Nested Designs (cont.)

NRES 710

Last compiled: 2024-11-20

So last class we discussed nested designs, and specifically how they can handle pseudoreplication and random effects that are nested within each other.

Today I want to show you the upper limits of what this design can handle.

## Split-plot Design

Let’s consider a field experiment with six fields and four variables:

We want this to be a fully-factorial design, where we will cross every single one of these variables with every other variable.

* E.g., bison, no fertilizer, medium water, no pesticide, etc.
* 2 bison \* 2 fertilizer \* 3 water \* 2 pesticides = 24 plots within each field.
* We need to break up each field into 24 blocks to do all of these combinations.

If we created a tiny block (1/24th of field) and put a bison into it, in two months when we come back to measure the effects, the bison will be dead and all the biomass will be gone.

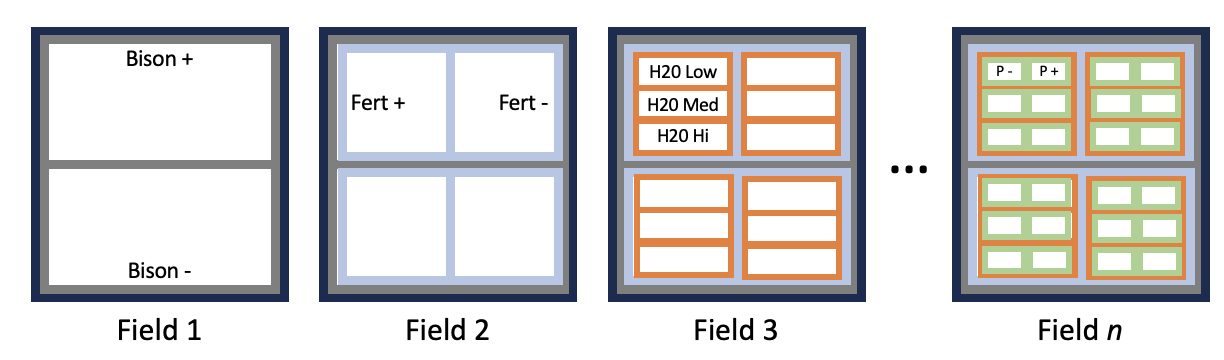
So an alternative way to approach this would be to split each field in half. Each half will be randomly assigned to have a bison or not.

Now, we’ll take every ‘bison plot’, split those in half, and randomly assign fertilizer or no fertilizer treatment.

Now, we’ll take every fertilizer plot, split those in three, and randomly assign water: low, medium, high.

Last, we’ll take each water plot, split those in half, and randomly assign pesticide/no pesticide. This is what this would look like:

*Brian draw this on the board: field 1, 2, 3, … 6.; with design illustrated in different fields with different colored markers*



How many rows would our Excel file have for this design?

* 6 fields
* 2 bison treatments
* 2 fertilizer treatments
* 3 water treatments
* 2 pesticide treatments
* 6 fields \* 2 bison treatments \* 2 fertilizer treatments \* 3 water treatments \* 2 pesticide treatments = **144 biomass measurements**

We will 144 biomass measurements; one each from every pesticide plot.

This is called a ‘split-plot design’! Lots of plot splitting happening here.

**Q:** We have a 144 samples to look at the effect of pesticide. Do we have 144 samples to look at the effect of water?

* We have 72 water plots.
* We have 24 fertilizer plots.
* We have 12 bison plots.

This is important! While we have 144 rows, our sample sizes differ depending the effect we are testing for.

We could simplify the model and just test for an effect of bison, for example. But then we would introduce a lot of pseudoreplication into the design. Therefore, we have to account for these nested random effects to account for that pseudoreplication.

We may want to look at the effect of these all at the same time, because we want to account for swamping.

A scary thing about this design is that the people who do these experiments are interested in interactions. Two-way interactions (e.g., bison, fertilizer; water, fertilizer; fertilizer, pesticides), three-way interactions (e.g., bison, water, pesticide), or four-way interactions…??

But the important part of this design is we need to account for pseudoreplication. We can do that using random effects in our model, that are nested according to the split-plot structure of our experiment.

