Table of Contents

Sources of Material for these Notes

1. William D. Dupont *Statistical Modeling for Biomedical Researchers*. 2002, New York: Cambridge Univ. Press, Chapter 11.
2. Glenn A. Walker *Common Statistical Methods for Clinical Research with SAS Examples*, 2nd Edition, 2002: SAS Institute, Chapter 8.
3. Julian J. Faraway *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. 2006, Boca Raton, FL: Chapman & Hall, Chapter 9.

Online references include, but are not limited to:

4. http://ww2.coastal.edu/kingw/statistics/R-tutorials/repeated.html
5. http://cran.r-project.org/doc/contrib/Lemon-kickstart/kr_repms.html
6. UCLA's discussion is nice – visit http://goo.gl/8Y8bhF
7. http://www.r-statistics.com/2010/04/repeated-measures-anova-with-r-tutorials/

Longitudinal Data

We turn now to study designs where the same subject is observed repeatedly over time. In analyzing these data, we must take into consideration the fact that the error components of repeated observations on the same subject are usually correlated. In a repeated-measures experiment, the fundamental unit of observation is the subject. We seek to make inferences about members of a target population who are treated in the same way as our study subjects. Using our usual fixed-effect ANOVA approaches to analyze repeated-measures data can lead to wildly exaggerated estimates of statistical significance since we have many more observations than patients.

An Example: Forearm Blood Flow, Race and Dose of Isoproterenol

Lang et al. (1995) studied the effect of isoproterenol, a β-adrenegic agonist, on forearm blood flow in a group of 21 normotensive men[1]. Nine subjects were black, 12 white. Each subject's blood flow was measured at baseline and then at escalating doses (10, 20, 60, 150, 300 and 400 ng/min) of isoproterenol. Here's one representation of the data:

id	race	fbf0	fbf10	fbf20	fbf60	fbf150	fbf300	fbf400
1	white	1.0	1.4	6.4	19.1	25.0	24.6	28.0
2	white	2.1	2.8	8.3	15.7	21.9	21.7	30.1
etc.								
22	black	2.1	1.9	3.0	4.8	7.4	16.7	21.2

This is called the **wide format** for these data, and is stored in the `isoproterenol.csv` data file on the course web site.

id	dose	race	fbf
1	0	1	1.0
1	10	1	1.4
1	20	1	6.4
1	60	1	19.1
1	150	1	25.0
1	300	1	24.6
1	400	1	28.0
2	0	1	2.1
21	400	2	21.2

In order to plot or work with the data in detail, we will also want to develop a **long format** version of the data, with one row for every forearm blood flow measurement.

That data set is called `isoproterenol-long1.csv` and it's also on the course web site.

Remember that Chang's R Graphics Cookbook has some nice material in Chapter 15 (sections 19 and 20) about melting data to move from wide format to long format, and casting data to convert from long format back to wide format. We discussed this in EPBI 431, of course.

Suppose, for instance, that we wanted to move from wide to long format here…

```
isop <- read.csv("isoproterenol.csv")
## Show how to convert from wide format in isop to long format in
isop.long1
library(reshape2)
isop.long1 <- melt(isop, id.vars=c("id", "race"),
     measure.vars=c("fbf0", "fbf10", "fbf20", "fbf60", "fbf150",
     "fbf300", "fbf400"), variable.name="dose.raw", value.name="fbf")
```

[1] Source: Dupont, p.338 – although I dropped one case with missing values of blood flow for simplicity.

```
> summary(isop.long1)
       id         race      dose.raw       fbf        
 Min.   : 1   black:63   fbf0  :21   Min.   : 1.000  
 1st Qu.: 6   white:84   fbf10 :21   1st Qu.: 3.025  
 Median :11              fbf20 :21   Median : 5.800  
 Mean   :11              fbf60 :21   Mean   : 9.443  
 3rd Qu.:16              fbf150:21   3rd Qu.:12.750  
 Max.   :21              fbf300:21   Max.   :43.300  
                         fbf400:21                   
```

The problem now is that I don't like those levels for `dose.raw`, and would like instead to have a numeric representation of the dose. First, I'll build a new factor variable, `dose1`, that contains the numeric representation, then I'll use a tricky approach to convert those results to something R correctly recognizes as a number, and put that in `dose.n`

```
library(plyr)
isop.long1$dose1 <- mapvalues(isop.long1$dose.raw,
      c("fbf0", "fbf10", "fbf20", "fbf60", "fbf150", "fbf300",
      "fbf400"), c("0", "10", "20", "60", "150", "300", "400"))
isop.long1$dose.n <-
      as.numeric(levels(isop.long1$dose1))[isop.long1$dose1]
```

```
summary(isop.long1)
       id         race      dose.raw       fbf         dose1        dose.n     
 Min.   : 1   black:63   fbf0  :21   Min.   : 1.000   0  :21   Min.   :  0.0  
 1st Qu.: 6   white:84   fbf10 :21   1st Qu.: 3.025   10 :21   1st Qu.: 10.0  
 Median :11              fbf20 :21   Median : 5.800   20 :21   Median : 60.0  
 Mean   :11              fbf60 :21   Mean   : 9.443   60 :21   Mean   :134.3  
 3rd Qu.:16              fbf150:21   3rd Qu.:12.750   150:21   3rd Qu.:300.0  
 Max.   :21              fbf300:21   Max.   :43.300   300:21   Max.   :400.0  
                         fbf400:21                    400:21                  
```

Or, we could just import the `isoproterenol-long1.csv` data file from the web.

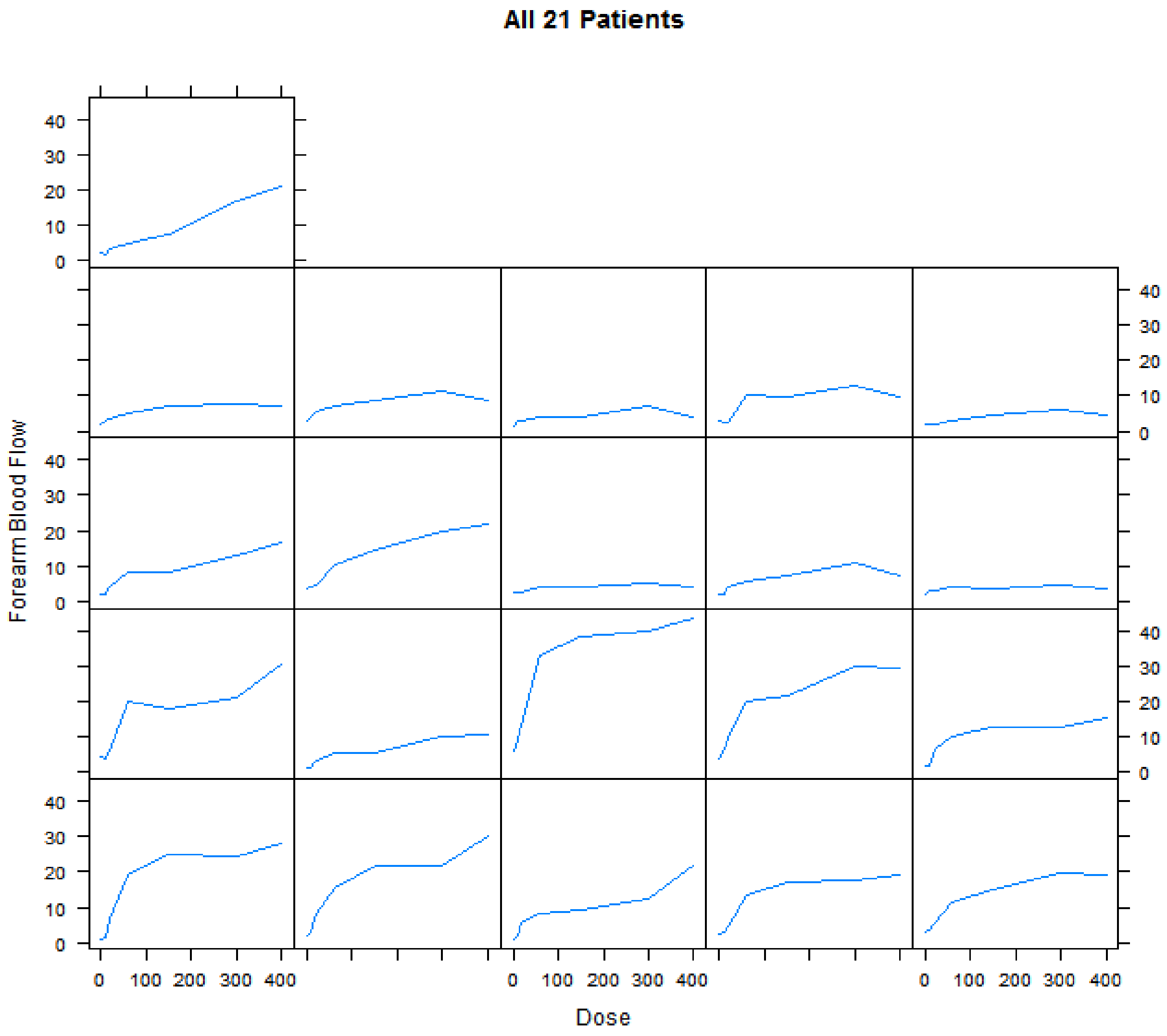
```
isop.long1web <- read.csv("isoproterenol-long1.csv")
```

