Experimental Design and Data Analysis, Lecture 9

Eduard Belitser

VU Amsterdam

Lecture overview

1. ANCOVA
2. prediction and feature selection in linear regression:
   - lasso
   - ridge
   - elastic net
3. multiple testing procedures, FDR control

analysis of covariance (ANCOVA)

Setting

An experiment with:

- a numerical outcome Y;
- a factor that can be fixed at I levels.
- a numerical explanatory variable X.

we want to study influence of fac. and num. exp. var. together on response/outcome

Often the dependence of Y on the numerical variable X is a-priori evident, and the variable is included only to increase the precision of the analysis.

EXAMPLE Experiment to investigate the strength of a wire as dependent on the type of material used and its thickness. (Thickness could not be controlled.)

fac.

response

num. var = measured

EXAMPLE Experiment where a subject must press a green or red button if there is a car in the picture shown on the screen, with outcome reaction time, factors presence or not of an auditory stimulus and explanatory variable age of the subject.

Design

list. units over level fac.s

- Select NI experimental units randomly from the population of interest.
- Measure the X of each unit.
- Assign level i of the factor randomly to N units.
- Perform the experiment NI times independently.

then we obs. what comes out = response var.

Randomization is as for one-factor experiments (1-way ANOVA).

It's like 1-way ANOVA but there is also var. X which we measure

The model and hypothesis to test

i is also present here b/c val. of exp. var. could vary for each lvl. of fac.

we also measure Xs for each level i

Data: $(Y_{i1}, X_{i1}), (Y_{i2}, X_{i2}), \ldots, (Y_{iN}, X_{iN}),\ i = 1, 2, \ldots, I.$

I samples = each sample is obs. of level i

ith sample → obs. when we assign level i

The linear ANCOVA model assumes that

response

fac. num. var.

$$Y_{ik} = \mu + \alpha_i + \beta X_{ik} + e_{ik}, \quad i = 1, \ldots, I, \quad k = 1, \ldots, N,$$

does var. play a role?

for errors (e_{ik}) that can be viewed a random sample from a normal population.

We want test the null hypothesis $H_0 : \alpha_i = 0,\ i = 1, 2, \ldots, I,$ and $H_0 : \beta = 0.$

is fac. important?

We also want to estimate the parameters $\alpha_1, \ldots, \alpha_I$ and β.

Any ANCOVA/ANOVA can always be seen as linear regression $Y = Z\gamma + e$ with the certain design matrix Z and parameter vector γ. For example, for $I = 2$, $N = 3$, → 3 obs. for each lvl.

μ 1 α_1 1 α_2 0 β

$$Y = \begin{pmatrix} Y_{11} \\ Y_{12} \\ Y_{13} \\ Y_{21} \\ Y_{22} \\ Y_{23} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & X_{11} \\ 1 & 1 & 0 & X_{12} \\ 1 & 1 & 0 & X_{13} \\ 1 & 0 & 1 & X_{21} \\ 1 & 0 & 1 & X_{22} \\ 1 & 0 & 1 & X_{23} \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \beta \end{pmatrix} + \begin{pmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \end{pmatrix} = Z\gamma + e.$$

α_1, α_2

6 obs. put on top of each other

- Z = design matrix
- γ → contains param.s

Os and 1s = has no mean. unlike fac.s and var.s

The first column of Z is related to the intercept μ, the next two are "dummy" variables related to ANOVA part α_1, α_2, the last is related to the linear regression part β.

* fac. enters model w/param.s as many as levels.

Os and 1s b/c you eith. have α_1 or α_2

Analysis in R: data input

The data frame contains the data about the strength of a fiber made on 3 different machines. Thickness cannot be controlled, but measured.

```
> fiber=read.table("fiber.txt",header=TRUE); fiber
   strength thickness type
1        36        20    1
2        41        25    1
3        39        24    1
4        42        25    1
5        49        32    1
6        40        22    2
7        48        28    2
8        39        22    2
9        45        30    2
10       44        28    2
11       35        21    3
12       37        23    3
13       42        26    3
14       34        21    3
15       32        15    3
```

