

Repeated measures analysis

Repeated measures

- More than one response *measurement* for each subject, same thing at different times
- Generalization of matched pairs (“matched triples”, etc.).
- Expect measurements on same subject to be correlated, so assumptions of independence will fail.
- *Repeated measures*. *Profile analysis* uses Manova (set up).
- Another approach uses *mixed models* (random effects).
- Variation: each subject does all treatments at different times (called *crossover design*).

Packages

```
library(car)
library(tidyverse)
library(lme4) # for mixed models later
```

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"
dogs <- read_table(my_url)
dogs
```

```
# A tibble: 8 x 7
  dog    drug      x     lh0     lh1     lh3     lh5
  <chr> <chr>   <chr> <dbl> <dbl> <dbl> <dbl>
1 A     Morphine N     -3.22 -1.61 -2.3  -2.53
2 B     Morphine N     -3.91 -2.81 -3.91 -3.91
3 C     Morphine N     -2.66  0.34 -0.73 -1.43
4 D     Morphine N     -1.77 -0.56 -1.05 -1.43
5 E     Trimethaphan N     -3.51 -0.48 -1.17 -1.51
6 F     Trimethaphan N     -3.51  0.05 -0.31 -0.51
7 G     Trimethaphan N     -2.66 -0.19  0.07 -0.22
8 H     Trimethaphan N     -2.41  1.14  0.72  0.21
```

Setting things up

```
response <- with(dogs, cbind(lh0, lh1, lh3, lh5))  
response
```

	lh0	lh1	lh3	lh5
[1,]	-3.22	-1.61	-2.30	-2.53
[2,]	-3.91	-2.81	-3.91	-3.91
[3,]	-2.66	0.34	-0.73	-1.43
[4,]	-1.77	-0.56	-1.05	-1.43
[5,]	-3.51	-0.48	-1.17	-1.51
[6,]	-3.51	0.05	-0.31	-0.51
[7,]	-2.66	-0.19	0.07	-0.22
[8,]	-2.41	1.14	0.72	0.21

Another way to make response

```
dogs %>% select(starts_with("lh")) %>%
  as.matrix() -> response
response
```

	lh0	lh1	lh3	lh5
[1,]	-3.22	-1.61	-2.30	-2.53
[2,]	-3.91	-2.81	-3.91	-3.91
[3,]	-2.66	0.34	-0.73	-1.43
[4,]	-1.77	-0.56	-1.05	-1.43
[5,]	-3.51	-0.48	-1.17	-1.51
[6,]	-3.51	0.05	-0.31	-0.51
[7,]	-2.66	-0.19	0.07	-0.22
[8,]	-2.41	1.14	0.72	0.21

The repeated measures MANOVA

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

```
times <- colnames(response)  
times
```

```
[1] "lh0" "lh1" "lh3" "lh5"
```

```
times.df <- data.frame(times=factor(times))  
times.df
```

```
  times  
1   lh0  
2   lh1  
3   lh3  
4   lh5
```

Fitting the model

```
dogs.1 <- lm(response ~ drug, data = dogs)
dogs.2 <- Manova(dogs.1,
  idata = times.df,
  idesign = ~times
)
```

The output (there is a lot)

- normally you just run

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

and pull out what you need to answer the question.

- But you can grab just individual pieces as shown below:

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

```
[1] "type"                 "repeated"             "multivariate.tests"  
[4] "univariate.tests"    "pval.adjustments"   "sphericity.tests"  
[7] "SSPE"
```

What there is here

- three sets of tests, for
 - ▶ times; drug; their interaction
- two *types* of test for each of these:
 - ▶ univariate; multivariate
- univariate is more powerful *if* it applies; if it doesn't, can make adjustments to it

Sphericity

- The thing that decides whether the univariate tests apply is called “sphericity”.
- This holds if the outcomes have equal variance (to each other) and have the same (positive) correlation across subjects.
- Tested using Mauchly's test (part of output)
- If sphericity rejected, there are adjustments to the univariate P-values due to Huynh-Feldt and Greenhouse-Geisser. Huynh-Feldt better if responses not actually normal (safer).

Sphericity tests

