ANOVA models

In previous slides, we discussed the use of categorical variables in multivariate regression. Often, these are encoded as indicator columns in the design matrix.

In [1]:

```
options(repr.plot.width=4, repr.plot.height=4)
```

In [2]:

```
url = 'http://stats191.stanford.edu/data/salary.table'
salary.table = read.table(url, header=T)
salary.table$E = factor(salary.table$E)
salary.table$M = factor(salary.table$M)
salary.lm = lm(S ~ X + E + M, salary.table)
head(model.matrix(salary.lm))
```

	(Intercept)	X	E2	E3	M1
1	1	1	0	0	1
2	1	1	0	1	0
3	1	1	0	1	1
4	1	1	1	0	0
5	1	1	0	1	0
6	1	2	1	0	1

ANOVA models

- Often, especially in experimental settings, we record *only* categorical variables.
- Such models are often referred to *ANOVA (Analysis of Variance)* models.
- These are generalizations of our favorite example, the two sample t-test.

Example: recovery time

- Suppose we want to understand the relationship between recovery time after surgery based on an patient's prior fitness.
- We group patients into three fitness levels: below average, average, above average.
- If you are in better shape before surgery, does it take less time to recover?

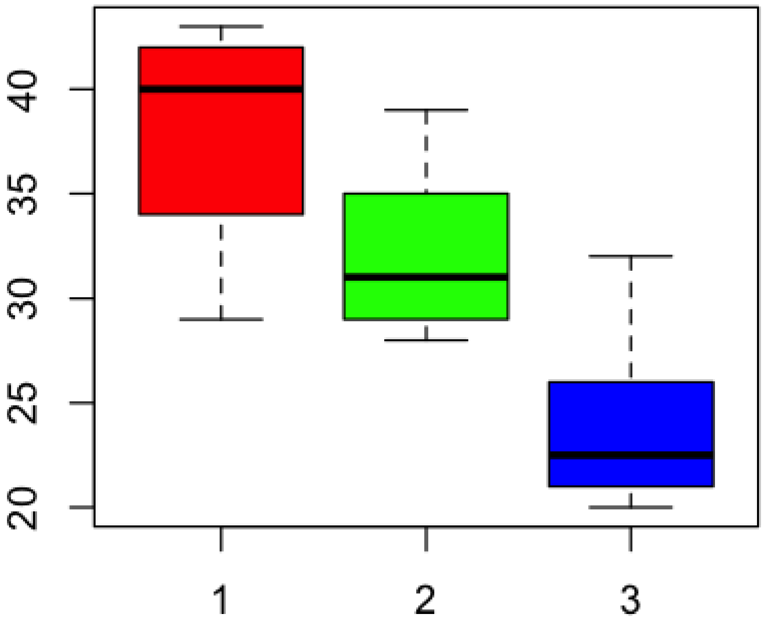
In [3]:

```
url = 'http://stats191.stanford.edu/data/rehab.csv'
rehab.table = read.table(url, header=T, sep=',')
rehab.table$Fitness <- factor(rehab.table$Fitness)
head(rehab.table)
```

Fitness	Time
1	29
1	42
1	38
1	40
1	43
1	40

In [4]:

```
attach(rehab.table)
boxplot(Time ~ Fitness, col=c('red','green','blue'))
```



One-way ANOVA

- First generalization of two sample t-test: more than two groups.
- Observations are broken up into r groups with $n_i, 1 \leq i \leq r$ observations per group.
- Model:

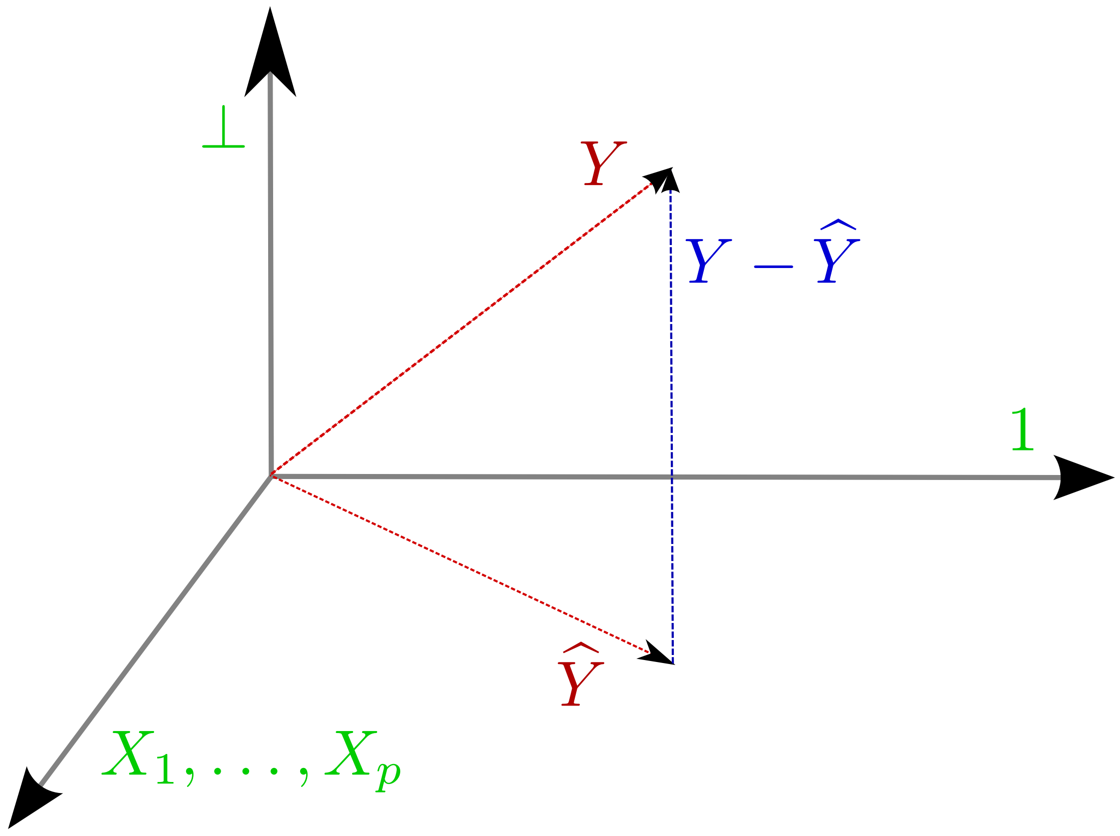
$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij}, \qquad \varepsilon_{ij} \overset{IID}{\sim} N(0, \sigma^2).$$

- Constraint: $\sum_{i=1}^r \alpha_i = 0$. This constraint is needed for “identifiability”. This is “equivalent” to only adding $r-1$ columns to the design matrix for this qualitative variable.

One-way ANOVA

- This is not the same *parameterization* we get when only adding $r-1$ 0-1 columns, but it gives the same *model*.
- The estimates of α can be obtained from the estimates of β using R's default parameters.
- For a more detailed exploration into R's creation of design matrices, try reading the following tutorial on design matrices (http://nbviewer.ipython.org/github/fperez/nipy-notebooks/blob/master/exploring_r_formula.ipynb).

Remember, it's still a model (i.e. a plane)



Fitting the model

- Model is easy to fit:
$$\widehat{Y}_{ij} = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij} = \overline{Y}_{i\cdot}$$
If observation is in i-th group: predicted mean is just the sample mean of observations in i-th group.
- Simplest question: is there any group (main) effect?
$$H_0 : \alpha_1 = \cdots = \alpha_r = 0?$$
- Test is based on F-test with full model vs. reduced model. Reduced model just has an intercept.
- Other questions: is the effect the same in groups 1 and 2?
$$H_0 : \alpha_1 = \alpha_2?$$

In [5]:

```
rehab.lm <- lm(Time ~ Fitness)
summary(rehab.lm)
```

```
Call:
lm(formula = Time ~ Fitness)

Residuals:
   Min     1Q Median     3Q    Max 
  -9.0   -3.0   -0.5    3.0    8.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   38.000      1.574  24.149  < 2e-16 ***
Fitness2      -6.000      2.111  -2.842  0.00976 ** 
Fitness3     -14.000      2.404  -5.824 8.81e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.451 on 21 degrees of freedom
Multiple R-squared:  0.6176,	Adjusted R-squared:  0.5812 
F-statistic: 16.96 on 2 and 21 DF,  p-value: 4.129e-05
```

In [6]:

```
print(predict(rehab.lm, list(Fitness=factor(c(1,2,3)))))
c(mean(Time[Fitness == 1]), mean(Time[Fitness == 2]), mean(Time[Fitness == 3]))
```

```
 1  2  3 
38 32 24 
```

38 32 24

Recall that the rows of the `Coefficients` table above do not correspond to the α parameter. For one thing, we would see three α's and their sum would have to be equal to 0.

Also, the design matrix is the indicator coding we saw last time.

In [7]:

```
head(model.matrix(rehab.lm))
```

	(Intercept)	Fitness2	Fitness3
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0

- There are ways to get *different* design matrices by using the `contrasts` argument. This is a bit above our pay grade at the moment.
- Upon inspection of the design matrix above, we see that the `(Intercept)` coefficient corresponds to the mean in `Fitness==1`, while `Fitness==2` coefficient corresponds to the difference between the groups `Fitness==2` and `Fitness==1`.