In what follows, I'll use this version from the web.

```
attach(isop.long1web)
summary(isop.long1web)
       id          dose          race          fbf        
 Min.   : 1   Min.   :  0.0   black:63   Min.   : 1.000  
 1st Qu.: 6   1st Qu.: 10.0   white:84   1st Qu.: 3.000  
 Median :11   Median : 60.0              Median : 5.800  
 Mean   :11   Mean   :134.3              Mean   : 9.443  
 3rd Qu.:16   3rd Qu.:300.0              3rd Qu.:12.750  
 Max.   :21   Max.   :400.0              Max.   :43.300  
```

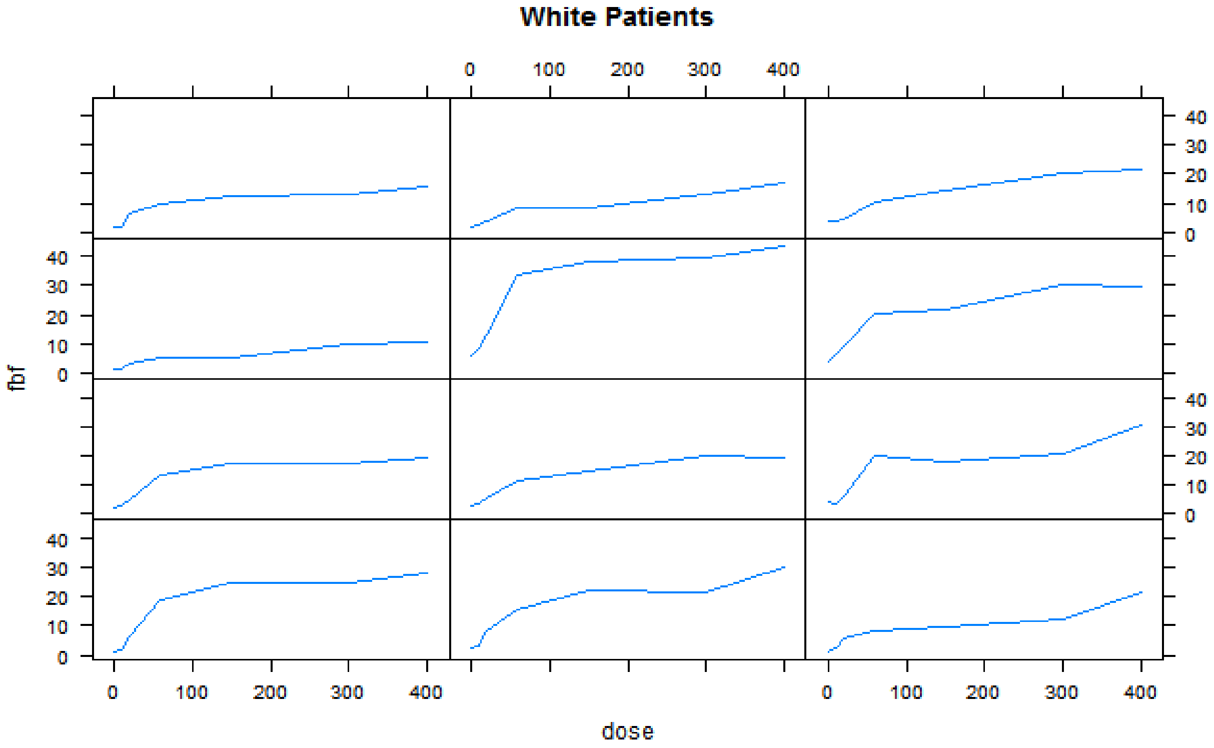
Now, let's plot the individual results. One way to do this, for a small data set like this, is with a **lattice plot**.

```
library(lattice)
xyplot(fbf ~ dose | id, isop.long1web, type="l", strip=FALSE,
main="All 21 Patients", ylab="Forearm Blood Flow", xlab="Dose")
```

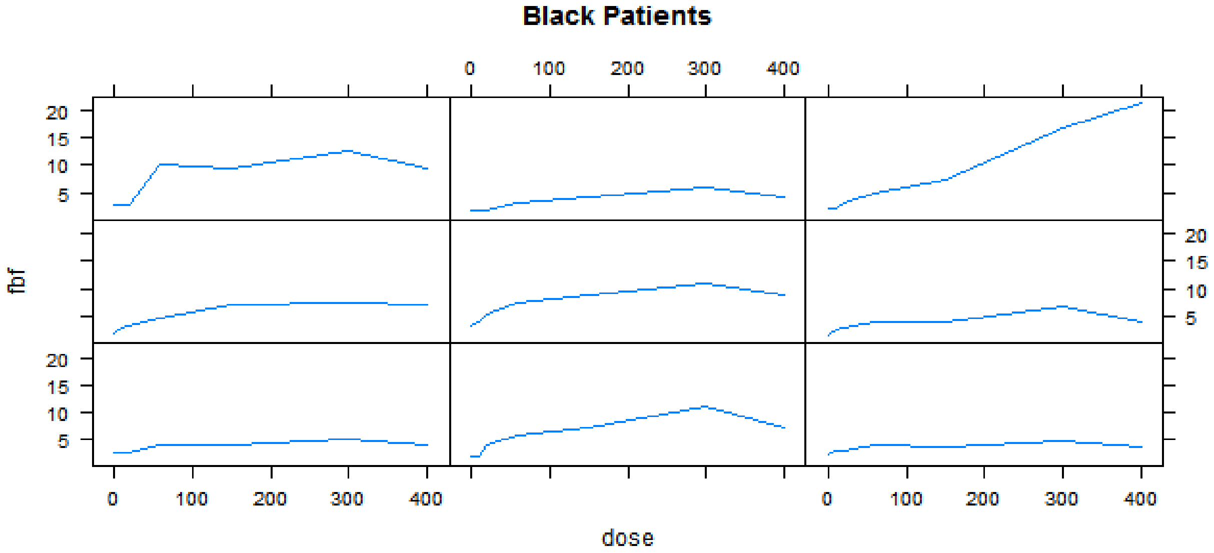


We might also plot the white and black patients in separate lattice plots.

```
xyplot(fbf ~ dose | id, isop.long1web, type="l",
       subset=(race=="white"), strip=FALSE, main="White Patients")
```



```
xyplot(fbf ~ dose | id, isop.long1web, type="l",
       subset=(race=="black"), strip=FALSE, main="Black Patients")
```

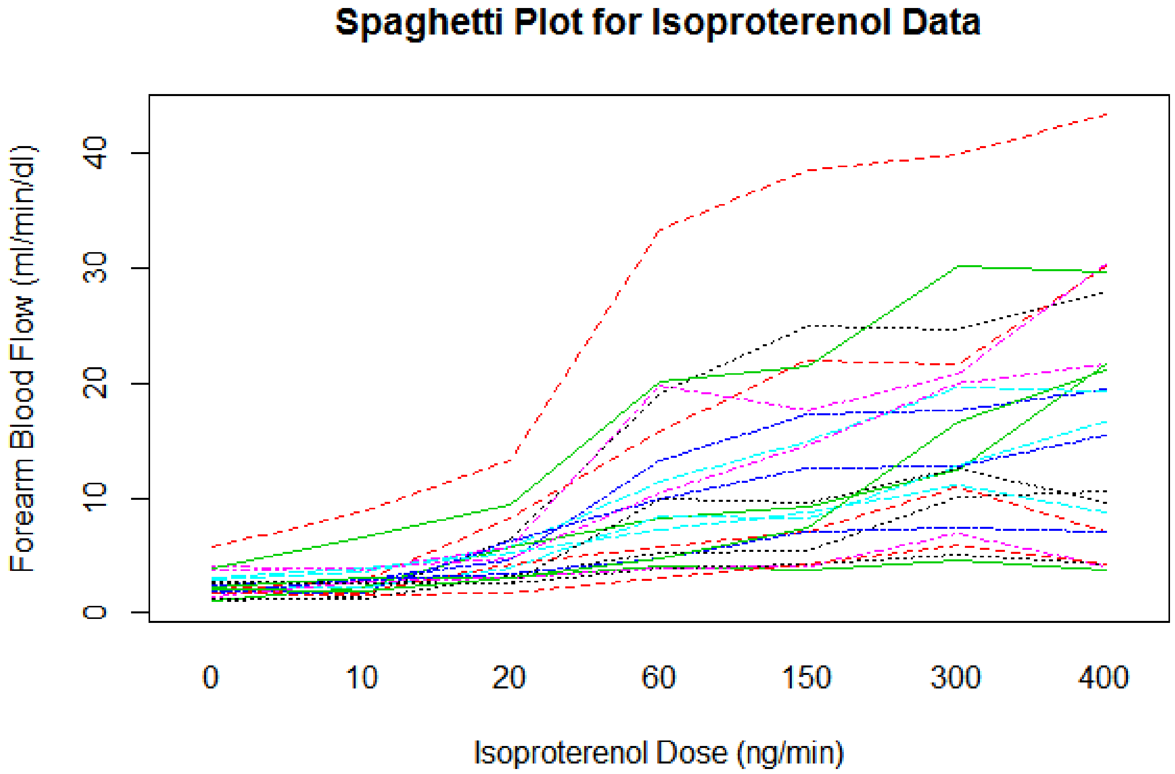


This works pretty well, if you have a small number of subjects.