```
summary(dogs.2)$sphericity.tests
```

	Test statistic	p-value
times	0.12334	0.084567
drug:times	0.12334	0.084567

Sphericity is not rejected; proceed to univariate tests.

Univariate tests

```
summary(dogs.2)$univariate.tests
```

	Sum Sq	num Df	Error SS	den Df	F value	Pr(>F)					
(Intercept)	71.342	1	22.1026	6	19.3664	0.004565 **					
drug	11.520	1	22.1026	6	3.1272	0.127406					
times	26.160	3	2.2534	18	69.6546	4.215e-10 ***					
drug:times	5.111	3	2.2534	18	13.6095	7.050e-05 ***					

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

- Significant interaction between drug and time: the pattern of log-histamine over time is different for the different drugs.

If sphericity had been rejected

then we would use the H-F adjusted P-values:

```
summary(dogs.2)$pval.adjustments
```

```
GG eps    Pr(>F[GG])      HF eps    Pr(>F[HF])
times     0.5261798 3.744618e-06 0.6822614 1.843418e-07
drug:times 0.5261798 2.348896e-03 0.6822614 7.307096e-04
attr(,"na.action")
(Intercept)          drug
                  1           2
attr(,"class")
[1] "omit"
```

In this case (because sphericity was not rejected), these are very similar to the ones from the univariate tests, and the conclusion (significant interaction) was the same.

Comments

- If the interaction had not been significant:
 - ▶ cannot remove interaction with time
 - ▶ so look at univariate (or adjusted for sphericity) tests of main effects in model with non-significant interaction

Next

- investigate interaction with graph
- but dataframe has several observations per line (“wide”).
- Plotting works with data in “long format”: one response per line.
- The responses are log-histamine at different times, labelled 1h-something. Call them all 1h 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")  
  
# A tibble: 32 x 5  
# ... with 1 row omitted  
  dog    drug      x     time     lh  
  <chr> <chr>     <chr> <chr> <dbl>  
1 A      Morphine N   lh0    -3.22  
2 A      Morphine N   lh1    -1.61  
3 A      Morphine N   lh3    -2.3  
4 A      Morphine N   lh5    -2.53  
5 B      Morphine N   lh0    -3.91  
6 B      Morphine N   lh1    -2.81  
7 B      Morphine N   lh3    -3.91  
8 B      Morphine N   lh5    -3.91  
9 C      Morphine N   lh0    -2.66  
10 C     Morphine N  lh1     0.34  
# i 22 more rows
```

Getting the times

Not quite right: want new variable containing just number in time:
parse_number. (Top 5 rows shown.)

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

```
# A tibble: 5 x 6
  dog    drug     x    timex     lh   time
  <chr> <chr> <chr> <chr> <dbl> <dbl>
1 A      Morphine N    lh0    -3.22     0
2 A      Morphine N    lh1    -1.61     1
3 A      Morphine N    lh3    -2.3      3
4 A      Morphine N    lh5    -2.53     5
5 B      Morphine N    lh0    -3.91     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` (which we actually *do* use later).
- 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

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

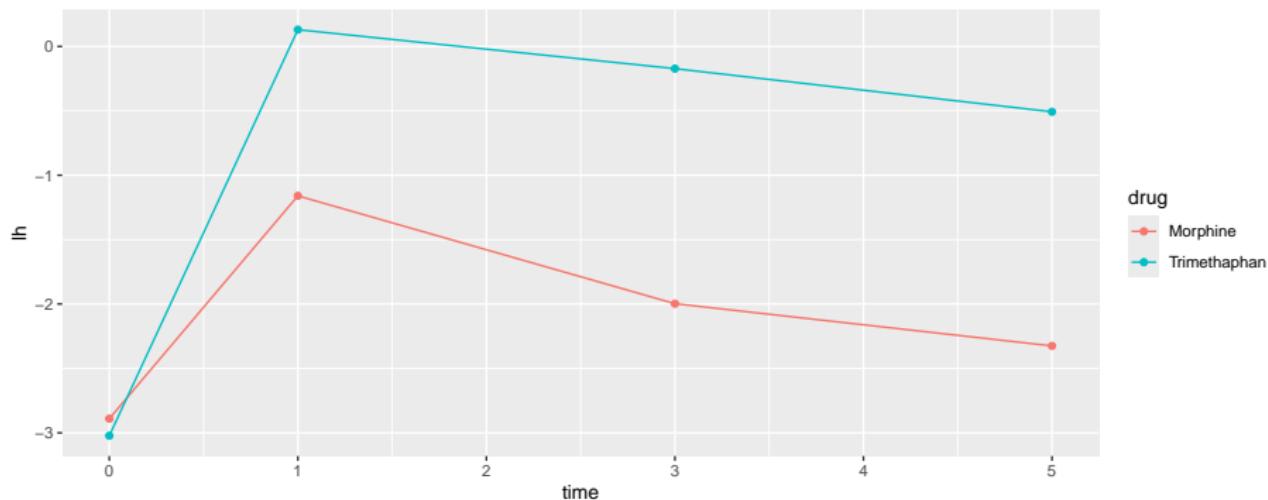
Comments

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 1h 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 (univariate)

