1 Overview

2 ANOVA: refresher

3 Two Independent Variables

4 Two-Way ANOVA

5 Interpretation of the Results

6 Examples in R

6.1 Selecting Most Significant Cross-Terms

6.2 Case of No Interaction

6.3 Categorical and Continuous Variables

Week 8 Notes Part 1 2-way ANOVA

Code ▾

Author: Brendan Gongol

Last update: 27 May, 2023

1 Overview

In this Note we return to ANOVA models. We first review one-way ANOVA, which we are familiar with from the previous weeks. Then we set up a stage for the analysis of the dependence of a random variable on two independent explanatory variables and study how toassess the significance of such models. The section is concluded with examples of using two-way ANOVA with categorical and continuous variables.

2 ANOVA: refresher

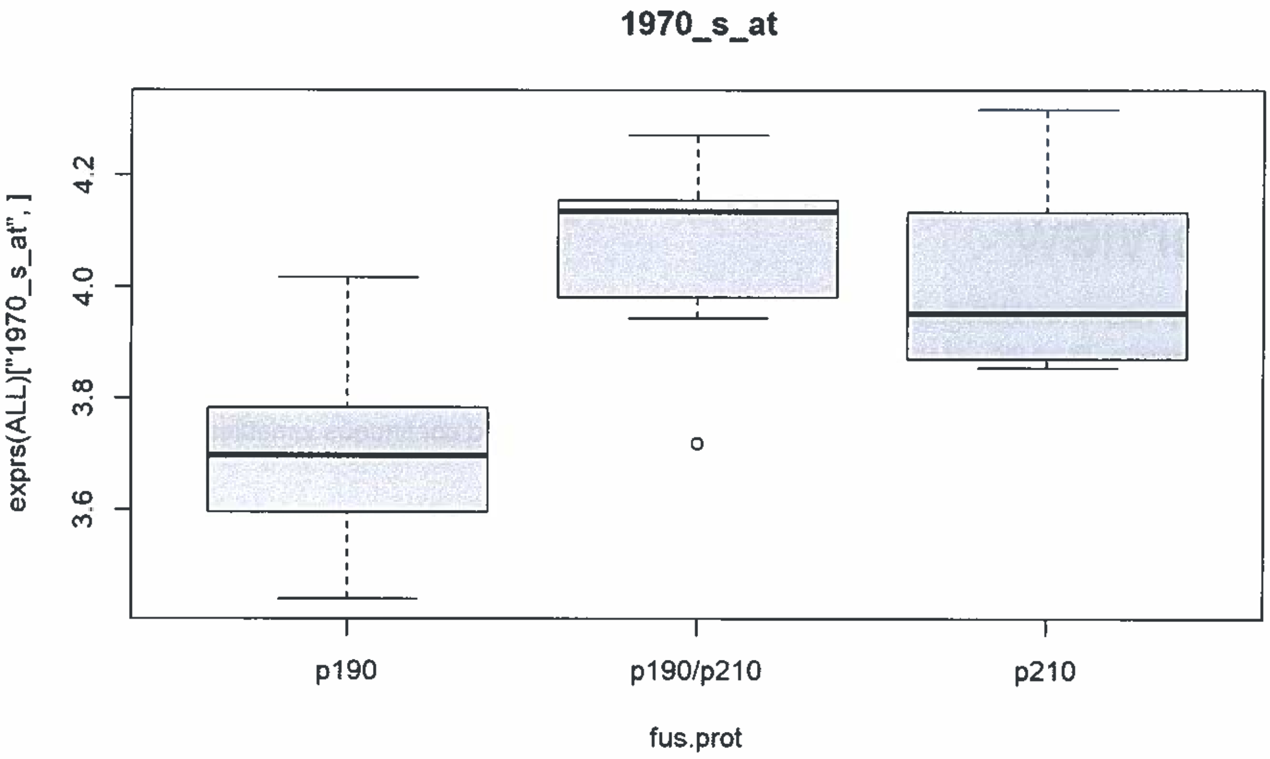
In the previous weeks we have learned about one-way ANOVA. As a brief refresher (revisit older lecture notes if you need to!): Analysis Of Variance (ANOVA) is a very general approach, which in its most generic form aims at assessing how the total variability in the dependent variable Y can (or cannot) be explained by/attributed to the independent variable(s) X. For instance, if $y = ax$ (strict equality, no random noise/other factors at play at all), then the variance of Y is fully explained by X. Indeed, if we were to measure Y only, we would observe a range of seemingly random values following some distribution. The observed “variance” of Y reflects that range. However, if we had sufficient insight and also measured X, we could discover the above linear dependence and realize that each and every change in measured value of Y is completely determined and driven by (i.e. “explained by”) a change in X. In real life, we usually deal with some intermediate situation, of course: some part of the observed changes in measured values of Y from case to case are determined by changes in X, and some other part remains unexplained (noise: actual value of y differs from the one deterministically predicted from x). ANOVA assesses just this balance between explained/unexplained variance. With such general definition of the approach, you can imagine that there exists a very large number of methods and flavors related to ANOVA, and you will be correct.

You should also remember that the “explanatory” variable X does not have to be continuous or even numerical, but can also be categorical. The interpretation is the same: we assign observed values of Y to different classes determined by observed values (levels) of the categorical variable X and ask whether such assignment “explains” away sufficiently large fraction of the total variance in Y: the change(s) in the (mean) values of Y between such classes/groups are “explained” (by the group membership: e.g. males vs females, or smokers vs non-smokers); the remaining variation of the values of Y within each group is “unexplained”. This is one-way ANOVA.

Here is some code to further remind you how we go about one-way ANOVA, i.e. the case when the response variable depends on a single independent variable. In this example we assess the dependence of expression level of a particular gene on the status of BCR/ABL1 fusion protein:

Hide

```
library(ALL); data(ALL)
fus.prot <- pData(ALL)$"fusion protein" # extract data…
boxplot(exprs(ALL)["1970_s_at",]~fus.prot,main="1970_s_at")
```



Hide

```
lm.1970.s.at.fp <- lm(exprs(ALL)["1970_s_at",]~fus.prot) # fit model
summary(lm.1970.s.at.fp)$coef
```

```
##                     Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)        3.7052303 0.03782947 97.945612 3.692547e-39
## fus.protp190/p210 0.3614408 0.06687368  5.404829 7.427066e-06
## fus.protp210      0.3054996 0.06687368  4.568308 7.859778e-05
```

Hide

```
anova(lm.1970.s.at.fp) # get anova p-value, and we are done!
```

```
## Analysis of Variance Table
##
## Response: exprs(ALL)["1970_s_at", ]
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## fus.prot   2 0.92909 0.46455  19.095 4.477e-06 ***
## Residuals 30 0.72984 0.02433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hide

```
df.tmp <- data.frame(model.matrix(lm.1970.s.at.fp),
 fus.prot[!is.na(fus.prot)])
colnames(df.tmp) <- c("Intercept","p190.p210","p210","FP")
df.tmp[1:10,]
```