## Statistical model

* between-field error
* between-bison plot error (*residual error for the bison effect*)
* between-fertilizer plot error (*residual error for the fertilizer effect*)
* between-water plot error (*residual error for the water effect*)
* residual error

**Q:** Why is there no pesticide random effect? Can’t include it because each pesticide treatment is only measured once. Maybe if we had a repeated-measures through time for pesticides, then we could have a pesticide effect. But we don’t.

**Q:** Are we double-counting fixed effects as random effects? No. The fixed-effect for bison is different from the field|bison random effect. The fixed effect compares all bison plots to all no-bison plots. The random effect is different because *bison is nested within field*, and every field has a different effect, so the bison effects are different within every different field.

Last thing: what if we wanted to test for interactions…?

Let’s say we added an interaction term, . What is the meaning of ?

* When we add , then becomes the effect of bison in unfertilized plots, becomes the effect of fertilizer in no-bison plots, and becomes the difference in the effect of bison between fertilizer and no fertilizer plots.

Let’s say we added an interaction term, . What is the meaning of ?

* When we add , then becomes the effect of pesticide in unfertilized plots, and becomes the effect of pesticides in fertilized plots. So becomes the difference in the effect of pesticide between fertilizer and no fertilizer plots.

Let’s say we added another interaction term, . What is the meaning of …?!

* We’ve just changed the meaning of … before it was the difference between the effect of bison in fertilizer and unfertilized plots. Now that we have the interaction, is the difference in the effect of bison between fertilized and unfertilized plots *in no-pesticide plots*.
* Which means, is… the difference in the difference in the effect of bison between fertilizer and no fertilizer plots *between pesticide plots and no pesticide plots*.

Graphically, this might look like…

*Brian draw graphs on the board. Create one panel with bison x fertilizer, with bison positive effect, no fertilizer effect, and a bison x fertilizer interaction. Create another panel with bison x pesticide; bison positive effect, no pesticide effect, but then a negative bison x pesticide effect that is beta8…???*

The take-home message: this gets confusing really quickly… don’t do three-way interactions. It can be done, but it’s difficult to interpret.

## Split-plot analysis in R

Let’s examine how to analyze these data, without interactions, in R. Let’s take a look at a dataset:

# Load the data  
datum <- read.csv("lecture\_19\_dataset1.csv")  
  
# Examine it  
head(datum, 10)

I simulated data with the split-plot design we explored on the board, but I didn’t go all the way out to the pesticide treatment because that was ridiculously complex.

Instead, there are 6 fields, that are split in half (bison, no bison), which are split in half (fertilizer, no fertilizer), which are split into thirds (low, medium, and high water).

We can dive into what the true values are for each effect, but I am not actually super worried about that. Instead, this is a pseudoreplication problem, and we want to make sure that we specify this model properly so that R gets the correct sample size for each effect…!

However, to start, let’s assume that we did not recognize the split-plot design of the model. Instead, we fit a simple ‘lme()’ with only a random-effect of field.

# Linear-mixed effects model w/ random effect of field  
library(nlme)  
results <- lme(Biomass ~ Bison + Fertilizer + Water, data = datum, random = ~1|Field)  
summary(results)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Field  
## (Intercept) Residual  
## StdDev: 2.742 2.216  
##   
## Fixed effects: Biomass ~ Bison + Fertilizer + Water   
## Correlation:   
## (Intr) Bison Frtlzr WatrLw  
## Bison -0.207   
## Fertilizer -0.207 0.000   
## WaterLow -0.253 0.000 0.000   
## WaterMedium -0.253 0.000 0.000 0.500  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.7159 -0.7222 0.1008 0.6732 2.1067   
##   
## Number of Observations: 72  
## Number of Groups: 6

# 72 observations with 6 groups (fields)... does not account for pseudoreplication properly, we need better nesting

The summary says we have 72 observations with 6 groups (fields). Take a look at the Degrees of Freedom (DF). We can go into the exact math of degrees of freedom, but it’s complicated and I don’t pretend to understand it entirely.