3 types

continous var. = X

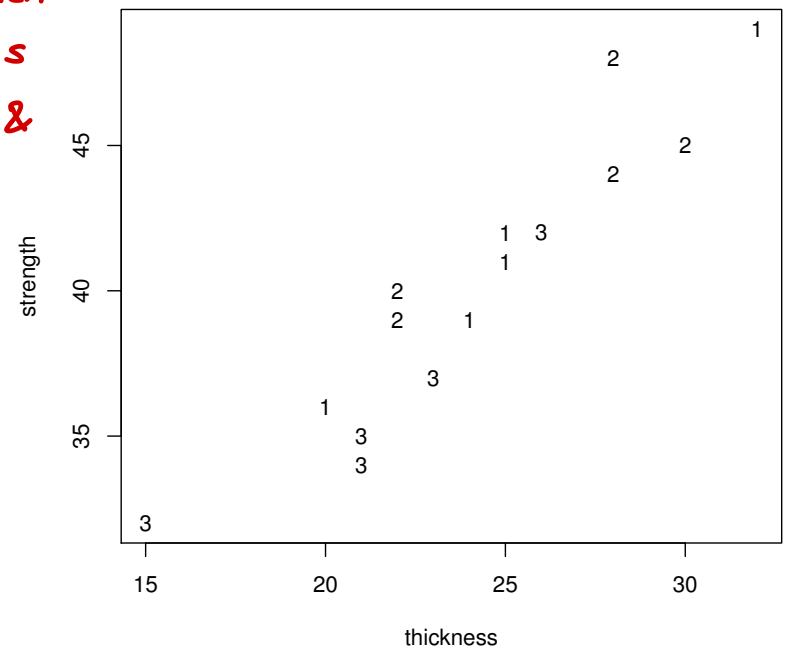
Analysis in R: graphics

```
> plot(strength~thickness,pch=as.character(type))
```

informative plot of data = you can recover all info

you will know which thickness corresp.s to what strength & type

⊙ non-inform. plots → boxplot = you see some rep. of data but it is not all you have as obs. → loss of info



Strength clearly increases with thickness. Its dependence on type is not so clear.

Analysis in R: testing (1)

type = num. → if you don't decl. as fac. then it will be treat. as num. var.

```
> fiber$type=as.factor(fiber$type)
> anova(lm(strength~type,data=fiber))
```

1-way ANOVA

```
[ some output deleted ]
          Df Sum Sq Mean Sq F value  Pr(>F)
type       2  140.4  70.200  4.0893 0.04423 *
```

b/c even if we are interest. in type, we should take thick. into acc. as it will be an imp. var.

Factor type is significant, but one-way ANOVA with only factor type is not correct!

```
> fiber1=lm(strength~thickness+type,data=fiber)  # type second!
> anova(fiber1)     # only p-value for type is relevant
[ some output deleted ]
          Df  Sum Sq Mean Sq  F value    Pr(>F)
thickness  1 305.130 305.130 119.9330 2.96e-07 ***
type       2  13.284   6.642   2.6106    0.1181
Residuals 11  27.986   2.544
```

now order matters → we are interest. in type so we put it 2nd.

diff. p-val. b/c we take thick. into acc.

not right/relevant p-val. for thick. b/c it's not 2nd

Factor type is now insignificant. The output of ANCOVA depends on the order of the variables in the model formula. The correct p-value for type is obtained with strength∼thickness+type, not with strength∼type+thickness. Alternative: use drop1 instead of anova, see next slide.

Analysis in R: testing (2)

```
> drop1(fiber1,test="F")        # here all p-values are relevant
Single term deletions

Model:
strength ~ thickness + type
          Df Sum of Sq     RSS    AIC F value    Pr(>F)
<none>                  27.986 17.355
thickness  1   178.014 206.000 45.297 69.9694 4.264e-06 ***
type       2    13.284  41.270 19.181  2.6106    0.1181
```

we get both p-val.s

p-val. for thick. is right one = same as val. we get if we do type + thick.

The command drop1 is very handy: it performs the tests for the both models, strength∼thickness+type and strength∼type+thickness at once, whereas the *p*-values in the output of anova are sequential, as in a step-up strategy. This problem does not arise in (balanced) ANOVA or linear regression, but it does in an unbalanced ANOVA, ANCOVA and mixed models. Another (and the best) way to get correct p-values, e.g., for the factor type: `fiber2=lm(strength∼thickness,data=fiber)`, then `anova(fiber2,fiber1)` will give the right p-values for the factor type.

test whether cert. var. is imp. by taking full model w/ var. and the model w/out var. → then applying ANOVA

Analysis in R: estimation

create model

```
> fiber1=lm(strength~thickness+type,data=fiber); summary(fiber1)
[ some output deleted ]
            Estimate Std.Error t value Pr(>|t|)
(Intercept)  17.360    2.961    5.862  0.000109 ***
thickness     0.954    0.114    8.365  4.26e-06 ***
type2         1.037    1.013    1.024  0.328012
type3        -1.584    1.107   -1.431  0.180292
```

(Intercept) μ; thickness β; type2 $\alpha_2 - \alpha_1 = \alpha_2$; type3 $\alpha_3 - \alpha_1 = \alpha_3$ (0)

est. β = pos. val. → thicker the fiber, stroger it is → we reject hypoth. that $\beta = 0$.