```
summary(dogs.2)$sphericity.tests
```

	Test statistic	p-value
times	0.57597	0.25176
drug:times	0.57597	0.25176

```
# summary(dogs.2)$pval.adjustments  
summary(dogs.2)$univariate.tests
```

	Sum Sq	num Df	Error SS	den Df	F value	Pr(>F)					
(Intercept)	24.2607	1	20.1874	6	7.2106	0.03628 *					
drug	16.2197	1	20.1874	6	4.8207	0.07053 .					
times	3.3250	2	0.7301	12	27.3251	3.406e-05 ***					
drug:times	0.3764	2	0.7301	12	3.0929	0.08254 .					

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

Comments

- sphericity: no problem (P-value 0.25)
- univariate test for interaction no longer significant (P-value 0.082)
- look at main effects:
 - ▶ strong significance of time, even after taking out time 0
 - ▶ actually *not* significant drug effect, despite interaction plot

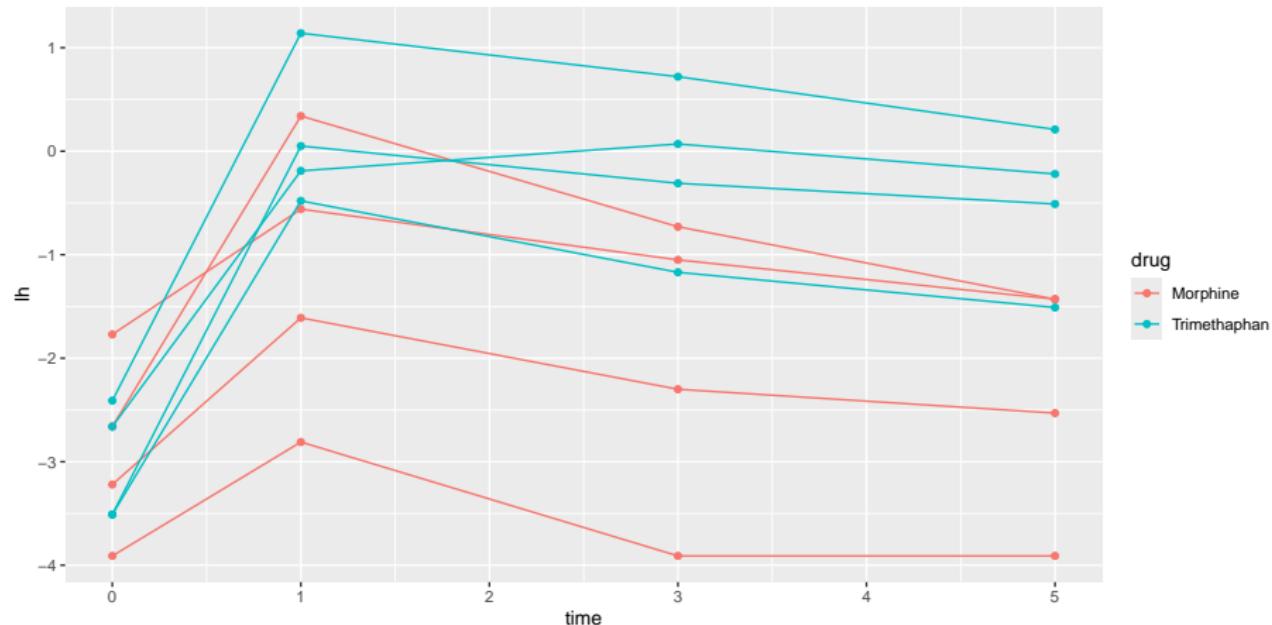
Non-significant drug effect reasonable?

- Plot *actual data*: lh against days, labelling observations by drug: “spaghetti plot”.
- Uses long data frame:
 - ▶ Plot (time, lh) points coloured by *drug*
 - ▶ connecting measurements for each *dog* by lines.
 - ▶ Hence, group = dog, but colour = drug:

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

The spaghetti plot

g



Comments

- For each dog over time, gradual decrease in log-histamine from time 1: significant time effect after we took out time 0.
- Pattern about same for each dog, regardless of drug, hence non-significant interaction.
- Most trimethaphan dogs (blue) have higher log-histamine throughout (time 1 and after), some morphine dogs (red) have lower.
- *But* two morphine dogs have log-histamine profiles like trimethaphan dogs. This ambiguity probably why drug effect not quite significant.

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 (uses lme4)

