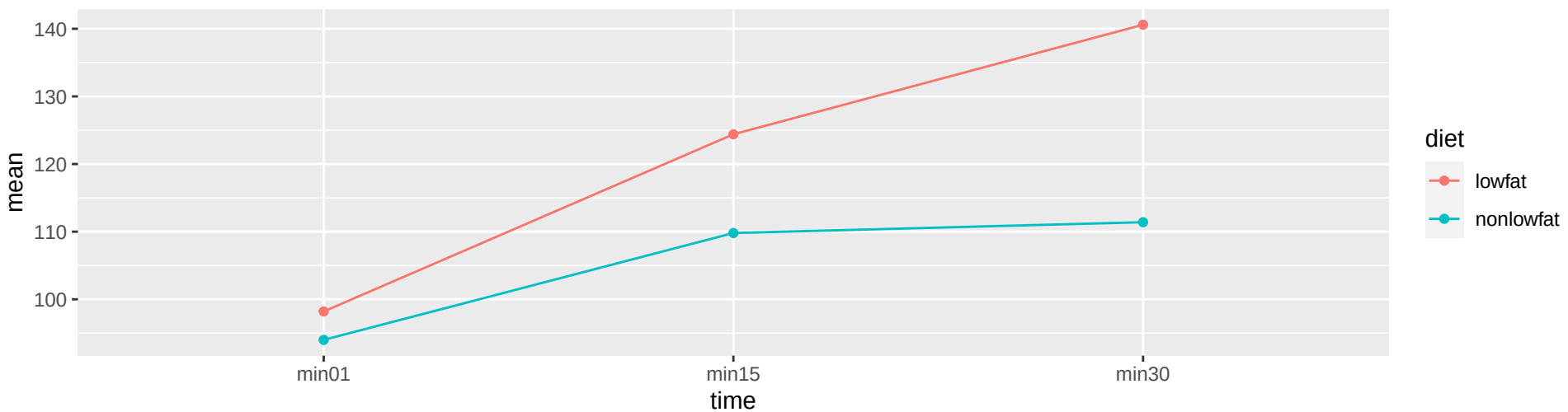
time	diet	mean	sd
min01	lowfat	98.2	3.701351
min01	nonlowfat	94.0	4.527693
min15	lowfat	124.4	8.619745
min15	nonlowfat	109.8	13.122500
min30	lowfat	140.6	7.197222
min30	nonlowfat	111.4	7.924645

- Pulse rates at any given time higher for `lowfat` (diet effect),
- Pulse rates increase over time of exercise (time effect),
- but the *amount by which pulse rate higher* for a diet depends on time: `diet` by `time` interaction.

Interaction plot

- We went to trouble of finding means by group, so making interaction plot is now mainly easy:

```
ggplot(summ, aes(x = time, y = mean, colour = diet,
                 group = diet)) + geom_point() + geom_line()
```



Comment on interaction plot

- The lines are not parallel, so there is interaction between diet and time for the runners.
- The effect of time on pulse rate is different for the two diets, even though all the subjects here were running.