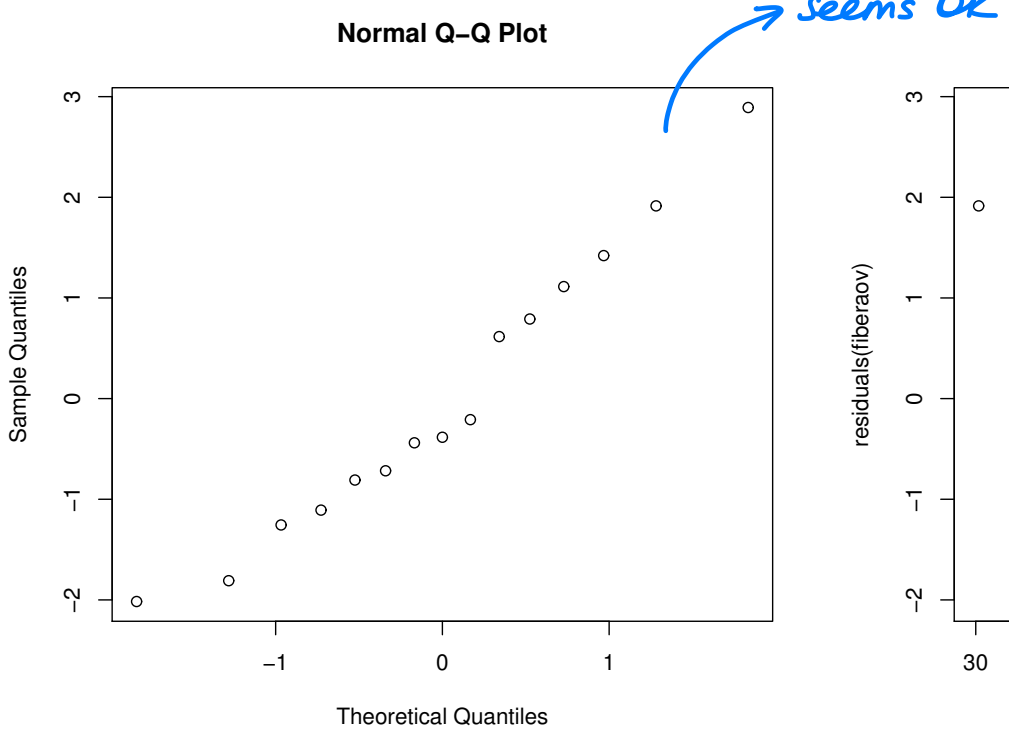
This shows the coefficient estimates $\hat{\mu}$, $\hat{\beta}$, $\hat{\alpha}_2$ and $\hat{\alpha}_3$ ($\hat{\alpha}_1 = 0$ as this is the default treatment parameterization). Their confidence intervals can be obtained by `confint(fiber1)`. As $\hat{\beta} = 0.954 > 0$, the thicker the fiber, the stronger it is, the strongest type of fieber is `type2`, although factor `type` is now insignificant. As, in case of anova and linear model, the rest concerns testing the individual hypothesis about the coefficients being zero. For example, the p-value for testing $H_0 : \beta = 0$ (the coefficient for `thickness` variable is 4.26e-06, hence $H_0 : \beta = 0$ is rejected.