```
# dogs.long including time zero with categorical timex  
dogs.3 <- lmer(lh ~ drug * timex + (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.3, test="Chisq")
```

Single term deletions

Model:

```
lh ~ drug * timex + (1 | dog)
      npar      AIC      LRT    Pr(Chi)
<none>        62.167
drug:timex     3 84.589 28.422 2.962e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Interaction very significant. Including time zero, the pattern of log-histamine over time is different for the two drugs (as we found before).

Omitting time zero

Let's pretend we are working at $\alpha = 0.01$:

```
dogs.long %>% filter(timex != "lh0") -> dogs.long.no0
dogs.4 <- lmer(lh ~ drug * timex + (1|dog), data=dogs.long.no0
drop1(dogs.4, test = "Chisq")
```

Single term deletions

Model:

```
lh ~ drug * timex + (1 | dog)
      npar      AIC      LRT Pr(Chi)
<none>        42.119
drug:timex    2 44.771 6.6518 0.03594 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Interaction is not quite significant at $\alpha = 0.01$. So we could remove it.

Removing the interaction

```
dogs.5 <- update(dogs.4, . ~ . - drug:timex)
drop1(dogs.5, test = "Chisq")
```

Single term deletions

Model:

```
lh ~ drug + timex + (1 | dog)
      npar      AIC      LRT  Pr(Chi)
<none>    44.771
drug       1 47.489  4.7176  0.02985 *
timex      2 62.972 22.2011 1.51e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

- Definitely an effect of time, but drug is not quite significant (at $\alpha = 0.01$).
- More or less same conclusions as from MANOVA.

The exercise data

- 30 people took part in an exercise study.
- Each subject randomly assigned to one of two diets ("low fat" or "non-low fat") and to one of three exercise programs ("at rest", "walking", "running").
- $2 \times 3 = 6$ experimental treatments, and thus each one replicated $30/6 = 5$ times. (Two-way ANOVA, so far?)
- However, each subject had pulse rate measured at three different times (1, 15 and 30 minutes after starting their exercise), so have repeated measures.

Reading the data

Separated by *tabs*:

```
url <- "http://ritsokiguess.site/datafiles/exercise2.txt"
exercise.long <- read_tsv(url)
exercise.long %>% slice(1:7) # top 7 rows
```

```
# A tibble: 7 x 5
  id diet      exertype pulse time
  <dbl> <chr>    <chr>    <dbl> <chr>
1 1 nonlowfat atrest     85 min01
2 1 nonlowfat atrest     85 min15
3 1 nonlowfat atrest     88 min30
4 2 nonlowfat atrest     90 min01
5 2 nonlowfat atrest     92 min15
6 2 nonlowfat atrest     93 min30
7 3 nonlowfat atrest     97 min01
```