However, a good rule of thumb is that the DF has to be less than the number of samples that should be used. In this case, it is using at least 62 samples when estimating each of these effects. In fact, it’s using all 72 (‘72 observations’ at the bottom).

**Q:** How many samples do we have for ‘bison’? In truth, we actually have 12 bison plots: 6 with bison, and 6 without. So the true Degrees of Freedom for bison should be less than 12. But it says 62, which means we are pseudoreplicating with respect to bison, and also fertilizer as well. However, we have not pseudoreplicated with respect to ‘water’, as all 72 samples are used to test the effect of water.

Part of the definition of a split-plot design is that *you have different sample sizes for different variables*. So we need to remove this pseudoreplication to correctly measure and test these effects.

### Split-plot Design  
# Linear-mixed effects model w/ random, nested effects  
results2 <- lme(Biomass ~ Bison + Fertilizer + Water, data = datum, random = ~1|Field/Bison/Fertilizer)  
summary(results2)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Field  
## (Intercept)  
## StdDev: 2.252  
##   
## Formula: ~1 | Bison %in% Field  
## (Intercept)  
## StdDev: 2.083  
##   
## Formula: ~1 | Fertilizer %in% Bison %in% Field  
## (Intercept) Residual  
## StdDev: 1.557 0.9668  
##   
## Fixed effects: Biomass ~ Bison + Fertilizer + Water   
## Correlation:   
## (Intr) Bison Frtlzr WatrLw  
## Bison -0.496   
## Fertilizer -0.243 0.000   
## WaterLow -0.100 0.000 0.000   
## WaterMedium -0.100 0.000 0.000 0.500  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.00393 -0.60943 -0.05625 0.55974 2.16708   
##   
## Number of Observations: 72  
## Number of Groups:   
## Field Bison %in% Field Fertilizer %in% Bison %in% Field   
## 6 12 24

# The random effects are nested with field > bison > fertilizer. Water is not included, because it is our 'ultimate sample' and therefore is the residual error.

Let’s take a look at the bottom of this. It says we have 72 observations (correct), 24 fertilizer plots (correct), 12 bison plots (correct), and 6 fields (correct)!

If we scroll up and look at the Degrees of Freedom, it’s using 46 for water, 11 for fertilizer, and 5 for bison. All of these are less than the number of groups from below, which is good. It is correctly measuring the amount of replication in our model; we are using the correct sample size for each of our particular variables.

We also get random effects for each variable. Reading them from the bottom upward:

* Residual error: error for the water treatment
* Intercept error: variation in biomass due to fertilizer treatment
* Bison %in% field: variation in biomass due to bison effect
* Field: variation in biomass due to field

**Questions?**

You may think this is crazy – but many of you may have more difficult designs than this!

This design is fully factorial, and balanced, and isn’t missing data due to field constraints, lazy technicians, or other weird problems that inevitably arise.

What happens if we add in ‘water’ as a final, nested random effect?

# Linear-mixed effects model w/ random, nested effects  
results3 <- lme(Biomass ~ Bison + Fertilizer + Water, data = datum, random = ~1|Field/Bison/Fertilizer/Water)  
summary(results3)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Field  
## (Intercept)  
## StdDev: 2.252  
##   
## Formula: ~1 | Bison %in% Field  
## (Intercept)  
## StdDev: 2.083  
##   
## Formula: ~1 | Fertilizer %in% Bison %in% Field  
## (Intercept)  
## StdDev: 1.557  
##   
## Formula: ~1 | Water %in% Fertilizer %in% Bison %in% Field  
## (Intercept) Residual  
## StdDev: 0.7947 0.5506  
##   
## Fixed effects: Biomass ~ Bison + Fertilizer + Water   
## Correlation:   
## (Intr) Bison Frtlzr WatrLw  
## Bison -0.496   
## Fertilizer -0.243 0.000   
## WaterLow -0.100 0.000 0.000   
## WaterMedium -0.100 0.000 0.000 0.500  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.14125 -0.34707 -0.03203 0.31877 1.23417   
##   
## Number of Observations: 72  
## Number of Groups:   
## Field Bison %in% Field Fertilizer %in% Bison %in% Field   
## 6 12 24   
## Water %in% Fertilizer %in% Bison %in% Field   
## 72

# Water is included when it was not needed to be included.

For the most part, it gives us the exact same answer. R is smart enough to recognize that ‘water’ is the ‘ultimate sample’ and measures it as such: 72 samples for water in fert. in bison in field.