Analysis in R: diagnostics

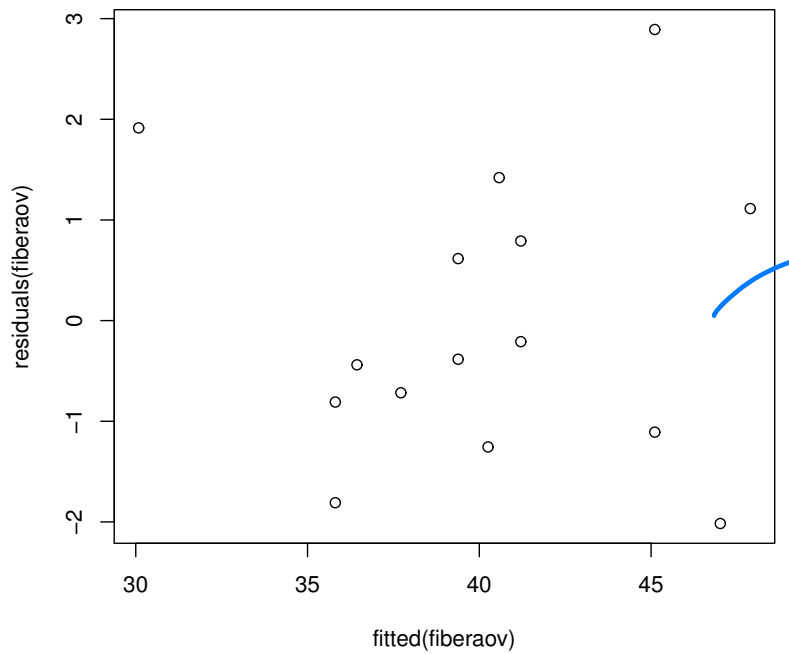
The residuals and fitted values can (and should) be investigated as usual.

```
> qqnorm(residuals(fiber1))
> plot(fitted(fiber1),residuals(fiber1))
```

check cond.s of model



seems OK



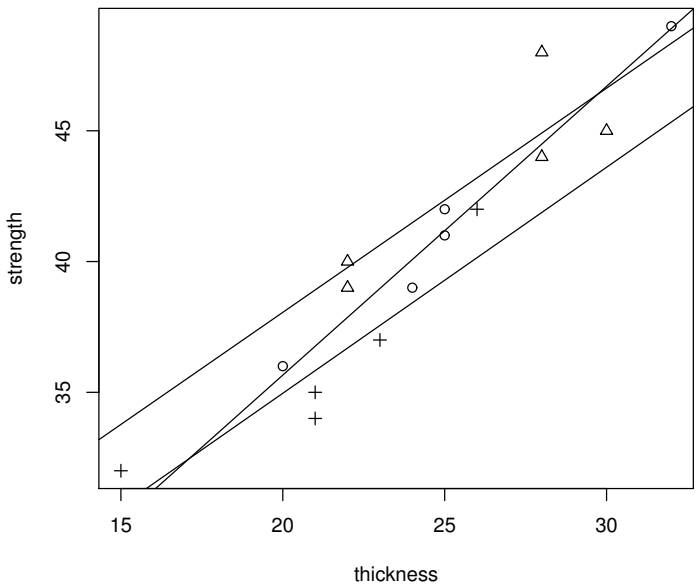
seems OK

Analysis in R: interaction between factor and predictor (1)

for each level i of fac. we have same β → influence of thick. is same for each group.

The model $Y_{ik} = \mu + \alpha_i + \beta X_{ik} + e_{ik}$ says that within each level i of the factor the dependence of Y on X is a straight line with the same slope.

```
> plot(strength~thickness,pch=unclass(type))
> for (i in 1:3) abline(lm(strength~thickness,data=fiber[fiber$type==i,]))
```



Plot shows no indication that the true lines would not be parallel. We can test for that as follows: fit the model with different slopes $\beta_1, \beta_2, \beta_3$ for each factor level $Y_{in} = \mu + \alpha_i + \beta_i X_{in} + e_{in}$, and then test $H_0 : \beta_1 = \beta_2 = \beta_3$. In other words, this is testing for the interaction between factor type and predictor thickness.

⊙ for each group, we can fit simple linear regr. → if they were parallel, the slope is same for all groups.
if their slope deviates → type influences the eff. of thick.
= there is interact. b.w. them

Analysis in R: interaction between factor and predictor (2)

Testing for the interaction between factor type and predictor thickness is done by including the interaction term type:thickness in the model.

```
> fiber3=lm(strength~type*thickness,data=fiber); anova(fiber3)
[ some output deleted ]
               Df  Sum Sq Mean Sq F value    Pr(>F)
type            2 140.400  70.200 25.0231 0.0002107 ***
thickness       1 178.014 178.014 63.4538 2.291e-05 ***
type:thickness  2   2.737   1.369  0.4878 0.6292895
Residuals       9  25.249   2.805
```

for here, it needs to be b.w. fac. and var. → it is interact. b.w. fac. and var. (interact. means slope becomes dep. on level)

don't pay att. to them

testing hypoth. that βs are all same.

p-val. is relatively big.

H_0 is not reject.

β is same for all levels of fac.