Spaghetti Plots

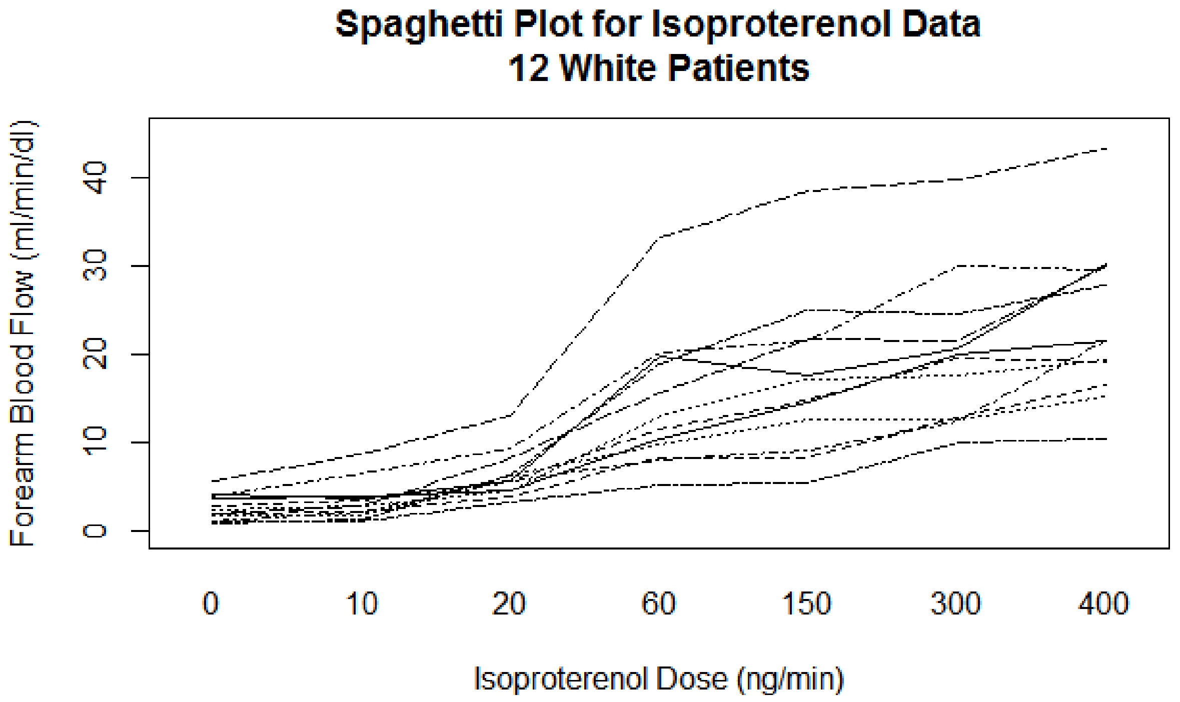
A common approach is to use a **spaghetti plot**, which can be formed using the `interaction.plot` function in R that we've used before.

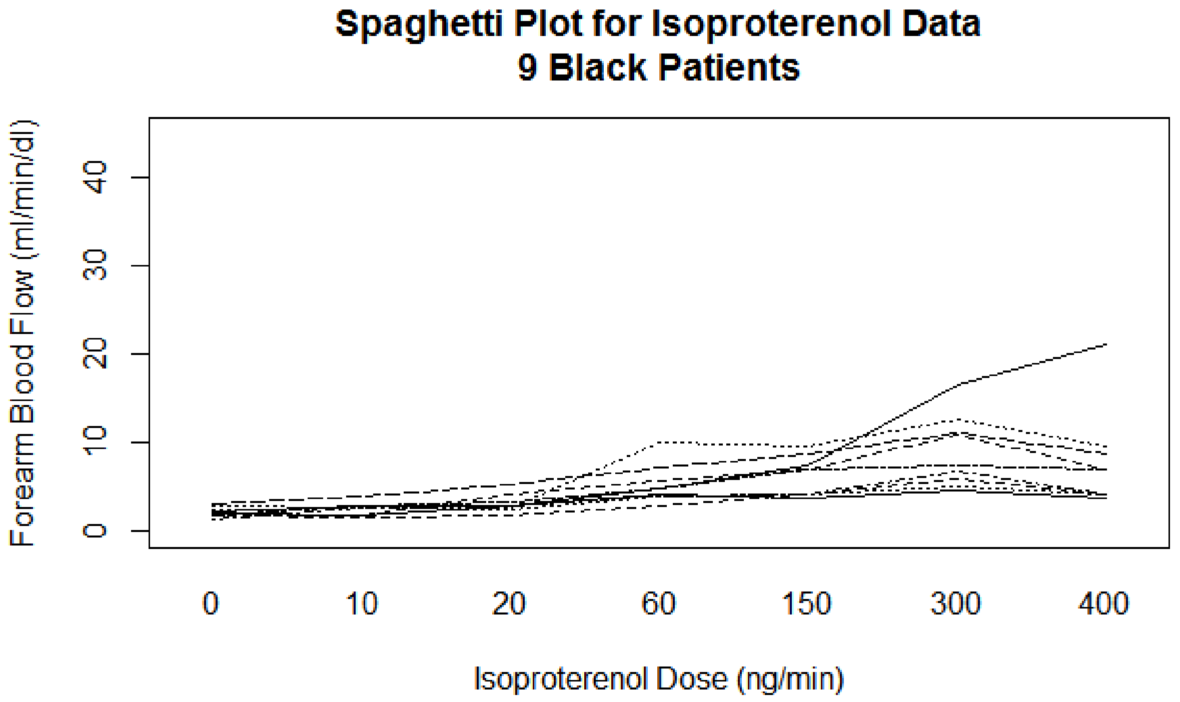
```
interaction.plot(dose, id, fbf, col=c(1:6), legend=FALSE,
                 main="Spaghetti Plot for Isoproterenol Data",
                 ylab="Forearm Blood Flow (ml/min/dl)",
                 xlab="Isoproterenol Dose (ng/min)")
```



We can produce separate spaghetti plots using this approach for the two race groups. Note how I'm forcing the y-axis to be the same for each of the plots below, so we can directly compare the White and Black data.

```
interaction.plot(dose[race=="white"], id[race=="white"],
     fbf[race=="white"], legend=FALSE, ylim=c(0,45), main="Spaghetti
     Plot for Isoproterenol Data\n13 White Patients", ylab="Forearm
     Blood Flow (ml/min/dl)", xlab="Isoproterenol Dose (ng/min)")

interaction.plot(dose [race=="black"], id[race=="black"],
     fbf[race=="black"], legend=FALSE, ylim=c(0,45), main="Spaghetti
     Plot for Isoproterenol Data\n9 Black Patients", ylab="Forearm
     Blood Flow (ml/min/dl)", xlab="Isoproterenol Dose (ng/min)")
```





Here, the lines cross some, but not much, indicating a high degree of correlation between observations from the same patient. The response of men to escalating doses of isoproterenol appeats to be greater in white patients than in black patients.

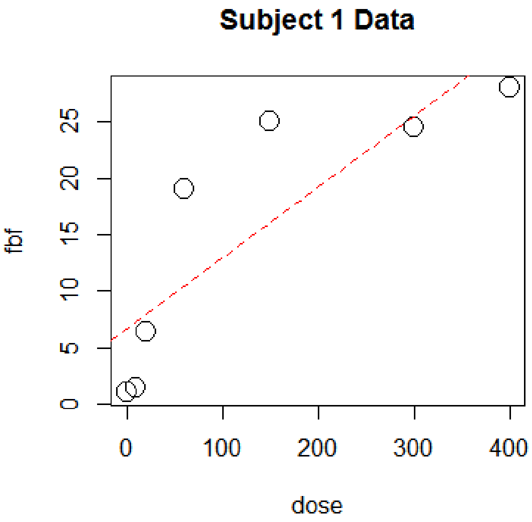
Response Feature Analysis: Fitting a Straight Line to Each Subject

The simplest approach to analyzing repeated measures data is a **response feature analysis**. The basic idea is to reduce the multiple responses on each subject to a single biologically meaningful response that captures the patient attribute of interest. This response measure is then analyzed in a fixed-effects one-way ANOVA, or similar procedure. Examples of a response feature that may be useful include a regression slope or other summary derived from the observations on an individual patient.