It doesn’t change any of our *t*-values or *p*-values.

It does do something weird, where it tries to partition the water error into error due to water treatment and some additional residual error. I don’t know how R tries to do this, but it does. But ultimately it doesn’t matter, because it is ultimately using the water-plot to look at the water-effect, with correct sample sizes, which is what we are after.

**Q1:** It is testing whether ‘high water’ is different from ‘low water’ and ‘medium water’. How do we test if ‘low water’ is different from ‘medium water’? **Re-level and re-run**

**Q2:** What if want to know if water ‘as a whole’ is significant? **F-drop test!** We run it again without water, and do an F-drop test to compare the two models. And remember, we have to change it to be a ‘maximum likelihood model’, rather than a REML model.

There is one last thing I want to mention about this model. Again, sometimes you all are going to have models more complicated than this! One of the reasons this model works is that R recognizes that all of the effects are linked to the plots that are nested within each other. But if you start having ‘missing data’ or other structural anomolies, it can be hard for R to recognize the design of your data and model. There are some other algorithms you can use:

# Help file for lme()  
help(lme)  
# Notice the optimMethod -- an alternative optimization methods that allows you to change how it recognizes the correct degrees of freedom.

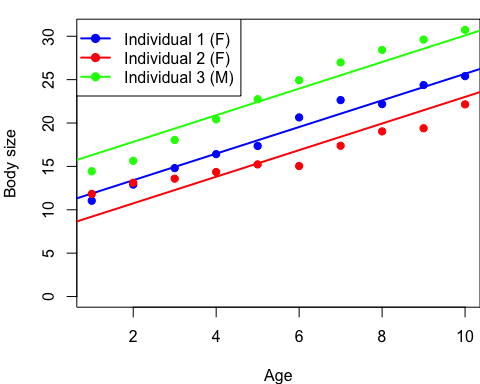
Notice the ‘optimMethod’ – an alternative optimization methods that helps the model recognize the correct degrees of freedom. Sometimes this can’t be fixed… but maybe it doesn’t matter. If your sample sizes are large (>100 for treatments), then it shouldn’t matter too much because the p-values converge at those larger sample sizes.

## Nested ANCOVA Design

Let’s do one more example, using a dataset we have seen before:

This is an example that we are pretty comfortable with.

We have previously assumed that every one of our points came from a different individual. But now let’s say that every point came from the *same individual*. (We also looked at this with tree sizes, as another example.)



**Q:** What did we call this particular design, where we measured individuals repeatedly over time. **Repeated measures design**.

Technically, we need to worry about autocorrelation… but there is another thing we need to be worried about that we have not so far.

Our individuals differ in sex: female, female, male, etc.

* **10 individuals**
* **1-10 years old**
* **100 total points**

**Q1:** What do you think our sample size is for looking at the effect of age on sample size?

* A lot of people might say ‘10’, because we only have ten individuals.
* **But it’s actually 100 samples for looking at the effect of age on size**.
* The reason is because: for every single point here, **the X-value changed for age**.
* That makes those points *true replicates* for looking at the effect of age on size.

Let’s take this to an extreme. Imagine we only had 1 individual, which we measured 10 times. Can we test for the effect of age on size?

* Yes, we can: each observation is a true replicate for looking at the effect of age on size.
* However, this is poor inference, because it’s hard to make conclusions beyond that single individual.
* But, it’s technically not pseudoreplication. We have true replicates, but we have weak inference.

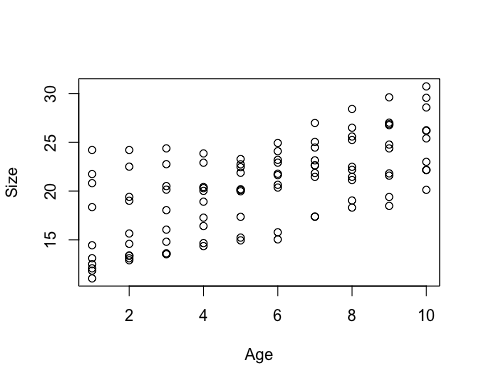
**Q2:** How many samples do we have to look at the effect of sex? **Ten**. The sex of the individual could not change, which makes each one of those points a *pseudoreplicate* for looking at the effect of sex.