Comments

- “Long format”, usually what we want.
- But for repeated measures analysis, we want *wide* format!
- Keep track of which is which:
 - ▶ Manova analysis: wider
 - ▶ graphs and lmer analysis: longer.
- `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) # random 5 rows
```

```
# A tibble: 5 x 6
  id diet      exertype min01 min15 min30
  <dbl> <chr>     <chr>    <dbl>   <dbl>   <dbl>
1 17 lowfat    walking    103     109     90
2 1 nonlowfat  atrest     85      85      88
3 8 lowfat     atrest     92      94      95
4 28 lowfat    running    103     124     140
5 9 lowfat     atrest     97      99      96
```

Setting up

- Make response variable from min01, min15, min30:

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

- Predict from diet, exertype, interaction using lm:

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

... continued

- Run this through Manova:

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

Sphericity tests

```
summary(exercise.2)$sphericity.tests
```

	Test statistic	p-value
times	0.92416	0.40372
diet:times	0.92416	0.40372
exertype:times	0.92416	0.40372
diet:exertype:times	0.92416	0.40372

No problem with sphericity; go to univariate tests.

Univariate tests

```
summary(exercise.2)$univariate.tests
```

	Sum Sq	num Df	Error SS	den Df	F value
(Intercept)	894608	1	2085.2	24	10296.6595
diet	1262	1	2085.2	24	14.5238
exertype	8326	2	2085.2	24	47.9152
diet:exertype	816	2	2085.2	24	4.6945
times	2067	2	1563.6	48	31.7206
diet:times	193	2	1563.6	48	2.9597
exertype:times	2723	4	1563.6	48	20.9005
diet:exertype:times	614	4	1563.6	48	4.7095
	Pr(>F)				
(Intercept)	< 2.2e-16	***			
diet	0.0008483	***			
exertype	4.166e-09	***			
diet:exertype	0.0190230	*			
times	1.662e-09	***			
diet:times	0.0613651	.			
exertype:times	4.992e-10	***			
diet:exertype:times	0.0027501	**			

Comments

- The three-way interaction is significant
- the effect of diet on pulse rate over time is different for the different exercise types

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.
- `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.)

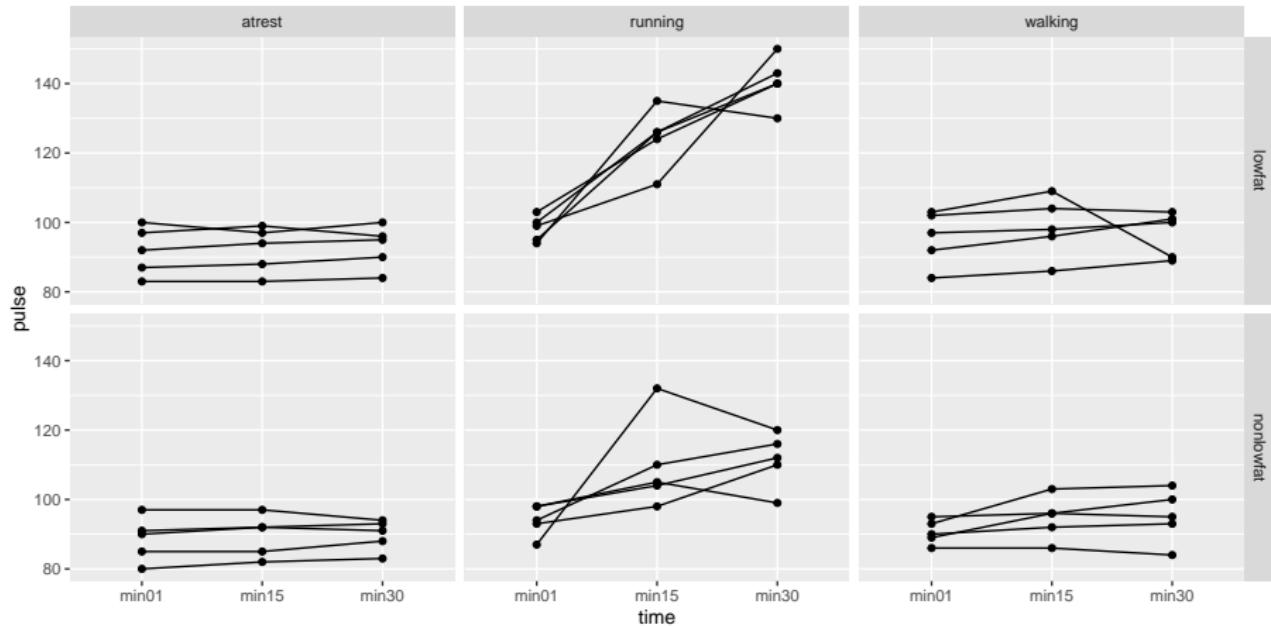
... continued

- 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)
```