```
##       Intercept p190.p210 p210        FP
## 01005         1         0    1      p210
## 03002         1         0    0      p190
## 08001         1         0    0      p190
## 08011         1         1    0 p190/p210
## 09008         1         0    0      p190
## 11005         1         0    0      p190
## 12006         1         0    1      p210
## 12012         1         0    0      p190
## 12026         1         1    0 p190/p210
## 14016         1         0    1      p210
```

3 Two Independent Variables

What happens if we have two independent variables that we want to use to explain changes in some response variable? Can we still use ANOVA in order to assess the dependence and how should we do that? Note that last week we already ran `anova()` on multivariate linear models, but the setting was a little different: the flavor of ANOVA we employed used the concept of nested models and added coefficients (corresponding to different explanatory variables) one by one, i.e. we were calculating how much of yet unexplained variance can be attributed to each next variable X_i. In contrast, here we look at classical two-way ANOVA (in the case of categorical variables), where variables X_i are evaluated simultaneously rather than in a succession. Since, as we will see, two-way ANOVA also evaluates interaction between the explanatory variables (as described below), it is also relevant for assessing continuous linear models that include cross-terms (e.g. y=*ax1*+*bx*2+*cx1x*2).

Let's start from a simple example of two independent categorical variables and a response variable that depends (presumably) on both. For the sake of example, let's imagine that we have two drugs A,B for controlling the ratio of good cholesterol to bad cholesterol, and we want to assess their effects and (possible) interaction. Note that:

- There is a study design decision: if we are absolutely sure (are we ever?) that there is no interaction between the drugs, we can run two independent studies: in one we give A vs placebo, in the other B vs placebo; in this case we could run a linear model or categorical one-way ANOVA (if we are comparing fixed dosage of a drug vs no drug) in each study separately (independently for drug A; then the same for drug B). Or we could simply run t-test in the categorical case (drug/no drug), as we know that withonly two classes (levels of categorical variable), ANOVA and t-test are equivalent.
- It is often unknown if there is an interaction or at least such possibility is worth entertaining. Here we understand interaction as, for instance, the effect of drug A being different in the presence or in the absence of drug B. Running separate linear model, one-way ANOVA or t-test in each arm of the study (drug A vs control, then drug B vs control) would not help in this case. In order to assess the interaction, we obviously need subjects who receive both drugs, and we need a statistically sound method for comparing those patients to drug A-only, drug B-only and controls.
- In this example we are not looking at (continuous) dosages of A, B (which would probably ask for a full-fledged linear model; ANOVA can actually handle that, just like it could handle linear model with one independent variable); here we employ the simplest possible (yet important) example in order to understand how things work, so we are considering the following design: drug A (at some fixed dosage)/no drug A, drug B/no drug B
- Hence we have two categorical variables (drug A, drug B) with two levels each (YES/NO), and the following combinations: noA+noB (control), A only, B only, A + B

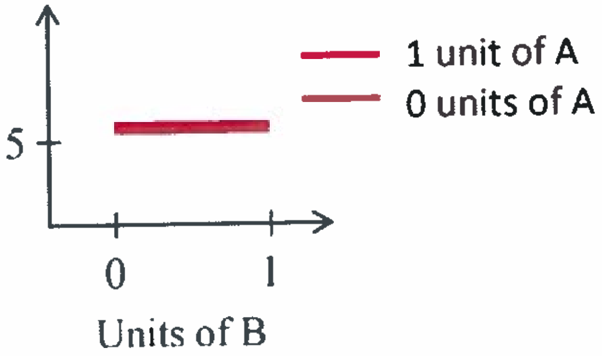
Let us consider possible scenarios with respect to the observations we can make in different groups (of course we would make multiple measurements in each group in order to assess the variance, but here we show only means):

3.1 Scenario 1: no effect.

If neither drug has an effect, our results may look like it is shown in the following table. Note that it looks similar to the 2x2 contingency table, but it is important that we are solving a different problem here. In the case when we have two two-level categorical variables W,Z and assess whether the corresponding labels are distributed independently in the population, we end up with a 2x2 contingency table that lists the counts of cases for each combination. In that case we would apply chi-square or fisher exact test to the contingency table in order to see if there is any significant bias with respect to the null hypothesis that states that labels are assigned randomly and independently. If we choose to speak in terms of associations, we would say that these tests look for one-way association between the two (categorical) variables: W~Z (e.g. patient sex vs complete/refractory remission status, see Week 5 Notes). In the case we are considering here, we also have two (categorical) variables A,B, but in addition we have the response variable that (presumably) depends on them. So the table lists not the counts, but values of the outcome variable, so the model we are investigating is $Y \sim A * B$ (in R notation this is a shortcut for saying that we look for complete cross-dependence on A,B and AB: $Y A + B + AB$).

		Drug B		
		0 Units	1 Unit	Row mean
Drug A	0 Units	5.1	5.1	5.1
	1 Unit	5.1	5.1	5.1
	Column mean	5.1	5.1	5.1

We can also display the results graphically (units of B are shown on the x-axis, and units of A are shown by two different lines, one for $A = 0$, the other for $A = 1$; the outcome (mean cholesterol ratio) is shown on y-axis). In the case of no effect, both lines (A/noA) run at the same y-coordinate (no effect of A, same mean cholesterol ratio) and parallel to x-axis (B also has no effect on Y):

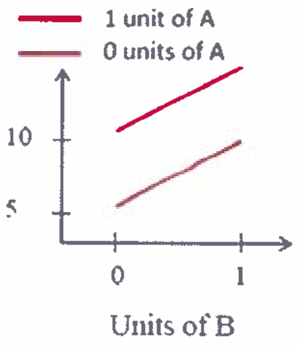


3.2 Scenario 2: independent effects.

In this scenario both drugs have an effect (we chose exactly the same effect of 5.1 for each, but they do not have to be the same of course); these effects are completely independent, i.e.they are additive: giving a patient drug A and drug B simply results in an effect of 5.1 (baseline)+5.1(drug A)+5.1 (drug B)=15.3.

		Drug B		
		0 Units	1 Unit	Row mean
Drug A	0 Units	5.1 baseline	10.2	7.65
	1 Unit	10.2	15.3 additive	12.75
	Column mean	7.65	12.75	10.2

Graphical representation of this situation looks as shown below. Note that there is an effect of B (lines have slope: adding B increases the outcome variable Y), there is an effect of A (there is a finite separation between lines: adding A increases outcome variable so the line shifts up), and the effects are independent (lines are parallel: it does not matter whether we add A at $B = 0$ or at $B = 1$ – the resulting change in Y is still the same).



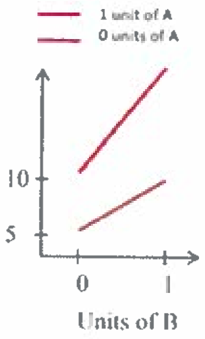
In the context of two-way ANOVA, variables A,B are often referred as row variable and column variable (which is row and which is column depends, of course, on how you set up the table and it does not really matter). The effects of each of the row/column variables are known as main effects. In order to detect main effects, we compare row means and column means. In other words, if we want to answer the question if there is a (separate) effect of A and (separate) effect of B, we ask if there is a significant difference between the row means of 7.65 and 12.75 and between the column means of 7.65 and 12.75 in the above table. Of course we need the variance (multiple measurements) in order to assess such significance – we cannot assess significance from the simplified table shown above as it only illustrates the concept and does not show how much noise we have in each group. Note that in order to assess those separate main effects we are averaging with respect to the levels of the "other" variable: for instance, assessment of the main effect of A involves comparing the outcome at A=0 (row mean - averaged with respect to all levels of B) to the outcome at $A = 1$ (also row mean - averaged with respect to all levels of B). This poses a certain problem, which we will see in next two scenarios: main effects (row means) are confounded by interactions.

3.3 Scenario 3: mutual enhancement

In this example we assume that both drugs have the same effect when given separately, but enhance each other's action when given together: the combined effect of the two drugs is greater than a simple sum of the effects. The outcome table in this case may look like this:

		Drug B		
		0 Units	1 Unit	Row mean
Drug A	0 Units	5.1 baseline	10.2 +5.1	7.65
	1 Unit	10.2 +5.1	20.3	15.25
	Column mean	7.65	15.25	11.45

The corresponding graphical representation is shown below. The interaction between the drugs manifests itself in non-parallel lines (up-shift resulting from giving a subject one unit of drug A is greater when this subject is also given one unit of drug B):



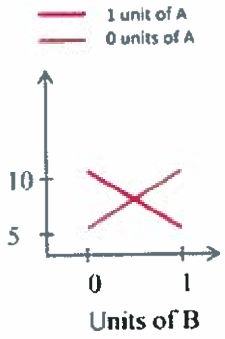
Note that main effects (differences between the two row means and between the two column means) are present in this scenario, but they are confounded by the interaction: for instance, row means for drug A are 7.65 ($A = 0$) vs 15.25 ($A = 1$) – difference of 7.6, while for drug A alone (with no drug B given, $B = 0$), the effect is 5.1 ($A = 0$) vs 10.2 ($A = 1$) – difference of only 5.1.

3.4 Scenario 4: negative interaction

Now suppose that both drugs have effect when given alone, but that they counteract and weaken each other when given together. In this situation, the outcome table may look as in the example below:

		Drug B		
		0 Units	1 Unit	Row mean
Drug A	0 Units	5.1	10.2	7.65