⊙ type * thick. = thick. * type → order doesn't matter

The model formula type*thickness, rather than type+thickness, describes the model $Y_{ik} = \mu + \alpha_i + \beta_i X_{ik} + e_{ik}$. Only the last p-value is relevant which always concerns interaction for models with interaction. We conclude from it that $H_0 : \beta_1 = \beta_2 = \beta_3$ is not rejected, i.e., there is no interaction between factor type and predictor thickness (or, the slopes for all groups are the same).

Analysis in R: interaction between factor and predictor (3)

```
> summary(fiber3)
[ some output deleted ]
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)μ       13.5722     4.9375   2.749 0.022520 *
type2               7.3421     7.6684   0.957 0.363355
type3               4.1068     6.6631   0.616 0.552932
thickness β         1.1043     0.1937   5.702 0.000294 ***
type2:thickness    -0.2471     0.2960  -0.835 0.425337
type3:thickness    -0.2401     0.2843  -0.845 0.420215
```

$\beta_2 - \beta_1 =$

$\beta_3 - \beta_1 =$

not signif. → there is no diff. b.w. type 1 – type 2 and type 1 – type 3

not signif. = there is no diff.

diff. b.w. type 2 – type 3 is unknown

The estimates of `type2:thickness` and `type3:thickness` give the estimated differences $\hat{\beta}_2 - \hat{\beta}_1$ and $\hat{\beta}_3 - \hat{\beta}_1$. The interaction term is not significant. So, no indication that the initial analysis is in trouble.

Prediction and feature selection in linear regression

Lasso, ridge and elastic net method (1)

- In this case we have only 4 variables to choose from, so we were able to identify the significant ones by a manual inspection of p-values.
- This will quickly become unfeasible if the number of predictors is big.
- An algorithm that could somehow automatically shrink the coefficients of the insignificant variables or (better!) set them to zero altogether?
- This is precisely what lasso and its close cousin, ridge regression, do.
- Lasso and ridge regularization work by adding a penalty term $\lambda P(\beta)$ to the mean residual sum of squares

to adding a lot of β= overfit. data

$$\frac{1}{N}\sum_{n=1}^{N}\left(Y_n-(\beta_0+\beta_1X_{n,1}+\ldots+\beta_pX_{n,p})\right)^2=\frac{\|Y-X\beta\|^2}{N}$$

when we try to find estim.s of β param.s we minimize this quad. func.

and minimizing the resulting sum $\frac{1}{N}\|Y-X\beta\|^2+\lambda P(\beta)$ ($2N$ can be used instead of N) with respect to $\beta=(\beta_0,\beta_1,\ldots,\beta_p)\in\mathbb{R}^{p+1}$:

pen. term dep. on β = the more you add β, this term will grow s.t. it reflects compl. of model

penalizes complexity

$$\frac{1}{N}\|Y-X\beta\|^2+\lambda P(\beta)\to\min_{\beta}$$

Lasso, ridge and elastic net methods (2)

- Lasso method: $P(\beta) = \|\beta\|_1 = \sum_{k=0}^{p} |\beta_k|$, i.e., → we take sum of abs. val.s of β

$$\min_{\beta} \left\{ \frac{\|Y - X\beta\|^2}{N} + \lambda \|\beta\|_1 \right\} = \min_{\beta} \left\{ \frac{\|Y - X\beta\|^2}{N} + \lambda \sum_{k=0}^{p} |\beta_k| \right\}.$$

- Ridge method: $P(\beta) = \|\beta\|_2^2 = \sum_{k=0}^{p} \beta_k^2$, i.e., → we take sum of sq. of abs. val.s of β

$$\min_{\beta} \left\{ \frac{\|Y - X\beta\|^2}{N} + \lambda \|\beta\|_2^2 \right\} = \min_{\beta} \left\{ \frac{\|Y - X\beta\|^2}{N} + \lambda \sum_{k=0}^{p} \beta_k^2 \right\}.$$

- Elastic net method: $P(\beta) = \alpha\|\beta\|_1 + (1-\alpha)\|\beta\|_2^2$ ($0 \leq \alpha \leq 1$ controls the "mix" of ridge and lasso regularisation, with $\alpha = 1$ being "pure" lasso and $\alpha = 0$ being "pure" ridge), i.e.,

combines lasso and ridge partly

$$\min_{\beta} \left\{ \frac{\|Y - X\beta\|^2}{N} + \lambda \left(\alpha \|\beta\|_1 + (1-\alpha)\|\beta\|_2^2 \right) \right\}.$$

lasso pen. part. — ridge pen. part

- Parameter $\lambda \geq 0$ is a free parameter which is usually selected by using a method called cross-validation.