ANOVA table

Much of the information in an ANOVA model is contained in the ANOVA table.

Source	SS	df	$\mathbb{E}(MS)$
Treatment	$SSTR = \sum_{i=1}^r n_i \left(\overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot}\right)^2$	r-1	$\sigma^2 + \frac{\sum_{i=1}^r n_i \alpha_i^2}{r-1}$
Error	$SSE = \sum_{i=1}^r \sum_{j=1}^{n_i} (Y_{ij} - \overline{Y}_{i\cdot})^2$	$\sum_{i=1}^r (n_i - 1)$	σ^2

In [8]:

```
anova(rehab.lm)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Fitness	2	672	336.00000	16.96154	4.129301e-05
Residuals	21	416	19.80952	NA	NA

- Note that $MSTR$ measures “variability” of the “cell” means. If there is a group effect we expect this to be large relative to MSE.
- We see that under $H_0 : \alpha_1 = \dots = \alpha_r = 0$, the expected value of $MSTR$ and MSE is σ^2. This tells us how to test H_0 using ratio of mean squares, i.e. an F test.

Testing for any main effect

- Rows in the ANOVA table are, in general, independent.
- Therefore, under H_0
  $$F = \frac{MSTR}{MSE} = \frac{\frac{SSTR}{df_{TR}}}{\frac{SSE}{df_E}} \sim F_{df_{TR}, df_E}$$
  the degrees of freedom come from the df column in previous table.
- Reject H_0 at level α if $F > F_{1-\alpha, df_{TR}, df_E}$.

```
In [9]:
```

```
F = 336.00 / 19.81
pval = 1 - pf(F, 2, 21)
print(data.frame(F,pval))
```

```
         F         pval
1 16.96113 4.129945e-05
```

Inference for linear combinations

- Suppose we want to ``infer'' something about
  $$\sum_{i=1}^r a_i \mu_i$$
  where $\mu_i = \mu + \alpha_i$ is the mean in the i-th group. For example:
  $$H_0 : \mu_1 - \mu_2 = 0 \qquad (\text{same as } H_0 : \alpha_1 - \alpha_2 = 0)?$$
- For example:

  Is there a difference between below average and average groups in terms of rehab time?

Inference for linear combinations

- We need to know
  $$\text{Var}\left(\sum_{i=1}^r a_i \overline{Y}_{i\cdot}\right) = \sigma^2 \sum_{i=1}^r \frac{a_i^2}{n_i}.$$
- After this, the usual confidence intervals and t-tests apply.

In [10]:

```
head(model.matrix(rehab.lm))
```

	(Intercept)	Fitness2	Fitness3
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0

This means that the coefficient Fitness2 is the estimated difference between the two groups.

In [11]:

```
detach(rehab.table)
```

Two-way ANOVA

Often, we will have more than one variable we are changing.

Example

After kidney failure, we suppose that the time of stay in hospital depends on weight gain between treatments and duration of treatment.

We will model the `log` number of days as a function of the other two factors.

Variable	Description
Days	Duration of hospital stay
Weight	How much weight is gained?
Duration	How long under treatment for kidney problems? (two levels)

In [12]:

```
url = 'http://statweb.stanford.edu/~jtaylo/stats191/data/kidney.table'
kidney.table = read.table(url, header=T)
kidney.table$D = factor(kidney.table$Duration)
kidney.table$W = factor(kidney.table$Weight)
kidney.table$logDays = log(kidney.table$Days + 1)
attach(kidney.table)
head(kidney.table)
```

Days	Duration	Weight	ID	D	W	logDays
0	1	1	1	1	1	0.0000000
2	1	1	2	1	1	1.0986123
1	1	1	3	1	1	0.6931472
3	1	1	4	1	1	1.3862944
0	1	1	5	1	1	0.0000000
2	1	1	6	1	1	1.0986123

Two-way ANOVA model

- Second generalization of t-test: more than one grouping variable.
- Two-way ANOVA model:
  - r groups in first factor
  - m groups in second factor
  - n_{ij} in each combination of factor variables.
- Model:
$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}, \qquad \varepsilon_{ijk} \sim N(0, \sigma^2).$$
- In kidney example, $r = 3$ (weight gain), $m = 2$ (duration of treatment), $n_{ij} = 10$ for all (i, j).

Questions of interest

Two-way ANOVA: main questions of interest

- Are there main effects for the grouping variables?
$$H_0 : \alpha_1 = \dots = \alpha_r = 0, \qquad H_0 : \beta_1 = \dots = \beta_m = 0.$$
- Are there interaction effects:
$$H_0 : (\alpha\beta)_{ij} = 0, 1 \leq i \leq r, 1 \leq j \leq m.$$

Interactions between factors

We've already seen these interactions in the IT salary example.

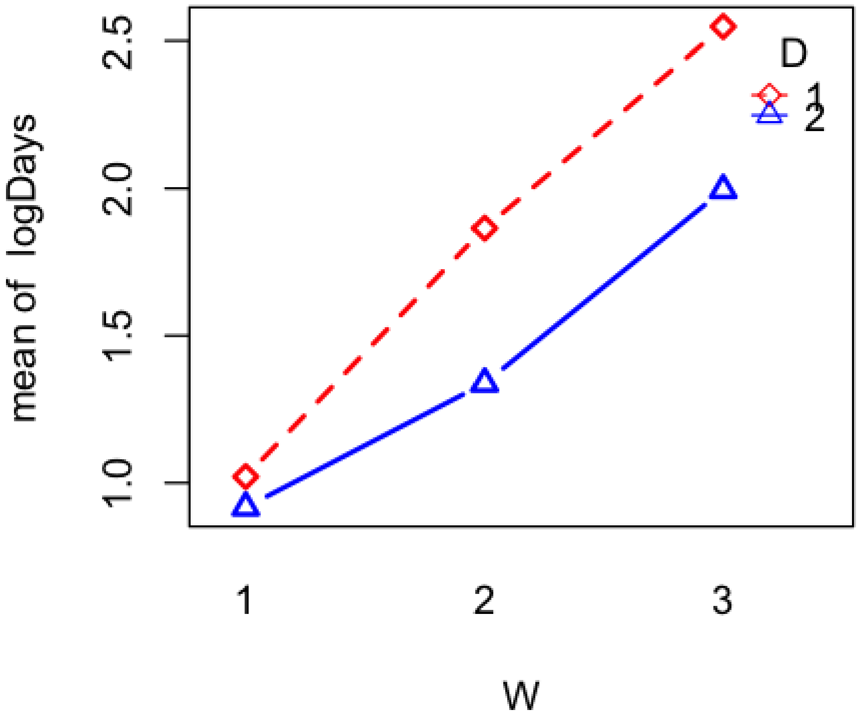
- An *additive model* says that the effects of the two factors occur additively -- such a model has no interactions.
- An *interaction* is present whenever the additive model does not hold.

Interaction plot

When these broken lines are not parallel, there is evidence of an interaction. The one thing missing from this plot are errorbars. The above broken lines are clearly not parallel but there is measurement error. If the error bars were large then we might consider there to be no interaction, otherwise we might.

In [13]:

```
interaction.plot(W, D, logDays, type='b', col=c('red',
                   'blue'), lwd=2, pch=c(23,24))
```



Parameterization

- Many constraints are needed, again for identifiability. Let's not worry too much about the details
- Constraints:
  - $\sum_{i=1}^r \alpha_i = 0$
  - $\sum_{j=1}^m \beta_j = 0$
  - $\sum_{j=1}^m (\alpha\beta)_{ij} = 0, 1 \leq i \leq r$
  - $\sum_{i=1}^r (\alpha\beta)_{ij} = 0, 1 \leq j \leq m.$
- We should convince ourselves that we know have exactly $r * m$ free parameters.

Fitting the model

- Easy to fit when $n_{ij} = n$ (balanced)

$$\widehat{Y}_{ijk} = \overline{Y}_{ij\cdot} = \frac{1}{n} \sum_{k=1}^{n} Y_{ijk}.$$

- Inference for combinations

$$\text{Var}\left(\sum_{i=1}^{r} \sum_{j=1}^{m} a_{ij} \overline{Y}_{ij\cdot}\right) = \frac{\sigma^2}{n} \cdot \sum_{i=1}^{r} \sum_{j=1}^{m} a_{ij}^2.$$

- Usual t-tests, confidence intervals.

In [14]:

```
kidney.lm = lm(logDays ~ D*W, contrasts=list(D='contr.sum', W='contr.sum'))
summary(kidney.lm)
```

```
Call:
lm(formula = logDays ~ D * W, contrasts = list(D = "contr.sum", 
    W = "contr.sum"))

Residuals:
     Min       1Q   Median       3Q      Max 
-1.33772 -0.51121  0.06302  0.62926  1.17950 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.61401    0.09459  17.063  < 2e-16 ***
D1           0.19747    0.09459   2.088   0.0416 *  
W1          -0.64496    0.13377  -4.821  1.2e-05 ***
W2          -0.01264    0.13377  -0.095   0.9251    
D1:W1       -0.14537    0.13377  -1.087   0.2820    
D1:W2        0.06618    0.13377   0.495   0.6228    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7327 on 54 degrees of freedom
Multiple R-squared:  0.4076,	Adjusted R-squared:  0.3528 
F-statistic: 7.431 on 5 and 54 DF,  p-value: 2.301e-05
```

Example

- Suppose we are interested in comparing the mean in $(D = 1, W = 3)$ and $(D = 2, W = 2)$ groups. The difference is

$$E(\bar{Y}_{13\cdot} - \bar{Y}_{22\cdot})$$

- By independence, its variance is

$$\text{Var}(\bar{Y}_{13\cdot}) + \text{Var}(\bar{Y}_{22\cdot}) = \frac{2\sigma^2}{n}.$$

In [15]:

```
estimates = predict(kidney.lm, list(D=factor(c(1,2)), W=factor(c(3,2))))
print(estimates)
sigma.hat = 0.7327 # from table above
n = 10 # ten observations per group
fit = estimates[1] - estimates[2]
upper = fit + qt(0.975, 54) * sqrt(2 * sigma.hat^2 / n)
lower = fit - qt(0.975 ,54) * sqrt(2 * sigma.hat^2 / n)
data.frame(fit,lower,upper)
```

```
       1        2 
2.548271 1.337719 
```

fit	lower	upper
1.210551	0.5536058	1.867497

In [16]:

```
head(model.matrix(kidney.lm))
```

	(Intercept)	D1	W1	W2	D1:W1	D1:W2
1	1	1	1	0	1	0
2	1	1	1	0	1	0
3	1	1	1	0	1	0
4	1	1	1	0	1	0
5	1	1	1	0	1	0
6	1	1	1	0	1	0

Finding predicted values

The most direct way to compute predicted values is using the `predict` function

In [17]:

```
predict(kidney.lm, list(D=factor(1),W=factor(1)), interval='confidence')
```

	fit	lwr	upr
1	1.021156	0.5566306	1.485681

ANOVA table

In the balanced case, everything can again be summarized from the ANOVA table

Source	SS	df	$\mathbb{E}(MS)$
A	$SSA = nm\sum_{i=1}^r \left(\overline{Y}_{i\cdot\cdot} - \overline{Y}_{\cdot\cdot\cdot}\right)^2$	r-1	$\sigma^2 + nm\frac{\sum_{i=1}^r \alpha_i^2}{r-1}$
B	$SSB = nr\sum_{j=1}^m \left(\overline{Y}_{\cdot j\cdot} - \overline{Y}_{\cdot\cdot\cdot}\right)^2$	m-1	$\sigma^2 + nr\frac{\sum_{j=1}^m \beta_j^2}{m-1}$
A:B	$SSAB = n\sum_{i=1}^r \sum_{j=1}^m \left(\overline{Y}_{ij\cdot} - \overline{Y}_{i\cdot\cdot} - \overline{Y}_{\cdot j\cdot} + \overline{Y}_{\cdot\cdot\cdot}\right)^2$	(m-1)(r-1)	$\sigma^2 + n\frac{\sum_{i=1}^r \sum_{j=1}^m (\alpha\beta)_{ij}^2}{(r-1)(m-1)}$
Error	$SSE = \sum_{i=1}^r \sum_{j=1}^m \sum_{k=1}^n (Y_{ijk} - \overline{Y}_{ij\cdot})^2$	(n-1)mr	σ^2

Tests using the ANOVA table

- Rows of the ANOVA table can be used to test various of the hypotheses we started out with.
- For instance, we see that under $H_0 : (\alpha\beta)_{ij} = 0, \forall i, j$ the expected value of $SSAB$ and SSE is σ^2 – use these for an F-test testing for an interaction.
- Under H_0

$$F = \frac{MSAB}{MSE} = \frac{\frac{SSAB}{(m-1)(r-1)}}{\frac{SSE}{(n-1)mr}} \sim F_{(m-1)(r-1),(n-1)mr}$$

In [18]:

```
anova(kidney.lm)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
D	1	2.3396928	2.3396928	4.3582928	4.156170e-02
W	2	16.9712909	8.4856454	15.8067448	3.944502e-06
D:W	2	0.6356584	0.3178292	0.5920404	5.567479e-01
Residuals	54	28.9891979	0.5368370	NA	NA

We can also test for interactions using our usual approach

In [19]:

```
anova(lm(logDays ~ D + W, kidney.table), kidney.lm)
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
56	29.62486	NA	NA	NA	NA
54	28.98920	2	0.6356584	0.5920404	0.5567479

Some caveats about `R` formulae

While we see that it is straightforward to form the interactions test using our usual `anova` function approach, we generally *cannot* test for main effects by this approach.

In [20]:

```
lm_no_main_Weight = lm(logDays ~ D + W:D)
anova(lm_no_main_Weight, kidney.lm)
anova(lm(logDays ~ D), lm(logDays ~ D + W))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
54	28.9892	NA	NA	NA	NA
54	28.9892	0	7.105427e-15	NA	NA

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
58	46.59615	NA	NA	NA	NA
56	29.62486	2	16.97129	16.04045	3.108672e-06

In fact, these models are identical in terms of their *planes* or their *fitted values*. What has happened is that `R` has formed a different design matrix using its rules for `formula` objects.