	1 Unit	10.2	5.1	7.65
	Column mean	7.65	7.65	7.65

The same data represented graphically will result in crossing lines:

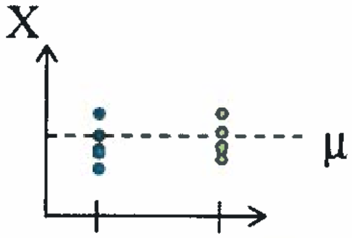


Note how main effects are absent in this scenario (of course it is a little doctored in order to make the effects cancel each other entirely, but in general we do expect weakening of main effects in the case of negative interaction). Indeed, while individual drugs do work, all row and column means are equal to 7.65 in the above table, i.e. main effects (effects of each individual drug) are again confounded by interaction.

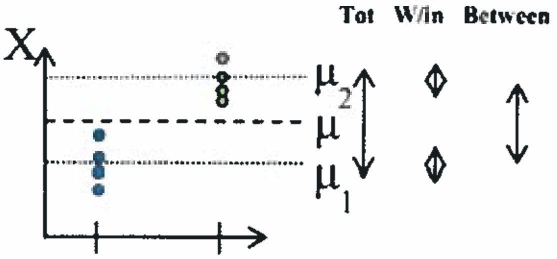
4 Two-Way ANOVA

Two-way ANOVA is a method for assessing the significance of the dependence between the response variable and two explanatory variables (categorical or continuous), as in the above examples. This method also directly quantifies the interaction between the explanatory variables.

Let us recall how a one-way ANOVA works. We have a total variance (sum of squares) SS_{total} and we split it into SS_{within} (within-group, unexplained variance) and $SS_{between}$ (between-group variance, i.e. the one we can explain by membership of the data points in specific groups with their own means), see the illustration below and/or consult Week 5 material:



No effect: none of the total variation in X can be ascribed to between-group difference: there is only the total mean of the whole dataset.



Effect present: part of total variation in X can be ascribed to between-group difference. Membership in groups 1/2 immediately "predicts" different means μ_1/μ_2, the rest is unexplained noise within each group.

In the case of two-way ANOVA, the idea is very similar, but we have more groups we can split the total variance into. The within-group variance (noise, unexplained variance) is computed in the same way as before: now we simply have table cells for groups (one cell for each combination of the levels of the two explanatory variables). Each cell (combination of levels) will have multiple measurements in any real and meaningful experiment (not just means as shown in the simplified tables above), and the sums of squares of the differences between measurements and corresponding cell means, further summed up across all cells, form the unexplained variance SS_{within}:

$$SS_{within} = \sum_A \sum_B \sum_i (X_{ABi} - \bar{X}_{AB})^2$$

where A,B run over levels of the two variables (rows and columns), i denotes i-th measurement in each cell, and $\bar{X}_{AB}$ is the mean of all the measurements in a given cell (row A, column B). The total variance is also defined in the same way as before: the sum of squared differences between the measurements and the total mean of all measurements, in all columns and rows:

$$SS_{total} = \sum_A \sum_B \sum_i (X_{ABi} - \bar{X})^2$$

where $\bar{X}$ is the grand-total mean. However, the between-group variance, $SS_{between} = SS_{total} - SS_{within}$ now has a more complex structure. Namely, it has contributions from both variables. Think of it in the following way: suppose we have two variables, A, B as in the examples above, one of them (let's say it is a row variable) has an effect, and the other is completely irrelevant (i.e. not associated with an outcome in any way). Then we should see a difference between the row means (because the value of variable A matters, as we assumed), but not between the column means. Knowing whether the measurement belongs to one or the other group (level) with respect to A, i.e. what row it belongs to, helps us predict the expected value (row mean) for that measurement. Hence, the explained variance for each datapoint is the (squared) difference between the individual row means and the mean across all rows (which is a total mean, of course):

where A_n is the number of the measurements in row A and $\bar{X}_{A*}$ are row means. Note that this expression is very similar to $SS_{between}$ we used in one way ANOVA (see Week 5 Notes). In the opposite situation, if we had two variables A, B such that A is irrelevant (no association with the outcome) and B has an effect, then row means would be the same, but column means would be different and would contribute to the explained variance:

$$SS_{col} = \sum_A n_B(\bar{X}_{*B} - \bar{X})^2$$

where n_B are counts of measurements in each column and $\bar{X}_{*B}$ are column means. Both variables A and B contribute to the explained variance, so that $SS_{col} + SS_{row}$ is a part of $SS_{between}$. If there is no interaction between A,B, this would be the only part (levels of each of the two variables independently predict the outcome, as in Scenario 2 above). However, it is possible, as we have seen, that there is an effect of both variables and yet the row means and column means are still the same (Scenario 4 above). This can happen only due to interaction between the two variables, and this part of variance is the last missing part of $SS_{between}$ (since this is a regular, predictable effect, it just cannot be predicted based on independent outcomes due to A only or B only):

$$SS_{RxC} = SS_{between} - SS_{row} - SS_{col} = SS_{total} - SS_{within} - SS_{row} - SS_{col}$$

where SS_{RxC} ("row x column") describes contribution due to the interaction.

Now we have all the pieces needed to run two-way ANOVA. In the same way as in one-way ANOVA, we first need to calculate mean squares (MS) from the sums of squares (SS) above. Then, in two way ANOVA we run three F-tests: one for each of the main effects (rows and columns), and one for the interaction:

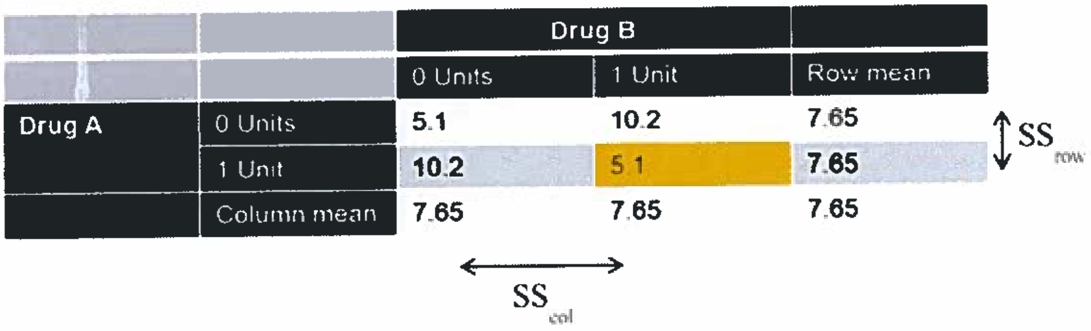
- Frow=MSrow/MSwithin (main effect of row variable)
- Fcols=MScol/MSwithin (main effect of column variable)
- FRxC=MSRxC/MSwithin (effect of interaction)

The number of degrees of freedom for each of the terms considered above is listed in the following table (similar arguments as the one used in Week 5 Notes can be applied to explain why the degrees of freedom are what they are, and of course they can be rigorously derived, but this is a tedious and not very interesting exercise, which is beyond our scope).

Total	$df_T = N_T - 1$	Note that $df_T = df_{wg} + df_{bg}$
within-groups (error)	$df_{wg} = N_T - r^*c$	
between- groups	$df_{bg} = r^*c - 1$	
rows	$df_{rows} = r - 1$	Note that $df_{bg} = df_{rows} + df_{cols} + df_{rxc}$
columns	$df_{cols} = c - 1$	
interaction	$df_{rxc} = (r-1)(c-1)$	

In the table above, N_T is the total number of measurements across all groups (combinations of levels); r is the number of rows (number of levels of the row variable), and c is the number of columns (number of levels of the column variable) – note that in all our examples we used $r = c = 2$, but both variables can have multiple levels in two-way ANOVA.

The following diagram uses the outcome table from the previous discussion and summarizes the components of the total variance introduced in this section:



		Drug B		
		0 Units	1 Unit	Row mean
Drug A	0 Units	5.1	10.2	7.65
	1 Unit	10.2	5.1	7.65
	Column mean	7.65	7.65	7.65

SS_{row}
SS_{col}

$SS_{between} = SS_{Total} - SS_{within}$: between all 4 groups $SS_{RxC} = SS_{between} - SS_{row} - SS$

5 Interpretation of the Results

It is worth reiterating what our examples from section 3 taught us:

- Main effects are confounded by the interaction
- We need to run a test (two-way ANOVA) in order to detect potential interactions.
- If the interactions are present, main effects are hard to interpret or meaningless altogether.
- If we discover interaction, we may want to stratify our data and run the independent tests (in order to get better idea of the independent effect of each factor).

Note that if we do not suspect an interaction, "main effect" is what we are measuring for a single variable (hence the name). In the example with two drugs, we have had an experimental design (called factorial design) aimed at looking for interactions, but consider the following example: we study the effect of the drug A on the cholesterol ratio in subjects from a general population and observe no effect. But then we start suspecting that the drug might work differently in males and females. So we stratify the population and now we have 2x2 outcome table, just like in the previous section: (1 unit of drug A, 0 units of drug A) x (Male, Female). Before we stratified our data, we looked at one variable (drug A/no drug A) only and what we measured, unsuspectingly, for the effect of drug was the row average in our 2x2 outcome table (i.e. average across males/females, because we did not use this stratification in our initial experimental design in the first place) - see the tables in the section 3, you can imagine M/F labels instead of drug B/no drug B. In our current example, if upon introducing the second variable we observe strong interaction (similar to the Scenario 4 above: for instance drug A improves ratio in males and worsens it in females), then the overall mean effects are meaningless of course, and we need to stratify the population and assess the effect of drug separately in males and females.

6 Examples in R

Let us now look how all this machinery works in R, on a real dataset. We start again with the simplest example of two categorical variables: patient sex and disease relapse status (relapsed or not: true/false) in ALL dataset (both variables available in pData() table), and we will study the effect of these variables on gene expression. For this exercise we will use one pre-selected gene in order to illustrate how ANOVA itself works; we will look into gene selection later.

Hide