Lasso, ridge and elastic net methods

- Ridge regression enforces the β coefficients to be lower, but it does not enforce them to be zero. That is, it will not get rid of irrelevant features but rather minimize their impact on the trained model.
- Lasso method overcomes the disadvantage of ridge regression by setting the coefficients β to zero if they are not relevant. One usually ends up with fewer features included in the model than you started with, which is an advantage.
- The R-package `glmnet` implements the elastic net method (for any $0 \leq \alpha \leq 1$) by R-function `glmnet`, with particular cases `ridge` ($\alpha = 0$) and `lasso` ($\alpha = 1$).
- The choice of λ is done by the cross-validation method, implemented by the R-function `cv.glmnet`.

⊙ bigger lambda becomes = more penalty we put on complexity → corresp. coeff.s become smaller, some disappear = become 0

Analysis in R: generic code for lasso (ridge and elastic net)

Suppose we have a data frame named data, with its first column being the response variable, and the remaining columns are the features to select from.

```
>library(glmnet)
>x=as.matrix(data[,-1]) #remove the response variable
>y=as.double(as.matrix(data[,1])) #only the response variable
>train=sample(1:nrow(x),0.67*nrow(x)) # train by using 2/3 of the data
>x.train=x[train,]; y.train=y[train]  # data to train
>x.test=x[-train,]; y.test=y[-train] # data to test the prediction quality
>lasso.mod=glmnet(x.train,y.train,alpha=1)
>cv.lasso=cv.glmnet(x.train,y.train,alpha=1,type.measure='mse')
>plot(lasso.mod,label=T,xvar="lambda")  #have a look at the lasso path
>plot(cv.lasso) # the best lambda by cross-validation
>plot(cv.lasso$glmnet.fit,xvar="lambda",label=T)
>lambda.min=lasso.cv$lambda.min; lambda.1se=lasso.cv$lambda.1se
>coef(lasso.model,s=lasso.cv$lambda.min) #beta's for the best lambda
>y.pred=predict(lasso.model,s=lambda.min,newx=x.test) #predict for test
>mse.lasso=mean((y.test-y.pred)^2) #mse for the predicted test rows
```

other col.s = predictors

1st col. = resp.

2/3 of rows = train

other rows = test

$\alpha = 1$ to insist on lasso

cross-valid. to choose lambda

plot shows val.s of var.s based on λ

mean sq. err.

`lambda.min` is the value of λ that gives minimum mean cross-validated error. The other λ saved is `lambda.1se`, which gives the most regularized model such that error is within one standard error of the minimum.

smallest lambda

next lambda = preferred

multiple comparisons

Multiple testing

e.g. we want to test $\beta_1 = 0$, $\beta_2 = 0$, $\beta_3 = 0$... together

- H_0 is falsely rejected (type I error) with probability at most α_{ind} $(= 0.05)$.
- Given 2 null hypotheses there are 2 possibilities to make such an error. The probability of at least 1 error is then at most $0.05 + 0.05 = 0.1$.

In principle your errors will add up.

- Suppose for each of m null hypotheses $H_{0,1}, \ldots, H_{0,m}$, the probability of type I error is at most α_{ind}, then the probability of at least 1 error is at most $m\alpha_{ind}$. Indeed,

Bonferroni bound

$$P(\text{at least one } H_{0,i} \text{ is rejected}) \leq \sum_{i=1}^{m} P(H_{0,i} \text{ is rejected}) \leq m\alpha_{ind}.$$

- $P(\text{at least one } H_{0,i} \text{ is rejected})$ is called family-wise error rate (FWER).
- To provide FWER ≤ 0.05, we can impose $\alpha_{ind} \leq \frac{0.05}{m}$ for all $H_{0,i}$. Indeed,

$$\text{FWER} \leq m\alpha_{ind} \leq m\frac{0.05}{m} = 0.05.$$

if you want to control this by 0.05, then each indiv. needs to be control. by 0.05/m (m = # of hypoth. you want to control)

you want to control this

Multiple testing: Bonferroni correction

★ hardly any features will be decl. significant.