To start our modeling, we might begin by ignoring the race information, and instead fitting a line to each subject, predicting their forearm blood flow as a function of the dose of isoproterenol received, starting with the first subject.

```
linmod <- lm(fbf ~ dose,
 subset=(id==1), isop.long1web)
coef(linmod)
(Intercept)        dose
 6.63155995  0.06285009
```

```
plot(fbf ~ dose, subset=(id==1),
 data=isop.long1web, cex=2,
 main="Subject 1 Data")
abline(linmod, col="red", lty=2)
```



We now fit a line to each of the 21 subjects in the data, and plot the results:

```
slopes <- numeric(21); intercepts <- numeric(21)
for(i in 1:21){
  lmod <- lm(fbf ~ dose, subset=(id==i), isop.long1web)
  intercepts[i] <- coef(lmod)[1]
  slopes[i] <- coef(lmod)[2]
}
```

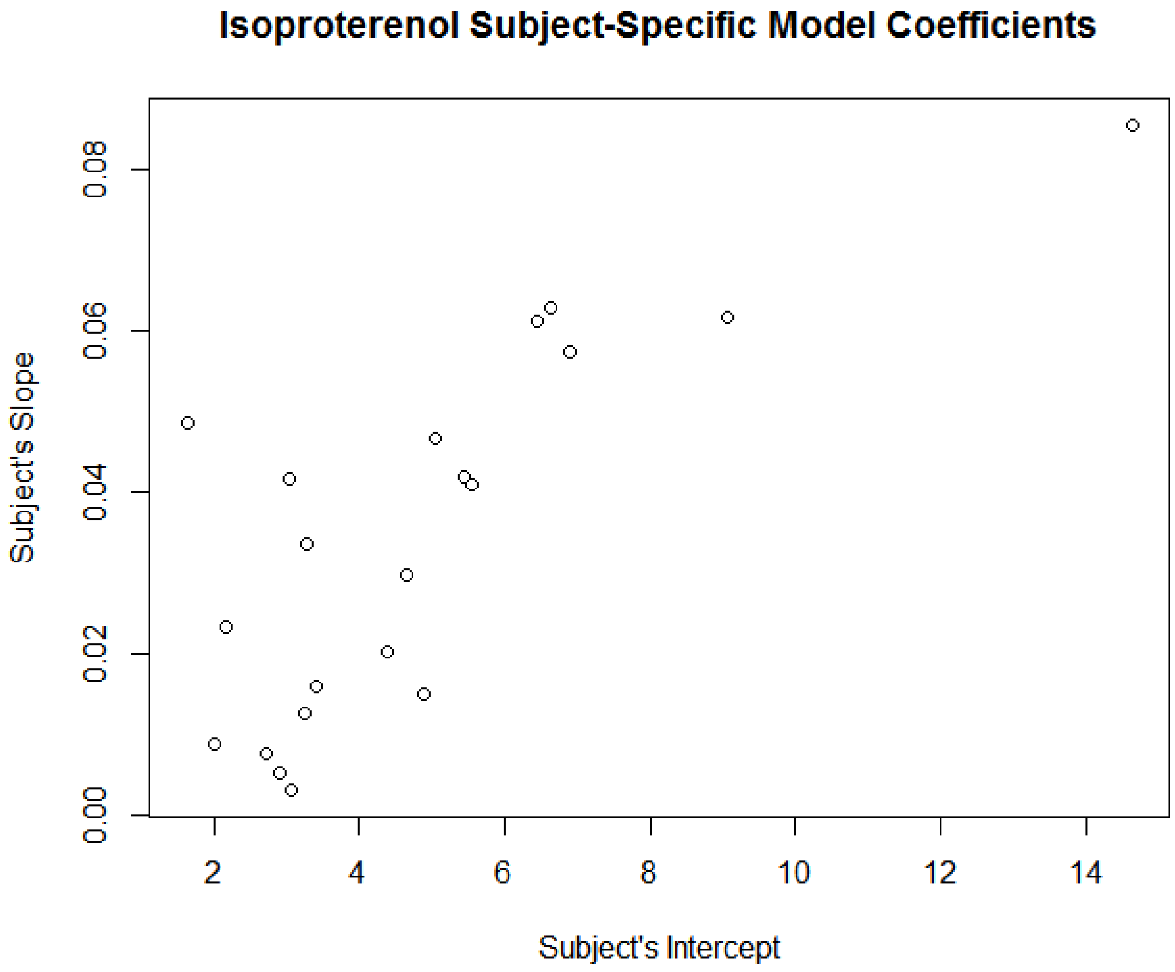
The first part of that command group specifies that we're going to need room for 21 slopes and 21 intercepts, because we have 21 subjects in the data.

Then, we fit a separate linear model for each subject, and store the coefficients in the intercepts and slopes variables, respectively.

Plotting the Subject-Specific Slopes and Intercepts

We can plot the subject-specific slopes and intercepts in a scatterplot, as follows.

```
plot(intercepts, slopes, xlab="Subject's Intercept", ylab="Subject's
Slope", main="Isoproterenol Subject-Specific Model Coefficients" )
```



What can we conclude from this graph?

Building the New Data Frame, with Subject-Specific Summaries

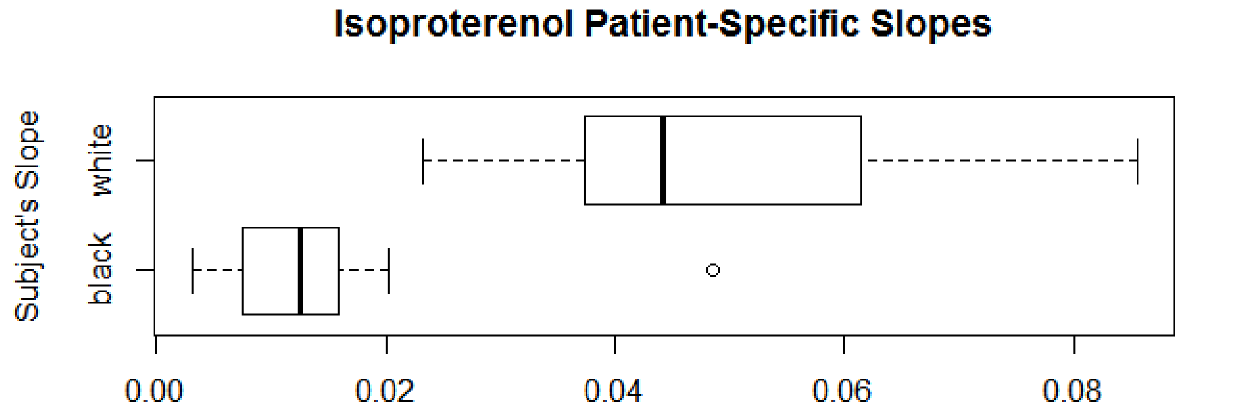
We can then build a data frame (called regres.df) which includes the slope, intercept, and also the race information for each subject using this little trick to gather the race information from the original data frame into a new variable, which I'll call prace.

```
prace <- isop.long1web$race[match(1:21, isop.long1web$id)]
regres.df <- data.frame(subject=1:21, slopes, intercepts, prace)
```

Comparing the two Race Groups in terms of Regression Coefficients

We now have a data frame which includes the slope and intercept, as well as the race, for each of the 21 patients. So we can compare, for instance, the slopes by race.

```
boxplot(slopes ~ prace, horizontal=TRUE, data=regres.df,
 ylab="Subject's Slope", main="Isoproterenol Patient-Specific Slopes")
```



```
t.test(slopes ~ prace, data=regres.df)
Welch Two Sample t-test
data:  slopes by prace
t = -4.9554, df = 18.936, p-value = 8.867e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: -0.04782043 -0.01941530
sample estimates: mean in group black mean in group white
                           0.01523086          0.04884872
```

We conclude that there is a statistically significantly larger response of whites to increasing doses of isoproterenol. Had there been three or more racial groups, we'd have used a one-way ANOVA or Kruskal-Wallis test. A rank sum test might also have been appropriate in this setting, due to the one large outlier in the "black" group.

Our response feature analysis establishes that there is a statistically significant difference in the response of blacks and whites to increasing doses of isoproterenol. Of course, one needs to interpret this cautiously. We cannot infer that this difference in response by race is of genetic origin, as genetic and environmental factors are so highly confounded in our society. It is certainly possible that race may be a marker of some environmental difference that explains these results.