```
# how many measurements (patients) do we have in each group defined
# by combinations of the two variables? :
table(pData(ALL)[,c("sex","relapse")])
```

```
##    relapse
## sex FALSE TRUE
##   F    11   17
##   M    23   48
```

Hide

```
df.tmp <- data.frame(
 expr.158.at=exprs(ALL)["158_at",],
 sex=pData(ALL)$sex,
 relapse=pData(ALL)$relapse)
# run linear model with interaction term and apply anova() -
# this will give us two-way ANOVA, just that simple:
anova(lm(expr.158.at~sex*relapse,df.tmp))
```

multiply independents.

```
## Analysis of Variance Table
##
## Response: expr.158.at
##             Df Sum Sq Mean Sq F value    Pr(>F)
## sex          1 0.0009 0.00089  0.0172 0.8959027
## relapse      1 0.0092 0.00922  0.1780 0.6740456
## sex:relapse  1 0.7044 0.70436 13.5955 0.0003775 ***
## Residuals   95 4.9218 0.05181
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

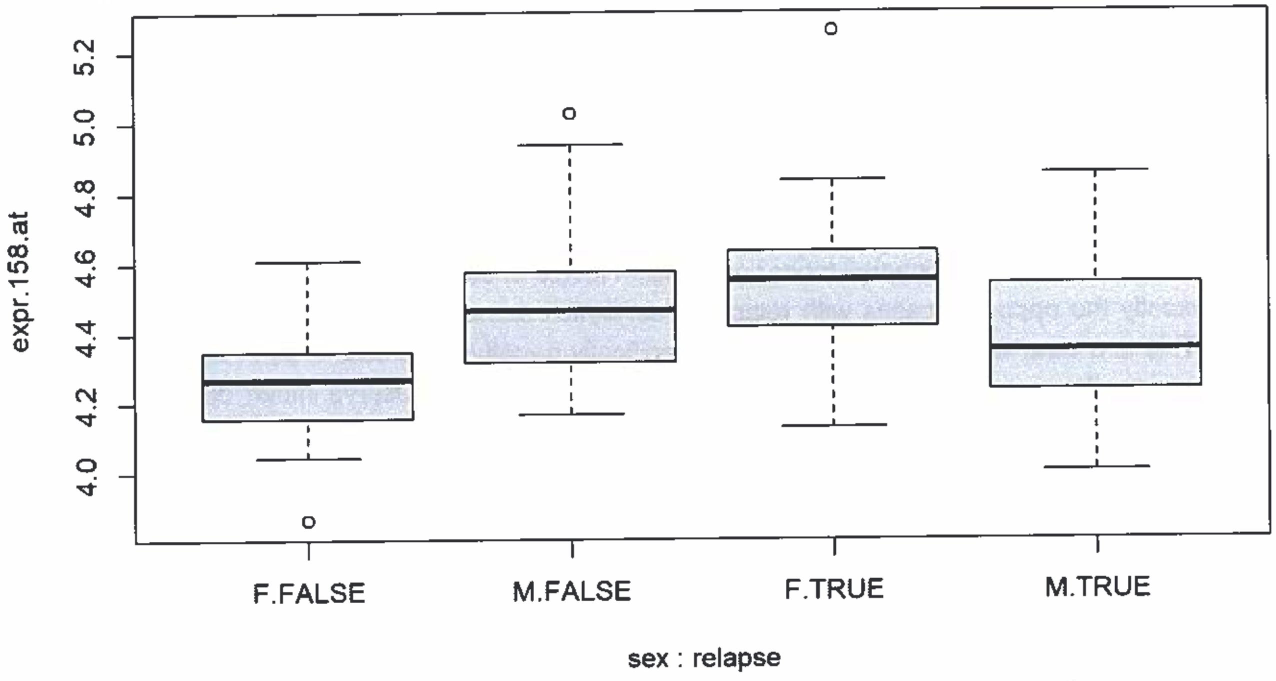
not significant as main effects.

← interaction is highly significant.

Hide

```
boxplot(expr.158.at~sex+relapse,df.tmp) # stratify by sex/relapse
```

the effect of relapse depends on sex (and visa-versa). main effects alone are misleading.