The graph(s)

g



Comments on graphs

- At rest: no change in pulse rate over time
- Walking: not much change in pulse rates over time.
- Running: overall increase in pulse rate over time, but increase stronger for lowfat group.
- No consistent effect of:
 - ▶ diet over all exercise groups.
 - ▶ exercise type over both diet groups.
 - ▶ 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
```

... continued

- 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
)
```

Sphericity tests

```
summary(runners.2)$sphericity.tests
```

	Test statistic	p-value
times	0.81647	0.4918
diet:times	0.81647	0.4918

- No problem, look at univariate tests.

Univariate tests

```
summary(runners.2)$univariate.tests
```

	Sum Sq	num Df	Error SS	den Df	F value	Pr(>F)	
(Intercept)	383522	1	339.2	8	9045.3333	1.668e-13	***
diet	1920	1	339.2	8	45.2830	0.0001482	***
times	4714	2	1242.0	16	30.3644	3.575e-06	***
diet:times	789	2	1242.0	16	5.0795	0.0195874	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1							

- Interaction still significant
 - ▶ dependence of pulse rate on time still different for the two diets

How is the effect of diet different over time?

- Table of means. Only I need long data for this:

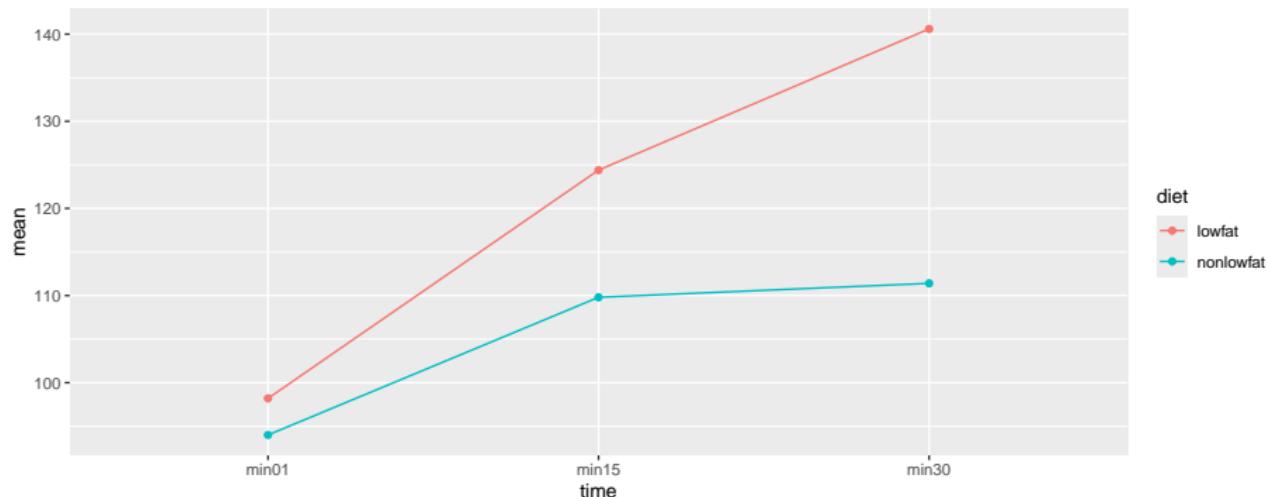
```
runners.wide %>%
  pivot_longer(starts_with("min"),
              names_to = "time", values_to = "pulse") %>%
  group_by(time, diet) %>%
  summarize(
    mean = mean(pulse),
    sd = sd(pulse)
  ) -> summ
```

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

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.