Looking into Dose Effects

We might also be seriously interested in determining which doses induce a significant effect. Given the differences between the trajectories of response to dosage by race that we have already observed, we probably want to separately look at white and black patients in this context. We could, for instance, separately consider results at each dosage in a series of two-sample (the two races) t tests.

```
dose0 <- data.frame(id=isop.long1web$id[dose==0],
      race=isop.long1web$race[dose==0], fbf=isop.long1web$fbf[dose==0])
```

```
summary(dose0)
       id          race          fbf       
 Min.   : 1   black: 9   Min.   :1.000  
 1st Qu.: 6   white:12   1st Qu.:1.800  
 Median :11              Median :2.100  
 Mean   :11              Mean   :2.467  
 3rd Qu.:16              3rd Qu.:2.900  
 Max.   :21              Max.   :5.800  
```

```
t.test(fbf ~ race, data=dose0)

      Welch Two Sample t-test

data:  fbf by race
t = -1.0618, df = 14.774, p-value = 0.3054
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.4632059  0.4909837
sample estimates:
mean in group black mean in group white 
           2.188889            2.675000 
```

And so on, for each possible dosage (0, 10, 20, 60, 150, 300 and 400)

Here's a summary table of results comparing forearm blood flow by race within each available dosage.

Dose	White Mean FBF	Black Mean FBF	Welch test p value	95% CI (B – W)
0	2.68	2.19	0.31	(-1.5, 0.5)
10	3.42	2.58	0.24	(-2.3, 0.6)
20	6.46	3.19	0.002	(-5.1, -1.4)
60	14.59	5.31	0.001	(-14.3, -4.3)
150	17.25	6.24	0.001	(-16.8, -5.2)
300	20.20	9.04	0.001	(-17.0, -5.3)
400	23.83	7.78	< 0.001	(-22.6, -9.5)

What conclusions can we draw here?

Repeated Measures Analysis of Variance

Repeated measures refer to multiple measurements taken from the same subject, often serial evaluations over time. Any appropriate analysis must take advantage of what is known about the correlation structure of these "longitudinal data", which can no longer be assumed to be independent.

A common analytic goal with these data is to describe a mean outcome (response) profile over time, or to compare intervention groups on the basis of their mean response profile. This happens, for instance, if we're trying to determine whether the onset of effect or the rate of improvement due to one treatment is faster than that of a competitor. Comparison of response profiles is based on F tests from a repeated-measures analysis of variance.

There are several analytic approaches for handling repeated measures. A "univariate" approach re-uses the same ANOVA concepts we discussed previously. In contrast, a "multivariate" approach treats the repeated measurements as a multivariate outcome. We'll show examples of both methods.

In general, we have g independent groups of subjects, each of whom are subjected to repeated measurements of the same outcome variable, y, at t equally spaced time periods. Our analysis of variance will partition the variability we observe into that attributable to the group level, to the patients (within the groups), to the time period, and to a group-by-time interaction term.

Repeated Measures ANOVA Table

The ANOVA summary will look like this, if N is the total number of observations.

Source	df	SS	MS	F
GROUP	g-1	SSG	MSG	F_G = MSG/MSP(G)
Patient (within Group)	N-g	SSP(G)	MSP(G)	
TIME	t-1	SST	MST	F_T = MST/MSE
GROUP by TIME	(g-1)(t-1)	SSGT	MSGT	F_{GT} = MSGT/MSE
Error	(N-g)(t-1)	SSE	MSE	
Total	Nt-1	TOT(SS)		

The samples are not independent within the group-time cells, because measurements over time on the same subject are correlated.

Arthritic Discomfort Following Vaccine – Univariate approach, Balanced Design

A pilot study was conducted in 8 patients to evaluate the effect of a new vaccine on discomfort due to arthritic outbreaks[2]. Four patients were randomly assigned to receive an active vaccine, while the other four received a placebo. The patients were asked to return to the clinic monthly for three months and evaluate their comfort level (0 = no discomfort, 10 = maximum discomfort) with routine daily chores during the previous month. Eligibility criteria required patients to have a rating of at least 8 in the month prior to vaccination. Is there any evidence of a difference in response profiles between the active and placebo vaccines?

Vaccine	Patient #	Month 1	Month 2	Month 3
Active	101	6	3	0
	103	7	3	1
	104	4	1	2
	107	8	4	3
Placebo	102	6	5	5
	105	9	4	6
	106	5	3	4
	108	6	2	3

[2] Example comes from Walker, Example 8.1

Using a “univariate” repeated measures ANOVA, we’ll identify:

- Vaccine as the Group effect
- Month to represent the Time effect
- The Patient within Vaccine and Vaccine by Month effects are also included in the ANOVA.

The data setup we need will have one row for each measured outcome - like this:

id	vaccine	month	rating
101	Active	1	6
101	Active	2	3
101	Active	3	0
102	Placebo	1	6
102	Placebo	2	5
102	Placebo	3	5

Etc.

The arthr.csv data file on the course website contains this information.

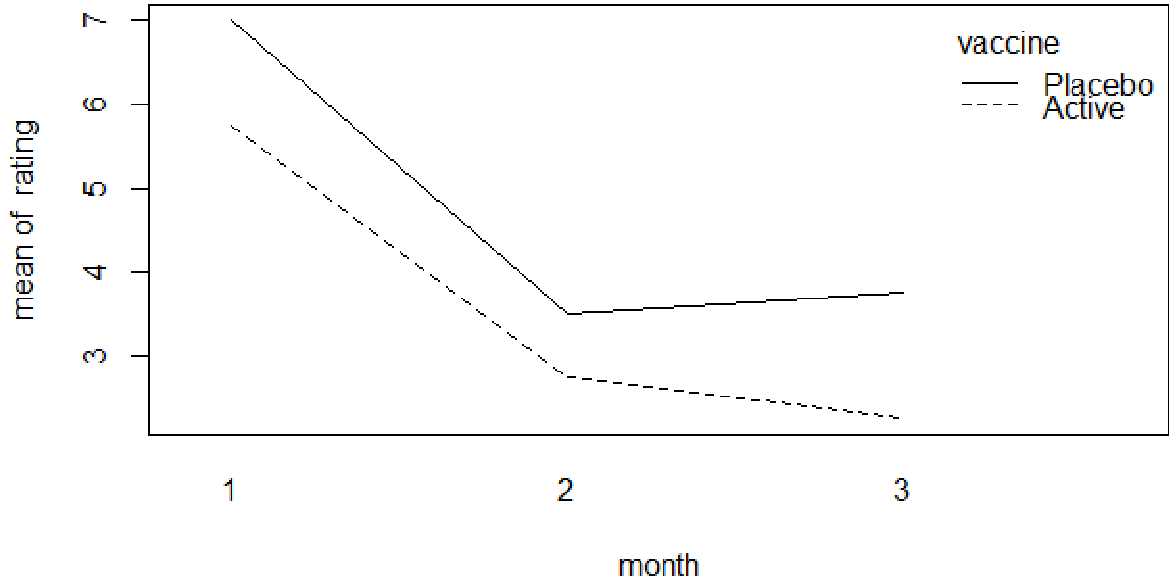
```
arthr <- read.csv("arthr.csv")

str(arthr)
'data.frame':     24 obs. of  4 variables:
 $ id     : int  101 101 101 102 102 102 103 103 103 104 ...
 $ vaccine: Factor w/ 2 levels "Active","Placebo": 1 1 1 2 2 2 1 1 1 1 ...
 $ month  : int  1 2 3 1 2 3 1 2 3 1 ...
 $ rating : int  6 3 0 7 3 1 4 1 2 8 ...
```

We need to convert id and month to factors, since vaccine is already a factor.

```
arthr <- within(arthr, {id <- factor(id)
                        month <- factor(month)
                        })
```

Now, we can build an interaction plot, and see what we're dealing with.

```
with(arthr, interaction.plot(month, vaccine, rating))
```



It looks like the effect of month (at least from month 1 to month 2) is a bigger effect than the difference between the vaccine groups.

```
arthr.aov <- aov(rating ~ vaccine * month + Error(id), data=arthr)
summary(arthr.aov)
```

```
Error: id
          Df Sum Sq Mean Sq F value Pr(>F)
vaccine    1  8.167   8.167    1.76  0.233
Residuals  6 27.833   4.639

Error: Within
              Df Sum Sq Mean Sq F value   Pr(>F)
month          2  58.58  29.292  17.430 0.000282 ***
vaccine:month  2   0.58   0.292   0.174 0.842745
Residuals     12  20.17   1.681
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The F test for the Vaccine effect is not significant, consequently in the interaction plot, we saw that the lines for the two groups are fairly close together. The within subject test indicates a very significant month effect – that is, the groups do change in rating over time. The lines are fairly parallel, which is consistent with a finding of a non-significant interaction effect.

Creating a Summary Table for the Univariate Repeated Measures ANOVA

R's ANOVA table for the `arthr` data can be converted into the following ANOVA summary – just add the df and SS together to get the Total results…

Source	**df**	**SS**	**MS**	**F**	**P**
Vaccine	1	8.167	8.167	1.76	0.23
Patient (within Vaccine)	6	27.833	4.639		
Month	2	58.583	29.292	17.43	0.0003
Vaccine x Month	2	0.583	0.292	0.17	0.84
Error	12	20.167	1.681		
Total	**23**	**115.33**			

The null hypothesis of similar response profiles over time for each of the vaccine groups is tested by the Vaccine-by-Month interaction. For two treatment groups, as in this example (two Vaccines), the hypothesis can be expressed as the simultaneous equality of Vaccine group differences at each Month. A one-way ANOVA could be used to test for the Vaccine effect at each Month, separately.

The Key Assumption: Compound Symmetry

The key assumption behind univariate analysis for repeated measures ANOVA is that of compound symmetry, i.e. that in addition to normality of the distribution of our outcome, and homogeneous population variances in each group, we must also assume that each pair of repeated measures has the same correlation.

This assumption is often not valid, especially if the trial is lengthy or if the time points are unequally spaced, since measurements taken farther apart in time may be less correlated than those taken closer together. Your conclusions using a univariate approach will not be correct if the compound symmetry assumption is substantially violated.

I should also mention that the method described above is designed for a balanced design, and that using repeated measures ANOVA in an unbalanced design is beyond the scope of this presentation.

A major limitation of this procedure is that you can't do post hoc tests on the repeated measures factor, which seems like a big limitation.

A Univariate Repeated-Measures ANOVA analysis for the Isoproterenol Data

We must be certain that the dose, race and id are all treated as factors by the ANOVA routine.