- Thus, a simple way to control the family-wise error rate FWER $\leq \alpha_{tot}$ for some overall level α_{tot} is to carry out each individual test with $\alpha_{ind} = \frac{\alpha_{tot}}{m}$, known as the Bonferroni correction.
- This is the same as to compare the individual p-values p_{ind} to $\alpha_{ind} = \frac{\alpha_{tot}}{m}$.
- Adjusted p-values for simultaneous tests p_{adj} are such that if every $H_{0,i}$ with $p_{adj} \leq \alpha_{tot}$ is rejected, then FEWR $\leq \alpha_{tot}$.
- Adjusted p-value according to Bonferroni correction is $p_{adj} = mp_{ind}$.
- In R, the adjusted p-values are called adjusted P-values for Multiple Comparisons, and are computed by `p.adjust`.
- Bonferroni correction is very conservative. Indeed, for reasonable α_{tot} (like 0.05)) and relatively large n (like $n = 100$), there will be very few simultaneously rejected $H_{0,i}$'s, because hardly ever we will have $100p_{ind} \leq 0.05$, or $p_{ind} \leq 0.00005$.

indiv. p-val.

→ you adjust your p-val.s and compare w/ α_{tot}

then we comp. to 0.05 → so p-indiv. should be v. small if we want it to be smaller than 0.05.

Multiple testing procedures for controlling FWER

Multiple testing arises when:

- there are many parameters of interest.
- investigating all differences $\alpha_i - \alpha_{i'}$ of a set of effects α_i in ANOVA.

The latter is the so called *"a-posteriori testing"*, performed following rejection of a composite hypothesis of the type $H_0 : \alpha_i = 0, i = 1, \ldots, I$.

Bonferroni correction is not the only method to control FWER, alternatives:

- Sidak correction (under indep. assump., slightly better than Bonferroni),
- Holm-Bonferroni method, better than Bonferroni (making it obsolete)
- Hochberg's step-up procedure
- Tukey's procedure (`library(multcomp)`, only for pairwise comparisons).
- some extensions of the above mentioned
- Similarly, one designs simultaneous confidence intervals for a set of parameters that have overall confidence level of $1 - \alpha_{tot}$.

you increase p-indv. but not too much, then compare it to 0.05

To implement these methods in R: fed with given individual p-values p_{ind} and a specified method, `p.adjust` gives the adjusted p-values p_{adj} (not for Sidak and Tukey's procedures) which should be compared to a specified significance level α_{tot}. The corresponding method rejects those hypothesis for which $p_{adj} \leq \alpha_{tot}$.

Individual p-values obtained in ANOVA

Recall the data pvc on the production of the plastic PVC, where 3 operators used 8 different devices called resin to produce PVC of size psize.

```
> pvc$operator=as.factor(pvc$operator); pvc$resin=as.factor(pvc$resin)
> pvcaov=lm(psize~operator*resin,data=pvc); summary(pvcaov)
[ some output deleted ]
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        36.2500     0.8598  42.164  < 2e-16 ***
operator2          -0.8500     1.2159  -0.699 0.491216
operator3          -0.9500     1.2159  -0.781 0.442245
resin2             -1.1000     1.2159  -0.905 0.374615
resin3             -5.5500     1.2159  -4.565 0.000126 ***
[ some output deleted ]
resin8              0.5500     1.2159   0.452 0.655078
operator2:resin2    1.0500     1.7195   0.611 0.547175
[ some output deleted ]
operator3:resin8   -2.7000     1.7195  -1.570 0.129454
```

coeff.s

indv. p-val.s

$\alpha_2 - \alpha_1$

$\alpha_3 - \alpha_1$

p-val. for only $H_0 : \beta_2 = \beta_1$

we want simult. p-val.s for mult. testing

The p-values produced above are not simultaneous. The p-values in the lines resin2, resin3,... are for testing the individual hypotheses $H_0 : \beta_2 = \beta_1, H_0 : \beta_3 = \beta_1, \ldots$.

Multiple testing in R by Tukey's method

```
> library(multcomp)
> pvcmult=glht(pvcaov,linfct=mcp(resin="Tukey"))
> summary(pvcmult)
             Estimate Std. Error t value Pr(>|t|)
2 - 1 == 0    -1.100      1.216  -0.905   0.9827
3 - 1 == 0    -5.550      1.216  -4.565    <0.01 **
4 - 1 == 0    -6.550      1.216  -5.387    <0.01 ***
5 - 1 == 0    -4.400      1.216  -3.619   0.0251 *
6 - 1 == 0    -6.050      1.216  -4.976    <0.01 ***
7 - 1 == 0    -3.350      1.216  -2.755   0.1538
8 - 1 == 0     0.550      1.216   0.452   0.9998
[ some output deleted ]
8 - 6 == 0     6.600      1.216   5.428    <0.01 ***
8 - 7 == 0     3.900      1.216   3.208   0.0625 .
```

create model

Tukey correction on all compar.s of resin coeff.s simult.

now p-val.s are simult.

significant = there is diff.

not signif. = they are not diff.