In [21]:

```
lm1 = lm(logDays ~ D + W:D)
lm2 = lm(logDays ~ D + W:D + W)
sum((resid(lm1) - resid(lm2))^2)
```

```
3.53473626413167e-29
```

ANOVA tables in general

So far, we have used `anova` to compare two models. In this section, we produced tables for just 1 model. This also works for *any* regression model, though we have to be a little careful about interpretation.

Let's revisit the job aptitude test data from last section.

In [22]:

```
url = 'http://stats191.stanford.edu/data/jobtest.table'
jobtest.table <- read.table(url, header=T)
jobtest.table$MINORITY <- factor(jobtest.table$MINORITY)
jobtest.lm = lm(JPERF ~ TEST:MINORITY + MINORITY + TEST, jobtest.table)
summary(jobtest.lm)
```

```
Call:
lm(formula = JPERF ~ TEST:MINORITY + MINORITY + TEST, data = jobtest.tabl
e)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0734 -1.0594 -0.2548  1.2830  2.1980 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)  
(Intercept)      2.0103     1.0501   1.914   0.0736 .
MINORITY1       -1.9132     1.5403  -1.242   0.2321  
TEST             1.3134     0.6704   1.959   0.0677 .
TEST:MINORITY1   1.9975     0.9544   2.093   0.0527 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.407 on 16 degrees of freedom
Multiple R-squared:  0.6643,	Adjusted R-squared:  0.6013 
F-statistic: 10.55 on 3 and 16 DF,  p-value: 0.0004511
```

Now, let's look at the anova output. We'll see the results don't match.

In [23]:

```
anova(jobtest.lm)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
MINORITY	1	8.051805	8.051805	4.069719	0.0607562318
TEST	1	45.917904	45.917904	23.208829	0.0001894279
TEST:MINORITY	1	8.666073	8.666073	4.380196	0.0526501180
Residuals	16	31.655473	1.978467	NA	NA

The difference is how the `Sum Sq` columns is created. In the anova output, terms in the response are added sequentially.

We can see this by comparing these two models directly. The `F` statistic doesn't agree because the `MSE` above is computed in the *fullest* model, but the `Sum of Sq` is correct.

```
In [24]:
```

```
anova(lm(JPERF ~ TEST, jobtest.table),
      lm(JPERF ~ TEST + MINORITY, jobtest.table))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
18	45.56830	NA	NA	NA	NA
17	40.32155	1	5.246751	2.212087	0.1552463

Similarly, the first `Sum Sq` in `anova` can be found by:

```
In [25]:
```

```
anova(lm(JPERF ~ 1, jobtest.table), lm(JPERF ~ TEST, jobtest.table))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
19	94.29126	NA	NA	NA	NA
18	45.56830	1	48.72296	19.24613	0.0003555104

There are ways to produce an *ANOVA* table whose p-values agree with `summary`. This is done by an ANOVA table that uses Type-III sum of squares.

```
In [26]:
```

```
library(car)
Anova(jobtest.lm, type=3)
```

```
Loading required package: carData
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	7.250560	1	3.664736	0.07363289
MINORITY	3.052180	1	1.542699	0.23211490
TEST	7.594407	1	3.838531	0.06774914
TEST:MINORITY	8.666073	1	4.380196	0.05265012
Residuals	31.655473	16	NA	NA

In [27]:

```
summary(jobtest.lm)
```

```
Call:
lm(formula = JPERF ~ TEST:MINORITY + MINORITY + TEST, data = jobtest.tabl
e)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0734 -1.0594 -0.2548  1.2830  2.1980 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)  
(Intercept)      2.0103     1.0501   1.914   0.0736 .
MINORITY1       -1.9132     1.5403  -1.242   0.2321  
TEST             1.3134     0.6704   1.959   0.0677 .
TEST:MINORITY1   1.9975     0.9544   2.093   0.0527 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.407 on 16 degrees of freedom
Multiple R-squared:  0.6643,    Adjusted R-squared:  0.6013 
F-statistic: 10.55 on 3 and 16 DF,  p-value: 0.0004511
```

Fixed and random effects

- In kidney & rehab examples, the categorical variables are well-defined categories: below average fitness, long duration, etc.
- In some designs, the categorical variable is “subject”.
- Simplest example: repeated measures, where more than one (identical) measurement is taken on the same individual.
- In this case, the “group” effect α_i is best thought of as random because we only sample a subset of the entire population.