```
isop.aov <- aov(fbf ~ factor(dose) * race + Error(factor(id)),
      data=isop.long1web)
summary(isop.aov)
```

```
Error: factor(id)
          Df Sum Sq Mean Sq F value  Pr(>F)   
race       1   1994  1993.6   15.02 0.00102 **
Residuals 19   2522   132.7                  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Within
                   Df Sum Sq Mean Sq F value   Pr(>F)    
factor(dose)        6   4446     741   67.12  < 2e-16 ***
factor(dose):race   6   1098     183   16.57 1.15e-13 ***
Residuals         114   1259      11                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Source	**df**	**SS**	**MS**	**F**	**P**
Dose	1	1994	1993.6	15.02	.001
Patient (within Dose)	19	2522	132.7		
Race	6	4446	741	67.12	< .001
Dose x Race	6	1098	183	16.57	< .001
Error	114	1259	11		
Total	**146**	**11319**			

What conclusions would we draw here about the isoproterenol data?

Linear Mixed Effects Modeling

Another more general approach that doesn't require the compound symmetry assumption is to use a Linear Mixed Effects model, as in the `nlme` package. The approach here is to treat the subjects (`id`) as a random factor, which is nested within the `vaccine` and `month` results. So our analysis boils down to a pair of fixed factors (`vaccine` and `month`) and their interaction, and a random subject (`id`) factor.

This approach controls for non-independence among the repeated observations for each individual, but it does so in a conceptually different way. Rather than just estimate the correlation among an individual's repeated observations, it actually adds one or more random effects for Individuals to the model. The model equation therefore includes extra parameters to include any random effects. They take the form of additional residual terms, each of which has its own variance to be estimated. This means the model is controlling for the effects of each individual.

The simplest mixed model, the random intercept model, controls for the fact that some individuals always have higher values than others. By controlling for this variation, we've taken it out of the original residual[3].

```
library(nlme)
arthr.lme <- lme(rating ~ vaccine * month, data=arthr,
      random = ~ 1 | id)
summary(arthr.lme)
Linear mixed-effects model fit by REML
 Data: arthr
      AIC      BIC    logLik
  90.8359 97.95887 -37.41795

Random effects:
 Formula: ~1 | id
        (Intercept) Residual
StdDev:   0.9930301 1.296363

Fixed effects: rating ~ vaccine * month
                      Value Std.Error DF   t-value p-value
(Intercept)            5.75 0.8164964 12  7.042285  0.0000
vaccinePlacebo         1.25 1.1547003  6  1.082532  0.3206
month2                -3.00 0.9166670 12 -3.272726  0.0067
month3                -3.50 0.9166670 12 -3.818181  0.0024
vaccinePlacebo:month2 -0.50 1.2963628 12 -0.385694  0.7065
vaccinePlacebo:month3  0.25 1.2963628 12  0.192847  0.8503
```

[3] See http://www.theanalysisfactor.com/repeated-measures-approaches/

```
Correlation:
                      (Intr) vccnPl month2 month3 vccP:2
vaccinePlacebo        -0.707                            
month2                -0.561  0.397                     
month3                -0.561  0.397  0.500              
vaccinePlacebo:month2  0.397 -0.561 -0.707 -0.354       
vaccinePlacebo:month3  0.397 -0.561 -0.354 -0.707  0.500

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max 
-1.5883921 -0.6518705  0.0675541  0.5690145  1.1449586 

Number of Observations: 24   Number of Groups: 8 
```

```
anova(arthr.lme)
              numDF denDF  F-value p-value
(Intercept)       1    12 89.82047  <.0001
vaccine           1     6  1.76048  0.2328
month             2    12 17.42974  0.0003
vaccine:month     2    12  0.17355  0.8427
```

Again, we conclude that the months (both 2 and 3) differ from month 1, but neither the vaccine factor nor the interaction terms are significant.

We'll see another way to look at this sort of analysis, using the `lme4` library, in a moment.

A Linear Mixed Model to Look at our Isoproterenol Responses

Suppose that the relationship between forearm blood flow and dose of isoproterenol can be partly predicted by the subject's race. The variation may be partitioned into two components. Clearly, there are other factors that will affect a subject's forearm blood flow. These factors may cause the FBF to be higher or lower, generally, or to cause the FBF rate to grow at a faster or slower rate as the dose is increased. We can model this variation with a random intercept and slope, respectively, for each subject. We also expect that there will be some variation within each subject.

For simplicity's sake, we sometimes assume initially that this error is homogeneous and uncorrelated. This yields the following **linear mixed model**.

```
library(lme4)
mmod <- lmer(fbf ~ dose*race +(dose|id), data=isop.long1web)

summary(mmod)
Linear mixed model fit by REML ['lmerMod']
Formula: fbf ~ dose * race + (dose | id)
   Data: isop.long1web
REML criterion at convergence: 854.3

Scaled residuals: 
    Min      1Q  Median      3Q     Max 
-1.8735 -0.5955 -0.0595  0.3828  4.3503 

Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 id       (Intercept) 4.993e+00 2.23440      
          dose        2.071e-04 0.01439  1.00
 Residual             1.261e+01 3.55142      
Number of obs: 147, groups:  id, 21

Fixed effects:
               Estimate Std. Error t value
(Intercept)    3.145190   0.960720   3.274
dose           0.015231   0.005686   2.679
racewhite      2.927267   1.270913   2.303
dose:racewhite 0.033618   0.007522   4.469

Correlation of Fixed Effects:
            (Intr) dose   racwht
dose         0.425              
racewhite   -0.756 -0.321       
dose:racwht -0.321 -0.756  0.425
```

Conclusions?

Generalized Estimating Equations: The Basics

For repeated measures, a natural choice of modeling strategy is to consider an approach called **generalized estimating equations**. Within a GEE structure, one can mirror the analyses performed above (in our linear mixed effects models) in terms of specifying the fixed effects, and identifying a grouping variable (now with an id argument), but now we can also consider models, where, for instance, a binary outcome is of interest.

The tradeoffs are as follows.

[1] Only simple groups are allowed in this setting, while nested grouping variables cannot be accommodated easily in the gee function I am about to display. So if we have patients, within providers, within health systems, and we want to account for all we're going to need a more sophisticated modeling structure.

[2] We must specify a correlation structure within each group. If we choose no correlation, the problem reduces to a standard GLM. Several choices are available. In some cases, it seems reasonable to assume that any pair of observations from the same subject have the same correlation. This is known as an **exchangeable** correlation, or, equivalently as compound symmetry. Another common choice is an **AR-1** model for the correlation structure – this is most appropriate when measurements are spread out over a long time horizon and we believe that consecutive measurements will be more correlated than measurements separated in time. Sometimes, we build a model without making any meaningful assumptions about the correlation structure, with the **unstructured** approach, but this tends to reduce statistical power. Other options include independence, fixed, stat_M_dep, and non_stat_M-dep structures.

GEE produces both naïve standard errors and Z statistics (those obtained by assuming the proposed correlation structure is correct) and robust versions of the same quantities. **You want the robust ones**, and here's why. GEE has the property that even if the proposed correlation structure is incorrect, the fixed effect estimates are still consistent, but the naïve standard errors are improved, in most cases, by the use of a what's called a sandwich estimator. This sandwich estimator gives us the robust standard errors.

A GEE models the data at the population level. The coefficient estimates (βs) in a GEE represent the effect of the predictors averaged across all individuals with the same predictor values. As a result, the estimates for a GEE are often smaller (in absolute terms) than those obtained from some sort of mixed model. GEEs do not use random effects, but instead model the correlation at the marginal or correlation level.

Next, I'll demonstrate GEE analyses briefly, and compare some results, across potential specifications of the correlation structure.