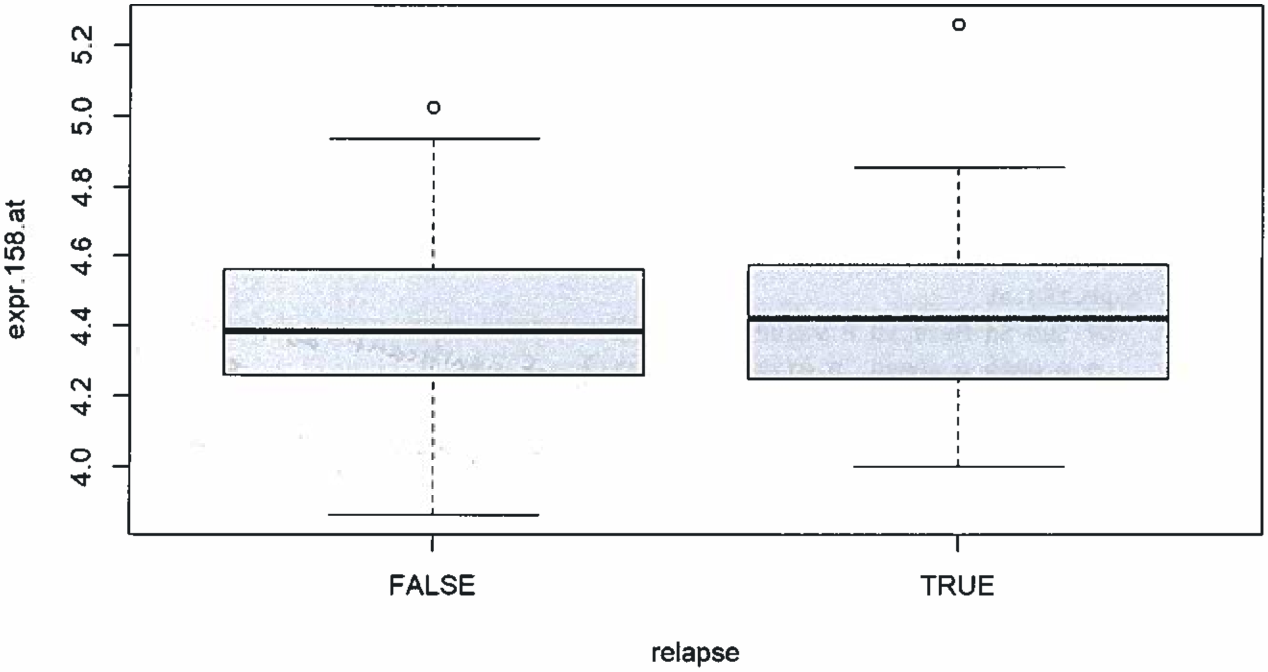


looking at expression stratified by BOTH relapse and sex: in females relapse ↑ expression. In males relapse ↓ expression.

Hide

```
boxplot(expr.158.at~relapse,df.tmp) # stratify by relapse only
```

If you only stratify by relapse you see no effect - they cancel each other out - this is why interaction matters.



In the code example above, we fit a linear model on two explanatory variables with interaction: `sex*relapse` is a shortcut used in R to denote that we look for the dependence on sex, relapse, and their interaction ("product") `sex:relapse` , so that $Z \sim X * Y$ is equivalent to $Z \sim X + Y + X : Y$ in R. Applying `anova()` to the fit is all we need to do in order to compute the two-way ANOVA F-statistics and p-values (the function is smart enough to understand that the fitted model object returned by `lm()` contains a fit on two explanatory variables and will do the right thing). Note that we have a significant interaction term in our model. As we have discussed earlier, this indicates that main effects are not very meaningful (they are not significant at all in the `anova()` output above, for either sex or relapse, but this does not mean much).

In order to better illustrate what is happening with the data, we first plot distributions of the expression levels of the selected gene stratified by both variables. Note that in females the cases with relapse status=true exhibit noticeably (and likely significantly) higher expression levels of the gene we are studying. In males the situation is exactly the opposite: cases with relapse=true exhibit somewhat lower expression levels of the same gene. This is a clear case of interaction (and specifically negative interaction, as in Scenario 4 in the earlier discussion). If we were to look at "main effects", we would not observe much difference as the second plot illustrates: if we ignore patient sex and stratify expression levels by relapse status only, we would not see any difference! This is consistent with the `anova()` report that tells us that the main effect is indeed insignificant. In the presence of interaction, this "insignificant" p-value does not mean that 'relapse' variable indeed does not have an effect on expression level of the gene. It means only that this variable alone does not have much effect (as the right panel plot in the figure above clearly demonstrates), but it can still have very distinct effect together with the second variable (that's what the interaction is).

Let us now look at the default design matrix used by `lm()` in the example we are studying (it has many rows so we look at a two small pieces):

Hide

```
model.matrix(lm(expr.158.at~sex*relapse,df.tmp))[1:5,]
```

```
##       (Intercept) sexM relapseTRUE sexM:relapseTRUE
## 01005           1    1           0                0
## 01010           1    1           1                1
## 03002           1    0           1                0
## 04006           1    1           1                1
## 04007           1    1           1                1
```

Hide

```
model.matrix(lm(expr.158.at~sex*relapse,df.tmp))[40:45,]
```

```
##       (Intercept) sexM relapseTRUE sexM:relapseTRUE
## 27003           1    0           1                0
## 28003           1    1           0                0
## 28005           1    1           1                1
## 28006           1    1           1                1
## 28007           1    0           0                0
## 28019           1    1           0                0
```

Hide

```
summary(lm(expr.158.at~sex*relapse,df.tmp))$coef
```

```
##                    Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)       4.2480781 0.06862825 61.899844 1.390822e-78
## sexM              0.2490433 0.08344080  2.984670 3.609273e-03
## relapseTRUE       0.2919891 0.08807598  3.315196 1.297679e-03
## sexM:relapseTRUE -0.3882837 0.10530549 -3.687212 3.774553e-04
```

Hide

```
mean(df.tmp$expr.158.at[df.tmp$sex=="F" & df.tmp$relapse==F],na.rm=T)
```

```
## [1] 4.248078
```

intercept = Female & not relapsed.

Hide

```
mean(df.tmp$expr.158.at[df.tmp$sex=="M" & df.tmp$relapse==F],na.rm=T)
```

```
## [1] 4.497121
```

The interpretation is essentially the same as in the case of a single variable. Indeed, in the latter case, for a single (categorical) variable X with k levels the design matrix had k columns. The "intercept" column in the default matrix (with all 1's in it) described the mean in the first group (first level of X), and the remaining columns contained 1's only in the cases (patients) belonging to the corresponding group (level of X); the corresponding coefficient of the fitted model was the offset of the mean value in that group with respect to the mean value in the first group (you may want to review the Notes on design matrices from Week 7).

In our present case, we have two categorical variables, each with two levels, so our outcome table contains the total of 2x2=4 groups defined by all combinations of the levels of input variables (just like the drug A/drug B examples we were discussing at the beginning). Hence, when using only 0/1 indicators, the design matrix must contain four columns in order to unambiguously describe which of the four groups each patient belongs to (note that this is not binary arithmetic but indicators: in binary arithmetic we could use one bit to encode a variable with 2 levels, or two bits for a variable with four levels -00, 01, 10, 11; but here we need instead a matrix that can be multiplied by the column of the model coefficients, see Week 7).

There are still different ways to define the four indicator columns (again, as discussed and demonstrated in Week 7 for the case of a single variable), and depending on the choice, the fitted coefficients would have different meanings. The default choice made by `lm()` in the example shown above is as follows: the "intercept" column (all 1's) corresponds to the group `sexF:relapseFALSE`. We can verify that by explicitly computing the mean expression level in that group (see the code fragment above): that mean is equal to the fitted "intercept" reported by the `summary()` of our fitted model. The next two columns ("sexM", "relapseTRUE"), have value set to 1 for the patients, in which the corresponding level of the categorical variable differs from the "background" used as the intercept. Thus, column `sexM=1` alone indicates `sexM:relapseFALSE`, and column `relapseTrue=1` indicates `sexF:relapseTRUE`. The fitted coefficients represent offsets of the corresponding group means from the intercept (mean of `sexF:relapseFALSE`). The last command in the code fragment above illustrates this for one of the columns: the offset of the mean expression in `M/FALSE` from the `F/FALSE` (the intercept) is 4.497121-4.248078= 0.249043, exactly what is reported for the fitted 'sexM' coefficient in the summary.

The last column in the design matrix is tricky: it represents the interaction and it is equal to 1 only when all other columns are equal to 1. Hence this column does represent M/TRUE, but the corresponding coefficient is the offset of the mean in that group from 4.248078 (intercept) + 0.2490433 (sexM) + 0.2919891 (relapseTRUE) – which would be the sum of the independent effects of the two variables. You can better understand this if you multiply design matrix by the vector of coefficients and write down the system of resulting case-by-case equations (just like we did in Week 7); you can also verify this numerically by computing the corresponding means, similarly to the sample code shown above.

6.1 Selecting Most Significant Cross-Terms

Let us now try searching for genes with most significant effects in the two-variable settings. In the following example we will be interested in the interaction term, and we will be using `lmFit()` in order to run significance analysis on all genes in the ALL dataset at once.