When to use random effects?

- A “group” effect is random if we can think of the levels we observe in that group to be samples from a larger population.
- Example: if collecting data from different medical centers, “center” might be thought of as random.
- Example: if surveying students on different campuses, “campus” may be a random effect.

Example: sodium content in beer

- How much sodium is there in North American beer? How much does this vary by brand?
- Observations: for 6 brands of beer, we recorded the sodium content of 8 12 ounce bottles.
- Questions of interest: what is the “grand mean” sodium content? How much variability is there from brand to brand?
- “Individuals” in this case are brands, repeated measures are the 8 bottles.

In [28]:

```
url = 'http://stats191.stanford.edu/data/sodium.table'
sodium.table = read.table(url, header=T)
sodium.table$brand = factor(sodium.table$brand)
sodium.lm = lm(sodium ~ brand, sodium.table)
anova(sodium.lm)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
brand	5	854.5292	170.9058333	238.7112	1.083746e-29
Residuals	42	30.0700	0.7159524	NA	NA

One-way random effects model

- Assuming that cell-sizes are the same, i.e. equal observations for each “subject” (brand of beer).
- Observations
$$Y_{ij} \sim \mu + \alpha_i + \varepsilon_{ij}, 1 \leq i \leq r, 1 \leq j \leq n$$
- $\varepsilon_{ij} \sim N(0, \sigma^2_{\epsilon}), 1 \leq i \leq r, 1 \leq j \leq n$
- $\alpha_i \sim N(0, \sigma^2_{\alpha}), 1 \leq i \leq r.$
- Parameters:
    - μ is the population mean;
    - σ^2_{ϵ} is the measurement variance (i.e. how variable are the readings from the machine that reads the sodium content?);
    - σ^2_{α} is the population variance (i.e. how variable is the sodium content of beer across brands).

Modelling the variance

- In random effects model, the observations are no longer independent (even if ε's are independent
$$\text{Cov}(Y_{ij}, Y_{i'j'}) = \left(\sigma^2_{\alpha} + \sigma^2_{\epsilon} \delta_{j,j'}\right) \delta_{i,i'}.$$
- In more complicated models, this makes ``maximum likelihood estimation'' more complicated: least squares is no longer the best solution.
- **It's no longer just a plane!**

- This model has a very simple model for the *mean*, it just has a slightly more complex model for the *variance*.
- Shortly we'll see other more complex models of the variance:
    - Weighted Least Squares
    - Correlated Errors

Fitting the model

The *MLE (Maximum Likelihood Estimator)* is found by minimizing

$$\begin{aligned} -2 \log \ell(\mu, \sigma^2_{\epsilon}, \sigma^2_{\alpha} | Y) = \sum_{i=1}^r & \biggl[(Y_i - \mu)^T (\sigma^2_{\epsilon} I_{n_i \times n_i} + \sigma^2_{\alpha} 1 1^T)^{-1} (Y_i - \mu) \\ & + \log\left(\det(\sigma^2_{\epsilon} I_{n_i \times n_i} + \sigma^2_{\alpha} 1 1^T)\right) \biggr]. \end{aligned}$$

THe function $\ell(\mu, \sigma^2_{\epsilon}, \sigma^2_{\alpha})$ is called the *likelihood function*.

Fitting the model in balanced design

Only one parameter in the mean function $\mu.$

- When cell sizes are the same (balanced),
$$\widehat{\mu} = \overline{Y}_{\cdot\cdot} = \frac{1}{nr}\sum_{i,j} Y_{ij}.$$
Unbalanced models: use numerical optimizer.
- This also changes estimates of σ^2_ϵ -- see ANOVA table. We might guess that $df = nr-1$ and
$$\widehat{\sigma}^2 = \frac{1}{nr-1}\sum_{i,j}(Y_{ij} - \overline{Y}_{\cdot\cdot})^2.$$
This is not correct.

ANOVA table

Again, the information needed can be summarized in an ANOVA table.

Source	SS	df	$\mathbb{E}(MS)$
Treatment	$SSTR = \sum_{i=1}^r n_i\left(\overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot}\right)^2$	r-1	$\sigma^2_\epsilon + n\sigma^2_\alpha$
Error	$SSE = \sum_{i=1}^r \sum_{j=1}^{n_i}(Y_{ij} - \overline{Y}_{i\cdot})^2$	$\sum_{i=1}^r(n_i - 1)$	σ^2_ϵ

- ANOVA table is still useful to setup tests: the same F statistics for fixed or random will work here.
- Test for random effect: $H_0 : \sigma^2_\alpha = 0$ based on
$$F = \frac{MSTR}{MSE} \sim F_{r-1,(n-1)r} \qquad \text{under } H_0.$$

Degrees of freedom

- Why $r-1$ degrees of freedom?
- Imagine we could record an infinite number of observations for each individual, so that $\overline{Y}_{i\cdot} \to \mu + \alpha_i$.
- To learn anything about $\mu.$ we still only have r observations $(\mu_1, \dots, \mu_r)$.
- Sampling more within an individual cannot narrow the CI for $\mu.$

Inference for μ

- Easy to check that

$$\begin{aligned} E(\overline{Y}_{\cdot\cdot}) &= \mu \\ \text{Var}(\overline{Y}_{\cdot\cdot}) &= \frac{\sigma^2_{\epsilon} + n\sigma^2_{\alpha}}{rn}. \end{aligned}$$

- To come up with a t statistic that we can use for test, CIs, we need to find an estimate of $\text{Var}(\overline{Y}_{\cdot\cdot})$.
- ANOVA table says $E(MSTR) = n\sigma^2_{\alpha} + \sigma^2_{\epsilon}$ which suggests

$$\frac{\overline{Y}_{\cdot\cdot} - \mu_{\cdot}}{\sqrt{\frac{MSTR}{rn}}} \sim t_{r-1}.$$

Estimating σ^2_{α}

We have seen estimates of μ and σ^2_{ϵ}. Only one parameter remains.

- Based on the ANOVA table, we see that

$$\sigma^2_{\alpha} = \frac{1}{n}(\mathbb{E}(MSTR) - \mathbb{E}(MSE)).$$

- This suggests the estimate

$$\hat{\sigma^2}_{\alpha} = \frac{1}{n}(MSTR - MSE).$$

- However, this estimate can be negative!
- Many such computational difficulties arise in random (and mixed) effects models.

Mixed effects model

- The one-way random effects ANOVA is a special case of a so-called *mixed effects* model:

$$\begin{aligned} Y_{n\times 1} &= X_{n\times p}\beta_{p\times 1} + Z_{n\times q}\gamma_{q\times 1} \\ \gamma &\sim N(0, \Sigma). \end{aligned}$$

- Various models also consider restrictions on Σ (e.g. diagonal, unrestricted, block diagonal, etc.)
- Our multiple linear regression model is a (very simple) mixed-effects model with $q = n$,

$$\begin{aligned} Z &= I_{n\times n} \\ \Sigma &= \sigma^2 I_{n\times n}. \end{aligned}$$

Using mixed effects models: `lme`

In [29]:

```
library(nlme)
sodium.lme = lme(fixed=sodium~1,random=~1|brand, data=sodium.table)
summary(sodium.lme)
```

```
Linear mixed-effects model fit by REML
 Data: sodium.table
      AIC      BIC    logLik
  154.923 160.4735 -74.46152

Random effects:
 Formula: ~1 | brand
        (Intercept)  Residual
StdDev:    4.612346 0.8461397

Fixed effects: sodium ~ 1
               Value Std.Error DF  t-value p-value
(Intercept) 17.62917  1.886939 42 9.342733       0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.90551291 -0.68337933  0.08232268  0.79246858  1.64968961

Number of Observations: 48
Number of Groups: 6
```

For reasons I'm not sure of, the degrees of freedom don't agree with our ANOVA, though we do find the correct `SE` for our estimate of μ:

In [30]:

```
MSTR = anova(sodium.lm)$Mean[1]
sqrt(MSTR/48)
```

1.88693884226396

The intervals formed by `lme` use the 42 degrees of freedom, but are otherwise the same:

```
In [31]:
```

```
intervals(sodium.lme)
```

```
Approximate 95% confidence intervals

 Fixed effects:
               lower     est.    upper
(Intercept) 13.82117 17.62917 21.43716
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: brand
                   lower     est.    upper
sd((Intercept)) 2.475221 4.612346 8.594683

 Within-group standard error:
    lower      est.     upper 
0.6832445 0.8461397 1.0478715 
```

```
In [32]:
```

```
center = mean(sodium.table$sodium)
lwr = center - sqrt(MSTR / 48) * qt(0.975,42)
upr = center + sqrt(MSTR / 48) * qt(0.975,42)
data.frame(lwr, center, upr)
```

lwr	center	upr
13.82117	17.62917	21.43716

Using our degrees of freedom as 5 yields slightly wider intervals

```
In [33]:
```

```
center = mean(sodium.table$sodium)
lwr = center - sqrt(MSTR / 48) * qt(0.975,5)
upr = center + sqrt(MSTR / 48) * qt(0.975,5)
data.frame(lwr, center, upr)
```

lwr	center	upr
12.77864	17.62917	22.4797