So we'll compare four different choices of correlation structure within each patient, just for illustration. In practice, we'd simply select an approach (most of the time, either exchangeable or AR-1) and run it…

- *Exchangeable*, which will yield a working correlation matrix between observations (at different doses) for the same id that has the same correlation between doses for all subjects and all doses. Remember that this is the same as the assumption of compound symmetry that we applied in running our repeated measures ANOVA analyses.
- *AR-1* (autoregressive, order 1), which will yield a working correlation matrix between observations (at different doses) for the same id that has stronger correlations between doses that are close to each other than for doses that are more separated.
- *Independence*, which will yield a working correlation matrix between observations (at different doses) for the same id that has zero correlation between observations for the same patient. If this was an appropriate assumption for whatever reason, you'd probably use a GLM instead of a GEE model. In the example below, the independence assumption won't change the fixed effect estimates from what we see with an exchangeable structure.
- *Unstructured*, which will yield a working correlation matrix between observations (at different doses) for the same id that can take on any values at all. Often, R will have trouble (at least with the naïve approach to estimating standard errors) using this approach to actually fit data. Not assuming structure seems like an attractive option, but it really costs you in terms of power.

Returning to our Isoproterenol Example

In our isoproterenol example, we'll look at the `dose` and `race` fixed effects (and their interaction) and the clusters of patient responses are defined by the `id` variable. We have a continuous outcome, so we'll use a `Gaussian` family.

GEE for our Isoproterenol Responses: Exchangeable Correlation Structure

```
library(gee)

### Assuming an exchangeable correlation structure for each patient
gg.ex <- gee(fbf ~ dose*race, id=id, family=gaussian,
      corstr="exchangeable", data=isop.long1web)
summary(gg.ex)
GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Identity
 Variance to Mean Relation: Gaussian
 Correlation Structure:     Exchangeable

Call: gee(formula = fbf ~ dose * race, id = id, data = isop.long1web,
    family = gaussian, corstr = "exchangeable")

Summary of Residuals:
        Min          1Q      Median          3Q         Max
-15.0119466  -3.0876955  -0.7997657   1.6701818  25.1002343

Coefficients:
                 Estimate  Naive S.E.   Naive z Robust S.E. Robust z
(Intercept)    3.14518969 1.474574700 2.132947 0.324067395 9.705357
dose           0.01523086 0.003655278 4.166812 0.004289151 3.551019
racewhite      2.92726740 1.950678973 1.500640 0.965466600 3.031972
dose:racewhite 0.03361787 0.004835478 6.952336 0.006450897 5.211347

Estimated Scale Parameter:  32.89985
Number of Iterations:  1

Working Correlation
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 1.0000000 0.4503899 0.4503899 0.4503899 0.4503899 0.4503899 0.4503899
[2,] 0.4503899 1.0000000 0.4503899 0.4503899 0.4503899 0.4503899 0.4503899
[3,] 0.4503899 0.4503899 1.0000000 0.4503899 0.4503899 0.4503899 0.4503899
[4,] 0.4503899 0.4503899 0.4503899 1.0000000 0.4503899 0.4503899 0.4503899
[5,] 0.4503899 0.4503899 0.4503899 0.4503899 1.0000000 0.4503899 0.4503899
[6,] 0.4503899 0.4503899 0.4503899 0.4503899 0.4503899 1.0000000 0.4503899
[7,] 0.4503899 0.4503899 0.4503899 0.4503899 0.4503899 0.4503899 1.0000000
```

In this model, we see a significant effect[4] for dose (Z = 3.55, *p* = 0.00039), race (Z = 3.03, *p* = 0.00244) and their interaction (Z = 5.21, *p* = 1.8 x 10^{-7}) using the robust Z scores.

[4] Recall that the two-tailed p value for Z = 5.21, for example, can be obtained with `2*(1-pnorm(5.21))`

GEE for our Isoproterenol Responses: AR-1 Correlation Structure

```
### GEE model, assuming an order 1 autoregressive correlation
      structure for each patient
gg.ar1 <- gee(fbf ~ dose*race, id=id, family=gaussian, corstr="AR-M",
      Mv=1, data=isop.long1web)
summary(gg.ar1)
 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Identity
 Variance to Mean Relation: Gaussian
 Correlation Structure:     AR-M , M = 1

Call:
gee(formula = fbf ~ dose * race, id = id, data = isop.long1web,
    family = gaussian, corstr = "AR-M", Mv = 1)

Summary of Residuals:
        Min          1Q      Median          3Q         Max
-11.2421338  -2.5190368  -0.4308791   2.3468304  26.3593096


Coefficients:
                Estimate  Naive S.E.  Naive z Robust S.E. Robust z
(Intercept)    3.01091335 1.784670651 1.687097 0.328792432 9.157490
dose           0.01199657 0.004927176 2.434776 0.004605927 2.604595
racewhite      3.30891104 2.360897357 1.401548 1.002379459 3.301056
dose:racewhite 0.02680920 0.006518042 4.113076 0.005642987 4.750888

Estimated Scale Parameter:  35.1264
Number of Iterations:  4

Working Correlation
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 1.0000000 0.8233765 0.6779488 0.5582071 0.4596146 0.3784359 0.3115952
[2,] 0.8233765 1.0000000 0.8233765 0.6779488 0.5582071 0.4596146 0.3784359
[3,] 0.6779488 0.8233765 1.0000000 0.8233765 0.6779488 0.5582071 0.4596146
[4,] 0.5582071 0.6779488 0.8233765 1.0000000 0.8233765 0.6779488 0.5582071
[5,] 0.4596146 0.5582071 0.6779488 0.8233765 1.0000000 0.8233765 0.6779488
[6,] 0.3784359 0.4596146 0.5582071 0.6779488 0.8233765 1.0000000 0.8233765
[7,] 0.3115952 0.3784359 0.4596146 0.5582071 0.6779488 0.8233765 1.0000000
```

In this model, we again see a significant effect for dose (Z = 2.60, p = 0.0093), race (Z = 3.30, p = 0.00097) and their interaction (Z = 4.75, p = 2.0 x 10^{-6}) via the robust Z scores.

GEE for our Isoproterenol Responses: Independence Assumed

As mentioned above, assuming independence is rarely our best choice for GEE.

```
### GEE model, assuming independence across observations in each patient
gg.ind <- gee(fbf ~ dose*race, id=id, family=gaussian,
      corstr="independence", data=isop.long1web)
summary(gg.ind)

 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Identity
 Variance to Mean Relation: Gaussian
 Correlation Structure:     Independent

Call:
gee(formula = fbf ~ dose * race, id = id, data = isop.long1web,
    family = gaussian, corstr = "independence")

Summary of Residuals:
        Min          1Q      Median          3Q         Max
-15.0119466  -3.0876955  -0.7997657   1.6701818  25.1002343


Coefficients:
                 Estimate  Naive S.E.  Naive z Robust S.E. Robust z
(Intercept)    3.14518969 0.980099382 3.209052 0.324067395 9.705357
dose           0.01523086 0.004930524 3.089095 0.004289151 3.551019
racewhite      2.92726740 1.296549612 2.257737 0.965466600 3.031972
dose:racewhite 0.03361787 0.006522470 5.154162 0.006450897 5.211347

Estimated Scale Parameter:  32.89985
Number of Iterations:  1

Working Correlation
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    1    0    0    0    0    0    0
[2,]    0    1    0    0    0    0    0
[3,]    0    0    1    0    0    0    0
[4,]    0    0    0    1    0    0    0
[5,]    0    0    0    0    1    0    0
[6,]    0    0    0    0    0    1    0
[7,]    0    0    0    0    0    0    1
```

The robust Z scores for the fixed effects in this model are identical to those we obtained with the exchangeable correlation structure.

GEE for our Isoproterenol Responses: Unstructured Correlations Permitted

```
### GEE model, assuming an "unstructured" correlation structure for
      each patient
gg.uns <- gee(fbf ~ dose*race, id=id, family=gaussian,
corstr="unstructured", data=isop.long1web)
Warning message:
  Working correlation estimate not positive definite
summary(gg.uns)
 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Identity
 Variance to Mean Relation: Gaussian
 Correlation Structure:     Unstructured

Call:
gee(formula = fbf ~ dose * race, id = id, data = isop.long1web,
    family = gaussian, corstr = "unstructured")

Summary of Residuals:
       Min          1Q      Median          3Q         Max
-6.0528550 -1.6103828  0.1923158  5.1821134 30.4122519


Coefficients:
                  Estimate Naive S.E.  Naive z Robust S.E. Robust z
(Intercept)    3.276165993   1.234475 2.653893 0.365423280 8.965400
dose           0.008438394        NaN      NaN 0.004874824 1.731015
racewhite      3.776688984   1.633058 2.312649 1.123543620 3.361408
dose:racewhite 0.006148839        NaN      NaN 0.005535934 1.110714

Estimated Scale Parameter:  56.11716
Number of Iterations:  9

Working Correlation
          [,1]      [,2]     [,3]     [,4]     [,5]       [,6]       [,7]
[1,]  1.000000  0.210168 0.069828 -0.23282 -0.29558 -0.3342115 -0.4211242
[2,]  0.210168  1.000000 0.088471 -0.13688 -0.18366 -0.2120880 -0.2991987
[3,]  0.069829  0.088471 1.000000  0.11879  0.13699  0.1196425  0.1060706
[4,] -0.232818 -0.136881 0.118787  1.00000  1.23635  1.2750527  1.4157720
[5,] -0.295577 -0.183659 0.136985  1.23635  1.00000  1.5113303  1.6761013
[6,] -0.334211 -0.212088 0.119643  1.27505  1.51133  1.0000000  1.8304846
[7,] -0.421124 -0.299199 0.106071  1.41577  1.67610  1.8304846  1.0000000
```

Without an assumed correlation structure, we obtain different results. Our only significant effect is for race (Z = 3.36, *p* = 0.00078), but neither dose (Z = 1.73, *p* = 0.084) nor the interaction (Z = 1.11, *p* = 0.27) is statistically significant.