Hide

```
library(limma)
b.mask <- !is.na(pData(ALL)$sex)&!is.na(pData(ALL)$relapse)
sex.wo.na <- pData(ALL)$sex[b.mask]
relapse.wo.na <- pData(ALL)$relapse[b.mask]
exprs.wo.na <- exprs(ALL)[,b.mask]
design <- model.matrix(~sex.wo.na*relapse.wo.na)
design[1:5,]
```

b.mask is TRUE only for samples where both sex + relapse are not missing.

$ 4 categories.

F: No Relapse   M: no relapse
F: Relapse   M: relapse.

always 1. M:no relapse relapse male relapse.

```
##   (Intercept) sex.wo.naM relapse.wo.naTRUE sex.wo.naM:relapse.wo.naTRUE
## 1           1          1     x     0     =     0  = male False
## 2           1          1     x     1     =     1  = Male True
## 3           1          0     x     1     =     0  : Female True
## 4           1          1     x     1     =     1    Male True
## 5           1          1     x     1     =     1    Male True .
```

always 1

Hide

edata d matrix.

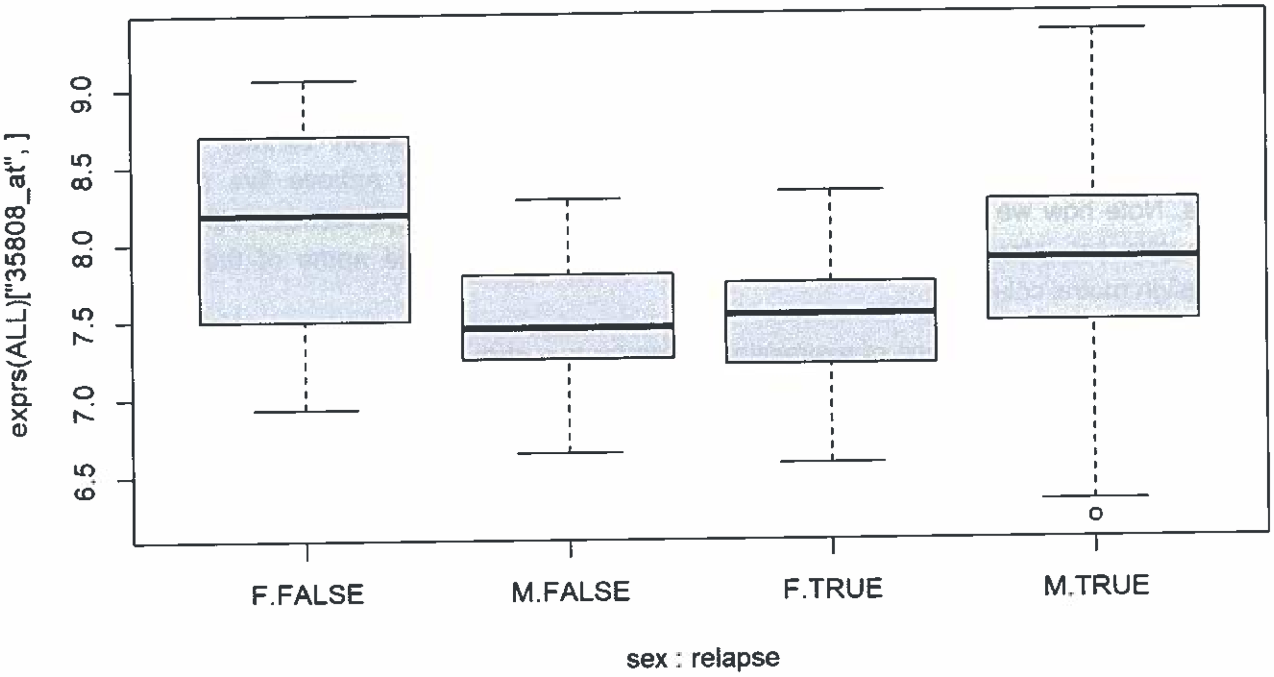
```
limma.fit.e <- eBayes(lmFit(exprs.wo.na,design))
#note that we ask for most significant cross-terms below:
topTable(limma.fit.e,"sex.wo.naM:relapse.wo.naTRUE",5)
```

Fits linear model for all genes at once. Each gene gets its own set of coefficients.

```
##                logFC  AveExpr         t      P.Value adj.P.Val         B
## 35808_at  0.9745584 7.748620  3.658608 0.0004108914 0.9997422 -2.666125
## 158_at   -0.3882837 4.430136 -3.653286 0.0004184541 0.9997422 -2.671738
## 33700_at -1.6192645 6.190601 -3.407525 0.0009530255 0.9997422 -2.925417
## 41071_at  1.7547541 6.840778  3.277095 0.0014523434 0.9997422 -3.055432
## 38474_at -0.2543228 3.340248 -3.124843 0.0023417392 0.9997422 -3.202862
```

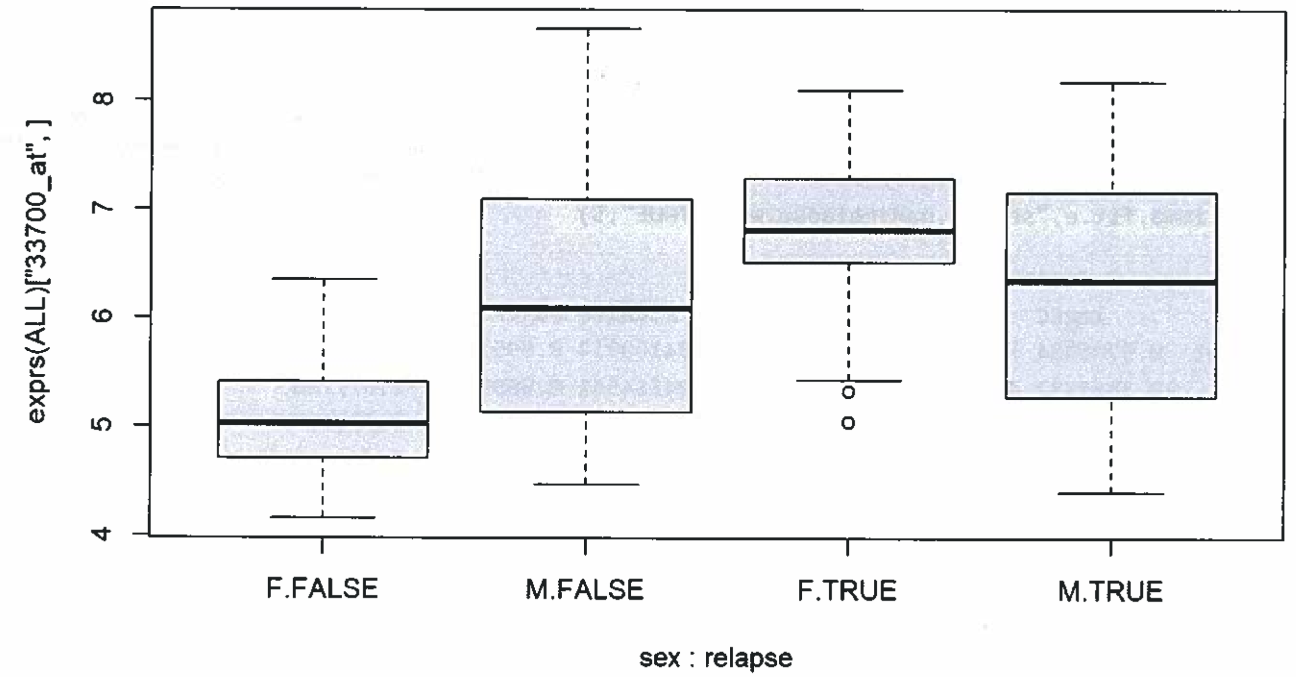
Hide

```
# we have looked at 158_at already, let's check other hits from topTable:
boxplot(exprs(ALL)["35808_at",]~sex+relapse,df.tmp)
```



Hide

```
boxplot(exprs(ALL)["33700_at",]~sex+relapse,df.tmp)
```



In the code fragment above, we first select subset of data that does not contain `NA` in either of the variables. Then we use `model.matrix()` function in order to generate (default) design matrix for the specified model: this is a new strategy, take a note of it, in the previous weeks we used to generate design matrices for `lmFit()` manually. Note the use of `model.matrix()` and the important fact that it not only can report the design matrix used by previously fitted object, but also can generate new design matrix simply from the formula.