**Q3:** How many samples do we have to look at an interaction – a sex by age interaction? 10 or 100?

* A little more complicated… the answer is **100 samples**. Each sample is actually a true replicate for looking at the interaction between age and size, *because age did change*.
* **Rule of thumb**: when it comes to interactions, whatever variable has the greatest number of samples – this becomes the number of samples for your interactions.

Let’s learn how to run this analysis in R:

Briefly looking at our data, we have 10 individuals that we have each measured 10 years, 5 females (0) and 5 males (1).



You can almost see that there are ten individuals here.

I am not going to worry about autocorrelation, because you know how I feel about it: it ain’t worth it!

In terms of nesting structure here, INDIVIDUAL is NOT nested within AGE, because every INDIVIDUAL got every AGE. But individual is nested within SEX, because not every individual got every sex. The design expands from top-to-bottom with: Sex < Individual < Age.

Sex is a fixed effect, so we won’t list it as a random effect. All we will do is list individual as random effect, and R will automatically recognize that individual is nested within sex. The key thing here: individual is our *sample* for looking at the effect of sex.

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Individual  
## (Intercept) Residual  
## StdDev: 2.981 2.075  
##   
## Fixed effects: Size ~ Age + Male   
## Correlation:   
## (Intr) Age   
## Age -0.280   
## Male -0.679 0.000  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.141547 -0.607007 0.005251 0.644986 2.208191   
##   
## Number of Observations: 100  
## Number of Groups: 10

Look at the degrees of freedom: it uses the appropriate less than 10 degrees of freedom for MALE (because we have 10 individuals), and then it uses less than 100 degrees of freedom for the effect of age (because we have 100 measurements at different ages).

Adding in the interaction term:

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Individual  
## (Intercept) Residual  
## StdDev: 3.025 1.279  
##   
## Fixed effects: Size ~ Age + Male + Age:Male   
## Correlation:   
## (Intr) Age Male   
## Age -0.246   
## Male -0.707 0.174   
## Age:Male 0.174 -0.707 -0.246  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.02368 -0.66348 -0.04287 0.67558 2.19235   
##   
## Number of Observations: 100  
## Number of Groups: 10

DF for Age:Male = 88 – so it used all 100 samples to look at the interaction between age and sex.

## Final design

I want to briefly describe one final design that is used by many graduate students. Let’s consider these variables:

We have worked with 20 individuals, given them the drug/placebo, and measured their pain through time: 0 hours, 2 hours, 4 hours, 8 hours.

We assume that everyone’s pain is the same at the beginning. After 2 hours, the placebo individuals have the same pain, and drugged individuals have less pain. And this continues in this pattern through time… until it ultimately converges together again at 8 hours.

Linear model:

We treat ‘Time’ as categorical, because then we no longer have to assume a linear relationship.

Do we have an interaction? **Absolutely!** We actually don’t care about the individual effects of drug or time – but what we really care about is the interaction effects through time.

**Q:** If we have 10 individuals, what’s our sample size for looking at the effect of drug? **10!** – but again, we don’t care about beta1. The interpretation for beta1 is the difference between drug and control at time 0. We already believe this to be ~zero.

Every single point is a true replicate for looking at the effect of time AND every single point is a true replicate for looking at the interaction between drug and time – which is our focal interest. We want to know how these lines diverge.

**Q:** If you get a significant interaction, what should you do? **Analyze things separately!** Start to analyze the different time-chunks separately. In each time-chunk, you model becomes:

In this case, you won’t have to include random effects, because you will only have 1 measurement for each individual in each time period, so pseudoreplication will not be an issue. You only need to include the random effect when you use the big, whole model.