GEE for our Vaccine Responses: Exchangeable Correlation Structure

Consider a GEE for the vaccine data. We'll assume an exchangeable correlation structure.

```
gg.1 <- gee(rating ~ vaccine * month, id=id, family=gaussian,
      corstr="exchangeable", data=arthr)
summary(gg.1)
GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
Model:   Link:                      Identity
         Variance to Mean Relation: Gaussian
         Correlation Structure:     Exchangeable
Call: gee(formula = rating ~ vaccine * month, id = id, data = arthr,
    family = gaussian, corstr = "exchangeable")

Summary of Residuals:    Min     1Q Median     3Q    Max
                      -2.750 -1.000  0.125  1.250  2.250
Coefficients:
                      Estimate Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept)               5.75  0.8164966  7.0422830   0.7395100  7.7754191
vaccinePlacebo            1.25  1.1547005  1.0825318   0.9601432  1.3018891
month2                   -3.00  0.9166667 -3.2727273   0.3535534 -8.4852814
month3                   -3.50  0.9166667 -3.8181818   1.0307764 -3.3954988
vaccinePlacebo:month2    -0.50  1.2963624 -0.3856946   0.8291562 -0.6030227
vaccinePlacebo:month3     0.25  1.2963624  0.1928473   1.3635890  0.1833397

Estimated Scale Parameter:  2.666667
Number of Iterations:  1

Working Correlation
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.3697917 0.3697917
[2,] 0.3697917 1.0000000 0.3697917
[3,] 0.3697917 0.3697917 1.0000000
```

Compare the GEE-based fixed effects estimates to those of our random intercept model, fitted previously, and repeated below. In the GEE, we see a bigger apparent Month 2 effect, in particular, after we apply the sandwich estimates for the robust standard errors. Note that the actual point estimates and naïve standard errors match precisely.

```
                      Value Std.Error DF   t-value p-value
(Intercept)            5.75 0.8164964 12  7.042285  0.0000
vaccinePlacebo         1.25 1.1547003  6  1.082532  0.3206
month2                -3.00 0.9166670 12 -3.272726  0.0067
month3                -3.50 0.9166670 12 -3.818181  0.0024
vaccinePlacebo:month2 -0.50 1.2963628 12 -0.385694  0.7065
vaccinePlacebo:month3  0.25 1.2963628 12  0.192847  0.8503
```

One Last GEE Example: the Ohio Children Wheezing Status data

R's faraway library contains a data frame called ohio, which has 2148 rows and 4 columns[5]. The data are part of a longitudinal study on the health effects of air pollution. Here, we are looking at 536 children from Steubenville, where children were in the study for four years (ages 7-10), and the response was whether they wheezed or not. The variables are:

- resp: an indicator of wheeze status (1=yes, 0=no)
- id: a numeric vector for subject id
- age: a numeric vector of age – 9 years, so -2 is 7 years old, 0 is 9 years old, etc.
- smoke: indicator of maternal smoking in the study's first year (1 = yes, 0 = no)

Let's fit a GEE model and assess the effects of age and maternal smoking on wheezing.

```
gg.oh1 <- gee(resp ~ age + smoke, id=id, family=binomial, corstr="AR-
      M", Mv=1, data=ohio)
summary(gg.oh1)
 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
Model:  Link:                      Logit
        Variance to Mean Relation: Binomial
        Correlation Structure:     AR-M , M = 1
Call: gee(formula = resp ~ age + smoke, id = id, data = ohio, family =
binomial, corstr = "AR-M", Mv = 1)

Summary of Residuals:
       Min         1Q     Median         3Q        Max
-0.1939063 -0.1586034 -0.1438830 -0.1178543  0.8821457

Coefficients:
              Estimate Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept) -1.8981575 0.10961955 -17.315867  0.11467812 -16.552045
age         -0.1147505 0.05586065  -2.054229  0.04493528  -2.553685
smoke        0.2438312 0.16620395   1.467060  0.17983107   1.355890

Estimated Scale Parameter:  1.016977
Number of Iterations:  3

Working Correlation
           [,1]      [,2]      [,3]       [,4]
[1,] 1.00000000 0.3989964 0.1591981 0.06351949
[2,] 0.39899643 1.0000000 0.3989964 0.15919815
[3,] 0.15919815 0.3989964 1.0000000 0.39899643
[4,] 0.06351949 0.1591981 0.3989964 1.00000000
```

[5] See ?ohio after loading the faraway library for more details.

Now, let's answer a few additional questions.

1. In this model (gg.oh1), what is the predicted probability that a 7-year-old with a smoking mother, wheezes?

log odds of wheezing = -1.898 – 0.115 (-2) + 0.244 (1) = -1.424

so, the odds of wheezing are exp(-1.424) = 0.241

and so the probability of wheezing is 0.241 / (1 + 0.241) = 0.194

2. In this model (gg.oh1), what indicates that a child who already wheezes is likely to continue to wheeze?

The correlation between a child's wheezing status in one year and their status in the next year is 0.399. If we used an exchangeable correlation structure instead, we would likely come to a similar conclusion, assuming a positive correlation.

Note: When we actually run the exchangeable correlation structure model, we get the following working correlation matrix.

```
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.3541398 0.3541398 0.3541398
[2,] 0.3541398 1.0000000 0.3541398 0.3541398
[3,] 0.3541398 0.3541398 1.0000000 0.3541398
[4,] 0.3541398 0.3541398 0.3541398 1.0000000
```

The fitted model under the exchangeable assumption turns out to be:

```
Coefficients:
              Estimate Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept) -1.8804277 0.11483941 -16.374411  0.11389291 -16.510489
age         -0.1133850 0.04354142  -2.604073  0.04385531  -2.585434
smoke        0.2650809 0.17700086   1.497625  0.17774655   1.491342
```

3. Suppose we repeat our analysis as a GLM assuming that the observations are independent, that is, that each single response value represents a different child. How would the results differ and which approach should be preferred?

Coefficients from the "independence" model follow:

```
              Estimate Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept) -1.8837347 0.08386590 -22.461271  0.11424020 -16.489245
age         -0.1134128 0.05409672  -2.096481  0.04387767  -2.584749
smoke        0.2721386 0.12350665   2.203433  0.17798185   1.529024
```

4. Suppose instead that we sum up the number of times wheezing is recorded for a child over the four measurements and model this as a function of the smoking status of the mother. What is the effect of smoking now? Which approach is preferable?

```
nohio <- reshape(ohio, idvar="id", direction="wide", timevar="age",
      v.names="resp")
nohio <- data.frame(smoke=nohio$smoke,
      wheeze=apply(nohio[,3:6],1,sum))
head(nohio,3)
   smoke wheeze
1      0      0
5      0      0
9      0      0

table(nohio$smoke, nohio$wheeze)
      0   1   2   3   4
  0 237  65  25  12  11
  1 118  32  19  11   7

t.test(nohio$wheeze ~ nohio$smoke)
Welch Two Sample t-test
data:  nohio$wheeze by nohio$smoke
t = -1.4854, df = 345.046, p-value = 0.1384
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: -0.33326665  0.04648284
sample estimates: mean in group 0 mean in group 1
                        0.5571429       0.7005348
```

If you don't care for this t test, we could instead consider running a linear model, an ordinal logistic regression, or perhaps a Poisson regression, as shown below.

```
summary(glm(wheeze ~ smoke, data=nohio, family=poisson))

Call: glm(formula = wheeze ~ smoke, family = poisson, data = nohio)

Deviance Residuals:     Min       1Q   Median       3Q      Max
                    -1.1837  -1.0556  -1.0556   0.5331   2.9806

Coefficients: Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.58493    0.07161  -8.168 3.13e-16 ***
smoke          0.22902    0.11297   2.027   0.0426 *

(Dispersion parameter for poisson family taken to be 1)
    Null deviance: 798.64  on 536  degrees of freedom
Residual deviance: 794.60  on 535  degrees of freedom
AIC: 1235.2
Number of Fisher Scoring iterations: 6
```