Adjusted p-values for simultaneous testing the null hypotheses $H_0 : \beta_2 = \beta_1$, $H_0 : \beta_3 = \beta_1$, $H_0 : \beta_4 = \beta_1$, ..., $H_0 : \beta_8 = \beta_7$, where β_j is the main effect of the jth level of `resin`. The probability that one or more of these would be less than 0.05 *while* the corresponding null hypothesis were true is less than 0.05. Thus we can "safely" say that *all* differences with p-value < 0.05 are nonzero.

False Discovery Rate (FDR)

- Procedures that control the FWER are considered too conservative for most cases of multiple testing (they lead to a substantial loss in power).
- Beter to control (and less stringent) is the False Discovery Rate (FDR) introduced by Benjamini and Hochberg (1995), the expected proportion of falsely rejected null hypothesis among the rejected hypotheses.
- Testing m hypotheses simultaneously (of which m_0 are true null hypotheses):

	H_0 is true	H_1 is true	Total
Procedure rejects H_0	V	S	R
Procedure does not reject H_0	U	T	$m - R$
Total	m_0	$m - m_0$	m

total num. of prod that rejects H_0

V is the number of false positives; T is the number of false negatives.

- Random variable R is observed and the number of hypothesis m is known.
- Random variables V, S, U, T are unobserved and the number of true hypothesis m_0 is unknown.
- FDR $= E\left(\frac{V}{R}\right)$, where we define FDR $= 0$ if $R = 0$ (then also $V = 0$).

BH and BY procedures to control FDR

- The Benjamini-Hochberg procedure ensures that its FDR is at most α:
  - Order the p-values $p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}$ and the null hypotheses $H_{0,(1)}, H_{0,(2)}, \ldots, H_{0,(m)}$ correspondingly;
  - If $k_{max} = \max_k \left(p_{(k)} \leq \frac{\alpha k}{m}\right)$ exists, reject $H_{0,(1)}, \ldots, H_{0,(k_{max})}$; otherwise reject nothing.
- The BH procedure is valid when the m tests are independent.
- Notice that $k_{max} = \max_k \left(p_{(k)} \leq \frac{\alpha k}{m}\right) = \max_k \left(\frac{mp_{(k)}}{k} \leq \alpha\right)$.
- Command `p.adjust` gives the adjusted ordered p-values $\frac{mp_{(k)}}{k}$, which should be compared to α, to control FDR up to level α.
- Benjamini-Yekutieli procedure (BY) is the generalization of BH procedure (for arbitrary dependence assumptions): instead of m one takes $mc(m)$ where $c(m) = \sum_{i=1}^{m} \frac{1}{i}$, so the BY procedure is a bit more conservative.
- `p.adjust` gives the adjusted ordered p-values also for the BY procedure.

Multiple testing in R

```
> p.raw=summary(pvcaov)$coef[,4] # vector of individual (raw) p-values
> p.raw=p.raw[order(p.raw)] # order the p-values
> p.val=as.data.frame(p.raw)
> p.val$Bonferroni=p.adjust(p.val$p.raw,method="bonferroni")
> p.val$Holm=p.adjust(p.val$p.raw,method="holm")
> p.val$Hochberg=p.adjust(p.val$p.raw,method="hochberg")
> p.val$BH=p.adjust(p.val$p.raw,method="BH")
> p.val$BY=p.adjust(p.val$p.raw,method="BY"); round(p.val,3)
                  p.raw Bonferroni  Holm Hochberg    BH    BY
(Intercept)       0.000      0.000 0.000    0.000 0.000 0.000
resin4            0.000      0.000 0.000    0.000 0.000 0.001
resin6            0.000      0.001 0.001    0.001 0.000 0.001
resin3            0.000      0.003 0.003    0.003 0.001 0.003
resin5            0.001      0.033 0.027    0.027 0.007 0.025
resin7            0.011      0.264 0.209    0.209 0.044 0.166
operator3:resin8  0.129      1.000 1.000    0.954 0.444 1.000
operator3:resin5  0.361      1.000 1.000    0.954 0.892 1.000
resin2            0.375      1.000 1.000    0.954 0.892 1.000
operator3         0.442      1.000 1.000    0.954 0.892 1.000
[ some output deleted ]
```

To wrap up

Today we learned:

- ANCOVA
- prediction and feature selection in linear regression
- multiple testing procedures, FDR control

Next time: Logistic regression, Poisson regression