With the matrix of expression levels for all genes and the design matrix, we run `lmFit()` in a usual way, adjust p-values with `eBayes()`, and finally run `topTable()` in order to retrieve five most significant dependences. Note how we tell `topTable()` to look for most significant dependences with respect to the specific term (we are interested in the interaction here, and we use the name of the corresponding coefficient/design matrix column).

Lastly, we plot stratified distributions of expression levels for two of the genes found by our analysis. Note different types of interactions discovered. In the left panel, the average expression across all groups with sex=F vs average expression across all groups with appear to be very close (look at the pair of the corresponding boxplots and find the average between the two, visually; of course you can confirm by plotting the corresponding boxplots or calculating the corresponding means in R). Similarly, average expression in all relapse=FALSE groups vs average expression in all relapse=TRUE groups appear to be the same as well. Thus, both main effects are apparently absent in this case, while both variables clearly have effect in properly stratified data, but the effects are opposite depending on the value of the other variable. In the right panel, averages of all F and all M are about the same (no main M/F effect), and the effect of relapse status is seen only in females; some main effect of relapse is still present in two-way analysis (try averaging, visually, between relapse =T and relapse=F), but it is partially washed away by the lack of relapse effect on males. It is the interaction term that lets us get to the effect in its “most pure” form.

6.2 Case of No Interaction

In the next example we will see the case where there is no significant interaction between the explanatory variables:

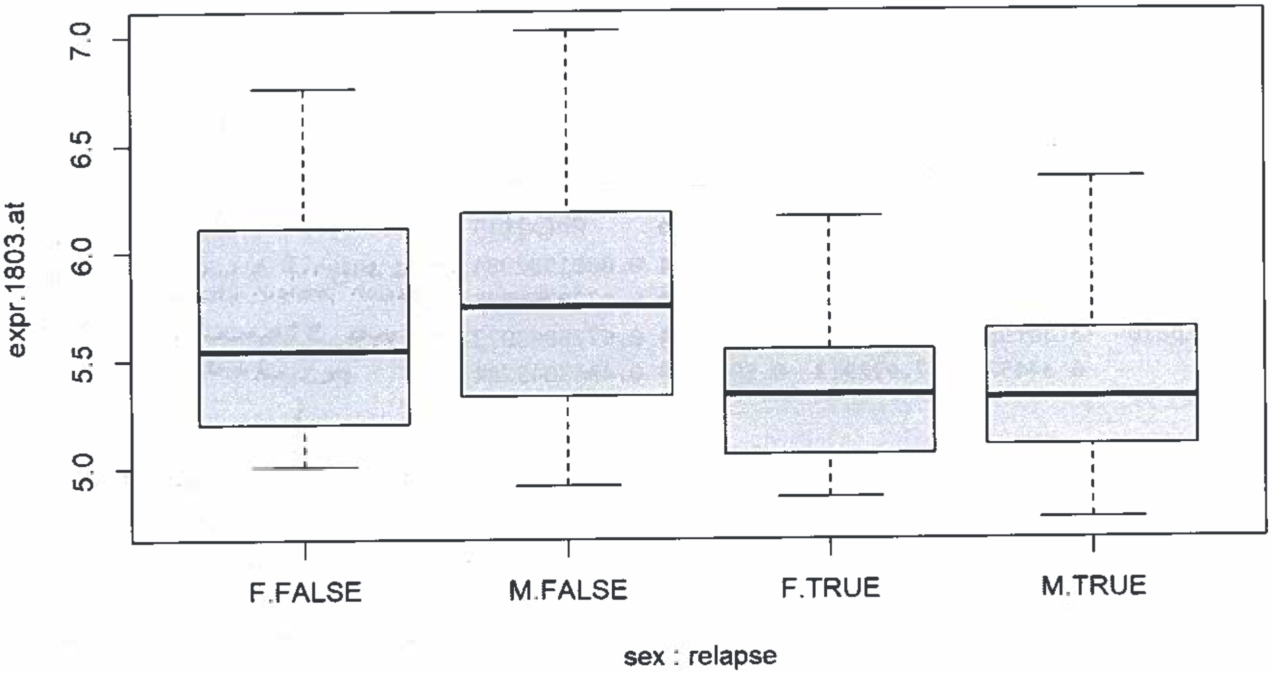
Hide

```
df.tmp <- data.frame(
 expr.1803.at=exprs(ALL)["1803_at",],
 sex=pData(ALL)$sex,
 relapse=pData(ALL)$relapse)
anova(lm(expr.1803.at~sex*relapse,df.tmp))
```

```
## Analysis of Variance Table
##
## Response: expr.1803.at
##             Df  Sum Sq Mean Sq F value    Pr(>F)    
## sex          1  0.0456  0.0456  0.2286    0.6337    
## relapse      1  3.4711  3.4711 17.3833 6.745e-05 ***
## sex:relapse  1  0.0045  0.0045  0.0227    0.8806    
## Residuals   95 18.9696  0.1997                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hide

```
boxplot(expr.1803.at~sex+relapse,df.tmp)
```



As usual, we pack data into a dataframe (optional step), fit a linear model on two explanatory variables with interaction term and run ANOVA on the resulting fitted object. For the gene selected for this example, the F-test p-value for the interaction term is not significant. Hence the main effects (if any) are meaningful and allow for straightforward interpretation. In this particular case, there is no effect of patient's sex on gene expression (insignificant p-value), but there is a strong effect of relapse status. The figure below shows the stratified distributions of gene expression levels for this example, and we can see that indeed only relapse has effect:

6.3 Categorical and Continuous Variables

We have seen examples of two continuous variables in the previous Week, so you may want to revisit the Notes on multiple regression. The idea and interpretations of the two-way ANOVA results in this case is the same, so it is not worth repeating here one more time. Instead, before we conclude the discussion of two-way ANOVA, let us look at the example of one continuous and one categorical variables. We will examine the dependence of days-to-remission on gene expression (continuous) and the status of BCR/ABL1 fusion protein (categorical). Here's the code (you may need to rerun a few initialization lines of code shown in earlier notes in order to get days-to-remission into your R session):

Hide

```
#### Load data (shown in previous notes) ####
ALL.pdat <- pData(ALL)
date.cr.chr <- as.character(ALL.pdat$date.cr)
diag.chr <- as.character(ALL.pdat$diagnosis)
date.cr.t <- strptime(date.cr.chr,"%m/%d/%Y")
diag.t <- strptime(diag.chr,"%m/%d/%Y")
days2remiss <- as.numeric(date.cr.t - diag.t)
x.d2r <- as.numeric(days2remiss)
exprs.34852 <- exprs(ALL)["34852_g_at",]
d2r.34852 <- data.frame(G=exprs.34852,D2R=x.d2r)
################################################
df.41690.fp <- data.frame(D2R=x.d2r,
 G=exprs(ALL)["41690_at",],
 FP=as.character(fus.prot))
summary(lm(D2R~G+FP,df.41690.fp))$coef # fit without interaction
```

looking at main effects only

additive - if we want to see interaction we multiply.

```
##                Estimate Std. Error      t value     Pr(>|t|)
## (Intercept) 102.247173  21.669715  4.7184364 0.0001982984
## G            -4.627271   2.924952 -1.5819990 0.1320750825
## FPp190/p210  -3.302635   7.792277 -0.4238344 0.6770008973
## FPp210        6.344573   7.922913  0.8007879 0.4343040784
```

← is signif = when gene exp = 0 and fusion protein status is the refrence, d2r does not equal zero. i.e. there is a baseline.

no significant results.

↓

affect of gene expression or fusion protein statuses is not significant in predicting days 2 remission when looking at main effects of each variable independently.

Hide

```
anova(lm(D2R~G+FP,df.41690.fp))
```

When considered separately (additively), neither gene expression of 41690_at or fusion protein status significantly affects days-to-remission.