So, this model looks scary at first… but if you identify the significant interaction effect, then you can say: “We found a significant interaction between drug and time; therefore, we analyzed the effect of drug separately during each time period.”

Many people often jump straight to analyzing the effect of drug separately during each time period, but this is a mistake. Sometimes the noise will not produce significant differences at individual time steps, but the global analysis with all the replicates can identify the significant divergence due to the interaction effect.

## Truth

Here is the code to create Truth for the two datasets analyzed here today.

################### 'Truth' ####################   
### Lecture 19: code to simulate data for class  
  
### Dataset 1  
# Simulate split-plot design data w/ three fixed effects: bison, fertilizer, water split across 6 fields  
set.seed(123)  
  
# Fields  
n\_fields <- 6  
n\_bison <- 2  
n\_fertilizer <- 2  
n\_water <- 3  
Field <- sort(rep(rep(1:n\_fields, n\_bison\*n\_fertilizer\*n\_water)))  
  
# Bison  
Bison <- rep(sort(rep(c(0, 1), n\_fertilizer\*n\_water)), n\_fields)  
  
# Fertilizer  
Fertilizer <- rep(sort(rep(c(0, 1), n\_water)), n\_fields)  
  
# Water  
Water <- rep(c("Low", "Medium", "High"), n\_fields \* n\_bison \* n\_fertilizer)  
Water <- factor(Water, levels = c("Low", "Medium", "High"))  
dummy <- data.frame(model.matrix(~ Water - 1))  
  
# Field error  
FieldError <- rep(rnorm(n\_fields, 0, 1), each = n\_bison \* n\_fertilizer \* n\_water)  
  
# Bison error  
BisonError <- rep(rnorm(n\_bison \* n\_fields, 0, 2), each = n\_fertilizer \* n\_water)  
  
# Fertilizer error  
FertilizerError <- rep(rnorm(n\_fertilizer \* n\_bison \* n\_fields, 0, 2), each = n\_water)  
  
# Residual error  
Error <- rnorm(length(Field), 0, 1)  
  
# Response variable -- biomass  
Biomass <- 20 + 10\*Bison + 10\*Fertilizer + 10\*dummy$WaterMedium + 10\*dummy$WaterHigh + FieldError + BisonError + FertilizerError + Error  
  
# Save the data  
datum <- data.frame(Field = Field, Bison = Bison, Fertilizer = Fertilizer, Water = Water, WaterMed = dummy$WaterMedium, WaterHigh = dummy$WaterHigh, FieldError = FieldError, BisonError = BisonError, FertError = FertilizerError, Error = Error, Biomass = Biomass)  
  
# Save the data  
write.csv(datum, "lecture\_19\_dataset1.csv", row.names = FALSE)  
  
  
### Dataset 2  
# Body size as a function of age and sex in multiple individuals  
# Set the seed for reproducibility  
set.seed(123)  
  
# Individuals  
n\_individuals <- 10  
n\_samples <- 10  
Individual <- sort(rep(1:n\_individuals, n\_samples))  
  
# Sex  
Male <- c(rep(rep(0, n\_samples), n\_individuals/2), rep(rep(1, n\_samples), n\_individuals/2))  
  
# Age  
Age <- rep(seq(1, n\_samples, 1), n\_individuals)  
  
# Error due to individual  
IndError <- rep(rnorm(n\_individuals, 0, 3), each = n\_samples)  
  
# Error within individual (residual error)  
# Simulate autocorrelated error - each value using the mean of the previous value  
Error <- matrix(NA, length(Age), 1)  
Error[1,1] <- rnorm(1, mean = 0, sd = 1)  
for (i in 2:length(Age)){  
 Error[i,1] <- rnorm(1, mean = Error[i-1, 1], sd = 1)  
}  
  
# Response variable - Size  
Size <- 10 + 1.5\*Age + 5\*Male + -1\*Male\*Age + IndError + Error  
  
# Save the data  
datum <- data.frame(Individual = Individual, Age = Age, Male = Male, IndError = IndError, Error = Error, Size = Size)  
  
# Save the data  
write.csv(datum, "lecture\_19\_dataset2.csv", row.names = FALSE)

[–go to next lecture–](lecture_20.html)