```
## Analysis of Variance Table
##
## Response: D2R
##           Df Sum Sq Mean Sq F value Pr(>F)
## G          1  458.2  458.16  2.2516 0.1518
## FP         2  241.1  120.55  0.5925 0.5640
## Residuals 17 3459.2  203.48
```

not signif. Independently, G + FP do not explain a significant amount of variation in days-to-remission.

3459.2 — the remaining total unexplained variance

203.48 ← average unexplained variance per observation.

Hide

```
summary(lm(D2R~G*FP,df.41690.fp))$coef # fit with interaction
```

Gene expression has a statistically significant apparent effect on D2R but only when you include the interaction w FP status in the model.

```
##                  Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)   135.0865469  17.062392  7.91721041 9.785597e-07
## G              -9.1498656   2.317100 -3.94884351 1.286545e-03
## FPp190/p210     4.3566644  47.351178  0.09200752 9.279096e-01
## FPp210       -179.5083035  38.921032 -4.61211575 3.387481e-04
## G:FPp190/p210  -0.7483812   6.173987 -0.12121523 9.051294e-01
## G:FPp210       23.8721749   4.965348  4.80775429 2.303580e-04
```

(Intercept) ← baseline.

G: for the refrence FP group (p190) - each unit increase in gene exprs decreases d2R by ~9.15 days

FPp210: Strongly lower baseline D2R for "p210" level compared to p190 when gene exprs = 0.

G:FPp210: Strong + significant interaction. The relationship b/w D2R + gene exprsn is very different in this group. Estimate is positive. In samples = the p210 FP status, for each unit increase in 41690 gene exprsn, the effect on D2R is 23.87 units higher than the ref group (p190).

* No sig difference in baseline D2R between ref FP status p190 and 190/210 when G = 0.

* No sig interaction b/w gene exprs + FP status 190/210.

Hide

```
anova(lm(D2R~G*FP,df.41690.fp))
```

```
## Analysis of Variance Table
##
## Response: D2R
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## G          1  458.16  458.16  5.1749 0.0380267 *
## FP         2  241.11  120.55  1.3616 0.2861543
## G:FP       2 2131.18 1065.59 12.0358 0.0007616 ***
## Residuals 15 1328.02   88.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

458.16 — the amount of var explained by gene exprs alone after accounting for the other terms in the model. it is significant but low, (11%)

Fusion Protein status alone is not signif.

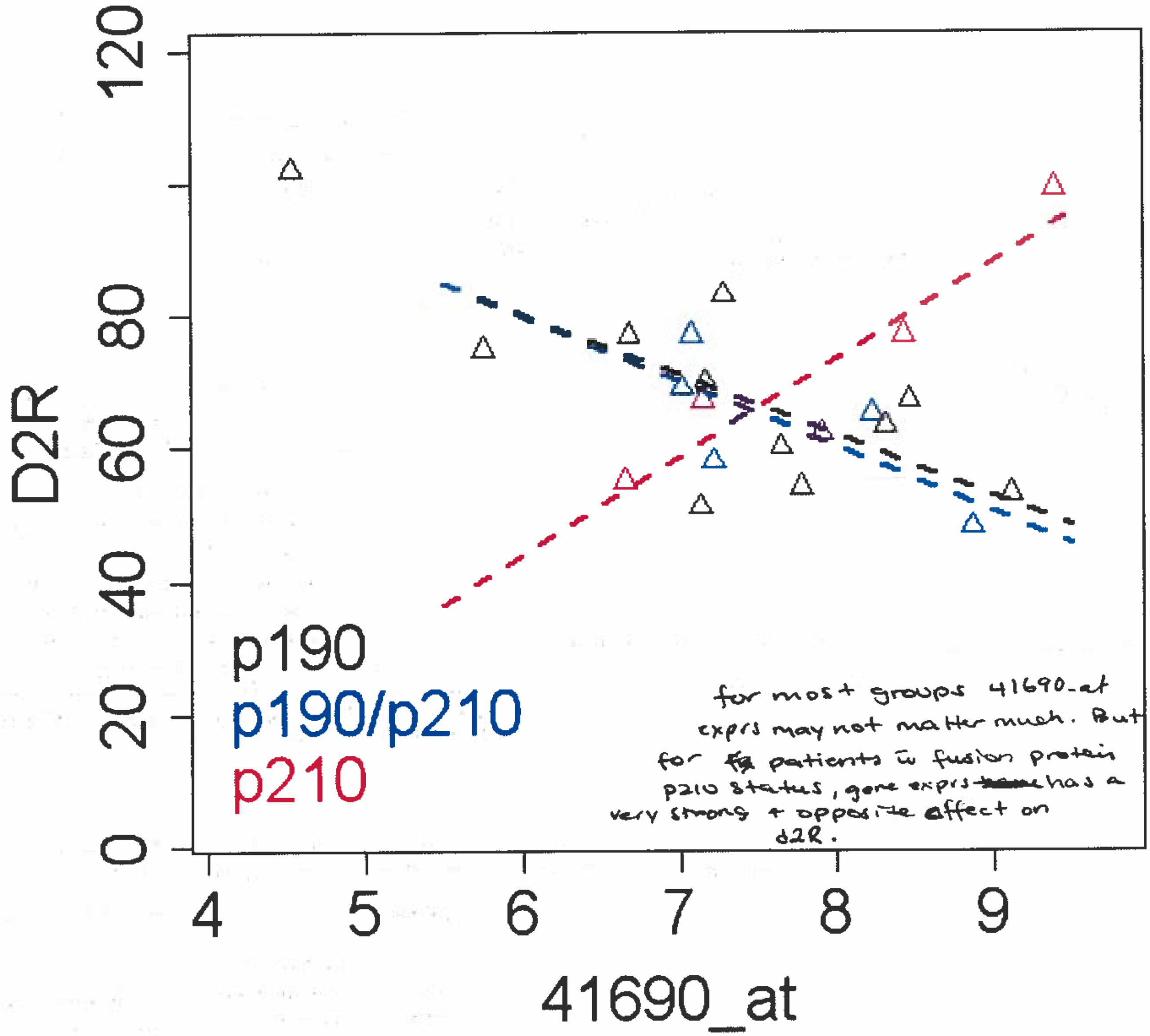
2131 is the var in D2R explained by the interaction - how the effect of gene exprs changes depending on FP status. (51% of variation)

The ANOVA shows that most of the explainable variation in D2R in this model comes from the interaction b/w gene exprs + FP status.

We start with fitting days-to-remission against expression levels of a particular gene and fusion protein status without any interaction terms (in other words we are specifically looking for purely additive effects). As the fit summary and ANOVA tell us, there is seemingly no dependence at all. However when we fit linear model with interaction, we obtain significant cross-term (and mildly significant main effect for gene expression). With such interaction term the lack of/weak main effects are not uncommon, so it is not surprising that we failed to detect any simple additive dependence with our original (inadequate) model. The Figure shown below illustrates the data structure in this example. Note that if we look at all gene expression levels (triangles of all colors), they indeed form a relatively shapeless and directionless cloud, so that there seems to be little, if any, association between gene expression and days-to-remission. However if we add the second variable (fusion status, shown in color), we see that the gene expression values stratify nicely and distinct (and opposite) dependences of `D2R` on expression in different classes are revealed. The strong negative interaction makes main effects (such as dependence on gene expression without regard for fusion status) next to non-existent.

PTO.

~~to plot this:~~



for most groups 41690_at exprs may not matter much. But for patients w/ fusion protein p210 status, gene exprs has a very strong + opposite affect on d2R.

The relationship b/w gene expression + d2r is not the same for all fusion protein groups. The p210 group behaves very differently and this is only revealed when we include interaction in our model.

See summary notes for code to make a plot like this based on the wk8_code.